Surfing the Blogosphere: Optimal Personalized Strategies for Searching the Web

Stratis Ioannidis, Laurent Massoulié

Thomson Technical Report
Number: **CR-PRL-2009-07-0001**
First Publication Date: Dec 21st, 2009

Abstract: We propose a distributed mechanism for finding websurfing strategies that is inspired by the StumbleUpon recommendation engine. Each day, a websurfer visits a sequence of websites recommended by our mechanism, and selects one that matches her daily interests. We formally show that even with this minimal feedback from the surfer—the selected website—our mechanism finds a websurfing strategy that matches the surfer's interests optimally. The surfer does not need to know—or declare—what her daily interests are before she is presented with content she likes. Moreover, our mechanism is content-agnostic: it is oblivious to the nature of the content the surfer selects.

In addition, we study how the performance of this mechanism can be improved if surfers with similar interests share their feedback. Such surfers can be found indirectly, *e.g.*, if they are all registered as friends in a social networking application. Our analysis characterizes the improvement in the mechanism's accuracy, based on the size of the group and the degree of similarity between the surfers' interests. In particular, we show that sharing feedback can significantly accelerate the convergence of our mechanism. Our results are derived analytically using stochastic approximation techniques, but are also validated through a numerical study.

THOMSON

1. INTRODUCTION

An increasing number of people use the Internet, as opposed to more traditional mass media, to learn about daily events. However, the sheer volume, diversity and varying quality of content on the web—ranging from personal blogs to professional news-sites—makes searching for current, interesting information a challenging task. As a result, online recommendation engines (such as, *e.g.*, Delicious[1], Digg[2], and Reddit[3]) giving daily suggestions to websurfers have recently proliferated.

The fundamental question the above recommendation engines are trying to answer, which also motivates the work presented in this paper, is the following: *on a given day, which websites should a websurfer visit in order to find content that she is interested in*? A challenge posed by the above problem is that both the interests of websurfers and the content published by websites may change from one day to the next. Moreover, eliciting a websurfer's interests is not always straightforward: the websurfer may not be able to declare what type of content she is interested in before viewing such content. For this reason, we would like to address the above question without requiring a-priori knowledge of the websurfer's interests on a given day.

To address this challenge, we adopt the approach employed by the recommendation engine StumbleUpon[4]. StumbleUpon implements a toolbar, giving websurfers the option between two buttons (see Fig. 1). The first, labeled "Stumble!", generates a randomly selected website. The second, labeled "I like this" allows surfers to indicate they like the website they see. Websurfers are thus presented with a sequence of different websites until one that matches their interests is found.

The exact algorithm used by Stumbleupon to recommend websites to its users is proprietary. In broad terms [1], StumbleUpon's server learns through a registered user's feedback which websites she likes; recommendations are then made by showing the user websites that have been recommended by other registered users with similar preferences. StumbleUpon also leverages *social-networking information* to make recommendations. In particular, registered users can declare their friends. Assuming that friends should have common interests, StumbleUpon exploits this information to recommend websites to users when selected by their friends. Through both of these techniques, StumbleUpon personalizes the search for content to a particular user. Moreover, it does so with minimal feedback; users need only declare which sites they like and, optionally, provide social networking information.

In this paper, we propose a distributed mechanism for determining which websites a surfer should visit that is inspired by StumbleUpon. In particular, our mechanism presents a sequence of websites to the surfer, and all a surfer needs to declare is whether she approves the websites shown to her. Like StumbleUpon, the websurfer need not know—nor explicitly declare—what her interests are before actually being presented with a website she likes. Our mechanism is *content-agnostic*: it is oblivious to the actual nature of the content shown to the websurfer by the websites she visits. Finally, contrary to StumbleUpon, our mechanism is distributed: no centralized data collection or processing is required.

Figure 1: The StumbleUpon toolbar. The websurfer is presented with a new website whenever she clicks the "Stumble!" button. If she finds the website shown interesting, she can register her approval by clicking "I like this".

Our first contribution is to show that, in spite of its simplicity, our mechanism computes a *surfing strategy*, *i.e.*, a strategy for visiting websites, that matches the websurfer's interests to the content presented by websites in an optimal way. More specifically, the surfing strategy produced by our mechanism maximizes a (non-increasing) function the number of websites the websurfer views before finding interesting content.

Our second contribution is to show that the performance of this mechanism can be significantly improved if feedback is aggregated among surfer communities. In particular, if surfers with similar interests share their feedback, the accuracy of the mechanism and the rate of its convergence can be significantly improved. To the best of our knowledge, this is the first time that the performance of such a mechanism is formally shown to improve through the use of community aggregated feedback.

The remainder of this technical report is organized as follows. In Section 2 we present related work in this area. In Section 3 we describe our proposed mechanism in detail. In Section 4 we state, without proofs, our main results regarding the performance of our mechanism; the full analysis can be found In Section 5. Section 6 provides a method for reducing the dimension of our problem. Finally, we validate our results through a numerical study in Section 7, and conclude in Section 8.

2. RELATED WORK

An altogether different approach to searching for content is adopted by search engines like, *e.g.*, Google or Blogscope [2]. Contrary to our scheme, search engines are centralized and content-aware: they require considerable storage and computational resources to store and process the web's contents. In addition, they respond to query submissions and, as such, cannot provide recommendations when a websurfer is not a-priori aware of what content interests her.

Surfers using the Delicious recommendation engine tag blogs with keywords; tagged content then appears on the Delicious website as an aid to daily websurfing. This approach relies on the participation of websurfers through tag submission, which they may not always be able or willing to perform; Digg and Reddit avoid this by letting surfers vote for websites they visit and displaying recent content with high voting scores. Such approaches are centralized but also lack *personalization*: in contrast, our scheme matches surfers' interests individually.

[1] http://delicious.com
[2] http://digg.com
[3] http://reddit.com
[4] http://www.stumbleupon.com

El-Arini et al. [3] propose a mechanism for recommending a list of websites to surfers that maximizes a personalized coverage function, indicating whether the suggested websites cover topics of interest to the surfer. A similar minimal feedback scheme as ours is employed, based on the approval and disapproval of suggested sites. However, the mechanism in [3] is not content-agnostic. It requires prior knowledge of the probability that a topic is covered by a proposed website, which, as in Google and Blogscope [2], is obtained by periodically crawling the web and processing the collected data in a centralized manner. Our work also differs by considering aggregation of feedback over communities.

Our mechanism has strong ties to collaborative filtering (see, *e.g.*, [4, 5]), which typically involves clustering users and making recommendations based on the suggestions of other cluster members. Our work is orthogonal: we are not concerned with whether communities are obtained through clustering or, *e.g.*, a social networking application. Irrespectively of how the community was formed, we formally characterize the performance gains of aggregating feedback over the community, in terms of its size and the degree of similarity between its members.

3. SYSTEM DESCRIPTION

3.1 Website Content and Topic Coverage

We model the web as a set of N websites. These websites maintain content that covers M different topics, such as, *e.g.*, sports, politics, movies, health, *etc.* Periodically, the content displayed at each website changes: we denote by $p_{w,f}$ the probability that the content presented in site w covers topic f, where $w = 1, \dots, N$ and $f = 1, \dots, M$. For each website w, we call the vector $\vec{p}_w = [p_{w,1}, \dots, p_{w,M}]$ as the *publishing strategy* of the website. In general, more than one topic (*e.g.*, sports *and* health) can be covered by a certain website at a given point in time; the expected number of topics covered by w will be $\sum_f p_{w,f}$.

We assume that for every topic f there exists a website w such that $p_{w,f} > 0$, *i.e.*, w covers f with positive probability. We also make the assumption that websites have high churn and their content changes frequently (as, *e.g.*, in news-sites like `nytimes.com` or popular blogs like `huffingtonpost.com`). As a result, we will assume that each time a website w is visited its current content is independent from content viewed during previous visits.

3.2 Surfer Interests and a Minimal Feedback Mechanism

We assume that each day a given websurfer is interested in topic f with some probability d_f, where $\sum_f d_f = 1$. We call $\vec{d} = [d_1, \dots, d_M]$ the *interest profile* of the surfer. The surfer visits websites daily by using a mechanism that works as follows: the mechanism recommends a sequence of websites to the surfer, and the surfer keeps viewing these sites until she finds a website covering the topic that interests her.

Like StumbleUpon, this mechanism can be implemented as a toolbar on the surfer's web-browser: the surfer would be presented with two different buttons, one called "next" and the other called "approve". Clicking the first would generate a new website, while clicking the second would indicate that a topic the surfer is interested in is found. However, unlike Stumbleupon, our mechanism is fully decentralized: the toolbar relies only on the surfer's feedback and there is no need for centralized data collection.

At each click of the "next" button, the website recommended is chosen by the mechanism according to a probability distribution over all sites: we denote by x_w the probability that the mechanism recommends website w, where $\sum_w x_w = 1$. We refer to the vector $\vec{x} = [x_1, \dots, x_N]$ as the *surfing strategy* of the mechanism. As we will see, the mechanism will use the surfer's feedback to update this strategy from one day to the next, in a manner we discuss in detail below.

3.3 Surfer Optimization

Let Y be the number of sites the surfer visits until it locates a topic it is interested in. Moreover, let $R(Y)$ be a function rating the performance of the system, given that the topic is found within Y steps. We treat R as a system parameter internal to the mechanism, and make the following assumption:

Assumption 1. *The rating function* $R : \mathbb{N} \to \mathbb{R}$ *is (a) non-increasing and (b) summable,* i.e., $\sum_{k=1}^{\infty} |R(k)| < \infty$.

Considering non-increasing rating functions is natural: the longer a surfer has to wait, the lower the system's rating should be. The assumption on the summability is technical; note that it holds if R has a finite support and, *e.g.*, visiting more than 30 websites gives a zero rating to the system.

Denote by $\mathbb{E}_{\vec{x}}[R(Y)]$ the expected system rating when the surfing strategy is $\vec{x}$. Ideally, assuming that the surfer's interest profile $\vec{d}$ and the website publishing strategies $\vec{p}_w$ are fixed, the mechanism recommending sites to the surfer should use a surfing strategy $\vec{x}$ that is a solution to the following problem:

Surfer Optimization

$$\text{Maximize:} \quad F(\vec{x}) = \mathbb{E}_{\vec{x}}[R(Y)], \tag{1a}$$

$$\text{subject to:} \quad \sum_w x_w = 1, \quad \text{and} \quad x_w \geq \epsilon, \ \forall w, \tag{1b}$$

for some $0 \leq \epsilon \leq \frac{1}{N}$. Intuitively, given that the rating of the system decreases with the number of visits the surfer makes, the mechanism should choose a surfing strategy that maximizes the expected rating by making the number of visits sufficiently small.

The optimization problem described by (1) is restricted to surfing strategies in which *every site is visited with probability at least* ϵ, *i.e.*, the feasible domain is restricted to

$$D_\epsilon = \{\vec{x} : \sum_w x_w = 1 \text{ and } x_w \geq \epsilon\}. \tag{2}$$

Note that $D_\epsilon \neq \emptyset$ iff $\epsilon \leq \frac{1}{N}$. Moreover, if $\epsilon < \epsilon'$, then $D_{\epsilon'} \subseteq D_\epsilon \subseteq D_0$, where D_0 includes all possible surfing strategies.

The main reason to restrict the domain to D_ϵ, where $\epsilon > 0$, has to do with website discovery—we want to ensure the websurfer has a chance to visit a website she likes. It is also useful for "tracking" [6] changes in $\vec{d}$ or $\vec{p}_w$: if a website is never visited, the mechanism will never have the opportunity to adapt to the fact that, *e.g.*, it has started covering sports, and thereby may now meet the interests of the surfer. We note that, if $\epsilon > 0$ is small, an optimal strategy in D_ϵ will not be much worse than the optimal strategy in D_0 (see also Lemma 3).

3.4 Updating the Surfing Strategy

The surfing strategy $\vec{x}$ can change from one day to the next based only on the minimal feedback given by the surfer; we denote by $\vec{x}(k)$, $k = 1, 2, \ldots$, the strategy on day k. Let

$$\Pi_{D_\epsilon}(\vec{x}) = \arg\min_{\vec{y} \in D_\epsilon} \|\vec{y} - \vec{x}\|_2, \quad \vec{x} \in \mathbb{R}^N \tag{3}$$

be the orthogonal projection of $\vec{x}$ to the domain D_ϵ. Since D_ϵ is closed and convex, Π_{D_ϵ} is well defined. Moreover, there are known algorithms (*e.g.*, [7, 8]) computing Π_{D_ϵ} in $O(N)$ time.

Based on the feedback of the surfer on the k-th day, the strategy $\vec{x}(k)$ is updated according to the following scheme:

$$\vec{x}(k+1) = \Pi_{D_\epsilon}\left(\vec{x}(k) + \gamma(k) \cdot \vec{g}(k)\right), \quad k = 1, 2, \ldots \tag{4}$$

The term $\gamma(k) \in \mathbb{R}_+$ is a gain factor; if it decreases with k, feedback given later has smaller impact on the surfing strategy. The term $\vec{g}(k) \in \mathbb{R}^N$ is a vector forcing the change on the surfing strategy based the surfer's feedback. In particular, $\vec{g}(k)$ has the following coordinates:

$$g_w(k) = \begin{cases} -\frac{Y(k)\Delta R(Y(k)+1)}{x_w(k)}, & \text{if the surfer selects } w \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

where $Y(k)$, $k = 1, 2, \ldots$ is the number of sites a surfer visits on the k-th day until it locates a topic she is interested in, and

$$\Delta R(k) = \begin{cases} R(1), & \text{if } k = 1, \\ R(k) - R(k-1), & \text{if } k > 1. \end{cases} \tag{6}$$

Note that, to compute $\vec{g}$, all the mechanism needs to know is Y, R and $\vec{x}$. The rating function R is a system parameter, while $\vec{x}$ is maintained by the mechanism and used to recommend websites. To compute Y, the mechanism needs to keep track of how many websites it has shown to the surfer; the only feedback given by the surfer is the indication that her interests have been met. In particular, the mechanism does not need to know—and the surfer does not need to declare—which is the interesting topic covered by website w. In fact, the mechanism is oblivious to even what the topics $f = 1, \ldots, M$ are.

3.5 Exploiting Surfer Communities

In our analysis, apart from considering the mechanism (4) when the surfer is isolated, we will also consider the case when the surfer is part of a surfer community $C = \{1, \ldots, |C|\}$. Each surfer $s \in C$ has its own interest profile $\vec{d}_s$. Each day the topic it is interested in is chosen independently of the topics that interest other surfers. Furthermore, each surfer executes the minimal feedback mechanism outlined in the previous section. The difference from the single-surfer case will be that these mechanisms will exploit the existence of the surfer community in the manner we describe below.

To begin with, all mechanisms implement a common surfing strategy $\vec{x}$ everywhere in the community. Moreover, at the end of each day, the feedback collected from each surfer is communicated among the mechanisms and is aggregated; the aggregated feedback is then used to update the common strategy.

More specifically, let $Y_s(k)$, $k = 1, 2, \ldots$ be the number of sites surfer $s \in C$ visits on the k-th day and $\vec{g}_s(k)$ be the change vector induced by the feedback of surfer s on day k, given by (5). The latter quantities are computed individually by each mechanism and, at the end of each day, are communicated among all surfers in C.

As in (1a), we denote by $F_s(\vec{x}) = \mathbb{E}_{\vec{x}}[R(Y_s)]$ the expected rating of the mechanism at surfer s. Moreover, we denote by

$$\vec{g}_C = \frac{1}{|C|}\sum_{s=1}^{|C|} \vec{g}_s \quad \text{and} \quad F_C(\vec{x}) = \frac{1}{|C|}\sum_{s=1}^{|C|} F_s(\vec{x}) \tag{7}$$

the average change vector and the average system rating among all surfers in the community.

The common surfing strategy is then updated as follows:

$$\vec{x}(k+1) = \Pi_{D_\epsilon}\left(\vec{x}(k) + \gamma(k) \cdot \vec{g}_C(k)\right), \; k = 1, 2, \ldots \tag{8}$$

In other words, the common surfing strategy is updated as in (4), the only difference being that, instead of the individual vectors $\vec{g}_s$, the community average $\vec{g}_C$ is used.

We will denote by δ_C the maximum l_1 distance between the interest profiles of surfers in the comminity, *i.e.*,

$$\delta_C = \max_{s,s' \in C} \left\| \vec{d}_s - \vec{d}_{s'} \right\|_1 = \max_{s,s' \in C} \sum_w |d_{s,w} - d_{s',w}|. \tag{9}$$

We will call δ_C the *diameter* of the community. Intuitively, we will be interested in the case where the diameter is small, as surfers in the same community should have similar interests.

3.6 Discussion on our Assumptions

The case where a surfer is interested in any of k different topics each day can easily be reduced to the single-topic case, by redefining the set of topics to be the set of topic k-tuples. Interestingly, even if our model is extended this way, mechanism (4) would remain unaltered: this is because, as noted, it is entirely oblivious to the nature of the topics f.

The assumption that websites display independent content across visits would not apply to websites that are updated infrequently. In practice, our mechanism should avoid visiting such sites more than once during a search, *e.g.*, through blacklisting them if the surfer does not approve them.

Our mechanism maintains a vector $\vec{x}$ whose dimension is N, the number of websites. Given the number of websites on the Internet, this is clearly impractical. Restricting $\vec{x}$ to a subset of all websites is not satisfactory, as it would limit the surfer's choices. In Section 6, we provide a more appealing solution to this problem by defining surfing strategies over *groups* of websites.

4. MAIN RESULTS

In this section we state our main results regarding the convergence of mechanism (4) to an optimal surfing strategy and the effect of aggregating community feedback through mechanism (8). We first state our results without proofs; the full analysis can be found in Section 5.

4.1 Convergence to an Optimal Surfing Strategy

Our first main result is to show that the minimal feedback mechanism we propose indeed computes an optimal surfing strategy. This is established for two different cases: one where the gain factors $\gamma(k)$ converge to zero, and one where they remain constant.

To begin with, by appropriately choosing $\gamma(k)$, we can guarantee that the mechanism converges to an optimal strategy, with probability one.

THEOREM 1 (CONVERGENCE W.P.1). *Let $\vec{x}(k)$ be updated by the mechanism* (4)*, where $\epsilon > 0$, $\gamma(k) = 1/k$ and Assumption 1 holds. Then $\lim_{k\to\infty} F(\vec{x}(k)) = \sup_{\vec{x}\in D_\epsilon} F(\vec{x})$, w.p.*1.

In other words, if the surfing strategy is updated according to our minimal feedback mechanism with $\gamma(k) = \frac{1}{k}$, the surfing strategy will eventually become optimal, *i.e.*, it will be a solution to the SURFER OPTIMIZATION problem.

Using decreasing gains can make our mechanism non-robust: if either the interest profile $\vec{d}$ of the surfer or the publishing strategies $\vec{p}_w$ change, and $\gamma(k)$ has effectively become zero, our mechanism will not be able to adapt. To be able to "track" [6] such changes, it is interesting to keep $\gamma(k)$ bounded away from zero.

In such a case, convergence to an optimal strategy cannot be guaranteed. However, one expects that if γ is small enough (4) does not deviate too far such a strategy. The following theorem implies that this intuition is indeed correct[5]:

THEOREM 2 (CONVERGENCE IN PROBABILITY). *Let $\vec{x}(k)$ be updated by the mechanism* (4)*, where $\epsilon > 0$, $\gamma(k) = \gamma > 0$ and Assumption 1 holds. Consider the continuous-time interpolated process $\vec{\xi}(t) = \vec{x}(\lfloor t/\gamma \rfloor)$, $t \in \mathbb{R}_+$. Then,*

$$\lim_{\gamma\to 0}\lim_{T\to\infty}\frac{1}{T}\int_{t=0}^{T} F(\vec{\xi}(t))dt = \sup_{\vec{x}\in D_\epsilon} F(\vec{x}), \quad \textit{in probability.}$$

In other words, by taking γ sufficiently small, we can make the time-average rating to be arbitrarily close to the one achieved by an optimal surfing strategy.

4.2 Exploiting Surfer Communities

Having established that our minimal feedback mechanism indeed computes an optimal surfing strategy, we look at its behavior if surfers form a community and implement a common strategy, updated according to (8).

Our first result is to show that, if the surfers have similar interest profiles, the mechanism (8) converges to a strategy that is not too far from each surfer's optimal surfing strategy.

THEOREM 3. *Let $\vec{x}(k)$ be updated by the mechanism* (8)*. Then, Theorems 1 and 2 hold if F is replaced by the average rating F_C, given by* (7)*. Moreover, under Assumption 1, if δ_C is the diameter of the community, then, for all $s \in C$,*

$$\left|\sup_{\vec{x}\in D_\epsilon} F_C(\vec{x}) - \sup_{\vec{x}\in D_\epsilon} F_s(\vec{x})\right| = O(\delta_C). \quad (10)$$

The theorem states that mechanism (8) maximizes the average rating among all surfers in the community, as opposed to each F_s individually. Most importantly, if surfers have similar interests (*i.e.*, the community diameter is small), the rating of the optimal strategy of an individual surfer will not be too far from the one achieved by algorithm (8).

On the other hand, receiving feedback from the entire community as in (8) can give a significantly improved performance over (4), as illustrated by the following theorem.

[5]Recall that a sequence of random variables X_n converges to X in probability when $n \to \infty$, for all $\delta > 0$, $\lim_{n\to\infty} \mathbf{P}(|X_n - X| > \delta) = 0$.

THEOREM 4. *Assume that the objective F_s of some $s \in C$ and the community average F_C have unique maxima at $\vec{x}^*_s$, $\vec{x}^*_C$, respectively, and that both are located in the relative interior of D_ϵ. Moreover, assume that $k^2|R(k)|$ is summable, and that the Hessian of F_s at $\vec{x}^*_s$ is invertible.*

If surfer s runs mechanism (4) *with $\epsilon > 0$ and $\gamma(k) = \gamma$, and $\vec{\xi}(t)$ is the continuous-time interpolated process $\vec{\xi}(t) = \vec{x}(\lfloor t/\gamma \rfloor)$, $t \in \mathbb{R}_+$, then*

$$\lim_{\gamma\to\infty}\lim_{T\to\infty}\mathbb{E}[(\|\vec{\xi}(T) - \vec{x}^*_s\|_2)^2/\gamma] = c, \quad (11)$$

where c some positive constant. If, on the other hand, s runs mechanism (8) *with the same $\epsilon > 0$ and $\gamma(k) = \gamma$, then, for small enough δ_C,*

$$\lim_{\gamma\to\infty}\lim_{T\to\infty}\mathbb{E}[(\|\vec{\xi}(T) - \vec{x}^*_C\|_2)^2/\gamma] = \frac{c}{|C|} + O(\sqrt{\delta_C}), \quad (12)$$

where $|C|$ the size of the community and δ_C the community diameter.

Recall from Theorem 2 that, if $\gamma(k) = \gamma$, (4) does not converge to a maximizer of F_s; Eq. (11) then states that, in steady state, its expected distance from the maximizer $\vec{x}^*_s$ will be of the order of $\sqrt{c\gamma}$, where c a constant. If, on the other hand, the surfing strategy is updated according to (8), Eq. (12) states that its expected distance from $\vec{x}^*_C$ will be of the order of $\sqrt{\frac{c\gamma}{|C|} + O(\sqrt{\delta_C})}$. Hence, if δ_C is small enough, *sharing feedback with the community decreases the expected distance from the maximizer by a factor of $\sqrt{|C|}$*. Keeping in mind that, from Theorem 3, the $F_s(\vec{x}^*_s)$ is not too far from $F(\vec{x}^*_C)$ (see also Corollary 1) Theorem 4 establishes that, for small enough community diameters, exchanging feedback among surfers in the community can have a significant improvement on the accuracy of the mechanism.

An alternative interpretation of Theorem 4 can be given in terms of speed of convergence. As the distance of (4) from $\vec{x}^*$ is of the order of $\sqrt{c\gamma}$, an obvious method for increasing the accuracy is by decreasing the gain factor γ. However, the smaller γ is, the smaller the "jumps" (4) makes and the longer it will take to get close to the maximizer. Theorem 4 therefore implies that, if δ_C is small, (8) can achieve the same accuracy as (4) but much faster, by using a greater γ.

Finally, Theorem 4 suggests an inherent trade-off appearing when the size of the community increases: the larger $|C|$ is, the smaller the first term of eq. (12) will be, indicating that obtaining feedback from many users will reduce the "noise" in our mechanism. On the other hand, increasing the community can also increase the community diameter δ_C, thus increasing the second term in (12). In this sense, depending on how δ_C relates to $|C|$, a maximal community size might exist, beyond which adding more surfers will not improve performance. We further investigate this in our numerical study in Section 7.

5. ANALYSIS

5.1 Convexity, Differentiability, and Near-Optimality

Our analysis will exploit the fact that SURFER OPTIMIZATION, described by (1), is a convex optimization problem. In particular, the expected rating is a concave function of the surfing strategy, as indicated by the following lemma.

LEMMA 1. *Under Assumption 1, for all $\epsilon \geq 0$, the objective function $F : D_\epsilon \to \mathbb{R}_+$, given by* (1a)*, is concave.*

PROOF. Recall that Y is the number of sites a surfer visits until it locates a topic it is interested in. Moreover, let Y_f the number of websites a surfer visits conditioned on the event that the topic it is interested in is f. Then, Y_f is a geometric random variable, *i.e.*,

$$\mathbf{P}(Y_f = \ell) = \rho_f (1 - \rho_f)^{\ell-1}, \quad \ell \geq 1, \tag{13}$$

where

$$\rho_f = \sum_w p_{w,f} x_w \tag{14}$$

is the probability that topic f is found in one step. As a result, F can be written as

$$F(\vec{x}) = \sum_{f=1}^{M} d_f \mathbb{E}_{\vec{x}}[R(Y_f)] \tag{15}$$

where $\mathbb{E}_{\vec{x}}[R(Y_f)] = \sum_{k=1}^{\infty} R(k)\rho_f(1-\rho_f)^{k-1}$. Observe that Assumption 1 immediately implies that

$$R \text{ is non-negative, bounded, and } \lim_{k\to\infty} R(k) = 0. \tag{16}$$

If $\mathbb{E}_{\vec{x}}[R(Y_f)]$ is a concave function of $\rho_f \in [0,1]$ then it is also a concave function of $\vec{x}$, as a composition of a concave and a linear function. The lemma then follows, as $F(\vec{x})$ would be concave as the sum of concave functions, by (15). If $\rho_f > 0$,

$$\mathbb{E}_{\vec{x}}[R(Y_f)] = \sum_{k=1}^{\infty} R(k)\rho_f(1-\rho_f)^{k-1} \overset{(6)}{=} \sum_{k=1}^{\infty}\sum_{\ell=1}^{k} \Delta R(\ell)\rho_f(1-\rho_f)^{k-1}$$

$$= \sum_{\ell=1}^{\infty}\sum_{k=\ell}^{\infty} \Delta R(\ell)\rho_f(1-\rho_f)^{k-1} = \sum_{\ell=1}^{\infty} \Delta R(\ell)(1-\rho_f)^{\ell-1} \tag{17}$$

Note that, by Assumption 1, $\Delta R(\ell) \leq 0$ for $\ell > 1$, while $\Delta R(1) = R(1) \geq 0$. As $(1-\rho_f)^{\ell-1}$ is a convex, decreasing function of $\rho_f \in [0,1]$ for all $\ell > 1$ and constant for $\ell = 1$, $\mathbb{E}_{\vec{x}}[R(Y_f)]$ is concave for $\rho_f \in (0,1]$, as the sum of concave functions and the constant $R(1)$; it is also continuous in $(0,1]$, by the monotone convergence theorem. By the same theorem, $\lim_{\rho_f\to 0}\sum_{k=1}^{\infty}\mathbb{E}_{\vec{x}}[R(Y_f)] = \sum_{k=1}^{\infty}\Delta R(k) = \lim_{\ell\to\infty} R(\ell) \overset{(16)}{=} 0$. Hence, $\mathbb{E}_{\vec{x}}[R(Y_f)]$ is continuous in all of $[0,1]$. As it is concave in $(0,1]$, continuity implies concavity on $[0,1]$. □

We can also compute the gradient of F.

LEMMA 2. *Under Assumption 1, for all $\epsilon \geq 0$, the objective function $F : D_\epsilon \to \mathbb{R}_+$ is continuously differentiable and*

$$\frac{\partial F}{\partial x_w} = \sum_f d_f p_{w,f} \cdot \begin{cases} -\frac{1}{\rho_f}\mathbb{E}[Y_f \Delta R(Y_f+1)], & \text{if } \rho_f > 0 \\ \sum_{k=1}^{\infty} R(k), & \text{if } \rho_f = 0, \end{cases} \tag{18}$$

where $\rho_f = \sum_f p_{w,f} x_w$.

PROOF. By (14), $\frac{\partial \mathbb{E}_{\vec{x}}[R(Y_f)]}{\partial x_w} = \frac{\partial \mathbb{E}[R(Y_f)]}{\partial \rho_f} \cdot p_{w,f}$. For ρ_f in $(0,1]$ we have by (17) that

$$\frac{\partial \mathbb{E}_{\vec{x}}[R(Y_f)]}{\partial \rho_f} = -\sum_{\ell=1}^{\infty}(\ell-1)\Delta R(\ell)(1-\rho_f)^{\ell-2}$$

$$\overset{k\equiv\ell-1}{=} -\sum_{k=1}^{\infty} k\Delta R(k+1)(1-\rho_f)^{k-1} = -\frac{\mathbb{E}[Y_f \Delta R(Y_f+1)]}{\rho_f} \tag{19}$$

This is continuous in $(0,1]$ by the monotone convergence theorem. By the same theorem,

$$\lim_{\rho_f\to 0}\frac{\partial \mathbb{E}_{\vec{x}}[R(Y_f)]}{\partial \rho_f} = -\sum_{k=1}^{\infty} k\Delta R(k+1) = \sum_{k=1}^{\infty} k[-\Delta R(k+1)]$$

$$= \sum_{k=1}^{\infty}\sum_{\ell=1}^{k} -\Delta R(k+1) = \sum_{\ell=1}^{\infty}\sum_{k=\ell}^{\infty} R(k) - R(k+1) = \sum_{\ell=1}^{\infty} R(\ell)$$

By (16), $\mathbb{E}_{\vec{x}}[R(Y_f)] = 0$ for $\rho = 0$. Hence,

$$\frac{\partial \mathbb{E}_{\vec{x}}[R(Y_f)]}{\partial \rho_f}\Big|_0 = \lim_{\rho_f\to 0}\frac{\mathbb{E}_{\vec{x}}[R(Y_f)](0) - \mathbb{E}_{\vec{x}}[R(Y_f)](\rho_f)}{\rho_f}$$

$$= \lim_{\rho_f\to 0}\frac{\sum_{k=1}^{\infty} R(k)\rho_f(1-\rho_f)^{k-1}}{\rho_f} = \sum_{k=0}^{\infty} R(k)$$

I.e., $\frac{\partial \mathbb{E}[R(Y_f)]}{\partial \rho_f}$ is continuous in $[0,1]$. □

An immediate implication is that F is Lipschitz continuous:

COROLLARY 1. *Under Assumption 1, for all $\epsilon \geq 0$, the objective function $F : D_\epsilon \to \mathbb{R}_+$ is Lipschitz continuous. In particular,*

$$|F(\vec{x}) - F(\vec{y})| \leq L\|\vec{x} - \vec{y}\|_1, \qquad \text{for all } \vec{x}, \vec{y} \in D_0,$$

where $L = \sum_{k=1}^{\infty} R(k) < \infty$. Lipschitz continuity in other norms follows by norm equivalence in $\mathbb{R}^N$.

PROOF. Observe that, since F is differentiable, by the mean value theorem $F(\vec{x}) - F(\vec{y}) = \nabla F(\vec{z})(\vec{x} - \vec{y})$ where $\vec{z} = \theta\vec{x} + (1-\theta)\vec{y} \in D_0$ for some $\theta \in [0,1]$. Thus, by Hölder's inequality $|F(\vec{x}) - F(\vec{y})| \leq \|\nabla F(\vec{z})\|_\infty \|\vec{x} - \vec{y}\|_1$. By (19), $\frac{\partial \mathbb{E}_{\vec{x}}[R(Y_f)]}{\partial \rho_f} \leq L$. Hence, $\|\nabla F(\vec{z})\|_\infty = \max_w \sum_f d_f p_{w,f} \frac{\partial \mathbb{E}_{\vec{x}}[R(Y_f)]}{\partial \rho_f} \leq L$ as $p_{w,f} \leq 1$ and $\sum_f d_f = 1$. □

The Lipschitz continuity of the objective function F can be used to understand the dependence of the optimal surfing strategy on ϵ. In particular, if ϵ is small enough, one does not lose too much by requiring that all sites are visited with probability at least ϵ. This is stated formally in the following lemma, whose proof can be found in Appendix A.

LEMMA 3. *Under Assumption 1, for any $0 \leq \epsilon \leq 1/N$, $\left|\sup_{\vec{x}\in D_\epsilon} F(\vec{x}) - \sup_{\vec{x}\in D_0} F(\vec{x})\right| = O(\epsilon)$.*

5.2 Estimating the Gradient

Having established the convexity and differentiability of F, we turn our attention to mechanism (4) and how it relates to the SURFER OPTIMIZATION problem. The following result states that the vector $\vec{g}$, given by (5), capturing the change of the surfing strategy based on the surfer's feedback, is in fact an unbiased estimator of the gradient ∇F. Hence, the mechanism (4) is a *stochastic approximation* algorithm [6], that solves SURFER OPTIMIZATION through a "randomized" gradient descent.

LEMMA 4. *Under Assumption 1, given that the surfing strategy of the surfer is $\vec{x}(k)$, the vector $\vec{g} \in \mathbb{R}^N$, can be written as*

$$\vec{g}(k) = \nabla F(\vec{x}(k)) + V(\vec{x}(k)) \tag{20}$$

where $\mathbb{E}[V(\vec{x}(k))]$ is the zero vector and $\mathbb{E}[\|V(\vec{x}(k))\|_2^2] \leq \frac{N}{\epsilon}\sum_{k=1}^{\infty}(k\Delta R(k))^2 < \infty$, for $\epsilon > 0$.

PROOF. Let E_f be the the site at which the topic is found, given that the topic sought for is f. Then, $\mathbf{P}(E_f = w) = x_w p_{w,f}/\rho_f$, where ρ_f is given by (14). Moreover, E_f and Y_f are independent as

$$\mathbf{P}(Y_f = \ell \cap E_f = w) \overset{(13),(14)}{=} (1-\rho_f)^{\ell-1} \cdot x_w p_{w,f}$$
$$= \rho_f (1-\rho_f)^{\ell-1} \cdot x_w p_{w,f}/\rho_f = \mathbf{P}(Y_f = \ell) \cdot \mathbf{P}(E_f = w) \quad (21)$$

By (5), we have that

$$\mathbb{E}[g_w] = -\sum_f d_f \frac{p_{w,f}}{\rho_f} \mathbb{E}[Y_f \Delta R(Y_f+1) \,|\, E_f = w] \overset{(21),(18)}{=} \frac{\partial F}{\partial x_w}$$

By (21), $\mathbb{E}[g_w(\vec{x})^2] = -\sum_f \frac{d_f p_{w,f} \mathbb{E}\left[\left(Y_f \Delta R(Y_f+1)\right)^2\right]}{\rho_f x_w}$ where $\mathbb{E}\left[(Y_f \Delta R(Y_f+1))^2\right]/\rho_f \leq C = \sum_{k=1}^{\infty} (k \Delta R(k))^2 < \infty$, as $\sum_{k=1}^{\infty} k \Delta R(k+1) = \sum_{k=1}^{\infty} R(k) < \infty$. As $p_{w,f} \leq 1$, $\sum d_f = 1$ and $x_w \geq \epsilon$ for all $\vec{x} \in D_\epsilon$, we have $\mathbb{E}[|g_w(\vec{x})|^2] \leq \frac{C}{\epsilon}$. The lemma thus follows as $\mathbb{E}[\|V(\vec{x})\|_2^2] = \sum_w \mathbb{E}[g_w(\vec{x})^2] - \sum_w \mathbb{E}[g_w(\vec{x})]^2 \leq \sum_w \mathbb{E}[g_w(\vec{x})^2]$. □

5.3 Proof of Theorem 1

Having established that SURFER OPTIMIZATION is a convex optimization problem, and that the change in the surfing strategy is an unbiased estimator of the gradient ∇F, we now prove Theorem 1. We essentially follow an "ODE method" [6], whereby the random evolution of the surfing strategy, captured by (4), is expressed in terms of the evolution of a deterministic solution to an ODE.

Recall that the feasible domain of (1) is D_ϵ, where $0 \leq \epsilon \leq \frac{1}{N}$, and let $\vec{x} \in D_\epsilon$. Given a vector $\vec{y}$, we define a mapping $Z : D_\epsilon \times \mathbb{R}^N \to \mathbb{R}^N$ as follows:

$$Z(\vec{x}, \vec{y}) = \lim_{\delta \to 0^+} \frac{\Pi_{D_\epsilon}(\vec{x} + \delta \vec{y}) - (\vec{x} + \delta \vec{y})}{\delta}. \quad (22)$$

Intuitively, for small δ, the quantity $\delta \cdot Z(\vec{x}, \vec{y})$ can be seen as the minimal force required to keep $\vec{x} + \delta \vec{y}$ in D_ϵ. The limit in the r.h.s. exists and, moreover, it can be expressed in terms of $\vec{x}$ and $\vec{y}$ in the following manner.

LEMMA 5. *Assume that $\vec{x} \in D_\epsilon$ for some $0 \leq \epsilon < \frac{1}{N}$. Let $A(\vec{x}) = \{w : x_w = \epsilon\}$ be the set of active constraints at $\vec{x}$. Then, there exists a set $B \subseteq A(\vec{x})$, such that $Z(\vec{x}, \vec{y}) = \vec{z}$ where*

$$z_w = \begin{cases} -y_w, & \text{if } w \in B, \\ -\frac{1}{|B^c|} \sum_{w' \in B^c} y_{w'}, & \text{if } w \in B^c. \end{cases} \quad (23)$$

Moreover, for every $w \in B$, $y_w - \frac{1}{|B^c|} \sum_{w' \in B^c} y_{w'} < 0$.

PROOF. Our proof will rely on the following property of the orthogonal projection Π_{D_ϵ}.

LEMMA 6. *Let $[N] = \{1, \ldots, N\}$, and assume that $0 \leq \epsilon < \frac{1}{N}$. Then, for every $\vec{x} \in \mathbb{R}^N$, there exists a set $A \subsetneq [N]$ (that depends on $\vec{x}$) such that for every $w \in A$, $x_w - \frac{\sum_{w' \in A^c} x_{w'} - 1 + N\epsilon}{|A^c|} < 0$ and $\Pi_{D_\epsilon}(\vec{x}) = \vec{y}$ where*

$$y_w = \begin{cases} \epsilon, & \text{if } w \in A, \\ x_w - \frac{\sum_{w' \in A^c} x_{w'} - 1 + N\epsilon}{|A^c|} + \epsilon, & \text{if } w \in A^c. \end{cases} \quad (24a)$$

This property follows from the simple geometry of D_ϵ. Its proof (and an $O(N)$ algorithm by Michelot [7] for computing the set A through repeated projections) is presented in Appendix C.

An immediate implication of the above property of Π_{D_ϵ} is that there exists a set B such that, for all $w \in B$,

$$x_w + \delta y_w - \frac{\sum_{w' \in B^c} (x_{w'} + \delta y_w) - 1 + N\epsilon}{|B^c|} < 0, \quad (25)$$

and, moreover, $\Pi_{D_\epsilon}(\vec{x} + \delta \vec{y}) = \vec{u}$, where

$$u_w = \begin{cases} \epsilon, & \text{if } w \in B, \\ x_w + \delta y_w - \frac{\sum_{w' \in B^c} (x_{w'} + \delta y_{w'}) - 1 + N\epsilon}{|B^c|} + \epsilon, & \text{o.w.} \end{cases} \quad (26)$$

Eq. (26) implies that B is a subset of $A(\vec{u}) = \{w : u_w = \epsilon\}$, the active constraints of $\vec{u}$. On the other hand, $A(\vec{u})$ is *upper semi-continuous* [6], *i.e.*, there exists $\delta' > 0$ s.t. for any $\vec{u} \in D_\epsilon$ with $\|\vec{x} - \vec{u}\| \leq \delta'$, $A(\vec{u}) \subseteq A(\vec{x})$. The projection on D_ϵ is Lipschitz continuous (in particular, it is non-expansive), so by making δ small enough we can make $\vec{u}$ to be arbitrarily close to $\vec{x}$. Upper semi-continuity therefore implies that, for δ small enough, $B \subseteq A(\vec{x})$. This, in turn, implies that

$$x_w = \epsilon, \text{ for all } w \in B, \quad \text{and} \quad \sum_{w' \in B^c} x_{w'} = 1 - |B|\epsilon. \quad (27)$$

The first statement of the lemma thus follows, as

$$\frac{1}{\delta}\left(\vec{u} - (\vec{x} + \delta \vec{y})\right) \overset{(26),(27)}{=} \begin{cases} -y_w, & w \in B \\ -\frac{1}{|B^c|} \sum_{w' \in B^c} y_{w'}, & w \in B^c. \end{cases}$$

In fact, we have proved that (23) holds not only in the limit, but for all small enough δ. The second statement follows from (25) and (27). □

Using the above lemma, can define a deterministic process that converges to a solution of SURFER OPTIMIZATION:

LEMMA 7. *Consider the following ODE*

$$\dot{\vec{x}} = \nabla F(\vec{x}) + Z(\vec{x}, \nabla F(\vec{x})) \quad (28)$$

and assume that the initial vector is in the feasible domain, i.e., *$\vec{x}(0) \in D_\epsilon$. Then, under Assumption 1, $F(\vec{x}(t)) \geq F(\vec{x}(t'))$ for all $t \geq t'$, and $\lim_{t \to \infty} F(\vec{x}(t)) = \sup_{\vec{x} \in D_\epsilon} F(\vec{x})$.*

PROOF. To begin with, assuming that $\vec{x}(0) \in D_\epsilon$, $\vec{x}(t) \in D_\epsilon$ for all $t \geq 0$, by the definition of the force $\vec{z}$. Since, by Lemmas 1 and 2, F is convex, increasing and continuously differentiable, a vector $\vec{x}^*$ will be a solution to the SURFER OPTIMIZATION problem, given by (1), if and only if there exist Lagrange multipliers $\lambda^*, \vec{\mu}^*$ that satisfy the KKT conditions:

$$\sum_\alpha x_a^* \leq 1, \quad \lambda^* (\sum_\alpha x_a^* - 1) = 0, \quad \text{and,}$$

$$\left.\frac{\partial F}{\partial x_a}\right|_{\vec{x}^*} - \lambda^* + \mu_a^* = 0, \quad x_a^* \geq \epsilon, \quad \mu_a^* (x_a^* - \epsilon) = 0, \quad \forall a \in [N]$$

where $[N] = \{1, \ldots, N\}$. The KKT conditions in turn imply that for an optimal solution $\vec{x}^*$ either $x_a^* = \epsilon$ or $\left.\frac{\partial F}{\partial x_a}\right|_{\vec{x}^*} = \lambda^*$, where λ^* the Lagrange multiplier of the constraint $\sum_\alpha x_a^* \leq 1$.

By Lemma 1, F is continuous, hence it attains a (finite) maximum in D_ϵ. Under (28),

$$\dot{F} = \nabla F(\vec{x}) \cdot \dot{\vec{x}} = \nabla F(\vec{x}) \cdot (\nabla F(\vec{x}) + Z(\vec{x}, \nabla F(\vec{x})))$$
$$\overset{(23)}{=} \sum_{a \in B^c(\vec{x})} \frac{\partial F}{\partial x_a} \left(\frac{\partial F}{\partial x_a} - \frac{1}{|B^c(\vec{x})|} \sum_{a' \in B^c(\vec{x})} \frac{\partial F}{\partial x_{a'}} \right)$$

(where $B(\vec{x})$ is some subset of the active constraints at $\vec{x}$),

$$= \sum_{a\in B^c(\vec{x})}\left(\frac{\partial F}{\partial x_a}\right)^2 - \frac{1}{|B^c(\vec{x})|}\left(\sum_{a'\in B^c(\vec{x})}\frac{\partial F}{\partial x_{a'}}\right)^2$$

$$= \sum_{i\in B^c(\vec{x})}\left(\frac{\partial F}{\partial x_a} - \frac{1}{|B^c(\vec{x})|}\sum_{a'\in B^c(\vec{x})}\frac{\partial F}{\partial x_{a'}}\right)^2 \geq 0$$

i.e., F is a Lyapunov function of (28), and this proves that $F(\vec{x}(t))$ is increasing in t. As not all constraints can be active for $\epsilon < \frac{1}{N}$, $B^c(\vec{x}) \neq \emptyset$. Thus, $\dot{F}$ is zero at some $\vec{x}^*$ if and only if $\frac{\partial F}{\partial x_a}\big|_{\vec{x}^*} = \frac{1}{|B^c(\vec{x}^*)|}\sum_{a'\in B^c(\vec{x}^*)}\frac{\partial F}{\partial x_{a'}}\big|_{\vec{x}^*}$, for every $a \in B^c(\vec{x}^*)$. Moreover, by Lemma 5, for every $a \in B(\vec{x}^*)$, $x_a^* = \epsilon$ and $\frac{\partial F}{\partial x_a}\big|_{\vec{x}^*} < \frac{1}{|B^c(\vec{x}^*)|}\sum_{a'\in B^c(\vec{x}^*)}\frac{\partial F}{\partial x_{a'}}\big|_{\vec{x}^*}$. Any such vector $\vec{x}^*$ will satisfy the KKT conditions with $\lambda^* = \frac{1}{|B^c(\vec{x}^*)|}\sum_{a'\in B^c(\vec{x}^*)}\frac{\partial F}{\partial x_{a'}}\big|_{\vec{x}^*}$ and $\mu_a^* = \max(\lambda^* - \frac{\partial F}{\partial x_a}\big|_{\vec{x}^*}, 0), \forall a \in [N]$, and thus is a maximizer of F. □

Note that the vector $\vec{x}(t)$ in the ODE (28) essentially follows the gradient $\nabla F(\vec{x})$; the term Z forces the evolution of $\vec{x}$ to be contained in D_ϵ.

The following is an immediate implication of Lemma 7.

LEMMA 8. *Assume that $\vec{x}(t)$, $t \in \mathbb{R}_+$, evolves according to the ODE* (28)*, and let $L_{D_\epsilon} = \lim_{t\to\infty}\bigcup_{\vec{y}\in D_\epsilon}\{\vec{x}(s), s \geq t : \vec{x}(0) = \vec{y}\}$ be the limit set [6] of* (28)*. Then, under Assumption 1, for every $\vec{x}\in L_{D_\epsilon}$, $F(\vec{x}) = \sup_{\vec{y}\in D_\epsilon} F(\vec{y})$.*

PROOF. Suppose that $F(\vec{x}) \neq F^* = \sup_{\vec{y}\in D_\epsilon} F(\vec{y})$ for some $\vec{x} \in L_{D_\epsilon}$. Then, there exists a sequence $\{t_n\}_{n\geq 1}$ and a $\vec{y} \in D_\epsilon$ such that $\vec{x}(0) = \vec{y}$ and $\lim_{n\to\infty}\vec{x}(t_n) = \vec{x}$, under some norm in $\mathbb{R}^N$. As F is continuous, $\lim_{n\to\infty} F(\vec{x}(t_n)) = F(\vec{x}) \neq F^*$, which contradicts Lemma 7. □

The reason we are interested in (28) is because it is a good "approximation" of the stochastic evolution of our mechanism (4). In particular, the following lemma states that the limit set of (4) will be included in L_{D_ϵ}:

LEMMA 9. *Let $\{\vec{x}(k)\}_{k\in\mathbb{N}}$ be the sequence defined by the stochastic approximation algorithm* (4)*, where $\gamma_k = \frac{1}{k}$ and $\epsilon > 0$. Then, under Assumption 1, w.p.1, the limit set of $\{\vec{x}(k)\}_{k\in\mathbb{N}}$ is a subset of the limit set of the ODE* (28).

PROOF. We have that $\sup_k \mathbb{E}[\|V(\vec{x}_k)\|_2^2] < \infty$, by Lemma 4, and $\lim_{k\to\infty}\gamma_k = 0$, $\sum_{k=1}^\infty \gamma_k = \infty$, $\sum_{k=1}^\infty \gamma_k^2 = \infty$. Moreover, F is continuously differentiable, by Lemma 2, and constant and equal to $\sup_{\vec{x}\in D_\epsilon} F(\vec{x})$ on all limit points of the ODE (28), by Lemma 8. Hence, the assumptions (A2.1)-(A2.6) of Theorem 2.1 in Chapter 5 of Kushner and Yin [6] are satisfied. As a result, w.p.1, the limit points of $\{\vec{x}_k\}_{k\in\mathbb{N}}$ belong to limit set of the ODE (28). □

Theorem 1 follows immediately from Lemmas 9 and 8. □

5.4 Proof of Theorem 2

Let L_{D_ϵ} be the limit points of the ODE (28), and, for some $\delta > 0$, let $N_\delta = \{\vec{x} \in D_\epsilon : \exists \vec{y} \in L_{D_\epsilon} \text{ s.t. } \|\vec{x} - \vec{y}\|_1 \leq \delta\}$ be the δ-neighborhood of L_{D_ϵ}. By Lemma 4, $\sup_k \mathbb{E}[\|V(\vec{x}_k)\|_2^2] < \infty$ and, by Lemma 2, F is continuously differentiable. Hence, assumptions (A1.1) to (A1.4) of Theorem 2.1 in Chapter 8 of Kushner and Yin [6] are satisfied. As a result,

$$\lim_{\gamma\to 0}\lim_{T\to\infty}\mathbf{P}\left(\frac{1}{T}\int_0^T \mathbb{1}_{\vec{\xi}(t)\notin N_\delta}dt > \delta'\right) = 0, \text{ for all } \delta, \delta' > 0. \quad (29)$$

I.e., the fraction of time the system spends in the δ-neighborhood of L_{D_ϵ} converges to one, in probability. On the other hand, for $F^* = \sup_{\vec{x}\in D_\epsilon} F(\vec{x})$ and $L = \sum_{k=1}^\infty R(k)$,

$$\left|F^* - \frac{1}{T}\int_0^T F(\vec{\xi}(t))dt\right| \leq \left|F^* - \frac{1}{T}\int_0^T F(\vec{\xi}(t))\mathbb{1}_{\vec{\xi}\notin N_\delta}dt\right| + \ldots$$
$$+ \left|\frac{1}{T}\int_0^T F(\vec{\xi}(t))\mathbb{1}_{\vec{x}\notin N_\delta}dt\right| \leq L\delta + \frac{L}{T}\int_0^T \mathbb{1}_{\vec{\xi}(t)\notin N_\delta}dt$$

by Corollary 1. Hence, for any $\delta', \delta > 0$,

$$\mathbf{P}\left(\left|F^* - \frac{1}{T}\int_0^T F(\vec{\xi}(t))dt\right| > \delta'\right) \leq \mathbf{P}\left(\frac{1}{T}\int_0^T \mathbb{1}_{\vec{x}\notin N_\delta}dt > \frac{\delta' - L\delta}{L}\right)$$

Theorem 2 thus follows from (29) by choosing $\delta < \delta'/L$. □

5.5 Proof of Theorem 3

The first part of Theorem 3 can be proved by replacing F with F_C and $\vec{g}$ with $\vec{g}_C = \frac{1}{|C|}\sum \vec{g}_s$ in the proofs of Theorems 1 and 2. For brevity, we simply give an outline of the proof below. To begin with, F_C is convex and continuously differentiable, as the average of convex and continuously differentiable functions. Lemma 2 implies that $\vec{g}_C(k) = \nabla F_C(\vec{x}(k)) + \bar{V}(\vec{x}(k))$ where $\bar{V}(\vec{x}(k))$ has zero expectation and bounded variance —the same as in Lemma (18), since $\vec{g}_r$ are independent. Moreover, as in Lemma 7, the solutions of the ODE

$$\dot{\vec{x}} = \nabla F_C(\vec{x}) + Z(\vec{x}, \nabla F_C(\vec{x})) \quad (30)$$

where Z is given by (22) converge to maximizers of F_C. Finally, Lemma 9 and Theorem 2 can be similarly extended by relating the evolution of (8) to the ODE (30).

We therefore turn our attention to proving (10). To begin with, observe that for every $\vec{x} \in D_\epsilon$,

$$F_C(\vec{x}) = \frac{1}{|C|}\sum_s\sum_f d_{s,f}\mathbb{E}_{\vec{x}}[R(Y_{s,f})] = \sum_f d_{C,f}\mathbb{E}_{\vec{x}}[R(Y_f)]$$

where $d_{C,f} = \frac{1}{|C|}\sum_s d_{s,f}$. This is because $\mathbb{E}_{\vec{x}}[R(Y_{s,f})]$ does not depend on s: conditioned on the surfing strategy, Y_f has the same distribution for all surfers $s \in C$. For the same reason, for all $\vec{x} \in D_\epsilon$ and all $s \in C$, and for $\bar{R} = R(1)$,

$$|F_C(\vec{x}) - F_s(\vec{x})| = \Big|\sum_f (d_{C,f} - d_{s,f})\mathbb{E}_{\vec{x}}[R(Y_f)]\Big| \leq \bar{R}\delta_C. \quad (31)$$

Let $\vec{x}^*$ be a maximizer of F_C and $\vec{\sigma}^*$ a maximizer of F_s. Then $F_C(\vec{\sigma}^*) - F_C(\vec{x}^*) \leq 0$ and $F_s(\vec{x}^*) - F_s(\vec{\sigma}^*) \leq 0$ while, from (31), $F_s(\vec{\sigma}^*) - F_C(\vec{\sigma}^*) \leq \bar{R}\delta_C$ and $F_C(\vec{x}^*) - F_s(\vec{x}^*) \leq \bar{R}\delta_C$. Adding the former inequalities to the latter yields (10). □

5.6 Proof of Theorem 4

Since the Hessian of F_s is invertible at $\vec{x}^*$, for small enough $\delta > 0$, there exists a constant Q such that if for all $\vec{x} \in D_\epsilon$ with $\|\vec{x} - \vec{x}_s^*\|_1 > \delta$, $F_s(\vec{x}_s^*) - F_s(\vec{x}) > Q\delta^2$. On the other hand, for any $\vec{x}$, $|F_s(\vec{x}) - F_C(\vec{x}| \leq L\delta_C$ and, therefore,

$$F_s(\vec{x}_C^*) + L\delta_C \geq F_C(\vec{x}_C^*) \geq F_C(\vec{x}_s^*) \geq F_s(\vec{x}_s^*) - L\delta_C,$$

hence $F_s(\vec{x}_s^*) - F_s(\vec{x}_C^*) \leq 2L\delta_C$. This implies that, for small enough δ_C, $\|\vec{x}_s^* - \vec{x}_C^*\|_1 \leq \sqrt{2L\delta_C/Q} = O(\sqrt{\delta_C})$.

Using a similar argument as in the proof of Lemma 2 and Corollary 1, one can show that if $k|R(k)|$ is summable, for all $s \in C$ the gradient ∇F_s will be Lipschitz continuous and that the same is true for the gradient ∇F_C of F_C. Moreover, it can be shown directly from Lemma 2 that $\|\nabla F_s(\vec{x}) - \nabla F_{s'}(\vec{x})\|_1 = O(\delta_C)$, for all $\vec{x} \in D_\epsilon$ and for all $s, s' \in C$.

Furthermore, by Lemma 2, the Hessian H_s of F_s has the following coordinates:

$$\frac{\partial^2 F_s}{\partial x_w \partial x_{w'}} = \sum_f d_{s,f} p_{w,f} p_{w',f} \begin{cases} \frac{\mathbb{E}[Y_f(Y_f+1)\Delta R(Y_f+2)]}{\rho_f}, \text{if } \rho_f > 0 \\ -2\sum_{k=1}^{\infty}(k-1)R(k), \text{o.w.} \end{cases}$$

which exists, is bounded and continuous provided that $kR(k)$ is summable. Similarly, the Hessian of F_C can be obtained from H_s by replacing $d_{s,f}$ by $d_{C,f}$. Again, one can show that if $k^2|R(k)|$ is summable then for all $s \in C$ the Hessian H_s will be Lipschitz continuous, w.r.t. the $\|\cdot\|_1$ norm[6], and that the same is true for the Hessian H_C of F_C. Moreover, it can again be shown directly from the above expression that, for all $\vec{x} \in D_\epsilon$, $\|H_s(\vec{x}) - H_C(\vec{x})\|_1 = O(\delta_C)$. As a result,

$$\begin{aligned} \|H_s(\vec{x}_s^*) - H_C(\vec{x}_C^*)\|_1 &\leq \|H_s(\vec{x}_s^*) - H_s(\vec{x}_C^*)\| + \ldots \\ &+ \|H_s(\vec{x}_C^*) - H_C(\vec{x}_C^*)\|_1 = O(\sqrt{\delta_C}) + O(\delta_C) \end{aligned} \quad (32)$$

The covariance of $\vec{g}_s$, denoted by

$$\Sigma_s = \mathbb{E}\left[(\vec{g}_s - \nabla F_s) \cdot (\vec{g}_s - \nabla F_s)^T\right]$$

can be shown to be a matrix of the form $[\sigma_{w,w'}]$ where

$$\sigma_{w,w'} = \begin{cases} -\frac{\partial F_s}{\partial x_w}\frac{\partial F_s}{\partial x_{w'}}, \text{ if } w \neq w' \\ \mathbb{E}[g_{s,w}^2] - \left(\frac{\partial F_s}{\partial x_w}\right)^2, \text{ if } w = w' \end{cases}$$

On the other hand, the covariance of $\vec{g}_C$,

$$\Sigma_C = \mathbb{E}\left[(\vec{g}_C - \nabla F_C) \cdot (\vec{g}_C - \nabla F_C)^T\right]$$

can be shown to be of the form $\Sigma_C = \frac{1}{|C|}\Sigma_s + O(\delta_C)$. Hence,

$$\begin{aligned} \|\frac{\Sigma_s(\vec{x}_s^*)}{|C|} - \Sigma_C(\vec{x}_C^*)\|_1 &\leq \frac{\|\Sigma_s(\vec{x}_s^*) - \Sigma_s(\vec{x}_C^*)\|}{|C|} + \ldots \\ + \|\frac{\Sigma_s(\vec{x}_C^*)}{|C|} - \Sigma_C(\vec{x}_C^*)\|_1 &= \frac{O(\sqrt{\delta_C})}{|C|} + O(\delta_C) \end{aligned} \quad (33)$$

From Theorem 1.1 in Chapter 10 of Kushner and Yin [6], if $\vec{x}(k)$ is updated at surfer s according to (4), the process $\frac{\xi(T+t)-\vec{x}^*}{\sqrt{\gamma}}$ converges weakly as $T \to \infty$ and $\gamma \to 0$ to $U_s(t)$ that satisfies

$$U_s(t) = U_s(0) + \int_0^t H_s(\vec{x}_s^*)U_s(\tau)d\tau + W_s(t)$$

where $W_s(t)$ is an N-dimensional Brownian motion with covariance matrix $\Sigma_s(\vec{x}_s^*)$. In particular, let $H_s(\vec{x}_s^*) = \Gamma_s\Phi_s\Gamma_s^T$, where Γ_s orthonormal and Φ_s a diagonal matrix. Such a diagonalization is possible as H_s is by definition Hermitian. Moreover, since by the hypothesis of the theorem $H_s(\vec{x}_s^*)$ is invertible, none of the diagonal elements ϕ_w of Φ_s will be zero. Then (see [9]), $U_s(t)$ will be a normally distributed random variable with covariance matrix $\bar{\Sigma}_s = \Gamma_s^T A \Gamma_s$ where $A = [a_{w,w'}]$ and $a_{w,w'} = [\Gamma_s\Sigma_s(\vec{x})\Gamma_s^T]_{w,w'}/(\phi_w + \phi_{w'})$. Eq. 11 thus follows by taking c to be the trace of $\bar{\Sigma}_s$.

Similarly, if $\vec{x}(k)$ is updated at surfer s according to (8), the process $\frac{\xi(T+t)-\vec{x}^*}{\sqrt{\gamma}}$ converges weakly as $T \to \infty$ and $\gamma \to 0$ to $U_C(t)$ that satisfies

$$U_C(t) = U_C(0) + \int_0^t H_C(\vec{x}_C^*)U_C(\tau)d\tau + W_C(t)$$

where $W_C(t)$ is an N-dimensional Brownian motion with covariance matrix $\Sigma_C(\vec{x}_C^*)$. From (32) and (33), for small enough δ_C, $H_C(\vec{x}_C^*)$ and $\Sigma_C(\vec{x}_C^*)$ are perturbed versions of $H_s(\vec{x}_s^*)$ and $\Sigma_s(\vec{x}_s^*)/|C|$, respectively. The theorem therefore follows by observing that the corresponding covariance matrix $\bar{\Sigma}_C$ will be a perturbed version of $\bar{\Sigma}_s/|C|$. For this, it is important that, as $H_s(\vec{x}_s^*)$ is invertible, both its eigenvectors Γ and its eigenvalues Φ will be Lipschitz continuous functions of $H_s(\vec{x}_s^*)$ [10]; in particular, for small enough δ_C, $H_C(\vec{x}_C^*)$ is also invertible. □

[6]Recall that the $\|\cdot\|_1$ norm of a matrix is the maximum of the $\|\cdot\|_1$ norm of among its columns.

6. DIMENSIONALITY REDUCTION

In this section, we outline a method for reducing the dimension of the surfing strategy. Consider a partition $P = \{B_1, B_2, \ldots, B_{N'}\}$ of the set $\{1, \ldots, N\}$ into N' groups, where $N' \ll N$. Moreover, let

$$\delta_P = \max_{i=1,\ldots,N'} \max_{w,w' \in B_i} \|\vec{p}_w - \vec{p}_{w'}\|_1$$

be the maximum l_1 distance between the publishing strategies of two websites belonging to the same group. Intuitively, if δ_P is small, websites in the same group publish content in a similar manner, and B_i can be seen as "communities" of websites.

Instead of implementing a surfing strategy over all sites, the mechanism can be indifferent towards sites from the same group. This can take place as follows. Consider strategies of the form $\vec{v} \in \mathbb{R}^{N'}$ defined *over the groups* B_i. To recommend a website, the mechanism first chooses a group B_i with probability v_i; then, it chooses a website uniformly from within this group and displays it to the surfer; as a result, each website in B_i is shown with the same probability, $v_i/|B_i|$. The strategy $\vec{v}$ is then updated according to (4), where websites are replaced by the groups B_i.

The above approach has the obvious advantage that the dimension of the vector $\vec{v}$ is reduced to N'. Let

$$D'_\epsilon = \{\vec{v} \in \mathbb{R}^{N'} : \sum_i^{N'} v_i = 1 \text{ and } v_i \geq \epsilon\}$$

be the set of strategies $\vec{v}$ defined over the partition P, and denote by

$$G(\vec{v}) = \mathbb{E}_{\vec{v}}[R(Y)]$$

the expected rating of the system when the strategy is $\vec{v}$.

The following theorem, whose proof is Appendix B, states that if δ_P is not too large—*i.e.*, the websites in the same group indeed have similar publishing strategies—the mechanism (4) will converge to a strategy close to the optimal.

THEOREM 5. *Let $\vec{v}(k)$ be updated by the mechanism (4), where $\vec{v}$ is defined as a strategy over the groups in partition P. Then, Theorems 1 and 2 hold if $\vec{x}$, $F(\vec{x})$ and D_ϵ are replaced by $\vec{v}$, $G(\vec{v})$, and D'_ϵ, respectively. Moreover, under Assumption 1,*

$$\left|\sup_{\vec{v}\in D'_\epsilon} G(\vec{v}) - \sup_{\vec{x}\in D_\epsilon} F(\vec{x})\right| = O(\delta_P). \quad (34)$$

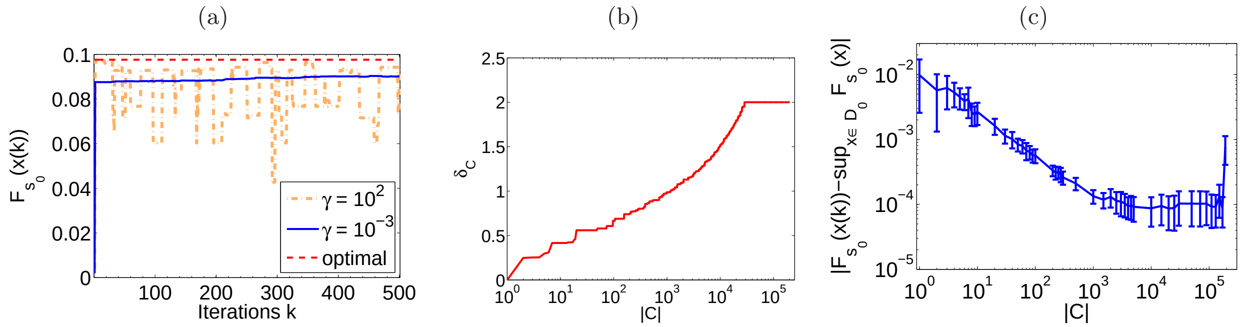

Figure 2: In figure (a), an execution of mechanism (4) by the central surfer s_0 is illustrated for $\gamma = 10^2$ and $\gamma = 10^{-3}$: within one iteration, the algorithm gets within %1 and %20 of the optimal value, respectively, and the larger value of γ also implies higher variability. In figure (b), the diameter δ_C, given by (9), is plotted as a function of the community size $|C|$. Finally, in (c), we plot the mean distance of $F_{s_0}(x(k))$ from the optimal value $\sup_{D_\epsilon} F_{s_0}(\vec{x})$, for $\gamma = 0.001$, averaged over 10,000 iterations. Each point corresponds to a different community size $|C|$, and the bars show the standard deviation. We observe that, initially, increasing the community size $|C|$ leads to an improvement on the accuracy of the mechanism. The benefit of aggregating feedback among the community disappears at about $|C| = 1500$; above $|C| = 4000$, the trend is reversed and the average distance from the optimal value starts to increase.

7. NUMERICAL STUDY

7.1 Delicious Dataset

We evaluated the performance of our mechanism through a numerical study using traces from Delicious. The trace we used was collected and made publicly available[7] by Göerlitz *et al.* [11]. It documents the tag assignments made by Delicious users between January 2006 and December 2007; for each tag assignment, the dataset contains the posting date, the user and website IDs and the actual tag.

We use this trace to obtain website publishing strategies and surfer (user) interest profiles as follows. First, due to memory constraints, we limit the dataset by focusing on the 20 most popular tags. We then further reduce the dataset by considering only tagging events for the 20 most tagged websites. This gives us a subset of the total trace, containing 20 websites, 19 tags (one was never used on the 20 most tagged websites), 203,084 users and 386,198 tagging events. We deduce the profile $\vec{d}_s$ of user s by taking $d_{s,f}$ to be proportional to the number of times s tagged a website with tag f. Similarly, we deduce the publishing strategy $\vec{p}_w$ of a site w by taking $p_{w,f}$ to be proportional to the number of times w was tagged by a user with tag f.

7.2 Evaluating the Effect of Community Feedback

To illustrate the behavior of our proposed mechanism, we selected the surfer s_0 that was positioned in the "center" of our dataset, *i.e.*, the average interest profile $\vec{d}_{C_0} = \sum_{C_0} \vec{d}_s/|C_0|$ over the entire dataset C_0 of 203,084 surfers.

Taking $\epsilon = 10^{-4}$ and $R(Y) = Y^{-2}$, we simulated two different executions of mechanism (4) as run by surfer s_0 with $\gamma = 100$ and $\gamma = 0.001$, respectively. These are illustrated in Fig. 6(a): in each case, we plot the expected reward $F_{s_0}(x(\vec{k}))$ for $k = 1, \ldots, 500$. We see that within only one iteration, the mechanism gets within %1 of the optimal value when $\gamma = 100$, and %20 when $\gamma = 0.001$. However, the larger value of γ also translates to higher variability.

To evaluate the effect of aggregating community feedback, we construct a community centered around s_0 as follows. Initially, our community C consists only of s_0. We then add other surfers sequentially, by taking each time the closest surfer to s_0 that is not already in C. In figure 6(b), we plot the diameter δ_C of the community, *i.e.*, the maximum l_1 distance between two users in C, as a function of the community size $|C|$. Note that the diameter cannot exceed two.

For several different values of the community $|C|$, we ran the mechanism (8), that aggregated community feedback at each iteration, for a total of $k = 10^5$ iterations. In each execution, we took $\gamma = 10^{-3}$, and started from an optimal strategy. In Figure 6(c), we plot the distance of $F_{s_0}(\vec{x}(k))$ from its optimal value in D_ϵ, averaged over all k iterations. This indicates the accuracy of the mechanism (8) in maximizing the rating as perceived by s_0.

As predicted by Theorem 4, the accuracy improves drastically with $|C|$—by two orders of magnitude—as long as the community diameter remains small. However, there exists a critical value, around $|C|$ =1,500, around which the benefits of increasing the community vanish. For $|C|$ larger than 4,000, δ_C becomes so large that the distance from the optimal value starts to increase. The above suggests that the ideal community size for surfer s_0 is between 1,500 and 4,500; based on the profiles of the other surfers extracted from the trace, aggregating feedback from larger communities will only decrease the accuracy of the system, as perceived by surfer s_0.

8. CONCLUSIONS

In this paper, we proposed a model and a mechanism for personalized randomized web-surfing, that operated with only minimal feedback from surfers. In addition, we illustrated the benefit of feedback aggregation over surfer communities: one of our key findings is the identification of a trade-off between the size of a community and its diameter, the impact of which we are able to quantify both analytically

[7] http://isweb.uni-koblenz.de/Research/DataSets

and numerically.

9. REFERENCES

[1] Stumbleupon's recommendation technology. [Online]. Available: http://www.stumbleupon.com/technology/

[2] N. Bansal and N. Koudas, "Searching the blogosphere," in *WebDB*, 2007.

[3] K. El-Arini, G. V. D. Shahaf, and C. Guestrin, "Turning down the noise in the blogosphere," in *KDD*, 2009.

[4] A. Das, M. Datar, A. Garg, and S. Rajaram, "Google news personalization: Scalable online colaborative filtering," in *WWW*, 2007.

[5] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–33, 2004.

[6] H. J. Kushner and G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*, 2nd ed. Springer, 2003.

[7] C. Michelot, "A finite algorithm for fiding the projection of a point onto the canonical simplex of $\mathbb{R}^n$," *Journal of Optimization Theory and Applications*, vol. 50, no. 1, pp. 195–200, 1986.

[8] N. Maculan, C. Santiago, E. Macambira, and M. Jardim, "An $O(n)$ algorithm for projecting a vector on the intersection of a hyperplane and a box in $\mathbb{R}^n$," *Journal of Optimization Theory and Applications*, vol. 117, no. 3, pp. 553–574, 2003.

[9] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks:shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.

[10] D. S. Watkins, *Fundamentals of matrix computations*. Wiley-Interscience, 2002.

[11] O. Görlitz, S. Sizov, and S. Staab, "Pints: Peer-to-peer infrastructure for tagging systems," in *IPTPS*, 2008.

APPENDIX

A. PROOF OF LEMMA 3

For every $\vec{x} \in D_0$, we show below that one can construct a $\vec{x}' \in D_\epsilon$ such that

$$\|\vec{x} - \vec{x}'\|_1 \le 2N\epsilon. \tag{35}$$

We use this result as follows. The optimality of $\vec{x}^*$ in D_0, Corollary 1 and (35) imply that there exists a $\vec{y} \in D_\epsilon$ such that $0 \le F(\vec{x}^*) - F(\vec{y}) = |F(\vec{x}^*) - F(\vec{y})| \le 2NL\epsilon$. By the optimality of $\vec{\sigma}^*$ in D_ϵ, we have that $F(\vec{y}) \le F(\vec{\sigma}^*)$ and the lemma follows.

It remains thus to prove (35). Assume that $\vec{x} \notin D_\epsilon$ (otherwise, (35) holds with $\vec{x}' = \vec{x}$). Let $[N] = \{1,\dots,N\}$, $I_{<\epsilon} = \{w \in [N] \text{ s.t. } x_w < \epsilon\}$, and $I_{\ge\epsilon} = [N] \setminus I_{<\epsilon} = \{w \in [N] \text{ s.t. } x_w \ge \epsilon\}$. Notice that $I_{<\epsilon} \ne \emptyset$ since $\vec{x} \notin D_\epsilon$ and $I_{\ge\epsilon} \ne \emptyset$ as $\epsilon \le \frac{1}{N}$. Let $\delta_w = x_w - \epsilon$. We have that $\sum_{w\in I_{\ge\epsilon}} \delta_w + \sum_{w\in I_{<\epsilon}} \delta_w = \sum_{w\in I_{<\epsilon}}(x_w-\epsilon) + \sum_{w'\in I_{\ge\epsilon}}(x_{w'}-\epsilon) = 1 - N\epsilon \ge 0$ as $\epsilon \le 1/N$. This implies that there exist a (possibly empty) set $A \subsetneq I_{\ge\epsilon}$ and a $u \in I_{\ge\epsilon} \setminus A$ such that

$$\sum_{w\in A}\delta_w < -\sum_{w\in I_{<\epsilon}}\delta_w \quad \text{and} \quad \delta_u + \sum_{w\in A}\delta_w \ge -\sum_{w\in I_{<\epsilon}}\delta_w. \tag{36}$$

Define

$$x'_w = \begin{cases} \epsilon, & w \in I_{<\epsilon} \cup A \\ x_u + (\sum_{w'\in I_{<\epsilon}}\delta_{w'} + \sum_{w'\in A}\delta_{w'}), & w = u \\ x_w, & w \in I_{\ge\epsilon} \setminus (A \cup \{u\}) \end{cases}$$

We have that

$$\sum_{i=1}^N x'_w = \sum_{w\in I_{<\epsilon}}\epsilon + \sum_{w\in A}\epsilon + x_k + \left(\sum_{w'\in I_{<\epsilon}}\delta_{w'} + \sum_{w'\in A}\delta_{w'}\right) + \sum_{w\in I_{\ge\epsilon}\setminus(A\cup\{u\})} x_w$$
$$= \sum_{w\in I_{<\epsilon}} x_w + \sum_{w\in A} x_w + x_k + \sum_{w\in I_{\ge\epsilon}\setminus(A\cup\{u\})} x_w = \sum_{i=1}^N x_w = 1.$$

For all $w \ne u$ we obviously have $x'_w \ge \epsilon$, while

$$x'_u = x_u + (\sum_{w\in I_{<\epsilon}}\delta_w + \sum_{w\in A}\delta_w) \overset{(36)}{\ge} x_u - \delta_u = \epsilon.$$

Hence, $\vec{x} \in D_\epsilon$. On the other hand

$$\|\vec{x} - \vec{x}'\|_1 = \sum_{w\in I_{<\epsilon}}(\epsilon - x_w) + \sum_{w\in A}(x_w - \epsilon) - (\sum_{w\in I_{<\epsilon}}\delta_w + \sum_{w\in A}\delta_w)$$
$$= -\sum_{w\in I_{<\epsilon}}\delta_w + \sum_{w\in A}\delta_w - \sum_{w\in I_{<\epsilon}}\delta_w - \sum_{w\in A}\delta_w = -2\sum_{w\in I_{<\epsilon}}\delta_w \le 2N\epsilon.$$

and (35) follows. □

B. PROOF OF THEOREM 5

The proofs of Theorems 1 and 2 hold as is, by replacing the set of websites with the partition P. We therefore focus on the proof of (34). We will show that, for every $\vec{x} \in D_\epsilon$, there exists a $\vec{v} \in D'_\epsilon$ such that

$$|F(\vec{x}) - G(\vec{v})| \le L\delta_p, \tag{37}$$

where $L = \sum_{\ell=1}^\infty R(\ell)$. Eq. (34) then follows as: if $\vec{x}^*$ is a maximizer of F, and $\vec{v}^*$ is a maximizer of G, $F(\vec{x}^*) \ge G(\vec{v}^*|$ as D'_ϵ can be seen as a restriction over all possible strategies in D'_ϵ. Hence

$$|F(\vec{x}^*) - G(\vec{v}^*)| = F(\vec{x}^*) - G(\vec{v}^*) \overset{(37)}{\le} L\delta_p + G(\vec{v}) - G(\vec{v}^*)$$

for some $v \in D'_\epsilon$, and $G(\vec{v}) - G(\vec{v}^*) \le 0$, by the optimality of $\vec{v}^*$.

It thus remains to prove (37). Indeed, take $\vec{v}$ such that $v_i = \sum_{w\in B_i} x_w$. Then, since $\vec{x} \in D_\epsilon$ then $\vec{v} \in D'_\epsilon$. Moreover, that $G(v)$ can be expressed in terms of F as $G(\vec{v}) = F(\vec{x}')$, where $x'_w = \frac{1}{|B_{i(w)}|} v_{i(w)} = \frac{1}{|B_{i(w)}|}\sum_{w'\in B_{i(w)}} x_{w'}$ and $i(w)$ is the group i s.t. $w \in B_i$.

$$|F(\vec{x}) - G(\vec{v})| = \left|F(\vec{x}) - F(\vec{x}')\right| \le \sum_f d_f |\mathbb{E}_{\vec{x}}[R(Y_f)] - \mathbb{E}_{\vec{x}'}[R(Y_f)]|$$

where Y_f the number of steps taken conditioned on the topic being f. Let $\rho_f(\vec{x}) = \sum_w p_{w,f} x_w$. From (19), we have that $\frac{\partial \mathbb{E}_x[R(Y_f)]}{\rho_f} \le L$. Hence, $\mathbb{E}_x[R(Y_f)]$ is Lipschitz in ρ_f with parameter L. This gives us

$$|F(\vec{x}) - G(\vec{v})| \le L\sum_f d_f |\rho_f(\vec{x}) - \rho_f(\vec{x}')|$$

We write $w \sim w'$ if they belong in the same group. Then

$$|\rho_f(\vec{x}) - \rho_f(\vec{x}')| = \left|\sum_w p_{w,f} x_w - \sum_w p_{w,f} x'_w\right|$$

$$= \left|\sum_w p_{w,f} x_w - \sum_w \sum_{w'} \mathbb{1}_{w\sim w'} \frac{p_{w,f}}{|B_{i(w)}|} x_{w'}\right|$$

$$= \left|\sum_w p_{w,f} x_w - \sum_w \sum_{w'} \mathbb{1}_{w\sim w'} \frac{p_{w,f}}{|B_{i(w')}|} x_{w'}\right|$$

$$\le \sum_w x_w \left|p_{w,f} - \frac{\sum_{w'\in B_{i(w)}} p_{w',f}}{|B_{i(w)}|}\right| \le \sum_w x_w \delta_p = \delta_p$$

and (37) follows. □

C. PROJECTION ON THE SIMPLEX

Given $r \ge 0$, let $V(r)$ be the following hyperplane:

$$V(r) = \{\vec{x} : \sum_{i=1}^{N} x_i = r\} \tag{38}$$

Moreover, let $D(r)$ be the (non-canonical) simplex with radius r, *i.e.*,

$$D(r) = \{\vec{x} : \sum_i x_i = r \text{ and } \vec{x} \ge 0\}. \tag{39}$$

Note that, under this notation, $D \equiv D(1)$.

Let $[N] = \{1, 2, \dots, N\}$ and, for any $A \subseteq [N]$, define

$$X_A = \{\vec{x} \in \mathbb{R}^N : x_i = 0 \text{ for all } i \in A\}, \tag{40}$$

$$V_A(r) = X_A \cap V(r)$$

$$= \{\vec{x}\in\mathbb{R}^N : \sum_{i=1}^{N} x_i = r \text{ and } x_i = 0 \text{ for all } i \in A\} \tag{41}$$

Note that, for every $A \subseteq [N]$, X_A is a linear subspace of $\mathbb{R}^N$. Moreover, if $A \subseteq B \subseteq [N]$, then X_B is a linear subspace of X_A and, thus, $V_B(r) \subseteq V_A(r)$.

For any A, the projection $\Pi_{V_A(r)}(\vec{x}) \mapsto \vec{y}$ is such that

$$y_i = \begin{cases} x_i - \frac{1}{|A^c|}\left(\sum_{j\in A^c} x_j - r\right), & \text{if } i \notin A \\ 0, & \text{if } i \in A \end{cases} \tag{42}$$

and, thus, can be computed in a straightforward manner.

Lemma 10. *For any $A \subseteq B \subseteq [N]$,*

$$\Pi_{V_B(r)}\left(\Pi_{V_A(r)}(\vec{x})\right) = \Pi_{V_B(r)}(\vec{x})$$

Proof. By (42) $\vec{y} = \left(\Pi_{V_A(r)}(\vec{x})\right)$ is such that

$$y_i = \begin{cases} x_i - \frac{1}{|A^c|}\left(\sum_{j\in A^c} x_j - r\right), & \text{if } i \notin A \\ 0, & \text{if } i \in A \end{cases}$$

and $\vec{z} = \left(\Pi_{V_B(r)}(\vec{y})\right)$ is

$$z_i = \begin{cases} y_i - \frac{1}{|B^c|}\left(\sum_{j\in B^c} y_j - r\right), & \text{if } i \notin B \\ 0, & \text{if } i \in B \end{cases}$$

$$\overset{B^c\subseteq A^c}{=} \begin{cases} x_i - \frac{\sum_{j\in A^c} x_j - r}{|A^c|} \\ \quad - \frac{\sum_{j\in B^c}\left(x_j - \frac{1}{|A^c|}\left(\sum_{k\in A^c} x_k - r\right)\right) - r}{|B^c|}, & \text{if } i \notin B \\ 0, & \text{if } i \in B \end{cases}$$

$$= \begin{cases} x_i - \frac{1}{|B^c|}\left(\sum_{k\in B^c} x_k - r\right), & \text{if } i \notin B \\ 0, & \text{if } i \in B \end{cases}$$ □

Consider now the following iterative projection algorithm.

```
1   function  y⃗ = Γ(x⃗)
2     A = ∅ ;
3     y⃗ := Π_{V_A(r)}(x⃗) ;
4     B := {i ∈ [N] : y_i < 0} ;
5     while  B ≠ ∅
6     begin
7        A := A ∪ B ;
8        y⃗ := Π_{V_A(r)}(y⃗) ;
9        B := {i ∈ [N] : y_i < 0}
10    end ;
```

Theorem 6 (Michelot [7]). *The iterative projection algorithm outlined above computes the projection on $D(r)$, i.e., $\Gamma(\cdot) \equiv \Pi_{D(r)}(\cdot)$.*

The proof can be found in [7]. What is important in the context of this work, is that the above algorithm gives us an insight in the nature of the projection $\vec{y} = \Pi_{D(r)}(\vec{x})$.

Lemma 11. *For any $\vec{x}$, the loop evaluating $\Gamma(\vec{x})$ terminates after at most N iterations. Moreover, there exists a set $A \subsetneq [N]$ (depending on $\vec{x}$) such that*

1. *$\Pi(\vec{x}) = \Pi_{V_{A(r)}}(\vec{x})$, and*
2. *if $i \in A$ then $x_i - \frac{1}{|A^c|}\left(\sum_{j\in A^c} x_j - r\right) < 0$.*

Proof. Let $A(0) = \emptyset$, $\vec{y}(0) = \Pi_{V_\emptyset(r)}(\vec{x})$, and $B(0) = \{i \in [N] : y_i(0) < 0\}$ be the initial values of $A, \vec{y}$ and B prior to the execution of the loop, (lines 2-4) and denote by $\vec{y}(j)$, $A(j)$ and $B(j)$ the corresponding quantities after the j-th iteration of the loop (*i.e.*, their values line 10). Note that

$$A(j) = A(j-1) \cup B(j-1),$$
$$y(j) = \Pi_{V_{A(j)}(\vec{y}(j))}, \text{ and}$$
$$B(j) = \{i \in [N] : y_i(j) < 0\}.$$

To see that the loop terminates after at most N iterations observe that $A(j)$ increases by at least one element with each iteration and $A(j) \cap B(j) = \emptyset$. Let $j_{\max} \le N$ be the last iteration of the while loop and note that

$$A(0) \subset A(1) \subset A(2) \subset \dots A(j_{\max}).$$

Then, the final vector $\vec{y}$ will be

$$\vec{y} = \Pi_{V_{A(j_{\max})}}(\Pi_{V_{A(j_{\max}-1)}}(\dots\Pi_{V A(0)}(\vec{x})\dots)) = \Pi_{V_{A(j_{\max})}}(\vec{x}).$$

by Lemma 10. We prove the last statement of the lemma by induction on $j = 0, 1, \dots, j_{\max}$. For $j = 0$, the statement is vacuously true as $A(0) = \emptyset$. Suppose now that for a certain $j \ge 1$,

$$x_i - \frac{1}{|A^c(j-1)|} \sum_{i'\in A^c(j-1)} x_{i'} < 0$$

for all $i \in A(j-1)$. Note that, by Lemma 10, $\vec{y}(j-1) = \Pi_{V_{A(j-1)}}(\vec{y}(j-2)) = \dots = \Pi_{V_{A(j-1)}}(\vec{x})$. That is

$$y_i(j-1) = \begin{cases} 0, & \text{if } i \in A(j-1) \\ x_i - \frac{\sum_{i'\in A^c(j-1)} x_{i'} - r}{|A^c(j-1)|}, & \text{if } i \in A^c(j-1) \end{cases}$$

The difference $A(j)\backslash A(j-1) = A^c(j-1)\backslash A^c(j) = B(j-1)$ consists of all $i \in [N]$ such that $y_i(j-1) < 0$. This is nonempty, as the j-th iteration takes place and, therefore

$$
\begin{aligned}
0 > \sum_{i\in B(j-1)} y_i(j-1) &= \sum_{i\in A^c(j-1)\setminus A^c(j)} y_i(j-1) \\
&= \sum_{i\in A^c(j-1)\setminus A^c(j)} \left[x_i - \frac{\sum_{i'\in A^c(j-1)} x_{i'} - r}{|A^c(j-1)|}\right] \\
&= \sum_{i\in A^c(j-1)\setminus A^c(j)} x_i - \frac{|A^c(j-1)\setminus A^c(j)|\left(\sum_{i'\in A^c(j-1)} x_{i'} - r\right)}{|A^c(j-1)|} \\
&= \frac{|A^c(j)|\sum_{i\in A^c(j-1)\setminus A^c(j)} x_i}{|A^c(j-1)|} \\
&\quad - \frac{|A^c(j-1)\setminus A^c(j)|\left(\sum_{i'\in A(j)} x_{i'} - r\right)}{|A^c(j-1)|}
\end{aligned}
$$

or,

$$
|A^c(j)|\sum_{i\in A^c(j-1)\setminus A^c(j)} x_i - |A^c(j-1)\setminus A^c(j)|\left(\sum_{i'\in A^c(j)} x_{i'} - r\right) < 0
$$

This, in turn, implies that

$$
\begin{aligned}
&\frac{1}{|A^c(j-1)|}\left(\sum_{i\in A^c(j-1)} x_i - r\right) - \frac{1}{|A^c(j)|}\left(\sum_{i\in A^c(j)} x_i - r\right) \\
&= \frac{|A^c(j)|(\sum_{i\in A^c(j-1)} x_i - r) - |A^c(j-1)|(\sum_{i\in A^c(j)} x_i - r)}{|A^c(j)||A^c(j-1)|} \\
&= \frac{1}{|A^c(j)||A^c(j-1)|}\cdot \\
&\left(|A^c(j)|\sum_{i\in A^c(j-1)\setminus A^c(j)} x_i - |A^c(j-1)\setminus A^c(j)|(\sum_{i\in A^c(j)} x_i - r)\right) < 0
\end{aligned}
$$

For all $k \in A(j-1)$

$$
x_k - \frac{\left(\sum_{i\in A^c(j)} x_i - r\right)}{|A^c(j)|} < x_k - \frac{\left(\sum_{i'\in A^c(j-1)} x_{i'} - r\right)}{|A^c(j-1)|} < 0.
$$

by the induction hypothesis. Similarly, for all $k \in A(j) \setminus A(j-1) = B(j-1)$,

$$
x_k - \frac{\left(\sum_{i\in A^c(j)} x_i - r\right)}{|A^c(j)|} < x_k - \frac{\left(\sum_{i'\in A^c(j-1)} x_{i'} - r\right)}{|A^c(j-1)|} < 0.
$$

as $y_k(j-1) < 0$ for all $k \in B(j-1)$. Hence the statement holds for all $k \in A(j)$. □

Lemma 6 for the case where $\epsilon = 0$ follows from Lemma 11 and Theorem 6 by setting $r = 1$. To deal with $\epsilon > 0$, observe that $\Pi_{D_\epsilon}(\cdot)$ is the unique solution of the following minimization problem

$$
\begin{aligned}
\text{Minimize:}\quad & \|\vec{y} - \vec{x}\|_2 \\
\text{subject to:}\quad & \sum_i y_i = 1 \\
& y_i \ge \epsilon, \quad \text{for all } i
\end{aligned}
$$

By setting $y'_i = y_i - \epsilon$, we can transform this to the equivalent problem

$$
\begin{aligned}
\text{Minimize:}\quad & \|\vec{y}' - \vec{x} + \epsilon\mathbf{1}\|_2 \\
\text{subject to:}\quad & \sum_i y'_i = 1 - N\epsilon \\
& y'_i \ge 0, \quad \text{for all } i
\end{aligned}
$$

Hence, projecting $\vec{x}$ to D_ϵ is equivalent to projecting $\vec{x} - \epsilon\mathbf{1}$ to $D(1 - N\epsilon)$, and can be solved using the algorithm outlined above. In particular, we have the following corollary of Lemma 11.

COROLLARY 2. *For every $\vec{x} \in \mathbb{R}^N$, there exists a set $A \subsetneq [N]$ (depending on $\vec{x}$) such that*

1. $\Pi_{D_\epsilon}(\vec{x}) = \Pi_{V_A(1-N\epsilon)}(\vec{x} - \epsilon\mathbf{1}) + \epsilon\mathbf{1}$, *and*
2. *if $i \in A$ then $x_i - \frac{1}{|A^c|}\left(\sum_{j\in A^c} x_j - 1 + N\epsilon\right) < 0$.*

PROOF. Follows from the fact that $\Pi_{D_\epsilon}(\vec{x}) = \Pi_{D(1-N\epsilon)}(\vec{x} - \epsilon\mathbf{1}) + \epsilon\mathbf{1}$, from Theorem 6 and from Lemma 11. □

Note that Lemma 6 follows as a direct consequence of the above corollary.