# Bounce Rate Optimization with Markov Traffic Models

**Steven Shao** YS2833@COLUMBIA.EDU
**James Xue** JX2218@COLUMBIA.EDU

## Abstract

Many online services like digital art galleries, media-sharing platforms, and even simply generic websites rely on connections between content to attract and engage users. A common scenario in these platforms that grow organically from semi-moderated or user-submitted content is the decision of how to connect the rest of the site to this new content.

For many services, standard metrics of site health and success include user engagement (how much a user interacts) or bounce rate (how quickly a user leaves the site). In this paper, we consider the problem of bounce rate optimization; given a new node with links to old nodes, the existing network must decide how to link back to this new node, so as to maximize the length of a typical user's stay on the site.

Following the steps of (Rosenfeld & Globerson, 2016), we formulate traffic as a Markov chain variant and attempt to maximize expected walk length in this setting. We achieve similar proofs and compare our algorithm against Rosenfeld's to observe that it performs favorably in different scenarios.

## 1. Introduction

### 1.1. Application to media-sharing platforms

To motivate our problem in a specific domain, consider non-hierarchical tags, keywords, or labels, which are frequently used paradigms in modern information systems to describe content. Tags are useful because they allow users to easily interact with the content in an application, enabling users to easily search by tags and find content relevant to their interests.

In this kind of system, users typically choose these tags themselves. In order to maximize the amount of engagement their new item or post will get, they have a strong incentive to choose these tags in a strategic way. Digital art galleries and media-sharing platforms such as Instagram and Twitter popularized these tags and allow users to easily navigate through vast amounts of content to reach items of interest.

However, looking beyond just tags, the general problem that we study—that of deciding which existing pages on a website should link back to a new page—is common to almost all content-sharing websites. Not only do media-sharing platforms which utilize tags, such as Instagram or Twitter, fall under this category, but so does any platform for content-sharing that seeks to maximize engagement, or increase the chances of users staying on the website.

### 1.2. Modeling as a Markov chain

In the original paper (Rosenfeld & Globerson, 2016), the idea of a user browsing a tagged information system as a random walk on a Markov chain is introduced. The task, which is elaborated on in Sec. 3, is to choose a subset of $k$ nodes for a new given node. The idea of an item-tag system is a special case of a generic system of nodes linking to other nodes, and this is a direction that we take in our approach as well. The items and tags are nodes which represent states in a Markov chain, where the transition probabilities encapsulate the probability of users going from an item page to a tag page.

### 1.3. Brief Outline

In this paper, we formulate our general problem with similarities and differences to the previous Markov chain approaches, in Sec. 3. In Sec. 4, we present our own proof of NP-hardness to show that our own extensions and modifications of the problem have not trivialized it. In Sec. 5, we summarize our algorithm, and we show that our algorithm performs well on synthetically generated data against different baselines with our experiments in Sec. 6. Finally, in Sec. 7, we give our concluding remarks.

## 2. Related Work

Perhaps the two domains closest to our project are link or tag recommendation and analysis, as in PageRank or (Rosenfeld & Globerson, 2016), as they also involve deciding how to structure some kind of network topology so as to maximize a user metric.

Optimal tag recommendation usually refers to predicting the set of tags that a user can attribute to an item in the context of collaborative tagging, where visitors are able to contribute to the tags for any item with an underlying 'good classification' goal in mind. Some popular methods for tag recommendation use techniques such as random walks (Hotho et al., 2006) or tensor factorization (Fang et al., 2015).

The above approaches address the goal of outputting a set of tags, essentially a task of prediction for item-user pairs. However, this differs from Rosenfeld's goal, which is an item-centric optimization task dictated by whether or not traffic *reaches* the item, and thus does not draw upon tag recommendation methods.

Rosenfeld formalizes their task of optimal tagging with an application of their Markov model on bipartite graphs, and also suggests an efficient algorithm with an approximation guarantee as a result of some theoretical properties of their objective function. In runtime, this greedy step can be further optimized with a few clever linear algebra techniques involving decomposition.

In the original paper, the efficient greedy algorithm outperformed other baselines, including random-walk based methods such as PageRank, in reaching a new state. These are some great results, but we were motivated to expand upon their approach based on some impracticalities we observed about their model. For example, systems are usually not incentivized to maximize engagement with just arbitrary new nodes, but rather total engagement with all the nodes as a whole. Optimizing for this objective is the approach that we consider, described in Sec. 3.

More similar work to Rosenfeld is the task of optimizing the PageRank of web pages. For example, in (Csáji et al., 2010), the problem of computing minimum and maximum PageRank values for "fragile" links is addressed. In (Litvak & Avrachenkov, 2004), additional outgoing links are shown to have an effect on PageRank value. Furthermore, in (Olsen et al., 2010), a constant-factor approximation algorithm is given for the link building problem, which is trying to maximize the PageRank value by adding at most $k$ links. The approach used by this paper as well as Rosenfeld's is similar in proving that the probability of reaching a web page is monotonic and submodular with respect to number of links. We have improved upon their proofs of both monotonicity and submodularity, and additionally present them in Sec. 4, but our primary focus was on modifying the model to optimize a system-wide metric rather than that of any particular node or page.

## 3. Problem Formulation

In (Rosenfeld & Globerson, 2016), the authors consider a general combinatorial optimization task over Markov chains before honing in on their specific case (since their application is item-tag systems, they eventually only consider bipartite graphs). Our formulation follows theirs closely; however, partly because we do not restrict ourselves to item-tag systems but rather on any network with traffic, our model contains several key differences.

As before, consider a Markov chain over $n+1$ states, and assume state $\sigma = n+1$ is a node, or in the web traffic scenario, a new webpage for which we would like to add a set of $k$ incoming transitions from the other $n$ states—the existing pages on the site.

We must choose a subset $S \subseteq [n]$ of $k$ states to maximize the expected length of a walk in this network. Our function to optimize differs slightly from theirs: if $\mathcal{P}(\rho(S), \pi)$ is the distribution of walks over a chain with transitions $\rho(S)$ and initial state distribution $\pi \in [0,1]^{n+1}$, our objective now is to retrieve

$$\max_{S \subseteq [n], |S| \leq k} \Pr\left[\|P\| \middle| P \sim \mathcal{P}(\rho(S), \pi)\right]$$

Crucially different is that in order to optimize lengths of stays on this network, a notion of *leaving* is necessary. To accomplish this, we relax the constraint on our Markov chains, so that the outgoing edge weights of each node must merely sum to *at most* one, rather than exactly one. The difference is then the probability a user leaves the site on that page.

To fully specify how $S$ affects the transition probabilities $\rho$, we again take an approach similar to the original paper. Each old node $i \in [n]$ has two sets of transition probabilities: $q_i, \bar{q}_i \sim [0,1]^{n+1}$, when $i$ is connected to $\sigma$ and when it isn't, respectively. $Q = [q_{ij}]$ and $\bar{Q} = [\bar{q}_{ij}]$ are the corresponding transition matrices.

The original paper did not require $\sigma$ to have any outgoing edges because it merely tracked paths up until they reached $\sigma$ for the first time. Diverging from this, our $\sigma$ also has a single list of outgoing transition weights $\bar{q}_\sigma = q_\sigma \in [0,1]^{n+1}$.

We now formalize the above along with some modest constraints (like in the original paper) that roughly mirror real user traffic:

1. Users cannot reach $\sigma$ from pages that don't connect: $\bar{q}_{i\sigma} = 0$ for all $i \in [n]$.

2. When a link to $\sigma$ is added, transitioning to $\sigma$ becomes more likely while to other states less likely: $\forall i, j \in [n], q_{ij} \le \bar{q}_{ij}$.

3. Connecting a node to $\sigma$ reduces the chance of a user leaving from that node[1]:

$$\sum_{j=1}^{n+1} q_{ij} \ge \sum_{j=1}^{n+1} \bar{q}_{ij}$$

4. Our final transition weights are thus determined:

$$\rho_{ij}(S) = \rho_S(i,j) = \begin{cases} q_{ij} & i \in S \\ \bar{q}_{ij} & i \notin S \end{cases}$$

# 4. Theoretical Results

## 4.1. NP-Hardness

To show that our modifications have not trivialized the problem, we present our own proof of NP-hardness by reducing from (the decision variant of) vertex cover.

**Theorem.** *Given an undirected graph $G = (V, E)$ with $n$ nodes as input to vertex cover, there exists a linking instance $(\pi, Q, \bar{Q})$ with $n+1$ nodes such that a cover of size $k$ exists if and only if a set of $k$ nodes can be picked so as to cause the expected path length to exceed a certain threshold.*

*Proof.* Given $G$, choose our Markov chain parameters as $\pi = \left(\frac{1}{n}, \ldots, \frac{1}{n}, 0\right)$ uniform over the old nodes, and

$$q_{ij} = \begin{cases} \varepsilon & i = \sigma, j = \sigma \\ 1 & i \ne \sigma, j = \sigma \\ 0 & \text{else} \end{cases} \quad \bar{q}_{ij}(S) = \begin{cases} \varepsilon & i = j = \sigma \\ \frac{\varepsilon}{\deg i} & \{i,j\} \in E \\ 0 & \text{else} \end{cases}$$

In other words, $\sigma$ always connects to itself with an $\varepsilon$ probability, every node connecting to $\sigma$ will lead users invariably to $\sigma$, and otherwise nodes connect to neighbours in $G$ with uniform weight such that the overall outgoing weight from a node is $\varepsilon$. Note that this reduction applies even to the simpler case when $\sigma$ is restricted *not* to connect to existing nodes.

These weights are chosen so that a path starting from a node in $S$ has at least one guaranteed 'step' of weight 1; at all other nodes, 'survival' chance is $\varepsilon$. Formally, we have the four cases:

1. If the first node is $\sigma$, the expected path length $\ell_\sigma$ is such that $\ell_\sigma = \varepsilon(1 + \ell_\sigma) \implies \ell_\sigma = \frac{\varepsilon}{1-\varepsilon}$.

2. If the first node $j$ is in $S$, since a user is guaranteed to head to $\sigma$, so $\ell_j = \ell_\sigma + 1 = \frac{1}{1-\varepsilon}$.

3. If the first node $i$ is in $[n] \setminus S$ and all its neighbours are in $S$, note that the probability of a path of length $k$ is $1\varepsilon^{k-1}(1-\varepsilon)$. This is because of $k-1$ $\varepsilon$-transitions, 1 guaranteed transition, and one death. So the expected path length is

$$\ell_i = \sum_{k=1}^{\infty} k\varepsilon^{k-1}(1-\varepsilon) = \frac{\varepsilon(2-\varepsilon)}{1-\varepsilon}$$

4. If the first node $i$ is in $[n] \setminus S$ and not all of its neighbours are in $S$, $\ell_i' < \frac{\varepsilon(2-\varepsilon)}{1-\varepsilon}$. This is because if every path leaving $i$ of length $k$ reached $\sigma$, the expected length would be $\sum_{k=1}^{\infty} k\varepsilon^{k-1}(1-\varepsilon)$ as before, but some of these paths do not reach $\sigma$ with nonzero probability $p$; since there's no free edge anymore, among those the expected path length is

$$\ell_i' = \sum_{k=1}^{\infty} k\varepsilon^{k}(1-\varepsilon) = \frac{\varepsilon^2(2-\varepsilon)}{1-\varepsilon} < \frac{\varepsilon(2-\varepsilon)}{1-\varepsilon}$$

This makes the overall expected path length, a weighted average of the two, less than in case 3.

Note that if $S$ is a vertex cover, case 4 never happens. The expected path length is then our desired threshold:

$$\frac{k}{n} \cdot \ell_j + \frac{n-k}{n} \cdot \ell_i = \frac{k + \varepsilon(n-k)(2-\varepsilon)}{n(1-\varepsilon)}$$

If not, case 4 happens with nonzero probability, causing the expected path length to dip below the threshold. □

## 4.2. Monotonicity and Submodularity

### 4.2.1. Monotonicity of Base Model

One of our contributions is a much simpler, alternative proof of monotonicity of the objective function in (Rosenfeld & Globerson, 2016) that avoids the convoluted casework and auxiliary definitions necessary in the original proof. Recall that instead of expected

[1] If our setting instead represents users leaving by a transition to a dead state $\emptyset$, this is just a special case of the point above, with $q_{i\emptyset} \le \bar{q}_{i\emptyset}$

path length, their model optimizes the probability of reaching $\sigma$:

$$c(S) = \Pr\left[P_t = \sigma \text{ for some } t \middle| P \sim \mathcal{P}(\rho(S), \pi)\right]$$

**Theorem.** *For all $S \subset [n]$ and $z \in [n] \setminus S$, $c(S) \leq c(S \cup \{z\})$.*

*Proof.* For each path $P$, denote its probability as

$$d_P(S) = \pi_{P_0} \prod_{i=1}^{\|P\|} \rho_S(P_{i-1}, P_i) \tag{1}$$

Then the target probability can be expressed explicitly through its complement as

$$c(S) = 1 - \sum_{P: P_t \neq \sigma} d_P(S) \tag{2}$$

In this form, the restriction that edges not going to $\sigma$ can only decrease in weight ($\bar{q}_{ij} \geq q_{ij}$ for $j \neq \sigma$) immediately implies that paths not touching $\sigma$ can only decrease in probability.

Formally, each path $P$ not passing $\sigma$ is such that $\rho_S(P_{i-1}, P_i) \geq \rho_{S\cup\{z\}}(P_{i-1}, P_i)$, with strict inequality only if $P_{i-1} = z$. The product of these inequalities yields $d_P(S) \geq d_P(S \cup \{z\})$, so that their sum over $P$ cannot increase. $c(S)$, the complement, therefore cannot decrease when $z$ is added to $S$. □

#### 4.2.2. Nonmonotonicity of our Model

Unfortunately, a similar argument does not work for expected length. If we try to write our objective in a similar form, we get

$$f(S) = \sum_P \|P\| \cdot \pi_{P_0} \prod_{i=1}^{\|P\|} \rho_S(P_{i-1}, P_i) \tag{3}$$

for which all $P$ that try to reach $\sigma$ from a node outside of $S$ have 0 probability. When $\{z\}$ is added, a subset of these paths probabilities become positive, but every other path that goes from $z$ to a non-$\sigma$ node drops in probability. Unlike in their model, there's no simple complement that enables a localized analysis corresponding directly to one of the model constraints set up earlier.

Indeed, under our current setting, expected length is *not* monotonic with respect to $S$, with several intuitive counterexamples.

**Example.** Consider a network with a single node $\tau$ besides $\sigma$ with weights

$$\bar{q}_{\tau\tau} = 1 - \varepsilon, q_{\tau\sigma} = 1, q_{\tau\tau} = q_{\sigma\sigma} = q_{\sigma\tau} = 0$$

$\tau$ might represent a very engaging webpage such that a link to $\sigma$ is compelling enough to drive substantial traffic there, but if $\sigma$ has boring content, it will cause users to leave quickly.

For small enough $\varepsilon$, the expected path length when $S = \emptyset$ is unbounded, but is 1 when $S = \{\tau\}$.

There are other nonmonotonic cases where the best $S$ is close to any fraction of the number of total nodes. Intuitively $\sigma$ could exist in a network with two clusters of size $n_1$ and $n_2$ that are 'maximally' and 'minimally' engaging respectively. If $\sigma$ is 'moderately' engaging and links to all the minimally engaging nodes, having those nodes link back to $\sigma$ would improve overall engagement but linking the maximally engaging ones would direct traffic to a poorer part of the network, so that the optimal size of $S$ is $n_2/(n_1 + n_2)$.

#### 4.2.3. Submodularity

Here we show a similar proof of submodularity of their model by looking at complements. While the proof is not substantially simpler, we take a completely different approach so as to highlight the qualitative behavior of these metrics.

We start by showing a multiplicative property of $d(\cdot)$.

**Lemma.** *$d(\cdot)$ from Equation 1 is such that for disjoint $S, T \subseteq [n]$, if $d(S) = \alpha \cdot d(\emptyset)$ and $d(T) = \beta \cdot d(\emptyset)$, then $d(S \cup T) = \alpha\beta \cdot d(\emptyset)$.*

*Proof.* If $\alpha = \frac{d(S)}{d(\emptyset)}$, then $\alpha$ can be computed thus:

$$\alpha = \frac{\pi_{P_0} \prod \rho_S(P_{i-1}, P_i)}{\pi_{P_0} \prod \rho_\emptyset(P_{i-1}, P_i)} = \prod_{i=1}^{\|P\|} \frac{\rho_S(P_{i-1}, P_i)}{\bar{q}(P_{i-1}, P_i)}$$

When $P_{i-1} \notin S$, the top is $\bar{q}$ and there's no effect on the product, so we only care when $P_{i-1} \in S$:

$$\alpha = \prod_{i: P_i \in S} \frac{q(P_i, P_{i+1})}{\bar{q}(P_i, P_{i+1})}$$

$\beta$ has a similar expression. Finally, we have

$$\begin{aligned}\frac{d(S \cup T)}{d(\emptyset)} &= \prod_{i: P_i \in S \cup T} \frac{q(P_i, P_{i+1})}{\bar{q}(P_i, P_{i+1})} \\ &= \left(\prod_{i: P_i \in S} \frac{q(P_i, P_{i+1})}{\bar{q}(P_i, P_{i+1})}\right)\left(\prod_{i: P_i \in T} \frac{q(P_i, P_{i+1})}{\bar{q}(P_i, P_{i+1})}\right) \\ &= \alpha\beta\end{aligned}$$

Where the second equality is due to $S$ and $T$ being disjoint so that individual factors are partitioned into exactly one of the two products. □

**Theorem.** *$c(\cdot)$ from Equation 2 is submodular, i.e.,*

$$c(S \cup \{z_1\}) + c(S \cup \{z_2\}) \geq c(S \cup \{z_1, z_2\}) + c(S)$$

*Proof.* It suffices to show that $-d(\cdot)$ is submodular, since $c(\cdot)$ is 1 more than their sum. The lemma gives

$$d(S \cup \{z_1\}) = d(\emptyset)\gamma\alpha \qquad d(S \cup \{z_1, z_2\}) = d(\emptyset)\gamma\alpha\beta$$
$$d(S \cup \{z_2\}) = d(\emptyset)\gamma\beta \qquad d(S) = d(\emptyset)\gamma$$

So our goal is to show that

$$-d(\emptyset)\gamma(\alpha + \beta) \geq -d(\emptyset)\gamma(\alpha\beta + 1)$$

which is equivalent to

$$d(\emptyset)\gamma(1 - \alpha)(1 - \beta) \geq 0$$

which is true as $d(\emptyset), \alpha, \beta, \gamma \in [0, 1]$. □

## 5. The Algorithm

While our results our not quite good enough to guarantee a $\left(1 - \frac{1}{e}\right)$-approximation greedy algorithm, a natural modification performs competitively as demonstrated by our experiments. We now discuss the implementation.

### 5.1. Evaluating the objective

It is possible to use Equation 3 as a starting point for evaluating the expected length of a path given $(\pi, \rho_S)$, but a much more useful perspective takes a recursive form. For each $i \in [n + 1]$, let $\ell_i(S)$ be the expected path length if starting at $i$, and $\boldsymbol{\ell}(S) = (\ell_1(S), \ldots, \ell_{n+1}(S))$. Then

$$\ell_i(S) = \sum_{j=1}^{n+1} \rho_S(i, j) \cdot (1 + \ell_j(S))$$

which produces $n+1$ linear equations in $n+1$ variables:

$$\begin{aligned} \boldsymbol{\ell}(S) &= \left( \sum_{j=1}^{n+1} \rho_S(i, j) \cdot (1 + \ell_j(S)) \right)_{i=1}^{n+1} \\ &= \rho_S \cdot (\mathbb{1} + \boldsymbol{\ell}(S)) \end{aligned}$$

which implies $\boldsymbol{\ell}(S) = (I - \rho_S)^{-1}(\rho_S \mathbb{1})$, where $\rho_S \mathbb{1}$ is the sum of the rows of $\rho_S$. The final expected length $\ell(S) = \pi \cdot \boldsymbol{\ell}(S)$.

### 5.2. Greedy updates

Our greedy algorithm is a generalization of the original paper's. Since our objective is not monotonic, we additionally use an local maximum early-stopping procedure that merely returns the current set $S$ if there's no single node to add to $S$ that would improve expected path length. Otherwise, the algorithm behaves the same as the original: add the single node that would locally lead to the greatest improvement in expected path.

**Algorithm 1** Greedy solver

```
function GreedyLinking(Q, Q̄, π, k)
    Initialize S = ∅
    Compute c ← ℓ(∅)
    for i ← 1, ..., k do
        for z ∈ [n] ∖ S do
            Compute v(z) ← ℓ(S ∪ {z})
        end for
        if max_z v(z) < c then
            return S
        end if
        S ← S ∪ argmax_z v(z)
        c ← max_z v(z)
    end for
    return S
end function
```

As in the original paper, since we're constantly evaluating this $\boldsymbol{\ell}(S)$ that requires a weight matrix inverse, it is helpful to use an $LUP$ decomposition for $(I - \rho_S)$. Computing $L, U$, and $P$ is expensive, but once this decomposition is found, inversion is fast. However, since successive calls to $\ell(S)$ only differ in one row from an earlier call, we can efficiently update the decomposition after every step instead of recomputing the matrices or the inverse for a faster amortized time.

## 6. Experiments

### 6.1. Setup

To test our algorithm, we generated several datasets and compared the performance of our algorithm on two metrics against those of several baselines, heuristics, and even the original paper's. Because of the theoretical and model-centered nature of our project, we primarily focused on a family of synthetic datasets, for which experiment figures below show a representative trend across runs. We now describe our generation process in detail.

1. $\pi$ is drawn over $\mathrm{Dir}(\mathbf{1}^{n+1})$ i.e., 'uniformly' over categorical distributions over all the nodes including $\sigma$.

2. For each starting node $i$, $\bar{Q}_i$ is similarly drawn over $\mathrm{Dir}(\mathbf{1}^n)$ (with $\bar{q}_{i\sigma} = 0$) and is then scaled by a factor $d \sim \mathrm{Beta}(\alpha, \beta)$ where $\alpha, \beta$ are varied.

3. To determine $Q_i$ for each $i$, we start with $\bar{Q}_i$ and scale it down by some $s \sim \text{Beta}(\alpha', \beta')$ and draw $q_{i\sigma}$ uniformly from $[s - sd, s - sd + d]$, an interval chosen so that $1 \geq \sum_j q_{ij} \geq \sum_j \bar{q}_{ij} = 1 - d$.

For various settings of parameters $\alpha, \beta, \alpha', \beta'$, we plotted our metric and the original paper's metric as a function of $k$, the size limit of $S$, over choices of $S$ from our algorithm, their algorithm, and several baselines.

A random $k$-out-of-$n$ was just a simple blind baseline, but the other two were reasonable heuristics for the two metrics we tested. Selecting the pages with the highest $\sum_j (q_i j - \bar{q}_{ij})$ instead picks pages for which adding $\sigma$ would improve that page's engagement the most. Selecting the highest row sums is a kind of trying to pick the most popular or engrossing webpage to link back to $\sigma$, to increase $\sigma$'s influence (and hopefully help both chances of reaching $\sigma$ and expected path length).

Due to time—and to some extent the obscurity of their description of their own preprocessing procedure—we were not able to reproduce the results they reported on the semi-synthetic datasets generated from user data. However, our second figure that evaluates our algorithm on *their* metric shows that the same trends they report are prevalent even in our datasets, so that to a rather large extent the conclusions drawn from these experiments do not depend much on actual data.

### 6.2. Results

For our experiment, we ran our algorithm for all values of $k$ from 1 to 25, identical to Rosenfeld's setup. The results are plotted shown. Each data point plotted is an average of 50 different trials, with the value of $n$, the number of nodes, fixed at 200.

As shown in Figures 1 and 2, our algorithm, in optimizing for the expected path length, performs better than both Rosenfeld's algorithm and the different baselines. In fact, Rosenfeld's algorithm actually performs worse than one of our baselines that selects the highest row differences between $Q_i$ and $\bar{Q}_i$.

In contrast, our algorithm only performs *slightly* worse than Rosenfeld's algorithm in probability of reaching $\sigma$ under a budget of $k$ tags. Therefore, when comparing these two algorithms, the experimental results show that optimizing for expected path length is robust and generalizable, in that it still performs well when measuring the probability of reaching $\sigma$, but not the converse: optimizing for reaching $\sigma$ does not generalize to expected path length.

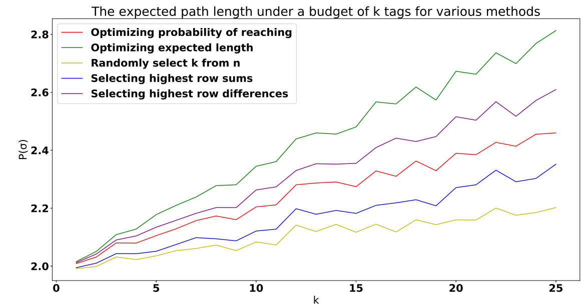


*Figure 1.* The expected path length under a budget of k tags for various methods

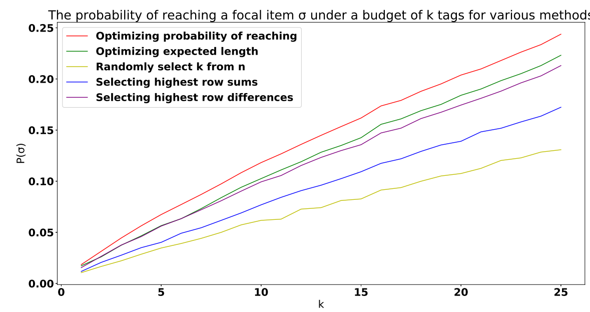


*Figure 2.* The probability of reaching a focal item $\sigma$ under a budget of k tags for various methods

## 7. Conclusions

Our project was driven by a motivation to investigate and improve on network traffic models. To this end, we looked at (Rosenfeld & Globerson, 2016), formulated alternative proofs to garner insight, and brainstormed ways to modify their problem to fit a different, possibly more common use-case while resembling the original setting enough to take advantage of their algorithms and results.

After coming up with, refining, and culling a few model modifications, we arrived at a variant with a different weight context and objective for which the problem is still NP-hard. This variant is also closer to real-world contexts, in that systems for link or tag recommendations are more incentivized to maximize overall engagement on their platform, rather than just likelihood of reaching a particular state.

While this objective turned out to be much more difficult to analyse and prove guarantees for—even with stronger setting restrictions—experiments have shown promise for our greedy algorithm, to the point that our performance is competitive even with the original paper's metric.

## Acknowledgments

We would like to thank our TAs, Jie Yuan and Da Tang, for their guidance throughout this project, as well as Professor Tony Jebara at Columbia University.

## References

Csáji, Balázs Csanád, Jungers, Raphaël M., and Blondel, Vincent D. Pagerank optimization in polynomial time by stochastic shortest path reformulation. In *ALT*, 2010.

Fang, Xiaomin, Pan, Rong, Cao, Guoxiang, He, Xiuqiang, and Dai, Wenyuan. Personalized tag recommendation through nonlinear tensor factorization using gaussian kernel. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pp. 439–445. AAAI Press, 2015. ISBN 0-262-51129-0.

Hotho, Andreas, Jschke, Robert, Schmitz, Christoph, and Stumme, Gerd. Folkrank: A ranking algorithm for folksonomies. In *Proc. FGIR 2006*, 2006.

Litvak, Nelly and Avrachenkov, Konstantin. The Effect of New Links on Google PageRank. Research Report RR-5256, INRIA, 2004.

Olsen, Martin, Viglas, Anastasios, and Zvedeniouk, Ilia. A constant-factor approximation algorithm for the link building problem. In Wu, Weili and Daescu, Ovidiu (eds.), *Combinatorial Optimization and Applications*, pp. 87–96, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-17461-2.

Rosenfeld, Nir and Globerson, Amir. Optimal tagging with markov chain optimization. In *Advances in Neural Information Processing Systems*, pp. 1307–1315, 2016.