Max-Intensity: Detecting Competitive Advertiser Communities in Sponsored Search Market

Wenchao Yu†, Ariyam Das†, Justin Wood†, Wei Wang†, Carlo Zaniolo†, and Ping Luo‡
†Department of Computer Science, University of California Los Angeles, CA 90095, USA
‡Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
{wenchaoyu, ariyam, juwood03, weiwang, zaniolo}@cs.ucla.edu, luop@ict.ac.cn

***Abstract*—In a sponsored search market, the problem of measuring the intensity of competition among advertisers is increasingly gaining prominence today. Usually, search providers want to monitor the advertiser communities that share common bidding keywords, so that they can intervene when competition slackens. However, to the best of our knowledge, not much research has been conducted in identifying advertiser communities and understanding competition within these communities. In this paper we introduce a novel approach to detect competitive communities in a weighted bi-partite network formed by advertisers and their bidding keywords. The proposed approach is based on an advertiser vertex metric called *intensity score*, which takes the following two factors into consideration: the competitors that bid on the same keywords, and the advertisers' consumption proportion within the community. Evidence shows that when market competition rises, the revenue for a search provider also increases. Our community detection algorithm *Max-Intensity* is designed to detect communities which have the maximum intensity score. In this paper, we conduct experiments and validate the performance of *Max-Intensity* on sponsored search advertising data. Compared to baseline methods, the communities detected by our algorithm have low Herfindahl-Hirschman index (HHI) and comprehensive concentration index (CCI), which demonstrates that the communities given by *Max-Intensity* can capture the structure of the competitive communities.**

I. Introduction

Search providers often divide the entire sponsored search market into different areas of business interests. Each such area, like healthcare, education, food and nutrition, etc. is formally known as a sector. Each sector has a wide spectrum of sponsored keywords that different advertisers bid on through keyword auctions. However, advertisers are eventually charged only when their sponsored ads are clicked by a user [1]. Typically, advertisers open their accounts with the search provider(s) and bid for a set of keywords which they consider to be relevant to their products [2]. Different advertisers, bidding on the same set of keywords, compete against each other for their ad slot in the search results. Thus, the search engine providers often ask themselves *"how is the best way to measure the competition among different advertisers in each sector?"*. This is a very important question from the search providers' perspective, as understanding the intensity of competition among different advertisers can help them monitor a sector better, and if necessary, implement changes in their current policy to increase their revenue.

In traditional retail markets, market concentration measures which are based on firms' profits or market shares [3]–[5] are used to estimate the competition. We can extend this notion to the sponsored search market as well, where we can view the fraction of the total "user clicks" that an advertiser gets on its sponsored ads as its corresponding market share in the sector. The fundamental principles of economic theory suggest that as market competition rises, the revenue for search providers should also increase [6]. This is easily verifiable from our results in Section II. However, analyzing market competition at the sector level over time may not always give adequate insights. For example, if the competition for a sector remains stagnant or increases over time, it does not necessarily mean that the competition among advertisers for different keywords will follow a similar trend. Even in such cases, there will be pockets of advertisers, fiercely competing against each other for certain keywords, thereby contributing towards a higher search engine revenue. Likewise, a low competition for an entire sector does not indicate that all keywords within that sector are experiencing high competition. There can be several groups of keywords which are not fairing well in the market and are hardly sought out by the advertisers. Thus, our primary motivation is to detect small meaningful communities of advertisers, competing against each other, *within* a sector. These communities offer an appropriate microscopic view that will allow us to extract relevant insights like finding core and fringe competitors of advertisers, studying how the competition evolves, etc.

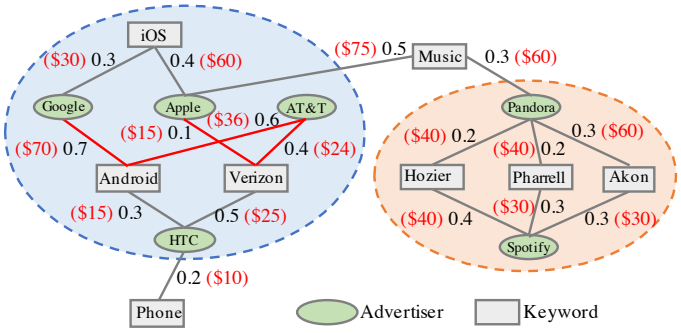

Fig. 1: Weighted advertiser-keyword bipartite graph

Traditional community detection algorithms [7]–[10], running on network graphs, allow us to discover groups of tightly connected vertices and their inter-relationships. In order to run these community detection methods, we modeled our sponsored search market as a bi-partite advertiser-keyword

network, as shown in Figure 1 (described in detail in Section III). However, recent studies in community detection mainly focus either on improving the existing modularity-based methods [11], [12], or proposing new metrics to better estimate the community structure [13]. But, for our advertiser-keyword network, we need to detect communities based on their internal competition with each other. Therefore, in this paper, we first introduce a novel scoring function called *intensity score* to measure the competitiveness of an advertiser. Thereafter, we propose a community detection algorithm *Max-Intensity* which detects highly competitive communities by maximizing the total intensity score within each community.

The principal contributions of this paper are summarized as follows:

1) We propose a new metric *intensity score* to represent the competition in an advertiser-keyword network (modeled as a weighted bi-partite graph).
2) Our proposed *Max-Intensity* algorithm, based on *intensity score*, detects competitive communities within a sector. To the best of our knowledge, this is the first competition based algorithm proposed in the literature.
3) We extend the concepts of market concentration measures from retail markets to calculate the competition in a sponsored search marketplace.

The rest of the paper is organized as follows. Section II shows the strong correlation between market competition and search provider revenue, and also explains the motivation to measure the competition within a sector. Section III describes our data model and the concepts that lead to the formulation of *intensity score*. Section IV presents the community detection algorithm *Max-Intensity*, followed by experimental results described in Section V. The survey of the related work is presented in Section VI. Finally, we conclude in Section VII.

For reproducible research, we make all of our code available online.[1]

II. Preliminaries

Market concentration measures are commonly used in retail markets to estimate the competition among the stakeholders. One such popular measure is the Herfindahl-Hirschman index (HHI) [4], which is widely regarded by economists as an excellent indicator for market competition. HHI for a market having N firms is calculated as,

$$\text{HHI} = \sum_{i=1}^{N} s_i^2 \tag{1}$$

where s_i is the market share of firm i in the market, and N is the number of firms. Thus, in a market with two firms each having 50% market share, the HHI will be $0.5^2 + 0.5^2 = 0.5$.

Another alternative market concentration measure commonly used for estimating competition is the comprehensive concentration index (CCI) [5]. CCI for a market having N firms is calculated as,

$$\text{CCI} = s_1 + \sum_{i=2}^{N} s_i^2 \times (2 - s_i) \tag{2}$$

where s_1 is the market share of the largest competitor. Thus, in a market with two competitors having shares of 40% and 60%, the CCI will be $0.6 + 0.4^2 \times (2 - 0.4) = 0.856$. CCI puts greater weight on the share of the largest firm as compared to HHI. But typically, a low value of HHI or CCI indicates high market competition, whereas a high HHI or CCI would practically indicate a monopoly. We use both HHI and CCI metrics to validate our results.

In order to apply HHI and CCI in the context of a sponsored search marketplace, we can consider "user clicks" as sales for an advertiser and the fraction of total clicks garnered by the advertiser on its sponsored ads as its market share. With these considerations, we calculate Spearman's rank correlation coefficient between HHI and the search engine revenue over seven weeks for different sectors. The results are summarized in Table I. For each sector, we average the user clicks and advertisers' consumption each week, and then calculate HHI of each sector (sector is a "market" here). It is evident from the table that there is a very strong negative correlation between the HHI for a sector and the revenue the search engine obtained from that sector. This indicates that an increase or decrease in the market competition can lead to a corresponding rise or drop in the search provider's revenue as well. Therefore, it is critical that search engine providers identify scenarios where market competition is becoming stagnant or decreasing so that it can come up with remedial strategies [3], such as free token distributions to intensify the competition.

However, sector level analysis offers a very broad macroscopic view. It is difficult to gain relevant insights into the market competition at the sector level. For example, a low HHI value does not suggest that all keywords within that sector are facing high competition, and vice versa. Thus, it is much more useful for search providers to find different pockets of advertisers and their corresponding keywords *within* a sector and track their competition over time. Our proposed community detection algorithm *Max-Intensity* identifies these competitive advertiser communities and their corresponding bidding keywords.

III. Data Modeling and Definitions

In this section, we first describe our data model and the intuition behind it. Then we formally define the *intensity score* and evaluate its boundary conditions.

A. Data Modeling

In order to deploy the community detection algorithms, we need to model the sponsored search market as a graph. In our model, we consider both keywords and advertisers as vertices. In this keyword-advertiser graph, an edge exists between a keyword and an advertiser, if the advertiser bids for that keyword and has paid some remuneration to the search provider for it. If an edge exists between an advertiser and a keyword, we say that the advertiser "consumed" the keyword and the corresponding "consumption" is measured by the amount paid by the advertiser to the search provider for that keyword. Since, in our model, there cannot be any advertiser-advertiser or keyword-keyword edges, we essentially have a bi-partite graph. However, all keywords are not equally important

[1] https://github.com/ucla-scai/Max-Intensity

TABLE I: Market competition versus search provider revenue

Sector	Criteria	Weekly Averaged Data							Correlation
Instruments	HHI	0.0006	0.0007	0.0007	0.0008	0.0019	0.0017	0.0006	-0.9611
	Revenue	6,059,822	5,876,791	5,494,558	4,614,660	1,593,922	625,657	5,196,103	
Finishing Materials	HHI	0.0009	0.0010	0.0012	0.0025	0.0027	0.0035	0.0011	-0.9417
	Revenue	7,438,719	7,143,498	6,151,826	4,988,395	2,103,629	1,740,237	6,392,471	
Industrial Chemicals	HHI	0.0010	0.0011	0.0011	0.0014	0.0018	0.0026	0.0012	-0.9533
	Revenue	3,175,138	3,042,216	2,862,213	2,347,107	951,526	574,986	2,760,362	
Machinery & Equipments	HHI	0.0003	0.0004	0.0004	0.0005	0.0008	0.0009	0.0004	-0.9976
	Revenue	5,397,096	5,196,760	4,721,726	3,909,659	1,363,837	789,953	4,650,112	
Metallic Materials	HHI	0.0004	0.0004	0.0005	0.0006	0.0011	0.0016	0.0006	-0.9415
	Revenue	3,441,036	3,330,836	2,920,052	2,177,350	701,702	433,938	3,024,788	
Specialty Hospital	HHI	0.0006	0.0006	0.0006	0.0007	0.0010	0.0014	0.0007	-0.9580
	Revenue	81,584,689	79,453,651	76,563,899	66,196,751	34,401,222	23,445,646	71,740,014	

to an advertiser. Therefore, we assign weights to the advertiser-keyword edge according to the remuneration the advertiser pays to the search provider for that keyword.

There is a significant drawback if we treat only advertisers as vertices in the graph. This is because the edges can be established between two advertisers vertices, if there is at least one mutual keyword that both these advertisers have consumed. In this case, there will be too many cliques which may not be meaningful communities. For example, consider a network where five advertisers share a common keyword, two among them share two additional keywords while the remaining three share only one additional keyword. These five advertisers in this model form a clique. However, even if these five advertisers had just one common keyword, they would still have formed a clique. Thus we will end up having too many redundant cliques that goes against the definition of communities (less external connections and more internal connections).

We now formally denote the weighted bi-partite advertiser-keyword graph by $G(A, K, E)$, where A is the set of *advertisers*, K is the set of *keywords* and E is the weighted edge set such that $A \cap K = \varnothing$ and $E \subseteq V \times K$. Let w_{ij} denote the weight of an edge between vertex $i \in A$ and vertex $j \in K$. In the context of our model, $w_{ij} > 0$ is the proportion of the money that advertiser i spent on keyword j. Thus we have

$$w_{ij} = csm_{ij} \times \frac{1}{\sum_{k=1}^{n_i} csm_{ik}} \tag{3}$$

where csm_{ij} (shown with red numbers in Figure 1) represents the consumption of advertiser i on keyword j and n_i is the total number of keywords consumed by advertiser i.

B. Competition Coefficient

Community detection methods [14] partition vertices in a graph into set of groups, also called communities, based on their inter-relationships. Let the subgraph $C(A_C, K_C, E_C)$ of an advertiser-keyword graph $G(A, K, E)$ be such a *community*. In order to measure the degree to which advertisers within a community tend to compete with each other, we propose a *competition coefficient* based on the internal competition within a community.

Definition 1 (Homogeneous Neighborhood)**.** *For a given vertex $u \in A_C \cup K_C$ in a bipartite subgraph C, we define its homogeneous neighbor set as $N(u) = \{v | (u,t) \in E_C \wedge (t,v) \in E_C, t \neq \varnothing, u \neq v\}$, which is a collection of homogeneous vertices.*

Definition 2 (Competition Coefficient)**.** *In a given bipartite subgraph $C(A_C, K_C, E_C)$, the competition coefficient of an advertiser vertex $i \in A_C$ is defined as follows:*

$$cc_i = \begin{cases} \frac{\sum_j \sum_k w_{jk}}{|N(i)|}, & \text{if } N(i) \neq \varnothing \\ 0 & \text{if } N(i) = \varnothing \end{cases} \tag{4}$$

where $j \in N(i)$, $k \in K_C$, $w_{ik} > 0$ and $w_{jk} > 0$.

Leveraging the concept of clustering coefficient in graph theory [15], we propose the competition coefficient to factor the competition an advertiser vertex will face in a bipartite graph. The definition shows that competition coefficient is the sum of the weighted edges of all the competitors that are bidding on the same keywords as the advertiser, normalized by its number of competitors. Consider the example given in Figure 1, the competition coefficient of vertex *HTC* in the community is $cc_{HTC} = \frac{0.7+0.1+0.6+0.4}{3} = 0.6$.

From this definition we can obtain that $cc_i \in [0, 1]$. The maximum value of 1 is obtained when all the homogeneous neighbors spend all their money on the same keywords that are bid on by advertiser i. The lower bound of competition coefficient is 0, which is obtained when no competitors exist ($N(i) = \varnothing$).

Theorem 1. *Given a community $C(A_C, K_C, E_C)$, advertiser $i \in A_C, cc_i = \varphi$. If a new competitor j is introduced in this community, the competition coefficient will increase if and only if*

$$\sum_{w_{ik}>0} w_{jk} > \varphi. \tag{5}$$

Proof: Let, $\sum_l \sum_k w_{lk} = \alpha$, where $l \in N(i), k \in K_C$. We have $\varphi = cc_i = \frac{\sum_l \sum_k w_{lk}}{|N(i)|} = \frac{\alpha}{|N(i)|}$. After adding a new competitor j, $\varphi' = \frac{\alpha + \sum_{w_{ik}>0} w_{jk}}{|N(i)|+1}$. If the competition coefficient increases, $\varphi' > \varphi$, then $\frac{\alpha+\sum_{w_{ik}>0} w_{jk}}{|N(i)|+1} > \frac{\alpha}{|N(i)|}$. Thus we can get $\sum_{w_{ik}>0} w_{jk} > \frac{\alpha}{|N(i)|} \times (|N(i)|+1) - \alpha = \frac{\alpha}{|N(i)|} = \varphi$. ■

C. Intensity Score

Based on the competition coefficient, we formulate the *intensity score* for an advertiser vertex. We assume that if the proportion of advertiser consumption for a keyword is high, then the advertiser's competitive capacity to this keyword is also high. To measure the intensity score of an advertiser within a community, we use the following two criteria:

1) The internal consumption proportion $\sum_{i\in A_C \wedge j\in K_C} w_{ij}$ of the advertiser i inside community C should be more than the maximum consumption proportion to a single external community $\max_{C'} \sum_{i\in A_C \wedge j'\in K_{C'}} w_{ij'}$. This criteria is represented in the intensity computation as the difference between the internal consumption proportion and the maximum external community consumption proportion $\sum_{i\in A_C \wedge j\in K_C} w_{ij} - \max_{C'} \sum_{i\in A_C \wedge j'\in K_{C'}} w_{ij'}$, where C' represents the external communities (more details in case study 1 of Section IV-C). This value will be between -1 when there are no competitors inside the community, and 1 when there are no competitors outside the community. This criteria emphasizes that a vertex is likely to be within a community if its sum of consumption proportions (edges) within the community is large (see Theorem 1).
2) Within a specific community, if the competition intensity is high, then the homogeneous neighbors of the advertiser vertex i should spend more money on the same keywords as i. In this case, its competition coefficient, cc_i, should be high. (more details in case study 2 of Section IV-C). This value ranges from 0 (not competitive) to 1 (full competition).

We aggregate these two criteria to formulate *intensity score* I_i of an advertiser vertex i in community $C(A_C, K_C, E_C)$ as follows:

$$I_i = cc_i + \lambda \left(\sum_{i\in A_C \wedge j\in K_C} w_{ij} - \max_{C'} \sum_{i\in A_C \wedge j'\in K_{C'}} w_{ij'} \right) \tag{6}$$

where $\lambda \geq 0$ is a tuning parameter. We test different values for λ in our experiments. Base on this formulation, the intensity score for HTC in Figure 1 is $I_{HTC} = cc_{HTC} + \lambda(0.3 + 0.5 - 0.2) = 1.2$ (we choose $\lambda = 1$ here).

The intuition behind this intensity measure is based on our observations that an advertiser vertex with high *intensity score* has lots of competitors within its community, and its internal consumption proportion is higher than the maximum consumption proportion to any single external community.

D. Boundary Conditions of Competition Intensity

For vertices that do not have any external connections, I_i is equal to the sum of internal coefficient and λ (i.e. $I_i = cc_i + \lambda$). I_i attains its maximum value of $1+\lambda$ when advertiser i has no external connections and faces full competition (all competitors spend all their money on the same keywords as advertiser i) inside the community. The theoretical lower bound of I_i is $-\lambda$, which is obtained when the competitors of advertiser i are only from outside of the community. For example, this is possible for singleton advertiser community which has only external connections. Therefore, for every advertiser vertex i, $I_i \in [-\lambda, \lambda + 1]$.

The intensity score of the community $C(A, K, E)$ is given by $I_C = \sum_{i\in A_C} \frac{1}{|A|} I_i, I_C \in [-\lambda, \lambda + 1]$. I_C will be closer to $\lambda + 1$ as more vertices inside the community high intensity score. This can happen only when the community has a strong internal structure without any external connections to its vertices and all the keywords inside the community are bid by all the advertisers. If C is a singleton community which has only one advertiser and has only external connections, then $I_C = -\lambda$.

IV. Proposed Approach

In this section, we present the formal definition of competition community detection problem, and develop a community detection algorithm called *Max-Intensity* that identifies communities by maximizing their intensity scores.

A. Objective

Our objective is to deal with the following problem: given a weighted bipartite graph $G(A, K, E)$, partition the vertices of G into subsets so as to maximize the *intensity score* in each detected community. The goal function is,

$$f_{obj} = \max \sum_{A_C \subseteq G} \sum_{i\in A_C} I_i \tag{7}$$

where $C(A_C, K_C, E_C)$ is the detected community. An edge between A and K in $G(A, K, E)$ amounts to an advertiser $a \in A$ bidding on a particular keyword $k \in K$. The weight of an edge $W_{a,k}$ represents the consumption of that particular bid normalized by total consumption for the advertiser. With this input set then the algorithm is set to iteratively assign keyword and advertiser vertices to communities in order to increase the overall intensity of the network.

B. Community Detection with Max-Intensity

Similar to finding subgraphs of a given size with some fitness measures larger than a threshold [14], our problem is also NP-complete. Therefore, we utilize a heuristic approach which strives to obtain a high value of intensity. To achieve this desiderata, the algorithm iterates through the entire advertiser set and tests the assignment of an advertiser to each of it's neighboring communities against no assignment at all. The advertiser is then assigned to the community which results in the highest intensity for the graph overall (pseudocode in Algorithm 1). After each assignment outside of an advertiser's current community, and initially, each keyword is assigned to the community which has the highest average intensity score, as shown in Algorithm 2. An iteration limit is set so as to avoid infinite looping in the event that the assignment of vertices oscillate. By iteratively assigning the advertiser and keyword vertices the algorithm will approach a network with a good intensity score.

C. Case Study

In this section, we study the behavior of *Max-Intensity* algorithm with two simple cases.

Case 1: In this case, we test the advertiser assignment regarding the consumption proportion. The initial choice of vertices in the graph is arbitrary but suppose without loss of generality the vertices are chosen in order (i.e. $A1, A2, ..., A5$). Initially each vertex is assigned to it's seed community ($C1, C2, ..., C5$). The intensity score of an advertiser vertex in it's seed community is $-\lambda$ due to hitting the lower boundary condition (Section III-D). Then the keywords are assigned to the community with the highest average intensity score. Again

Algorithm 1 Max Intensity

Input: A weighted bipartite graph G.
Output: Intensity of advertisers in G; Detected communities

procedure MAX_INTENSITY($G(A, K, E), \lambda$)
  Each vertex is assigned to its seed community
  $Assign\_Keywords(K)$
  $Sum \leftarrow -\lambda \times |A|$
  $Old\_Sum \leftarrow -1$
  do
    $Old\_Sum \leftarrow Sum$
    $Sum \leftarrow 0$
    for $a \in A$ **do**
      $cur\_i \leftarrow Intensity(a)$
      if $cur\_i == \lambda + 1$ **then**
        $Sum \leftarrow Sum + cur\_i$
        continue
      end if
      $cur\_i\_neig \leftarrow 0$
      for $l \in Neig(Neig(a))$ **do**
        $cur\_i\_neig \leftarrow cur\_i\_neig + Intensity(l)$
      end for

      for $C \in Comm(Neig(a))$ **do**
        Move a to community C
        $n\_i \leftarrow Intensity(a)$
        $n\_i\_neig \leftarrow 0$
        for $l \in Neig(Neig(a))$ **do**
          $n\_i\_neig \leftarrow n\_i\_neig + Intensity(l)$
        end for
        if $(cur\_i < n\_i)$ and $(cur\_i\_neig < n\_i\_neig)$ **then**
          $cur\_i \leftarrow n\_i$
          $Assign\_Keywords(K)$
        else
          replace a to its original community
        end if
      end for
      $Sum \leftarrow Sum + cur\_i$
    end for
  while not stopping criterion and $Sum \neq Old\_Sum$
  $Advertiser\_intensity = Sum/|A|$
  return $Advertiser\_intensity$
end procedure

Algorithm 2 Assign Keywords

Input: A partite set of keywords K.
Output: Assigned communities of K

procedure ASSIGN_KEYWORDS(K)
  for all $k \in K$ **do**
    $max\_i \leftarrow -\infty$
    for $C \in Comm(Neig(k))$ **do**
      $cur\_i \leftarrow 0$
      $A \leftarrow \{\, a \mid (a \in Neig(k)) \wedge (a$ is a memeber of $C)\,\}$
      for $a \in A$ **do**
        $cur\_i \leftarrow cur\_i + Intensity(a)$
      end for
      $cur\_i \leftarrow \frac{cur\_i}{|V|}$
      if $cur\_i > max\_i$ **then**
        $max\_i \leftarrow cur\_i$
        Move k to community C
      end if
    end for
  end for
end procedure

without loss of generality assume ties are assigned to the lower ordered vertex (i.e $A1 < A2 < ..., A5$). There are other intutive choices to make in breaking ties, such as assigning the keyword to the advertiser with a higher weight, but for simplicity we will just use this basic ordering. After initialization, each advertiser is assigned to a distinct community with $K1$ and $K2$ assigned to $A1$'s community ($C1$), $K3$ assigned to $A3$'s community ($C3$) and $K4$ assigned to $A4$'s community ($C4$). $C2$ and $C5$ are still singleton communities with $A2$ and $A5$ respectively. The algorithm begins the iteration, choosing

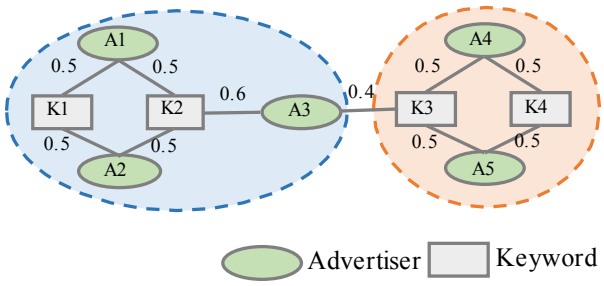

Fig. 2: Case study 1: Advertiser assignment regarding the consumption proportion (results generated from *Max-Intensity*)

$A1$ first. The intensity score will be calculated at this vertex and is given to be 1 because all attached keywords are within the same community. Next $A1$ will stay in $C1$ since all other assignments will lead to a minimum score. This is because all attached keywords are in $C1$, so an assignment to another community will lead to $cc_{A1} = 0$ ($N(A1) = \varnothing$). The intensity score thus becomes $-\lambda$, the leftmost bound. The next vertex chosen will be $A2$. The initial score for $A2$ is -1 because that is the maximum weight of the leaving edges to $C1$. Once $A2$ is placed into $C1$ then the intensity will increase and $A2$ will stay in $C1$. This will result in a call to $Assign\_Keywords(K)$ will place $K3$ into $C4$. $A3$ is next and will follow the same steps as $A2$, this time choosing $C4$. $A4$ will stay in $C4$ by the same reasoning as $A1$. The final advertiser, $A5$ is placed into $C4$ because it will increase the intensity score of the graph, ending the current iteration. The next iteration begins selecting the advertisers in order. $A1$ and $A2$ will remain in $C1$ since all of their keywords are contained in $C1$. $A3$, which at this time is in $C4$, has the choice to stay or move to $C1$. Since the intensity is higher for both $A3$ and $A3$'s neighbors to switch, $A3$ will be placed in $C1$. The remaining two advertisers will stay in their current communities. Since there are no other options, the algorithm will terminate and the result is shown in Figure 2.

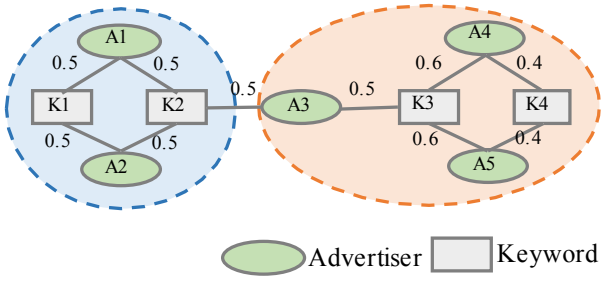

Fig. 3: Case study 2: Advertiser assignment regarding the competitors in the same community (results generated from *Max-Intensity*)

Case 2: In this case, we test the advertiser assignment regarding the competitors in the same community. The iteration will proceed as in Case 1 for the first iteration. On the second iteration, as in Case 1, $A1$ and $A2$ will stay in their communities. Once the vertex $A3$ is taken, the vertex will stay in it's own community rather than adding it to $C1$ which will not increase the intensity of $A3$. The algorithm will proceed

and will not change the communities of $A4$ and $A5$. This leaves the network partitioned into two communities as shown in Figure 3.

D. Effectiveness Analysis

To help in analyzing the complexity of the algorithm, the shorthand notation that is presented along with a description in Table II will be used.

TABLE II: Notations used in effectiveness analysis

Notation	Description
$\|V\|$	number of vertices in G
$\|A\|$	number of advertiser vertices in V
$\|K\|$	number of keyword vertices in V
$\|E\|$	number of edges in G
$\|C\|$	number of communities in G
M	max iteration

Since the algorithm runs in a max total of M times in the worst case this factor is run for each vertex assignment iteration. In the vertex assignment iteration the most dominate time factor comes from iterating through all the communities in the keyword edge set neighbors of an advertiser. This additional factor of $|C|$ will be multiplied to the intensity calculation of the neighbors of the adjoining advertisers of the keyword. This cost will be dominated by a move to a new community which will call the $Assign\_Keywords$ subroutine. Since the $Assign\_Keywords$ subroutine iterates over all keywords this factor of $|K|$ will be multiplied by the run through all communities multiplied by the intensity score of each advertiser neighbor. This total cost so far is $O(M \times |K| \times |C| \times |A| \times |C| \times cost(i))$, where $cost(i)$ is the cost of intensity score calculation which is dominated by the cost associated with calculating the competition coefficient and in the worst case is $|K| \times |A|$ giving a total worst case running time of $O(M \times |K| \times |C| \times |A| \times |C| \times |K| \times |A|) \Rightarrow O(M \times |K|^2 \times |C|^2 \times |A|^2)$. Although this running time is slow in the worst case, actual running times are much faster. This is due to the time complexity reaching its worst case only when an assignment to a new community is made which has the effect of reducing $|K|$, $|C|$, $|A|$ in subsequent iterations.

V. Evaluation

In this section, we provide a brief overview of our sponsored search advertising dataset collected from one search provider, the comparative methods and the evaluation metrics that we used in our experiments, and finally the performance analysis of our Max-Intensity algorithm.

A. Dataset Description

We conducted all our experiments on actual sponsored search advertising data collected for a period of two months. These datasets are structured and capture information about different advertisers in different sectors consuming various keywords. The overall data is available at two different granularities–one at the keyword level and the other at the advertiser level. Each of these datasets are described in Table III.

This data has been curated and anonymized to ensure business secrets and privacy information are not publicly available. For example, we do not know the exact keywords, but instead we have the keyword ids. This, however, does not concern us, since our model and community detection methodology do not require this information. Table IV shows the average statistics for daily sponsored search advertising dataset.

TABLE III: Sponsored search advertising dataset description

Dataset	Attribute	Description
Keyword Level	uid	advertiser's id
	kid	bidding keyword's id
	csp	total advertiser's consumption on this keyword
Advertiser Level	uid_sid	advertiser's id and sector's id
	clk	total user clicks
	csp	total consumption for the advertiser in this sector

TABLE IV: Daily sponsored search advertising dataset statistics

Network	Dataset	Average Statistics
$\|V\|$	advertisers	$\approx 200,000$
	keywords	$\approx 2,000,000$
$\|E\|$	consumption records	$\approx 6,000,000$

There are about 200 sectors in this dataset like IT, electronics, food, nourishment, etc. Figure 4 shows the advertiser distribution among all sectors. In our experiment, we selected the top four popular sectors.

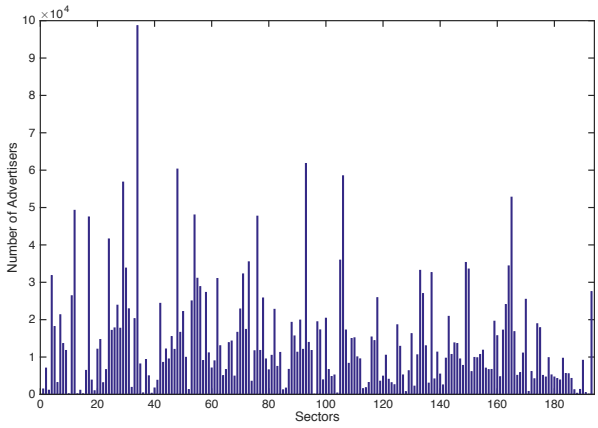

Fig. 4: Advertiser distribution among all sectors

Pre-processing. Given the nature of the dataset, modeling of the data can result in a set of highly disconnected and sparse subgraphs which act as noise in the mining analysis. This is due to sets of keywords being exclusively bid on by a small set of advertisers. Therefore, prior to constructing the advertiser-keyword graph, we pre-process the input to reduce the noise. We filter out the noise by building a set of trees using breadth-first search (BFS) [16] over the entire vertex set of the graph and then considering only those trees whose depth and vertex set size are above user-defined threshold. We demonstrate the filtering process through an example shown in Figure 5 and Figure 6. It is apparent from the figures that the strongly connected clusters in our advertiser-keyword graph are retained, while those subgraphs, which act as outliers, are removed.

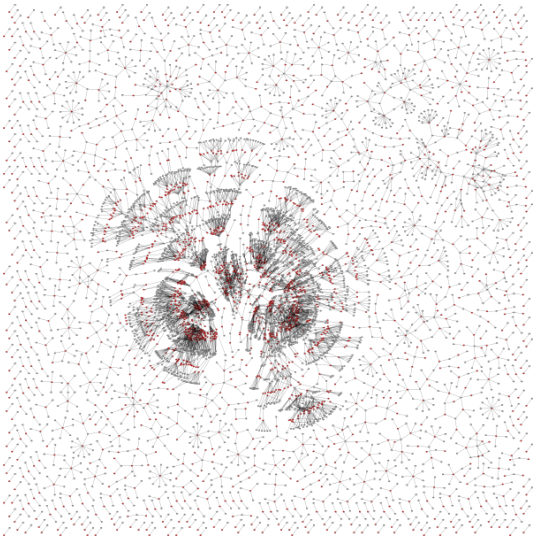

Fig. 5: Input graph before filtering (red vertex represents advertiser, grey vertex represents keyword)

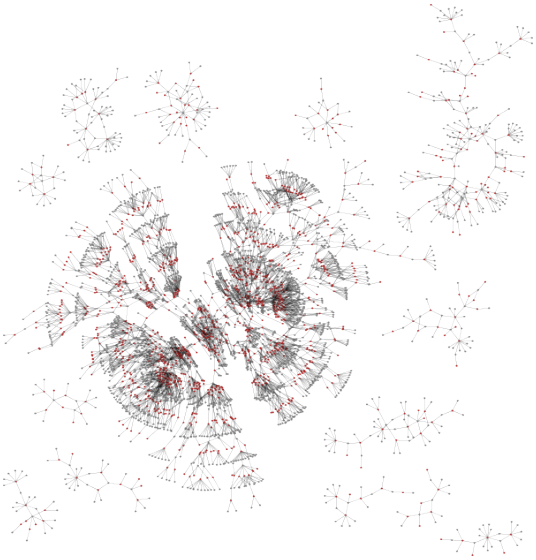

Fig. 6: Input graph after filtering (red vertex represents advertiser, grey vertex represents keyword)

B. Comparative Methods and Evaluation Metrics

In this subsection, we consider the canonical community detection algorithms as well as recent state-of-the-art algorithms. We then evaluate the quality of the detected communities by calculating their Herfindahl-Hirschman index (HHI) [4] and comprehensive concentration index (CCI) [5].

Comparative Methods. The comparative methods we use in this paper are summarized as follows.

1) Information Maps [8]: Information Maps (denoted as **Infomap**) is an information theoretic community detection approach that can be used with weighted and directed networks.
2) Multilevel [10]: **Multilevel** is a heuristic method based on modularity optimization that finds high modularity partitions of large networks in short time and discovers a hierarchical community structure for the network.
3) Leading Eigenvector [9]: The Leading Eigenvector method (denoted as **Eigen**) leverages a modularity matrix to maximize the modules in a network.
4) Max Permanence [13]: Max Permanence (denoted as **Max-Permanence**) detects the community structure by maximizing permanence score, a new vertex-based metric that can quantitatively give an estimate of the community-like structure of the network.
5) Bi-partite Permanence: This (denoted as **Bi-Permanence**) is a modified version of Max-Permanence [13] that detects the communities in a bipartite graph. Here, the modification is made to the permanence scoring function, keeping in mind the bi-partite nature of the graph, so that the score is evaluated at the neighbors of the vertex instead of at the vertex itself. In this case, the clustering coefficient for a vertex v becomes a function of v's neighbors' edges and of v's neighbors, both of which are located inside the community where v is assigned. The same boundary conditions of Permanence can also be applied to Bi-Permanence.

We used the *python-igraph* package for executing the canonical algorithms viz. *Infomap*, *Multilevel* and *Leading Eigenvector*, while we implemented *Max-Permanence*, *Bi-Permanence* and *Max-Intensity* algorithms in C#.

Evaluation Metrics. HHI and CCI will be used as evaluation metrics in this paper. According to the definition of HHI and CCI mentioned in Section II, smaller index scores indicate more competitive communities.

C. Experimental Results

In this section, we first evaluate the competitiveness of the communities detected from different algorithms by comparing their HHI and CCI scores. Then we test the performance of our Max-Intensity algorithm for different λ values. Lastly, we examine the computational complexity of our Max-Intensity algorithm.

Community Competitiveness Evaluation. To compare the competitiveness of communities detected by different algorithms, we run our experiments on one month dataset for the four sectors mentioned previously. Here we choose $\lambda = 1$ for all our experiments.

Figure 7 shows the cumulative proportion of communities against different HHI and CCI values for various algorithms averaged across four sectors. In this figure, x-axis represents the HHI (above) and CCI (below) values while y-axis represents the cumulative community proportion accordingly (averaged over four sectors). For example, a point $(0.4, 0.6)$ on the curve means 60% of all the detected communities by a certain method has HHI or CCI ≤ 0.4. According to the definitions, smaller HHI and CCI values indicate more competitive communities. We can see from the figure that the cumulative proportion of competitive communities detected by our algorithm *Max-Intensity* is significantly higher than other methods, particularly for lower HHI or CCI values (left of the vertical line, HHI=0.4 or CCI=0.4). The nature of the curves for *Max-Intensity*, *Bi-Permanence*, *Max-Permanence* and *Infomap* are very similar. However, the curves corresponding to

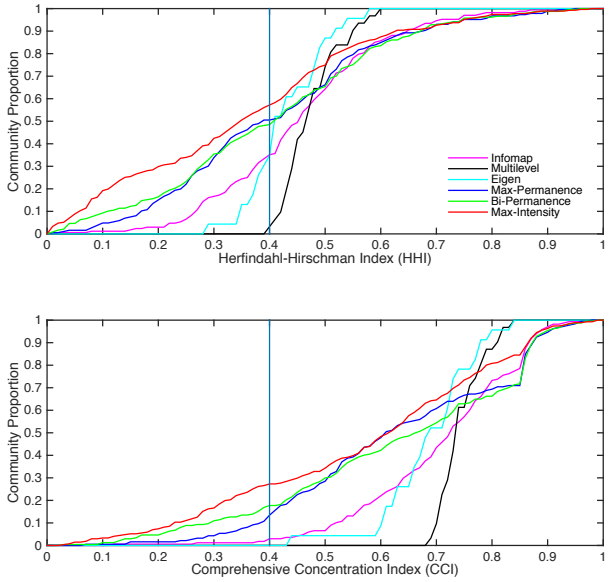

Fig. 7: Cumulative proportion of competition communities detected by each method

Multilevel and *Eigen* rise steeply from moderate HHI values and ends abruptly. This is because, both these methods detect very few communities, each with a large number of community members, which fail to capture the actual competition in the marketplace. In other words, both *Multilevel* and *Eigen* cannot detect either intense competitive communities or communities without significant competition, instead they merge smaller community structures to get an overall moderate HHI. This is primarily because modularity-based community detection algorithms (like *Multilevel* or *Eigen*) suffer from resolution limit problems and end up selecting very few but large communities. This is conclusively shown in table V.

TABLE V: Community number and average HHI/CCI of each method

Methods	#Comminities	Average HHI	Average CCI
Infomap	164	0.4542	0.7100
Multilevel	31	0.4730	0.7439
Eigen	23	0.4278	0.6868
Max-Permanence	271	0.4050	0.6292
Bi-Permanence	237	0.4008	0.6264
Max-Intensity	186	**0.3488**	**0.5725**
Average	152	0.4183	0.6615

Table V displays the average number of communities detected by each method and their corresponding average HHI and average CCI values (across all the communities). As shown in the table, *Max-Intensity* achieves the lowest average HHI and average CCI values.

We also tabulate the cumulative proportion of competitive communities under different HHI and CCI thresholds with their respective standard deviations from 0.1 to 0.4 with an interval of 0.1 in Table VI. We can observe from the table that, when HHI ≤ 0.1, the cumulative proportion of communities detected by *Max-Intensity* is 1.5 to 10 times higher than its closest competitors namely *Infomap*, *Max-Permanence* and *Bi-Permanence*. Similarly, when CCI ≤ 0.1, the cumulative proportion of communities detected by *Max-Intensity* is 1.8 to 20.5 times higher.

Parameter Analysis. The *intensity score* defined by Formula (6) is dependent on the parameter λ. The value of λ decides the relative importance between the competition coefficient and the difference between advertiser's consumption within the community and outside. If the value of the latter difference is high, it implies that the advertiser has less capacity to compete outside this community. By varying the parameter λ, we want to examine which component of the intensity score is more important to identify competitive communities in the sponsored search market. The special case of $\lambda = 1$ indicates that both these components are equally important. We choose different values of λ from 0.001 to 20 and compute the corresponding average HHI and CCI values. The results are summarized in Figure 8. It is evident from the figure that with $\lambda < 1$, we get more competitive communities (with low HHI and CCI values). However, the curves corresponding to $\lambda < 1$ follow a broad "U" pattern. This means that as λ increases from 0 to 1, the average HHI or CCI decreases till it attains its optimal value, after which it starts increasing. In this case, we found that $\lambda = 0.1$ gave the best HHI and CCI values. Likewise, for $\lambda \geq 1$, the average HHI or CCI values sharply increase. Thus, we can infer that the competition coefficient is significantly more important in the definition of the intensity score.

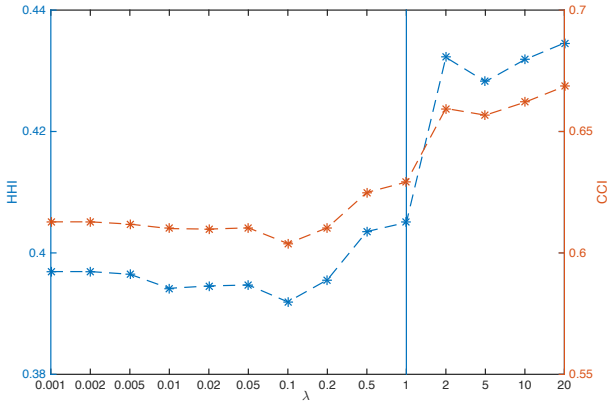

Fig. 8: Average HHI and CCI for different λ

Computational Analysis. Lastly, we examine the computational time for *Max-Intensity* and compare it with the *Bi-Permanence* algorithm which has the second best performance in competitive community detection results, as shown in Figure 7. In this experiment, we ran the algorithms on different graphs with varying number of edges. The running times of the algorithms averaged over 10 iterations are plotted against the number of edges in Figure 9. The results show that the running time of *Bi-Permanence* is only marginally faster than our *Max-Intensity* algorithm, since these two algorithms have similar time complexity. However, our method *Max-Intensity* can detect more competitive communities than *Bi-Permanence*.

VI. Related Work

In this section, we will highlight the literature that related to this paper. We discuss these in terms of community detection

TABLE VI: Cumulative proportion of competition communities under different HHI/CCI threshold

Methods	HHI				CCI			
	≤ 0.1	≤ 0.2	≤ 0.3	≤ 0.4	≤ 0.1	≤ 0.2	≤ 0.3	≤ 0.4
Infomap	1.62%±0.0060	4.55%±0.0147	17.02%±0.0242	39.43%±0.0688	0.15%±0.0031	1.00%±0.0047	1.47%±0.0031	4.09%±0.0071
Multilevel	0.00%±0.0000	0.00%±0.0000	0.00%±0.0000	10.88%±0.0655	0.00%±0.0000	0.00%±0.0000	0.00%±0.0000	0.00%±0.0000
Eigen	0.00%±0.0000	1.76%±0.0214	2.85%±0.0205	25.83%±0.0640	0.00%±0.0000	0.00%±0.0000	0.68%±0.0135	1.76%±0.0214
Max-Permanence	4.36%±0.0128	15.58%±0.0269	32.60%±0.0339	48.25%±0.0322	0.40%±0.0050	1.35%±0.0062	5.18%±0.0192	13.79%±0.0214
Bi-Permanence	10.45%±0.0175	19.93%±0.0355	34.80%±0.0340	53.21%±0.0417	1.73%±0.0051	5.64%±0.0143	12.16%±0.0140	19.12%±0.0392
Max-Intensity	**16.02%**±0.0218	**27.44%**±0.0188	**41.88%**±0.0111	**59.52%**±0.0170	**3.07%**±0.0037	**8.41%**±0.0071	**15.07%**±0.0116	**24.00%**±0.0250
Average	5.41%	11.54%	21.52%	39.52%	0.89%	2.73%	5.76%	10.46%

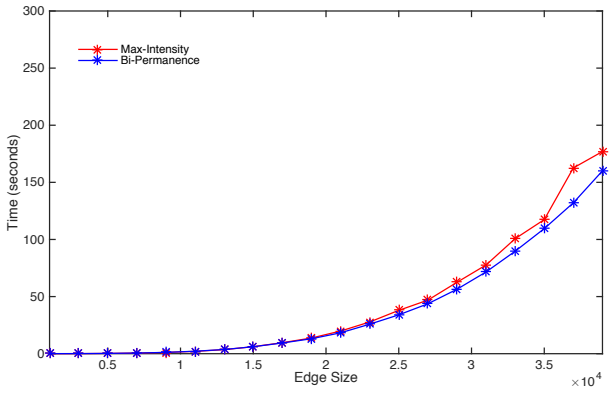

Fig. 9: Running time under different graph size

and competition measure which are mentioned in this paper.

A. Community Detection

Traditional methods involve clustering or partitioning the graph in a way as to discover communities. Often they require parameters which must be specified a priori, such as k-means clustering [17], or are highly dependent on a defined similarity measure as in hierarchical clustering [18]. Hierarchical clustering, which also has the advantage of detecting the hierarchical structure of communities, can be split into two categories: agglomerative, and divisive algorithms. Agglomerative algorithms are a bottom up approach where clusters are merged together using a similarity metric in a series of iterations. The approach of divisive algorithms is top down starting with large clusters and then iteratively breaking them apart. A well known and widely used community detection metric is modularity [7], which is a comparison of edges inside a cluster to the expected number of randomly distributed edges. Methods using this quality function attempt to maximize modularity of partitions in a graph. Since obtaining the maximal modularity has been shown to be a NP-Complete problem [19], approximation techniques must be used. This set of techniques include greedy, simulated annealing, extremal optimization, and spectral optimization techniques. Although the techniques can be quite good at estimating the maximal modularity, the metric may suffer some shortcomings in detecting "good" partitions [20]. Other methods such as spin models, random walks, and synchronizing can be described as a set of stochastic algorithms, and can be used for detecting communities. Spin models originated in statistical mechanics [21], and consist of a system of spins in different states. By applying these spin variables to vertices of a graph then clusters of the graph can be discovered by identifying like-valued spin clusters. In random walks [22] it is likely that a random walker will spend more time inside a community than crossing between different communities. One approach which exploits this assumption to find communities is to define a distance measure based on random walks [23] and then finding "close" vertices. The application to community detection involves placing oscillators at vertices, which are initially in random phases, and detecting which oscillators synchronize first [24].

To address the resolution limit problem of Modularity-based methods [7], recently Duan et al. proposed an approach to incorporate correlation analysis into the modularity-based method by subtly reformatting their math formulas and objective functions [12]. Recent studies also include *Permanence*, a vertex-based metric proposed to identify the community structures in the graph [13], aiming at estimating the internal and external connectivity of the vertex to individual communities. Another approach based on heat kernel is to compute this graph diffusion and use that to study the communities that it produces [11]. Besides, to identify the members of an potential but unlabeled community in large-scale social network, one can apply seed expansion to find remaining community members outside current dataset given sample community members [25].

B. Competition Measure

Sponsored search accounts for the overwhelming majority of income for search provider [1], [2], [26]. A better understanding about the bidding behavior of the advertisers in a search market plays an important role. Also, in order to test the policy effectiveness such as sending coupon codes to advertisers, search providers may want to measure the competition among them so that it can intervene when competition slackens. In sponsored search market, there is no particular measure that used to solve this issue. But here we can view the fraction of "user clicks" of each advertiser as its market share then deploy the traditional measures that used in competition analysis between companies.

Competition measures like price cost margin (PCM), Herfindahl-Hirschman index (HHI) and comprehensive concentration index (CCI) [3]–[5], [27], [28] are widely applied in technology management and related areas. However, the theoretical foundations of PCM as a competition measure are not robust [29], [30]. In order to assist the economic policy and research, a new measure is introduced based on firms' profit [3], which is showed to be more robust than PCM. Another robust competition measure relative profits (RP) shows that more intense competition increases the profits of a firm relative to a less efficient firm [31].

VII. Conclusion

In this paper, we developed a novel approach *Max-Intensity* to detect competitive communities for weighted bi-partite network formed by advertisers and their bidding keywords. This approach iteratively assigns keyword and advertiser vertices to communities in order to increase the overall intensity score of the network. The intensity score we proposed in this paper takes into consideration two factors: the competitors that bid the same keywords and the advertisers' consumption proportion within the community. We compared our algorithms with canonical community detection methods like *Infomap* [8], *Multilevel* [10] and *Eigen* [9], and the recent state-of-the-art methods like *Max-Permanence* [13] as well as the bipartite version of it (*Bi-Permanence*). We then used HHI and CCI as evaluation metrics to measure the competition within the detected communities. Compared to these baseline methods, the communities detected by *Max-Intensity* algorithm have the lowest HHI and CCI values, thereby demonstrating that our *Max-Intensity* identifies more competitive communities. In summary, this paper

1) proposes a new weighted bi-partite graph metric, *intensity score*, which captures the competition of the advertiser-keyword network better.
2) introduces a novel algorithm, *Max-Intensity*, based on *intensity score*, that detects advertiser communities with high competition.
3) applies the existing concepts of market concentration measures from retail markets to evaluate the competition among advertisers in sponsored search market.

Our future work is to extend the *Max-Intensity* algorithm to detect overlapping communities and further analyze the evolving patterns within the communities in each sector.

Acknowledgment

This work was partially supported by NSF IIS-1313606 and NSF IIS-1302698. We thank the anonymous reviewers for their careful reading and insightful comments on our manuscript.

References

[1] T. Graepel, J. Q. Candela, T. Borchert, and R. Herbrich, "Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine," in *Proceedings of the 27th International Conference on Machine Learning*, 2010, pp. 13–20.

[2] C. Perlich, B. Dalessandro, R. Hook, O. Stitelman, T. Raeder, and F. Provost, "Bid optimizing and inventory scoring in targeted online advertising," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 804–812.

[3] J. Boone, "A new way to measure competition," *The Economic Journal*, vol. 118, no. 531, pp. 1245–1261, 2008.

[4] A. O. Hirschman, "The paternity of an index," *The American Economic Review*, pp. 761–762, 1964.

[5] J. Horvath, "Suggestion for a comprehensive measure of concentration," *Southern Economic Journal*, pp. 446–452, 1970.

[6] C. Shapiro, "The 2010 horizontal merger guidelines: From hedgehog to fox in forty years," *Antitrust Law Journal*, pp. 49–107, 2010.

[7] M. E. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[8] M. Rosvall and C. Bergstrom, "Maps of information flow reveal community structure in complex networks," in *In Proceedings of the National Academy of Sciences USA*, 2007.

[9] M. E. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical review E*, vol. 74, no. 3, p. 036104, 2006.

[10] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.

[11] K. Kloster and D. F. Gleich, "Heat kernel based community detection," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 1386–1395.

[12] L. Duan, W. N. Street, Y. Liu, and H. Lu, "Community detection in graphs through correlation," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 1376–1385.

[13] T. Chakraborty, S. Srinivasan, N. Ganguly, A. Mukherjee, and S. Bhowmick, "On the permanence of vertices in network communities," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 1396–1405.

[14] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3, pp. 75–174, 2010.

[15] D. J. Watts and S. H. Strogatz, "Collective dynamics of âĂŸsmall-worldâĂŹnetworks," *nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[16] E. F. Moore, *The shortest path through a maze*. Bell Telephone System, 1959.

[17] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA., 1967, pp. 281–297.

[18] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani, *The elements of statistical learning*. Springer, 2009, vol. 2, no. 1.

[19] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner, *On modularity-np-completeness and beyond*. Citeseer, 2006.

[20] S. Fortunato and M. Barthélemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences*, vol. 104, no. 1, pp. 36–41, 2007.

[21] F.-Y. Wu, "The potts model," *Reviews of modern physics*, vol. 54, no. 1, p. 235, 1982.

[22] B. D. Hughes, "Random walks and random environments," 1996.

[23] H. Zhou and R. Lipowsky, "Network brownian motion: A new method to measure vertex-vertex proximity and to identify communities and subcommunities," in *Computational Science-ICCS 2004*. Springer, 2004, pp. 1062–1069.

[24] A. Arenas, A. Díaz-Guilera, and C. J. Pérez-Vicente, "Synchronization reveals topological scales in complex networks," *Physical review letters*, vol. 96, no. 11, p. 114102, 2006.

[25] I. M. Kloumann and J. M. Kleinberg, "Community membership identification from small seed sets," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 1366–1375.

[26] Y. Chen, W. Liu, J. Yi, A. Schwaighofer, and T. W. Yan, "Query clustering based on bid landscape for sponsored search auction optimization," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 1150–1158.

[27] J. M. Ferguson, *Advertising and Competition: Theory, Mesurement, Act*. Ballinger Publishing Company, 1974.

[28] P. Aghion, N. Bloom, R. Blundell, R. Griffith, and P. Howitt, "Competition and innovation: An inverted u relationship," National Bureau of Economic Research, Tech. Rep., 2002.

[29] R. Amir, "Market structure, scale economies and industry performance," *Scale Economies and Industry Performance*, 2003.

[30] J. Bulow and P. Klemperer, "Prices and the winner's curse," *RAND journal of Economics*, pp. 1–21, 2002.

[31] J. Boone, "Competition: Theoretical parameterizations and empirical measures," *Journal of Institutional and Theoretical Economics JITE*, vol. 164, no. 4, pp. 587–611, 2008.