Context-Aware Advertisement Recommendation for High-Speed Social News Feeding

Yuchen Li[‡], Dongxiang Zhang[§], Ziquan Lan[‡], Kian-Lee Tan[§‡]
[‡]*NUS Graduate School of Integrative Science and Engineering, National University of Singapore*
[§]*School of Computing, National University of Singapore*
{liyuchen,zhangdo,ziquan,tankl}@comp.nus.edu.sg

***Abstract*—Social media advertising is a multi-billion dollar market and has become the major revenue source for Facebook and Twitter. To deliver ads to potentially interested users, these social network platforms learn a prediction model for each user based on their personal interests. However, as user interests often evolve slowly, the user may end up receiving repetitive ads. In this paper, we propose a context-aware advertising framework that takes into account the relatively static personal interests as well as the dynamic news feed from friends to drive growth in the ad click-through rate. To meet the real-time requirement, we first propose an *online retrieval* strategy that finds k most relevant ads matching the dynamic context when a read operation is triggered. To avoid frequent retrieval when the context varies little, we propose a *safe region* method to quickly determine whether the top-k ads of a user are changed. Finally, we propose a hybrid model to combine the merits of both methods by analyzing the dynamism of news feed to determine an appropriate retrieval strategy. Extensive experiments conducted on multiple real social networks and ad datasets verified the efficiency and robustness of our hybrid model.**

I. Introduction

Social media ad spending has been rising dramatically in recent years and is expected to reach 24 billion in 2015 [1]. As the dominator in the market, Facebook made an ad revenue of 12.47 billion dollars in 2014, an increase of 58% year-over-year[2]. With the pay-per-click advertising methodology to assess the cost effectiveness, existing social network platforms place great emphasis on delivering matching ads to potentially interested users. They learn a prediction model for each user based on the personal interests and historical activities. When a user logins his/her account, the most relevant ads matching the learned model are embedded in the news feed and presented to the user. However, the model only captures the slowly evolving personal interests of a user, resulting in repetitious ad recommendation. In addition, recent research has shown that, people find targeted advertising to be intrusive since the ads are too relevant to their specific areas of interest [1].

To mitigate the issue, we propose a context-aware ad recommendation framework that takes into account the relatively static personal interests as well as the dynamic news feed from friends to drive growth in the ad click-through rate. We treat the news feed as a dynamic context that provides additional clue in the spatial, temporal and social dimensions for ad recommendation. For example, when a friend posts in Facebook the dining photos in a restaurant, relevant promotion coupons can be recommended. When a friend shows the status in hospital, displaying gift delivery ads is a good choice. Such motivation was also supported by a very recent work from Twitter [2] in which the contents in the tweet stream were taken into account to enhance the click-through prediction rate of advertising.

However, it is a rather challenging task to support social ad recommendation in a highly dynamic context. First, the posting rate and login frequency in Facebook and Twitter are very high. A new post will appear in all the friends' news feed and may cause their top-k relevant ads to be changed. Second, the ad repository is huge, e.g., Facebook has over 1 million advertisers[3], making the top-k query processing rather expensive when the read frequency is very high. To meet the real-time requirement, we first propose an *online retrieval* strategy that adopts existing top-k aggregation algorithms [3], [4] to find the most relevant ads matching the dynamic context when a read operation is triggered. However, when the context varies little, the online retrieval may retrieve the same set of top-k ads repetitively, which is a waste of CPU resources. Thus, we further propose a *safe region* method to quickly determine whether the top-k ads of a user are changed. We guarantee that as long as the dynamic context is located within the safe region, the top-k results remain the same and the cost of repetitive retrieval is saved. Finally, we observed that when the dynamic context vary dramatically, online retrieval is preferred because the safe region can only guarantee the safeness for a short period of time and requires frequent re-construction. Otherwise, safe region technique is a suitable choice. To combine the merits of both retrieval strategies, we propose a hybrid model that analyzes the dynamism of news feed for each user to determine which strategy should be applied.

To sum up, the contributions of this paper include:

1) We propose a new context-aware ad recommendation framework on social networks by considering both long-term user interests and highly dynamic contents in the news feed.
2) We present an online retrieval strategy that obtains k most relevant ads when a read operation is triggered.
3) We devise a safe region technique to avoid repetitive retrieval when the context varies little.
4) We propose a hybrid model to seamlessly combine the merits of the two retrieval strategies.

[1] http://blogs.wsj.com/cmo/2015/04/15/social-media-ad-spending-24-billion/
[2] http://investor.fb.com/releasedetail.cfm?ReleaseID=893395
[3] http://www.forbes.com/sites/roberthof/2013/06/18/you-know-whats-cool-1-million-advertisers-on-facebook/

5) We conduct extensive experiments on real social networks with billions of edges and real ad datasets with millions of tuples. The experimental results show that our hybrid method significantly outperforms the other two retrieval strategies up to 30x speedups.

We first review related works in Section II. The preliminaries of the context-aware ad recommendation are presented in Section III and we devise the online retrieval algorithm in Section IV. The safe region method is introduced in Section V. Subsequently, we propose the hybrid method in Section VI. The experimental results are reported in Section VII. Finally we conclude the paper in Section VIII.

II. Related Work

Pub/Sub System. Pub/sub system has been extensively studied in the past [5], [6], [7], [8], [9], with deployment in a variety of applications including stock market [5], E-commerce [8], location-based services [9], [7], [10], [11] and online advertisement [6]. However there are two main differences between these works and the context-aware ad recommendation. First, pub/sub systems typically focus on boolean expression matching which means there can be a potentially large number of matching events to a user's subscription, or in our case, the matching ads. This is not suitable for social marketing since users would be annoyed by too many ads. In the context-aware ad recommendation, we model it as a ranking problem where only the top-k relevant ads will be posted on news feeds. Second, these works assume subscriptions are static and the indices are built based on such assumptions to efficiently retrieve matching events. However, in our case the most relevant ads should not only match the static user interests but also the contents in the users' news feed. As news feeds change constantly for real world social network, existing solutions cannot be applied. Thus it calls for an efficient solution which caters to the dynamism of users' news feeds to retrieve the most relevant ads.

Top-K Aggregation Query. The other branch of existing work related to our problem is the top-k aggregation query [12], [3]. Consider a database $\mathbb{D}$ where each object $o = (x_1, x_2, \ldots, x_n)$ has n scores, one for each of its n attributes. Given a monotonic aggregation function f, where $f(o)$ or $f(x_1, x_2, \ldots, x_n)$ denote the overall score of object o, the top-k aggregation problem is to find a set of top-k objects in $\mathbb{D}$ with the highest overall scores. Many approaches such as Threshold Algorithm (TA), CA and their variants [3], [4], [13], [14] have been proposed. Since we consider in-memory recommendation without disk I/O, we adopt TA algorithm for top-k ad retrieval as it has been shown to be instance-optimal [3].

Local immutable region (LIR) [15] and gloabl immutable region (GIR) [16] are another two relevant works to our context-aware advertisement recommendation problem. For a given query vector with the respective top-k entities, LIR searches for a valid interval for a given dimension in the query vector such that the top-k entities remain the same, while all other dimension weights are kept constant. However, in our problem, the weights for different dimensions change simultaneously which LIR is unable to handle because of the local nature of LIRs. GIR is able to support simultaneous adjustments to multiple dimensions. Unfortunately, GIR is computationally expensive as it takes minutes or even hours to get the valid region for a given query vector with only 5-8 dimensions. This makes GIR infeasible to handle the dynamic nature of social news feeds. To overcome this issue, we design a series of techniques to quickly compute a subspace of GIR so that the maintenance cost is greatly reduced.

Microblog Search in Social Networks. There has been much effort made to address the problem of microblog search in social networks [17], [18], [19]. Chen et al. introduced a partial index named TI to enable instant keyword search for twitter [17]. Tao et al. proposed an index to search for the microblogs which are ranked by their provenance in the network [18]. Li et al. devised a 3D inverted index to enable efficient microblog search by considering content similarity, time freshness and social relevance [19]. However these indices are designed to search the microblogs whereas in our case the microblogs are used as queries to retrieve relevant ads. This means existing work cannot be applied to our context-aware ad recommendation problem since the dynamism of the query is not considered. Therefore our solution takes into consideration both the property of social graph and the dynamism of microblogs in the news feeds to deliver high-speed ad recommendation.

III. Preliminaries

We study the problem of context-aware ad recommendation for users in a directed social graph $G = (V, E)$. To capture both static and dynamic contexts of users, we model each user profile as a set of weighted topics that capture slowly evolving personal preference; as well as a pool of unread messages that update dynamically and rapidly. We treat the user profile construction as a black-box and any topic extraction or mining techniques [20], [21], [22], [23] can be adopted to transform the historical posts, sharing and other activities into a latent topic space T and the output is a weighted vector H_u with $|T|$ dimensions. Such topical distribution can be assumed to remain stable in a period of time [2].

Given an ad database A, our goal is to recommend k most relevant ads when a user requests for his news feed. Since the ads can also be projected into a $|T|$ dimensional topical vector, we follow previous works [21], [22], [23] to measure the relevance between an ad and the static user profile as

$$\phi_s(u, a) = \sum_{w \in T} \mathtt{rel}(u, w) \cdot \mathtt{rel}(a, w) \tag{1}$$

where $\mathtt{rel}(u, w) \in [0, 1]$ denotes the relevance between a user u and a topic in T.

Our context-aware ad recommendation also needs to take into account the dynamic news feed when measuring the relevance score between a user and an ad. We use a sliding window W_u to store m most recent posts disseminated to user u to serve as the dynamic context for ad recommendation. We apply the same topic modeling technique to project each post in the window to the latent topic space and use $\mathtt{rel}(d, w) \in [0, 1]$ to measure the relevance between a post and a topic. These scores are aggregated and normalized as follows to measure the contextual relevance w.r.t an ad.

$$\phi_d(u, a) = \frac{1}{m} \sum_{d \in W_u} \sum_{w \in T} \mathtt{rel}(d, w) \cdot \mathtt{rel}(a, w) \tag{2}$$

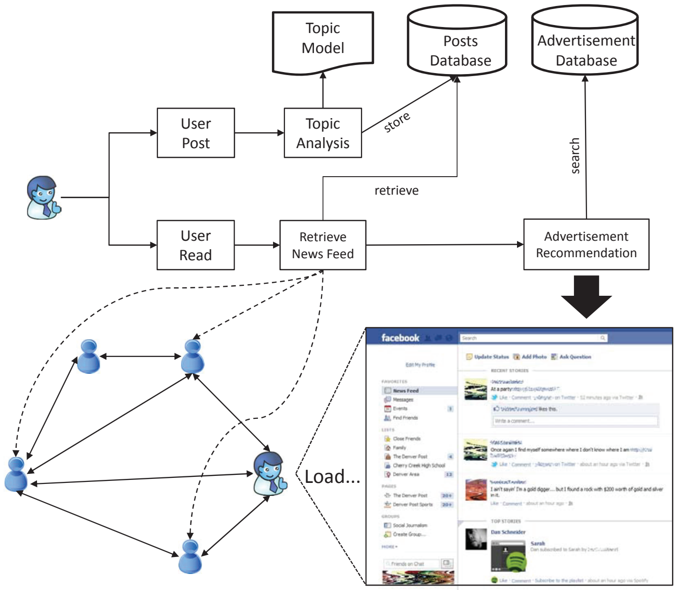

Fig. 1: System Overview of Context-Aware Advertisement Recommendation in Social Networks

Finally, the total relevance of an ad a and a user u is a linear combination of u's static interests and dynamic context:

$$\phi(u,a) = \alpha \cdot \phi_s(u,a) + (1-\alpha) \cdot \phi_d(u,a) \tag{3}$$

$\alpha \in [0,1]$ is a system parameter to balance the importance between personal interests and dynamic context and can be set based on the application requirements. In general, when α is close to 1, the user profile evolves slowly and less maintenance efforts are required. When α is close to 0, the ad recommendation relies mainly on the dynamic context, which makes efficiency a challenging issue. Based on the ranking function in Eqn. 3, we formally define our problem as follows:

Definition 1. *For any user u, the context-aware ad recommendation finds a set of ads, i.e. R, which has a size of k and satisfies $\phi(u,a) \geq \phi(u,a')$ $\forall a \in R \wedge \forall a' \in A \setminus R$.*

Fig. 1 illustrates the system overview. Each user in the social network is considered as both a subscriber and a publisher. When a user composes, shares or likes a post, we say the user, as a publisher, triggers a **write operation**. The new post is first sent for topic analysis, stored in the posts database and later may be retrieved to appear in the news feed of his social friends. When a user logins or refresh his/her news feed, we say the user, as a subscriber, triggers a **read operation**. Then, the posts from friends are retrieved and sorted chronologically and a sliding window containing m recent unread posts are returned. The topic distributions in the dynamic news feed are aggregated with the static personal profile vector as in the ranking function in Eqn. 3 to query the ad database. We call the aggregated vector **context-aware query vector**, denoted by Q_u. In the following section, we show how to handle such top-k query in a large ad database. All the frequent notations used in this paper are listed in Table I for ease of reference.

TABLE I: Frequent notations used across the paper

Notation	**meaning**
G, V, E	the social network, the vertex set and the edge set respectively.
T	the topic space.
A	the set of all ads
u, v	users in social networks.
m	number of posts that appeared in the current window of a user's news feed.
k	number of ads posted in a user's news feed.
$\texttt{rel}(u,w)$ $\texttt{rel}(d,w)$ $\texttt{rel}(a,w)$	the relevance of a user u, a post d and an advertisement a against a given topic w respectively.
α	the weight parameter between $[0,1]$ to balance the the importance between personal interests and dynamic context.
$\theta(x,y)$	the angle between two vectors x and y.
Q_u	the context aware query vector for user u.
Q_u^{lb}, Q_u^{ub}	upper and lower bound vectors for the safe region of user u.

IV. Online Retrieval Algorithm

Existing social ad recommendation systems learn a model for personal interests offline. Since the model is relatively static, the top-k relevant ads for each user can be computed offline and returned together with the news feed when a read operation is triggered. However, when the dynamic context is taken into account in the ranking function, we are unable to pre-compute the ads for each user because each write operation will cause the news feed of all the friends to vary and the incurred pre-computation cost is unaffordable. In this section, we introduce how to efficiently retrieve the top-k relevant ads on the fly.

When a user u triggers a read operation, we need to retrieve the unread posts from the neighbors of u, sort them in chronological order and obtain a window of m posts as the dynamic context. Then, a straightforward solution is to construct the query vector by combining the topic distributions in the dynamic window and static personal interests and issue a top-k query against the ad database. Without proper indexes, it needs to scan all the ads in order to find k of them with the highest relevance scores, incurring very high computation cost. To efficiently handle the top-k query processing, we propose to rewrite the ranking function in Eqn. 3 and apply existing aggregation methods. In particular, we have

$$\begin{aligned}&\phi(u,a) = \alpha \cdot \phi_s(u,a) + (1-\alpha) \cdot \phi_d(u,a)\\ &= \sum_{w \in T} \underbrace{\left[\alpha \cdot \texttt{rel}(u,w) + \frac{1-\alpha}{m} \sum_{d \in W_u} \texttt{rel}(d,w)\right]}_{Q_u(w)} \cdot \texttt{rel}(a,w)\end{aligned}$$

where $Q_u(w)$ is the aggregated relevance between user u and topic w and is set to $\alpha \cdot \texttt{rel}(u,w) + \frac{1-\alpha}{m} \sum_{d \in W_u} \texttt{rel}(d,w)$.

Now our ranking function $\phi(u,a)$ becomes an aggregation function among the partial relevance in each topic dimension. It consists of two terms $Q_u(w)$ and $\texttt{rel}(a,w)$. $\texttt{rel}(a,w)$ is independent of the dynamic context and can be computed and sorted offline. On the other hand, when the query user u is determined, $Q_u(w)$ becomes a constant and will not affect the order of $\texttt{rel}(a,w)$. Therefore, we can maintain $|T|$ inverted lists for each user, each sorted by $\texttt{rel}(a,w)$. When a read operation is triggered, we can retrieve the sorted lists and directly apply standard top-k aggregation techniques such as *Threshold Algorithm* (TA) [3]. It consists of two main steps

1) Perform a sorted access in parallel to each of the $|T|$ sorted lists. For each document accessed, perform a random access to other topics and compute the aggregated score of $\phi(u, a)$. If the computed aggregated score is one of the k highest we have seen so far, remember the ad and its score.
2) For each list L_i, let $high[i]$ be the score of the last ad seen under sorted access. Define the threshold value B_k to be the aggregated score of $high[i]$ by the aggregation function $\phi(u, a)$. As soon as at least k ads have been seen whose score is at least equal to B_k, the algorithm terminates.

Example 1. *Let the window size* $m = 3$*, the weighting parameter* $\alpha = 0.25$ *and the number of topics* $|T| = 2$*. Given a user u, let* $H_u = (0.4, 0.6)$ *be the topic distributions of his/her static interests. Suppose the topic distributions of the three posts in the window are* $(0.2, 0.8)$*,* $(0.1, 0.9)$ *and* $(1.0, 0)$ *respectively. When u triggers a read operation, the context-aware query vector* Q_u *is calculated as* $Q_u = 0.25 \cdot (0.4, 0.6) + \frac{1-0.25}{3}[(0.2, 0.8) + (0.1, 0.9) + (1.0, 0)] = (0.55, 0.45) = (0.425, 0.575)$*. Suppose* Q_u *is used to query an ad database with four tuples* $\{a_1 = (0.3, 0.9)$*,* $a_2 = (0.4, 0.7)$*,* $a_3 = (0.5, 0.8)$ *and* $a_4 = (1.0, 0)\}$*. To support top-k aggregation, we pre-compute two inverted lists* l_{w_1} *and* l_{w_2} *for the topics and get* $l_{w_1} = \{(a_4, 1.0)$*,* $(a_3, 0.5)$*,* $(a_2, 0.4)$*,* $(a_1, 0.3)\}$ *and* $l_{w_2} = \{(a_1, 0.9)$*,* $(a_3, 0.8)$*,* $(a_2, 0.7)$*,* $(a_1, 0.0)\}$*. By calling the TA algorithm presented above,* a_3 *will be returned as the most relevant ad if k is set to* 1.

V. Safe Region Algorithm

In a social network, the frequency of the read operations is normally much higher than the write operations. The famous 1-9-90 rule of Internet culture states that 90% of the participants of a community only view contents while the rest will edit(9%) or create(1%). Hence, for users who frequently login to check news updates from friends, the online retrieval algorithm in Section IV is not an appropriate choice. This is because the context may vary little in such a short period and it is a waste of CPU resources to repeatedly retrieve the same set of ads.

To tackle the issue, we propose a safe region algorithm that examines whether the top-k ads have changed since the previous user read requests. This can be done efficiently by maintaining a safe region for each user. As long as the new context-aware query vector triggered by a user read operation is still located in the safe region, the top-k ads can be directly presented to the user. Otherwise, we re-compute the new top-k results and update the safe region.

A. Safe Region Construction

In Eqn. 3, the ranking function aggregates the relevance of static user profile and dynamic sliding window in the news feed. The window contains w recent posts and can be represented in the form of a topic vector. When a new post is disseminated to a user, the oldest post in the window expires. The weight of the new window for each topic varies mildly and the top-k relevant ads may remain the same. Thus, by constructing a rectangle in the high-dimensional topic space such that whenever the topic vector of the new window is still located in the rectangle, the top-k relevant ads for the user will not change. We call the high-dimensional rectangle a safe region, denoted by $S = (Q_u^{lb}, Q_u^{ub})$, where Q_u^{lb} stores the lower bound of coordinates in all the dimensions and Q_u^{ub} stores the upper bound.

Algorithm 1: GSR(User u)

```
1  $R \leftarrow$ Use TA to compute the relevant ads against $Q_u$
2  $Q_u^{lb} \leftarrow Q_u, Q_u^{ub} \leftarrow Q_u$
3  while True do
4      $w \leftarrow$ DimensionSelect$(v)$
5      $\vec{\delta} \leftarrow \frac{1-\alpha}{m}\vec{e_w}$
6      for $a \in R$ do
7          $\phi(a) =$ MinS$(a, Q_u^{lb} - \vec{\delta}, Q_u^{ub} + \vec{\delta})$
8      $S_u \leftarrow \min\{\phi(a) | a \in R\}$
9      for $a \in A \setminus R$ do
10         $\phi(a) =$ MaxS$(a, Q_u^{lb} - \vec{\delta}, Q_u^{ub} + \vec{\delta})$
11     $S_l \leftarrow \max\{\phi(a) | a \in A \setminus R\}$
12     if $S_u \geq S_l$ then
13         $Q_u^{lb} \leftarrow Q_u^{lb} - \vec{\delta}$
14         $Q_u^{ub} \leftarrow Q_u^{ub} + \vec{\delta}$
15     else
16         return $(Q_u^{lb}, Q_u^{ub})$
```

In [16], Zhang et al. proposed GIR to compute the maximal safe region such that the topic vector update within the region incurs no change for the current top-k results. However, it is prohibitively expensive to construct the optimal safe region, especially in the high dimensional topic space. The method cannot meet the real time requirement in the social streaming environment. Thus, we propose a Greedy Safe Region (GSR) algorithm to incrementally build a safe region. As illustrated in Algorithm 1, we first store the top-k results for the current news feed window in R and initialize the safe region to be the context aware query vector Q_u (lines 1-2). In the following iterations, we pick the most promising topic/dimension to expand the current safe region (line 3). For each dimension, we first calculate the distance from Q_u to the boundaries of the current safe region (Q_u^{lb}, Q_u^{ub}) in that dimension. Then, we select the dimension with the minimum distance for safe region expansion.

For the selected dimension w, we examine whether it is safe to expand upwards and downwards by an expansion unit $\vec{\delta} = \frac{1-\alpha}{m}$, since $\frac{1-\alpha}{m}$ is the maximum possible change in $Q_u(w)$ for each new post. For the expanded safe region, if its minimum relevance to the current top-k ads, denoted by S_l, is still larger than the maximum relevance to those not in R, denoted by S_u, then the expansion is safe. Otherwise, the algorithm terminates and returns the safe region expanded in partial dimensions.

Theorem 1. *For a query vector* Q_u *with its bound vectors* Q_u^{lb} *and* Q_u^{ub} *returned by Algorithm 1, whenever* $Q_u^{lb}(w) \geq x(w) \geq Q_u^{ub}(w) \forall w \in T$*, it corresponds to the same set of top-k ads as* Q_u*.*

We omit the proof since it is trivial according to our explanation about the GSR algorithm.

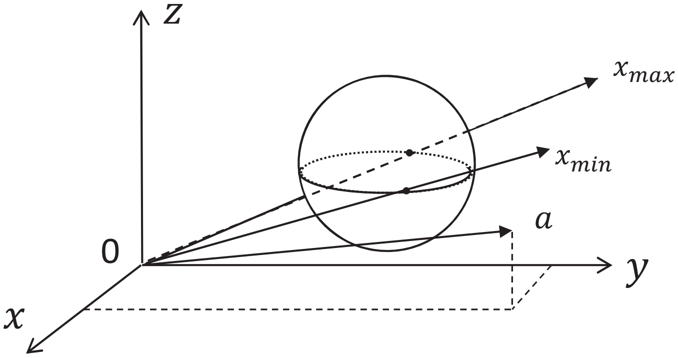

Fig. 2: Computing **MinS** and **MaxS** when the safe region sphere lies in the positive quadrant.

B. Computing MinS and MaxS

To obtain the values of S_u and S_l, we are required to evaluate the minimum and maximum relevance score between an ad and a safe region, denoted by **MinS** and **MaxS** respectively. We can formulate such a problem as the following:

$$\begin{aligned} &\min / \max \quad \sum_{w \in T} \texttt{rel}(a, w) \cdot \frac{x(w)}{\|x\|} \\ &\text{s.t.} \qquad Q_u^{lb}(w) \leq x(w) \leq Q_u^{ub}(w) \;\; \forall w \in T \end{aligned} \tag{4}$$

Note that in Eqn. 4, we divide the objective function by $\|x\|$. A success safe region expansion requires, for any query $x \in (Q_u^{lb}, Q_u^{ub})$, $x \cdot a_u \geq x \cdot a_l \; \forall a_u \in R$ and $\forall a_l \in A \backslash R$. This means the norm of x should not be taken into account of the relevance score between an ad and the safe region. If the normalization is not applied, **MinS** would choose Q_u^{lb} and **MaxS** would choose Q_u^{ub} as solutions of Eqn. 4. Since $\|Q_u^{lb}\| < \|Q_u^{ub}\|$, it incurs an underestimation of **MinS** and an overestimation of **MaxS**, resulting in a much smaller safe region.

Eqn. 4 is essentially an optimization problem of finding two vectors in the rectangular area defined by the safe region bound vectors (Q_u^{lb}, Q_u^{ub}), which have the minimum and the maximum cosine similarities respectively against an ad vector a. In other words, **MinS** and **MaxS** correspond to vectors x_{max} and x_{min} in the rectangular safe region which have the **maximum** and the **minimum** angles respectively to a. However, it is inefficient to find the exact solution due to the nonlinear term in the objective function. To solve the issue, we propose to use a sphere that encloses the safe region constructed so far. Based on the sphere, we calculate **MinS** and **MaxS** to determine a termination of the GSR algorithm or further expansion of the current safe region.

Let x_c be the vector that passes the origin and the center of the rectangular safe region, i.e. $x_c = \frac{1}{2}(Q_u^{lb} + Q_u^{ub})$. We apply a minimum sphere to enclose the safe region and replace x_{min} and x_{max} to be the minimum and maximum angles to the bounding sphere. The new angles from an ad a to x_{min} and x_{max} for the sphere can be computed as:

$$\theta(a, x_{min}) = \max\{\theta(a, x_c) - \arcsin(\frac{r}{\|x_c\|}), 0\} \tag{5}$$

$$\theta(a, x_{max}) = \theta(a, x_c) + \arcsin(\frac{r}{\|x_c\|}) \tag{6}$$

where $\theta(.,.)$ denotes the angle between two vectors and r is the radius of the spherical safe region, i.e $r = \|\frac{1}{2}(Q_u^{ub} - Q_u^{lb})\|$.

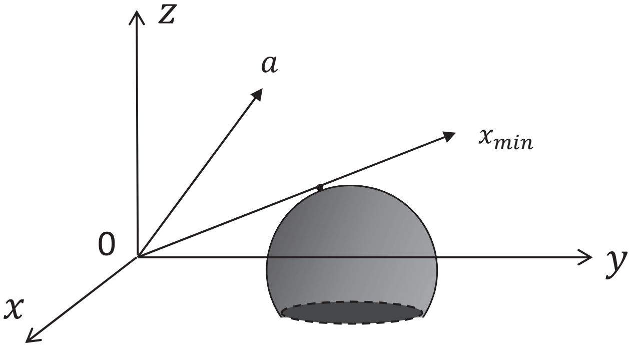

Fig. 3: Computing **MinS** and **MaxS** when the safe region sphere overlaps with x-y plane.

Fig. 2 presents illustrative examples of x_{min} and x_{max} in a 3-dimensional space. In Fig. 2, the whole spherical safe region is a ball that lies in the positive quadrant. It is visually intuitive about the min and max angles between a and the sphere region. In this case, **MinS** and **MaxS** are directly computed via Eqn 5 and Eqn. 6 respectively. Compared to using rectangle to derive **MinS** and **MaxS**, the computation becomes much more efficient but the constructed safe region may be slightly smaller. This is because the sphere encloses the rectangle and it results in smaller x_{min} and larger x_{max}, which consequently leads to larger **MinS** and smaller **MaxS**. In other words, the value of S_l increases but the value of S_u decreases, which makes the GSR algorithm more likely to terminate.

Fig. 3 illustrates a case worthy of our attention. The sphere region is now overlapped with x-y plane. In this case, we can still calculate x_{min} using Eqn. 5 because the ad a and query vector Q_u, which is the center of the safe region, are positive vectors and x_{min} is guaranteed to lie on top of the x-y plane. To calculate x_{max}, as the safe region sphere overlaps with x-y plane, we need to determine if x_{max} still lies on the surface of the sphere or the intersected area between the sphere and $x-y$ plane, which are shown in Fig. 3. The following theorem shows how to determine the location of x_{max} for an ad a.

Theorem 2. *For an ad vector a, let x^*_{max} be the vector obtained by directly applying **MinS** on the spherical safe region. Let I be an index set such that $I = \{i | x^*_{max}(i) < 0\}$ and $S(i)$ be the region where the sphere intersects with the plane $x(i) = 0 \; \forall i \in I$. If I is an empty set, x_{max} can be calculated by Eqn 6. Otherwise,x_{max} is obtained by:*

$$x_{max} = \underset{q}{\text{argmax}}\{\theta(a, q) | q \in S(i) \;\; \forall i \in I\}^4 \tag{7}$$

Readers can refer to the Appendix section for the proof.

Theorem 2 tells us that there are two cases when computing x_{max}. In the first case, if x_{max} lies in the positive quadrant, we compute x_{max} by Eqn. 6. In the second case, when x_{max} goes beyond the positive quadrant, x_{max} must lie in the intersected area between the sphere and boundaries of the positive quadrant. To obtain the exact location of x_{max} for the second case, we project the ad a onto the boundaries of the positive quadrant, where the boundaries contain a piece of

[4] For any point q, we also use q to represent the vector which passes through the origin and q as a point when there is no ambiguity, e.g. in $\theta(a, q)$, q means a vector whereas in $q \in S(i)$, q means a point.

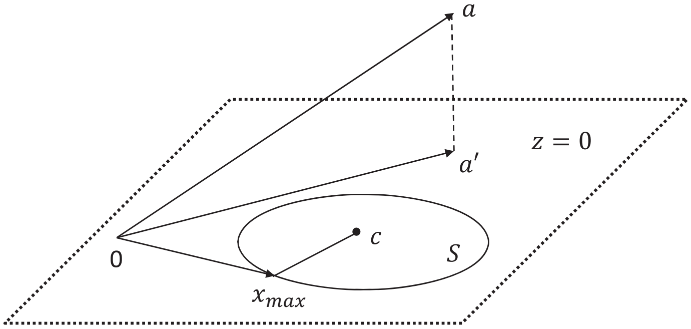

Fig. 4: Example of computing x_{max} when the safe region sphere overlaps with x-y plane.

the safe region sphere. It can be easily proven that x_{max} is exactly the point which has the maximum angle away from a's projection. Fig. 4 shows an example of identifying x_{max}. We plotted the intersection area S between x-y plane and the safe region sphere. a' is the projection of a onto x-y plane and we identify x_{max} from S as the point which has the maximum angle away from a' as shown in Fig. 4.

C. Safe Region Based Query Processing

Having discussed how the safe region can be constructed, we are now ready to present how to process incoming queries triggered by user read operations. We have proved in Theorem 1 that for any query vector x within the safe region formed by (Q_u^{lb}, Q_u^{ub}), the top-k results will be the same for all x. A naive query processing technique for a query Q is to simply check if $Q \geq Q_u^{lb} \wedge Q \leq Q_u^{ub}$. However such a checking rule is too strict and cannot handle the case where the safe region bound vectors and the query vector are not in the same scale. For example, there are two query vectors $Q = (0.3.0.5)$ and $Q^* = (0.15, 0.25)$. Both vectors will have the same top-k ads since $Q = 2 \cdot Q^*$ and the results are invariant under scalar multiplication of the query vector. Let a safe region be defined by $Q_u^{lb} = (0.1, 0.2)$, $Q_u^{ub} = (0.2, 0.4)$, Q is not bounded by (Q_u^{lb}, Q_u^{ub}). However it is easy to see that Q can indeed be processed by the safe region since Q and Q^* share the same result and Q^* is bounded by (Q_u^{lb}, Q_u^{ub}). If the naive checking rule is adopted, a large number of re-computations for new safe regions are needed. Thus, we assign a more flexible checking rule for query processing with the safe region.

Lemma 1. *For any query vector Q_u and a safe region formed by (Q_u^{lb}, Q_u^{ub}), if Q_u intersects the bounding sphere of the safe region, then Q_u will also be in the safe region.*

It is straightforward to prove Lemma 1 according to Theorem 1, therefore we omit the details here. To efficiently check if a query vector Q_u intersects with a sphere, we use Eqn. 5 where the ad vector a is replaced with Q_u. Such checking has a worst time complexity of $O(|T|)$ since we need to compute the angle between two vectors with at most $|T|$ dimensions.

D. Optimizations

To further improve the performance of the safe region method, we propose two optimization techniques. The first seeks to efficiently evaluate S_l and S_u in each iteration of GSR algorithm. The second aims to avoid the online retrieval and safe region re-construction cost when a query vector Q_u no longer exists in its original safe region.

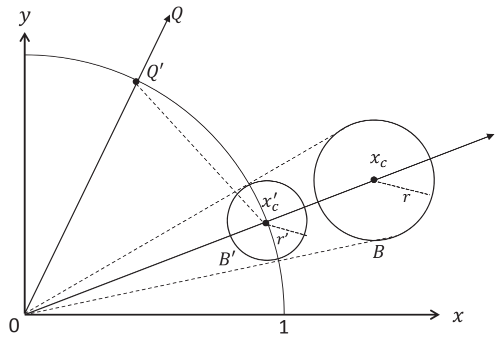

Fig. 5: Example of computing if a SSR contains a query vector

.

In Algorithm 1, we evaluate S_l by computing **MinS** for k times and S_u by computing **MaxS** $|A| - k$ times. This means evaluating S_l requires almost a scan of all ads, which is too computationally expensive to evaluate S_l for a large ad database. This motivates us to develop an efficient algorithm to compute **MaxS** only when necessary.

We design a TA-like approach to evaluate S_l. S_l is the score of the top-1 ad in $|A \setminus R|$, which has the highest **MaxS** score against a safe region. As the ads have been sorted w.r.t each topic by the ads' topic relevance scores, i.e $\text{rel}(a, w)$, we visit the ads in descending order of $\text{rel}(a, w)$ in topic w's inverted list and perform random access to other topics' inverted lists for computing **MaxS**. Meanwhile, we maintain an upper bound score b_w for the inverted list of each topic w and the maximum **MaxS** score of unvisited ads can be bounded by computing **MaxS** for $b = (b_1, .., b_{|T|})$ against the safe region. If the top-1 ad, which has the highest **MaxS** score among all visited ads, has larger **MaxS** score than that of b, we can terminate and return S_l. Such optimization will greatly reduce the number of **MaxS** computations in Lines 6-8 of Algorithm 1.

When the dynamic query vector Q_u deviates out of the safe region of user u, we need to adopt the *online retrieval* to obtain top-k ads on the fly and construct a new safe region in the meanwhile. Such computations are rather expensive. Thus for our second optimization, we propose a new idea to "salvage" the maintained safe regions as well as the associated top-k ads from other users. This is because all the query vectors in a safe region share the same top-k ads. When Q_u moves out of the safe region, we can search all the safe regions of other users. If we can find a safe region from user v that contains the new query vector Q_u of user u, its top-k ads are exactly the same as user u. Moreover, we can assign the safe region of v directly to user u. In this way, the cost of online retrieval and safe region computation can be saved.

To quickly identify whether the new query vector Q_u is contained in the safe regions of other users, we transform the problem into a standard range query in high dimensional space. As shown in Fig. 5, B is a safe region centered at x_c with radius r and Q is a query vector. Our original processing technique is to check if Q intersects with B. By mapping B to B' with new center x_c' and scaling Q to Q' where x_c' and Q' both lie on the boundary of the unit sphere, we transform the vector-sphere intersection problem into checking if the distance from Q' to x'_c is smaller than r'. This is because the minimum bounding convex cone is the same for both B and B'. After the transformation, our goal is to find the MBRs whose distance to a query point is smaller than the radius. The new problem can be efficiently solved by high-dimensional indexes such as k-d-tree [24] or iDistance [25].

VI. Hybrid Algorithm

In this section, we propose a hybrid model to combine the merits of *online retrieval* and *safe region*. Our strategy is to measure the dynamism of topic distributions in the streaming news feed of each user. If the topic distributions in a news feed vary dramatically as new posts flood in, we adopt the *online retrieval* method to avoid the cost of maintaining safe regions that update frequently. Otherwise, the topic distributions are relatively stable and the *safe region* method is suitable for the scenario.

A. Variance of Topic Distributions

To measure the variance of topic distributions, we use i.i.d Poisson process P_u of rate λ_u to model the generation of new posts in a user's news feed as it is frequently used to model the arrival of events. We assume that the number of topics in each post d from user u follows the discrete uniform distribution F_u with range $\{1, 2, ..., f_u\}$. The topics in d are then sampled via a multinomial distribution and each topic is selected with probability $p_{w,u}$. Let $D_{w,u}$ denote the total weightage of topic w in a post d from user u. The whole generative process to build a post for a user u is summarized as follows:

1) Draw a posting time from Poisson process $P_u \sim$ Possion(λ_u)
2) Draw the number of topics for a post $F_u \sim$ Uniform($1, 2, ..., f_u$)
3) Draw topics $D_{w,u}|F_u \sim$ Multinomial($p_{w,u}$)

For P_u, we can estimate the posting rate λ_u by u's historical posting times. For the parameters of F_u and $D_{w,u}$, we can estimate by analyzing all the topic vectors of u's posts.

Let $X_{w,v}$ be the random variable describing the weightage of topic w appears in a user v's sliding window. Given the aforementioned generative process, $X_{w,v}$ can be written as:

$$X_{w,v} = \sum_{n \in N(v)} \sum_{1 \le i \le M_{v,n}} D_{w,n}(F_n) \tag{8}$$

Where $N(v)$ is all v' neighbours and $M_{v,n}$ is a random variable describing how many posts are selected from a neighbour n to form the news feed window of m posts for user v. Then, based on Eqn. 8, the variance of a topic w in a user v's news feed can be defined as:

$$\mathrm{Var}[X_{w,v}] = \mathrm{Var}[\sum_{n \in N(v)} \sum_{1 \le i \le M_{v,n}} D_{w,n}(F_n)] \tag{9}$$

which can be further expanded to

$$\mathrm{Var}[X_{w,v}] = \sum_{n \in N(v)} \frac{(f_n+1)^2 p_{w,n}^2}{4} m\lambda_{v,n}(1-\lambda_{v,n}) - \sum\sum_{\substack{a,b \in N(v) \\ a \neq b}} m\lambda_{v,a}\lambda_{v,b} \frac{(f_a+1)(f_b+1)p_{w,a}p_{w,b}}{4} \tag{10}$$

The derivation process is presented in the Appendix. Here, $\lambda_{v,n}$ measures the probability of selecting a post from a neighbor n for user v. Since all the neighbours of v compose posts that follow i.i.d Possion processes and it has been shown in [26] that the sum of i.i.d Poisson distribution follows a multinomial distribution, we have $\lambda_{v,n} = \frac{\lambda_n}{\sum_{n' \in N(v)} \lambda_{n'}}$.

To calculate $\mathrm{Var}[X_{w,v}]$ according to Eqn. 10, we need to traverse all the pairs of neighbours (a, b) for each user v. Suppose the average node degree in a social network is z, the computation complexity is $O(z^2|V|)$, which is very high for dense social graphs. To reduce the computational cost, we rewrite the term in Eqn. 10 as:

$$\begin{aligned}
&\frac{1}{4}\sum\sum_{\substack{a,b \in N(v) \\ a \neq b}} m\lambda_{v,a}\lambda_{v,b}(f_a+1)(f_b+1)p_{w,a}p_{w,b} \\
&= \frac{1}{4}\sum_{a \in N(v)} m\lambda_{v,a}(f_a+1)p_{w,a} \sum_{\substack{b \in N(v) \\ a \neq b}} \lambda_{v,b}(f_b+1)p_{w,b} \\
&= \frac{1}{4}\sum_{a \in N(v)} m\lambda_{v,a}(f_a+1)p_{w,a} \big(\sum_{b \in N(v)} \lambda_{v,b}(f_b+1)p_{w,b} \big) \\
&- \frac{1}{4}\sum_{a \in N(v)} m\lambda_{v,a}^2(f_a+1)^2 p_{w,a}^2
\end{aligned}$$

In this way, we can pre-compute $\sum_{b \in N(v)} \lambda_{v,b}(f_b+1)p_{w,b}$ for each user v and the complexity is reduced to $O(z|V|)$.

B. Hybrid Retrieval Strategy

$\mathrm{Var}[X_{w,v}]$ only captures the variance of topic distributions in the news feed. We need to further combine it with the static personal interests to measure its impact in choosing an appropriate retrieval strategy. By applying coefficient of variation on the linear combination of static interests and dynamic topic distributions in the news feed, we have

$$\rho(v) = \max_{w \in T} \frac{\frac{1-\alpha}{m}\sqrt{\mathrm{Var}[X_{w,v}]}}{\alpha \cdot \texttt{rel}(u,w) + \frac{1-\alpha}{m} \cdot \mathbb{E}[X_{w,v}]} \tag{11}$$

In addition, the ratio of the read frequency of user v to the write frequency of v's neighbors will also affect the retrieval strategy selection and is ignored in the above model, which only considers the variance of topic distributions for a sequence of write operations. To bridge the gap, we extend $\rho(v)$ by taking the read frequency η_v of v into account.

$$\rho^*(v) = \frac{\sum_{n \in N(v)} \lambda_n}{\eta_v} \cdot \rho(v) \tag{12}$$

Finally, we can use $\rho^*(v)$ to determine the retrieval strategy for user v. If $\rho^*(v)$ is smaller than a pre-defined threshold ρ_{max}, we adopt the *safe region* strategy for user v. Otherwise *online retrieval* is used when v logins/refreshes its personal social page.

TABLE II: All parameter settings used in the experiments. The default values are highlighted.

Datasets	Twitter dataset	News dataset
#Users	10, 20, 30, **40** (M)	0.2, 0.6, 1, **1.4** (M)
#Edges	0.7, 1.1, 1.2, **1.3** (B)	1.0, 1.9, 2.6, **3.1** (M)
AvgDegree	76.4, 56.8, 46.1, **38.9**	5.2, 3.1, 2.6, **2.2**
α	0.1, 0.3, **0.5**, 0.7, 0.9	
k	1, 2, **3**, 4, 5	
R/W	1.0, 2.0, **4.0**, 8.0, 16.0	
#Topics	5, 10, **15**, 20, 25	

VII. Experimental Study

In this section, we study the performance of the three proposed methods on real social network datasets with billions of edges. In the following experiments, we focus on evaluating the efficiency of the proposed methods. The effectiveness evaluation is beyond the scope of the paper because we simply adopt the previous topical mining and relevance measurement techniques which have already been shown to be effective. In addition, the idea of considering newsfeed as a dynamic context was also supported by a recent work from Twitter [2].

We use **Online** to denote the online retrieval method presented in Section IV and use **GSR** and **Hybrid** to denote the methods proposed in Sections V and VI respectively. All the methods are implemented with C++ and run in memory on a CentOS server (Intel i7-3820 3.6GHz CPU with 8 cores and 60GB RAM).

Advertisement Datasets. We use Amazon products [27] and AOL keyword queries[5] as two representative ad repositories. The Amazon dataset consists of $548,552$ products associated with their metadata and review information; whereas in the AOL dataset, there are over 7 million keyword queries. We then apply existing topic modeling method [20] upon the products and keyword queries, resulting in ads in the form of probabilistic topic distributions with fixed number of dimensions.

Social Network Datasets. We use two real datasets, *Twitter* and *News*, from SNAP[6] to evaluate the performance in different structures of social networks. The *Twitter* dataset contains 41.6 million users and 476 million tweets; the *News* dataset contains 1.42 million websites extracted from a collection of 96 million online articles. For the News dataset, the vertex in the graph denotes a website whereas the edge means that there is a link from one website to another. To test the scalability with increasing graph sizes, we sampled 10M, 20M, 30M, 40M nodes for the *Twitter* dataset and 0.2M, 0.6M, 1M, 1.4M nodes for the News dataset.

To simulate a social network with high-speed news feeding, we extract 1 million articles or tweets to generate a sequence of write operations and the order is determined by the timestamp associated with each article or tweet. Each write operation consists of an article or tweet and the associated website or user in the network. The remaining items are used to model static personal interests. We simply apply LDA to the remaining articles or tweets of each user to construct the topic distributions. The input stream to the system consists a sequence of read and write operations. The user who triggers the read operation can be any node in the network. When a write operation arrives, we simply analyze the post and store it in the database. When a read operation arrives, we call the proposed methods to return top-k ads. Since our target is to guarantee the real-time delivery of relevant ads, we are interested to measure the average elapsed time in retrieving the top-k ads for each read operation in the following experiments.

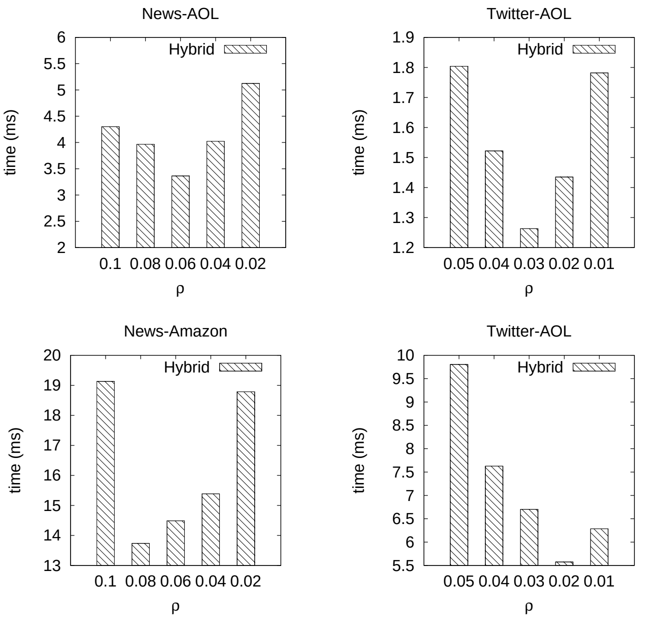

Fig. 6: Vary ρ

Parameters & Settings. As shown in Table II, we evaluate the scalability w.r.t. increasing α (the weight of static interests in the ranking function), k (the number of ads to be embedded in the news feed) and $|V|$ (the number of users in a social network). In addition, we simulate the user activities in social networks with different read/write ratio R/W. The read operation refers to a user login or refreshing the news feed. The write operation means a user composes, likes or shares a post. We will see that the proposed methods have different biases on this parameter. We also evaluate the number of topics from 5 to 25. For the number of posts in one's news feed window, we use the default value ($m = 20$) in Twitter and when a user logins, 20 latest articles/tweets are returned.

A. Tuning ρ

We first investigate the effect of threshold parameter ρ (Equation 11) that indicates how dynamic a user's news feed is in the **Hybrid** model. If ρ is large, the news feed of most users are considered as non-dynamic and they will adopt the **GSR** method for ad recommendation. There may be frequent update of their safe regions, incurring high CPU cost. If ρ is small, most users will adopt the **Online** method to retrieve the top-k ads when a read operation is triggered. CPU resources may be wasted if the read frequency is high but the top-k ads update infrequently. Therefore there exists a sweet spot for ρ. As shown in Fig. 6, we select 0.06, 0.08, 0.03 and 0.02 for News-AOL, News-Amazon, Twitter-AOL and Twitter-Amazon datasets respectively.

[5] http://www.gregsadetsky.com/aol-data/

[6] http://snap.stanford.edu/

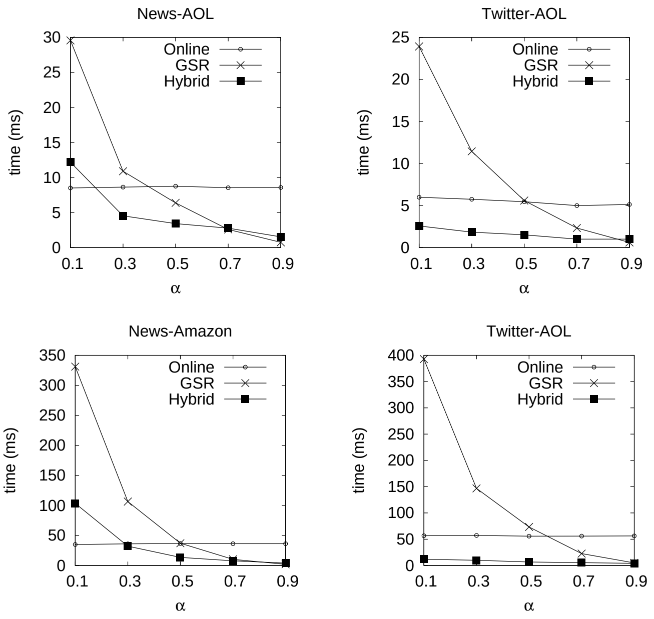

Fig. 7: Vary α

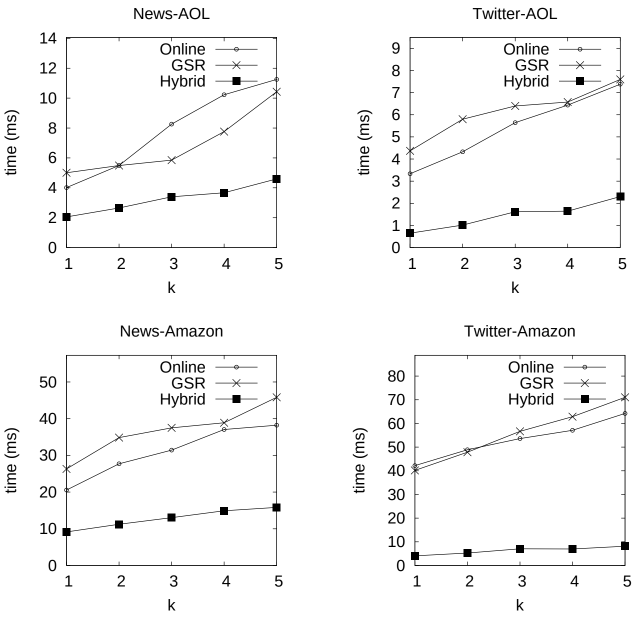

Fig. 8: Vary k

B. Varying α

In our ranking function for top-k ads retrieval, we consider the relevance to the static user interests and dynamic news feed that are combined linearly by the parameter α. In the first evaluation of the three proposed methods, we examine the performance w.r.t. varying α. As shown in Fig. 7, the performance of the **Online** method is invariant under different α for all datasets. This is because the **Online** method always recomputes the top-k ads whenever there is a read operation. The computation cost remains the same when α varies.

The **GSR** method shows superior performance over **Online** for large α. When α increases, the relevance score between an ad and a user is more likely to be dominated by the static user interests and less affected by the dynamic update in the news feed. Hence, the constructed safe region can last longer before its next re-construction. However, when α is very small, the **GSR** method becomes very sensitive to the news feed update and its performance can degrade to a point that it becomes inferior to the **Online** method. The arrival of new posts in the news feed incurs frequent re-construction of safe region which is more expensive than retrieving top-k ads in the **Online** method.

The **Hybrid** method combines the advantages of the **Online** and the **GSR** methods and shows superior performance. It can outperform **GSR** by up to 30x speedups and outperform **Online** by up to 11x speedups in our experiments. This is because the hybrid model can automatically select a retrieval strategy for each user based on our proposed cost model to optimize the performance. It can avoid repetitive retrieval of the same set of ads as in the **Online** method. It can also avoid frequent safe region re-construction as in the **GSR** method when the news feed updates at a high speed. Hence, we can see that its performance is not as sensitive to α as the **GSR** method. For different values of α, it can select a suitable retrieval strategy for each user. The experimental results verified the effectiveness of our proposed hybrid model.

We can also see that the ad database derived from the Amazon dataset results in slower performance than that from the AOL dataset. This is because the textual information in the Amazon dataset is more abundant. It contains product descriptions of books, music and movies while the ads in the AOL dataset are simply keyword search queries. The vectors of topic distributions in the AOL dataset is much more sparse with the values of many columns being or close to 0. It leads to an early termination of the TA algorithm to retrieve top-k ads, which is a component in all the three proposed methods.

Finally, as shown in the figure, when α varies, most users can adapt themselves by selecting a proper retrieval strategy. Another interesting observation is that the performance of **Hybrid** is more stable in the Twitter dataset than in the News dataset. This is because the posts in the News dataset have longer text and cover more topics. After aggregating the topic distributions in the window, the variation in the news feed would be more dramatical.

C. Increasing k

When we increase the number of recommended ads, i.e. k, it takes longer to perform recommendation for all three methods as shown in Fig. 8. First, all the methods need to retrieve top-k relevant ads using the TA algorithm. It is obvious that when k increases, the k-th score becomes larger and it needs to scan more items in the sorted lists until the k-th score is smaller than the upper bound of the unvisited items. Second, the effectiveness of a safe region is affected by k. Based on our observations on the experiments, when k is large, the k-th and $(k+1)$-th item become less distinguishable which makes it more difficult to construct an effective safe region. Nevertheless, **Hybrid** still significantly outperforms the other two methods and its speedup is 10x in the Twitter-Amazon dataset.

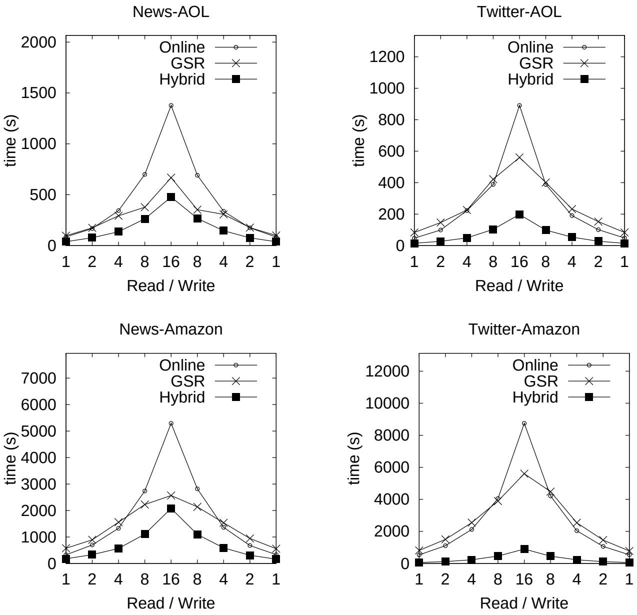

Fig. 9: Vary Read Write Ratio

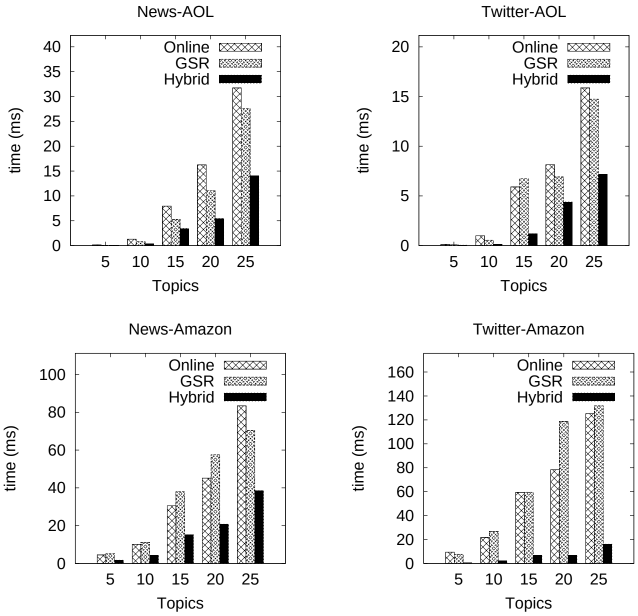

Fig. 10: Vary Number of Topics

D. Vary Read/Write Ratio

In this experiment, we examine the performance and robustness in a dynamic streaming environment with varying read/write ratio. In this experimental setup, we divide the 1 million sampled write operations into 10 blocks, each with 100K write operations. For each block, we manually control the read/write ratio inside. In particular, we first increase the ratio of each block from 1, 2, 4, 8 to 16 and then decrease afterwards. It means in the first block, we have 100K read operations and 100K write operations. When the ratio is 16, there are 1600K read operations with 100K write operations in one block.

We report the total running time of handling the read operations in Fig. 9. The running time for the write operations is the same for all the three methods and thus ignored. Since **Online** always retrieve the top-k results on the fly for a read operation, its running time drastically increases when there are more read operations. It is interesting to observe that **GSR** significantly outperforms **Online** when the read/write ratio is very high, say 8 or 16 in the figure. This is because the context for the next read operation varies little given such a high read/write ratio. The constructed safe region can support more read operations before its next re-construction. However, the total processing time of **GSR** still grows with the read frequency. When more read operations are triggered by randomly picked users, it becomes more likely to detect a user whose safe region requires re-construction. The throughput of **Hybrid** is higher than both **Online** and **GSR** and demonstrates higher adaptivity to the dynamic workloads. It periodically updates $\rho^*(v)$ for each user v to track the dynamism of v's news feed and apply a suitable retrieval strategy. Thus, we have seen that the performance of **Hybrid** is robust against streaming blocks with different read/write ratios.

E. Scalability

In the final set of experiments, we evaluate the scalability of the three proposed methods w.r.t increasing number of topics and graph size.

We increase the number of topics from 5 to 25 and the performance of all methods w.r.t increasing number of topics is presented in Fig. 10. It shows that all methods run slower when there are more topics. Higher dimensions in the topic distribution vectors will result in more computation cost in the TA algorithm as well as less effectiveness in the constructed safe regions. Nevertheless, **Hybrid** remains superior over the other two methods in all experiments and it shows that **Hybrid** is more capable to handle larger number of topics.

Lastly, we show the experimental results when varying graph sizes. Not surprisingly, **Online** is not affected by larger graphs since the retrieval of ads is independent of graph size. Surprisingly, we found that **GSR** and **Hybrid** show better performance when the graph size becomes larger. We interpret the reason as the following: although the social graph is larger, the average degree in a larger graph is actually smaller in our experiments as the graph statistics suggest in Table II. Smaller average degrees mean the news feeds are less dynamic since only the posts written by a neighbour on the social graph will appear in one's news feed. As safe region based methods are highly dependent on the dynamism of news feeds, it is intuitive to understand that they are more efficient with larger number of nodes in the social graph.

VIII. Conclusion

In this work, we studied the context-aware advertisement recommendation problem for high speed social news feeding. We first formulated a general ranking function of ads against each user in the social network by combing the his/her interests and dynamic contents in the news feed. The **Online** method was first proposed to retrieve a user's news feed and re-compute the recommended ads based on TA algorithm when

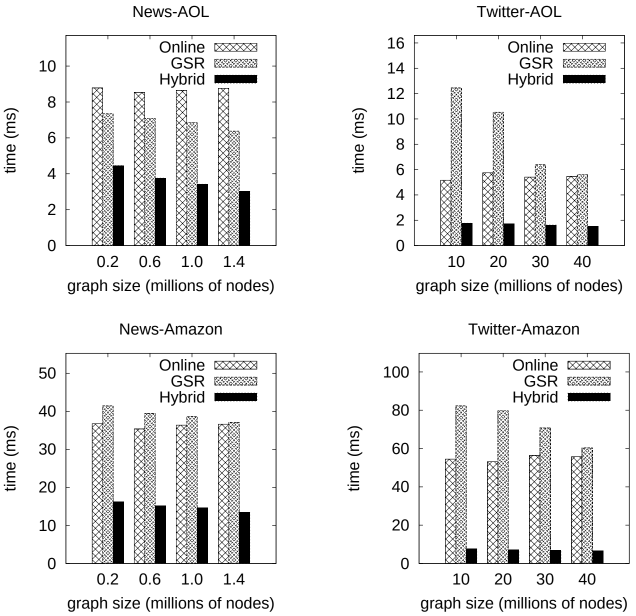

Fig. 11: Vary Graph Sizes

there is a read operation triggered. Then the **GSR** method is developed, which maintains a safe region and only recomputes the recommended ads whenever the safe region is found invalid against updated news feed. Subsequently, we developed the **Hybrid** method to analyze users in terms of the dynamism of their news feed and determine a suitable retrieval strategy so as to speedup the recommendation process. Extensive experiments on real world social networks and ad datasets have verified the efficiency and robustness of the hybrid model.

IX. Acknowledgement

This work is funded by the NExT Search Centre (grant R-252- 300-001-490), supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office. Yuchen Li and Ziquan Lan are partially supported by the NUS Graduate School of Integrative Science and Engineering, National University of Singapore.

References

[1] C. E. Tucker, "The economics of advertising and privacy," *International Journal of Industrial Organization*, vol. 30, no. 3, pp. 326–329, 2012.

[2] C. Li, Y. Lu, Q. Mei, D. Wang, and S. Pandey, "Click-through prediction for advertising in twitter timeline," in *KDD*, 2015, pp. 1959–1968.

[3] R. Fagin, A. Lotem, and M. Naor, "Optimal aggregation algorithms for middleware," in *SIGMOD*, 2001, pp. 102–113.

[4] I. F. Ilyas, G. Beskales, and M. A. Soliman, "A survey of top-k query processing techniques in relational database systems," *ACM Comput. Surv.*, vol. 40, no. 4, pp. 11:1–11:58, 2008.

[5] B. Chandramouli and J. Yang, "End-to-end support for joins in large-scale publish/subscribe systems," in *PVLDB*, vol. 1, no. 1, 2008, pp. 434–450.

[6] A. Machanavajjhala, E. Vee, M. N. Garofalakis, and J. Shanmugasundaram, "Scalable ranked publish/subscribe," in *PVLDB*, vol. 1, no. 1, 2008, pp. 451–462.

[7] L. Guo, L. Chen, D. Zhang, G. Li, K. Tan, and Z. Bao, "Elaps: An efficient location-aware pub/sub system," in *ICDE*, 2015, pp. 1504–1507.

[8] D. Zhang, C. Chan, and K. Tan, "An efficient publish/subscribe index for ecommerce databases," in *PVLDB*, vol. 7, no. 8, 2014, pp. 613–624.

[9] L. Guo, D. Zhang, G. Li, K. Tan, and Z. Bao, "Location-aware pub/sub system: When continuous moving queries meet dynamic event streams," in *SIGMOD*, 2015, pp. 843–857.

[10] W. Woerndl, C. Schueller, and R. Wojtech, "A hybrid recommender system for context-aware recommendations of mobile applications," in *ICDEW*, 2007, pp. 871–878.

[11] H. Zhu, E. Chen, H. Xiong, K. Yu, H. Cao, and J. Tian, "Mining mobile user preferences for personalized context-aware recommendation," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 4, pp. 58:1–58:27, 2014.

[12] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., 1986.

[13] D. Zhang, C. Chan, and K. Tan, "Processing spatial keyword query as a top-k aggregation query," in *SIGIR*, 2014, pp. 355–364.

[14] Y. Li, D. Zhang, and K. Tan, "Real-time targeted influence maximization for online advertisements," in *PVLDB*, vol. 8, no. 10, 2015, pp. 1070–1081.

[15] K. Mouratidis and H. Pang, "Computing immutable regions for subspace top-k queries," in *PVLDB*, vol. 6, no. 2, 2012, pp. 73–84.

[16] J. Zhang, K. Mouratidis, and H. Pang, "Global immutable region computation," in *SIGMOD*, 2014, pp. 1151–1162.

[17] C. Chen, F. Li, B. C. Ooi, and S. Wu, "Ti: An efficient indexing mechanism for real-time search on tweets," in *SIGMOD*, 2011, pp. 649–660.

[18] J. Yao, B. Cui, Z. Xue, and Q. Liu, "Provenance-based indexing support in micro-blog platforms," in *ICDE*, 2012, pp. 558–569.

[19] Y. Li, Z. Bao, G. Li, and K. Tan, "Real time personalized search on social networks," in *ICDE*, 2015, pp. 639–650.

[20] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[21] J. He, "A social network-based recommender system," Ph.D. dissertation, 2010.

[22] X. Yang, H. Steck, Y. Guo, and Y. Liu, "On top-k recommendation using social networks," in *RecSys*, 2012, pp. 67–74.

[23] A. Seth and J. Zhang, "A social network based approach to personalized recommendation of participatory media content," in *ICWSM*, 2008.

[24] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[25] H. V. Jagadish, B. C. Ooi, K.-L. Tan, C. Yu, and R. Zhang, "idistance: An adaptive b+-tree based indexing method for nearest neighbor search," *ACM Trans. Database Syst.*, vol. 30, no. 2, pp. 364–397, 2005.

[26] "Poisson approximations of multinomial distributions and point processes," *Journal of Multivariate Analysis*, vol. 25, no. 1, pp. 65–89, 1988.

[27] J. Leskovec, L. A. Adamic, and B. A. Huberman, "The dynamics of viral marketing," *ACM Trans. Web*, vol. 1, no. 1, 2007.

X. Appendix

Derivation of Eqn. 9: $\mathrm{Var}[X_{w,v}]$ is defined in Eqn. 9 as:

$$\mathrm{Var}[X_{w,v}] = \mathrm{Var}[\sum_{n\in N(v)} \sum_{1\le i\le M_{v,n}} D_{w,n}(F_n)]$$

By the definition of variance, we obtain the following expansion of $\mathrm{Var}[X_{w,v}]$:

$$\mathrm{Var}[X_{w,v}] = \sum_{n\in N(v)} \mathrm{Var}[\sum_{1\le i\le M_{v,n}} D_{w,n}(F_n)] \tag{13}$$

$$+ \sum_{\substack{a,b\in N(v)\\ a\neq b}}\sum \mathrm{Cov}[\sum_{1\le i\le M_{v,a}} D_{w,a}(F_a), \sum_{1\le i\le M_{v,b}} D_{w,b}(F_b)] \tag{14}$$

We compute Eqn. 13 and 14 separately. For a user v and a topic w, $\mathrm{Var}[\sum_{1\leq i\leq M_{v,n}} D_{w,n}(F_n)]$ can be expressed as:

$$\begin{aligned}
&\mathrm{Var}[\sum_{1\leq i\leq M_{v,n}} D_{w,n}(F_n)] \\
&= \mathbb{E}[\big(\sum_{1\leq i\leq M_{v,n}} D_{w,n}(F_n)\big)^2] - \mathbb{E}[\sum_{1\leq i\leq M_{v,n}} D_{w,n}(F_n)]^2 \\
&= \mathbb{E}[\mathbb{E}[\big(\sum_{1\leq i\leq M_{v,n}} D_{w,n}(F_n)\big)^2]|M_{v,n}] - \mathbb{E}[\mathbb{E}[\sum_{1\leq i\leq M_{v,n}} D_{w,n}(F_n)]|M_{v,n}]^2
\end{aligned}$$

Since we know that both $M_{v,n}$ is independent of $D_{w,n}(F_n)$ and each $D_{w,n}(F_n)$ are independent of each other:

$$\begin{aligned}
&\mathbb{E}[\mathbb{E}[\big(\sum_{1\leq i\leq M_{v,n}} D_{w,n}(F_n)\big)^2]|M_{v,n}] \\
&= \mathbb{E}[M_{v,n}^2 \mathbb{E}[D_{w,n}(F_n)]^2|M_{v,n}] \\
&= \mathbb{E}[D_{w,n}(F_n)]^2\mathbb{E}[M_{v,n}^2|M_{v,n}] = \mathbb{E}[D_{w,n}(F_n)]^2\mathbb{E}[M_{v,n}^2]
\end{aligned}$$

and in a similar way:

$$\mathbb{E}[\mathbb{E}[\sum_{1\leq i\leq M_{v,n}} D_{w,n}(F_n)]|M_{v,n}]^2 = \mathbb{E}[D_{w,n}(F_n)]^2\mathbb{E}[M_{v,n}]^2 \quad (15)$$

now we have:

$$\mathrm{Var}[\sum_{1\leq i\leq M_{v,n}} D_{w,n}(F_n)] = \mathbb{E}[D_{w,n}(F_n)]^2\mathrm{Var}[M_{v,n}]$$

Since $\mathrm{Var}[M_{v,n}] = m\lambda_{v,n}(1-\lambda_{v,n})$, we only need to derive the unknown term $\mathbb{E}[D_{w,n}(F_n)]$.

$$\mathbb{E}[D_{w,n}(F_n)] = \mathbb{E}[\mathbb{E}[D_{w,n}(F_n)]|F_n] = \frac{(f_n+1)p_{w,n}}{2}$$

Then, for any n and w, we are able to evaluate:

$$\mathrm{Var}[\sum_{1\leq i\leq M_{v,n}} D_{w,n}(F_n)] = \frac{(f_n+1)^2p_{w,n}^2}{4}m\lambda_{v,n}(1-\lambda_{v,n}) \quad (16)$$

Next we derive $\mathrm{Cov}[\sum_{1\leq i\leq M_{v,a}} D_{w,a}(F_a), \sum_{1\leq i\leq M_{v,b}} D_{w,b}(F_b)]$ for any $a, b \in N(v)$ and $a \neq b$. Let $A = \sum_{1\leq i\leq M_{v,a}} D_{w,a}(F_a)$ and $B = \sum_{1\leq i\leq M_{v,b}} D_{w,b}(F_b)$. It follows that:

$$\mathrm{Cov}[\sum_{1\leq i\leq M_{v,a}} D_{w,a}(F_a), \sum_{1\leq i\leq M_{v,b}} D_{w,b}(F_b)] = \mathbb{E}[AB] - \mathbb{E}[A]\mathbb{E}[B]$$

From Eqn. 15, we can get $\mathbb{E}[A] = \mathbb{E}[D_{w,a}(F_a)]\mathbb{E}[M_{v,a}]$ and $\mathbb{E}[A] = \mathbb{E}[D_{w,b}(F_b)]\mathbb{E}[M_{v,b}]$. The only left part is $\mathbb{E}[AB]$ which can be derived as the follows:

$$\begin{aligned}
&\mathbb{E}[AB] = \mathbb{E}[AB|M_{v,a}, M_{v,b}] \\
&= \mathbb{E}[\sum_{1\leq i\leq M_{v,a}} D_{w,a}(F_a) \cdot \sum_{1\leq i\leq M_{v,b}} D_{w,b}(F_b)|M_{v,a}, M_{v,b}] \\
&= \mathbb{E}[D_{w,a}(F_a)]\mathbb{E}[D_{w,b}(F_b)]\mathbb{E}[M_{v,a}M_{v,b}]
\end{aligned}$$

Then it is natural to have:

$$\begin{aligned}
&\mathrm{Cov}[A, B] = \mathbb{E}[D_{w,a}(F_a)]\mathbb{E}[D_{w,b}(F_b)]\mathrm{Cov}[M_{v,a}, M_{v,b}] \\
&= -\frac{(f_a+1)(f_b+1)p_{w,a}p_{w,b}}{4}m\lambda_{v,a}\lambda_{v,b} \quad (17)
\end{aligned}$$

By combining the above results from Eqn. 16 and 17, we can derive $\mathrm{Var}[X_{w,v}]$.

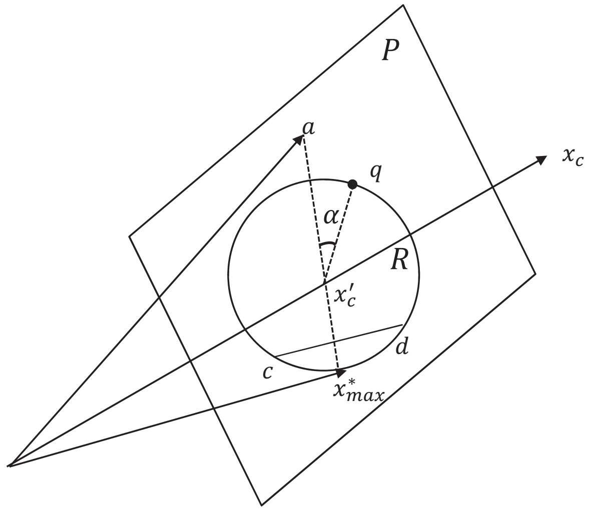

Fig. 12: The location of x_{max} for an ad against a safe region.

Proof of Theorem 2: For the ease of presentation, we prove Theorem 2 in 3D space and the proof can be generalized to any finite dimensional space. Given the spherical safe region B and the center of B, i.e. x_c, as shown in Figure 12, we draw a plane P which is normal to x_c and passes through x_{max}^* (Here x_{max}^* is used as the contact point between the sphere and the vector x_{max}^*). a is the point where the ad vector intersects with P and R is the intersecting circle region between B and P. It is easy to see that the line from a to x_{max}^* passes through the center of the intersecting circle plane between B and P, i.e. x_c'. This means any point q on the boundary of R will have shorter distance to a than that of x_{max}^*. Moreover, since x_{max}^* has a negative coordinate, we can find a line segment cd, which is the intersection between P and the boundary region S_i, that separates a and x_{max}^* on both sides of cd. For any point q on the boundary of R, α is the angle between a, x_c and q, x_c. Then the distance from a to q can be expressed as:

$$\xi(a,q) = \xi^2(a,x_c') + \xi^2(q,x_c') - 2\xi(a,x_c')\xi(q,x_c')cos\alpha$$

where $\xi(.,.)$ denotes the distance between two points. Therefore $\xi(a,q)$ is a continuous unimodal function w.r.t $\alpha \in [0, 2\pi)$. This means $\max\{\xi(a,c), \xi(a,d)\}$ is larger than any point q which is on the boundary of R and lies on the same side with a w.r.t line cd. This in term means $\theta(q,a) < \max\{\theta(a,c), \theta(a,d)\}$ because all points on R have the same distance to the origin and $\theta(q,a)$ is proportional to $\xi(q,a)$.

With the above proof, we have shown that, $\max\{\theta(a,c), \theta(a,d)\}$ is the maximum possible angle within the region R. However we have not shown for all points on B that such condition holds. Note that since R contains the point x_{max}^* which is a contact point between B and B's minimum bounding convex cone. Then all vectors from the origin to any points on B will pass through R. Thus we can conclude that $\max\{\theta(a,c), \theta(a,d)\}$ is the maximum possible angle among all points on B and prove Theorem 2.