1. Introduction

All social media users have encountered advertisements, which sometimes can be interesting, meaningful, relevant to our needs and often annoying. Haven't you ever wondered how an advertisement on for example Facebook or Twitter is so much corresponding to your current interests? Maybe you have thought, do they spy on me or are they stealing my information? In fact, there are calculations happening in the background, that you are not aware of. These calculations are studying what we are interest in and what not. In this paper I give you an overview of one of the most powerful models to handle and provide such advertisements in an efficient way.

The Researchers Yuchen Li, Dongxiang Zhang, Ziquan Lan and Kian-Lee Tan in the NUS Graduate School of Integrative Science and Engineering at the National University of Singapore came up with an idea and a challenge [1]. They wanted to develop a model, that can make user recommendations over social media more efficient, real time, less annoying and willing to make the user hit the advertisement icons, which satisfy his needs. Their idea came from the fact, that every person has his own static interest, so there is a possibility to make a system, that can recommend some advertisements to him. However, they have discovered that the system will not be accurate, since the user also has dynamic interests, which are changing with the news feeds, that he or she gets from friends. This new information could somehow change the interests in a way or another. Their challenge now is to combine the static interests and the dynamic interests into one model, that can recommend the most relevant advertisement, that meet the user's interests. They are aware that this model could be computationally expensive.

2. Related work

Advertisements became the major revenue source for social media platforms, even for the dominators of the market such as Facebook and Twitter it is a multi-billion-dollar market. In order to deliver ads to a potential interested user, Social networks have to learn a model to predict the user's interests, based on their personal static interests. It is not that efficient, since the user interests are growing slowly, thus the user may end up receiving repetitive ads. The group of researchers proposed a context aware advertisement framework, that combines the relatively personal interests and the dynamic news feed from friends to increase the possibility, that the user will hit the ads button. For example, when a friend shows the status in hospital, displaying gift delivery ads is a good choice. To do that they have proposed a hybrid model,

which combines the advantages of the online retrieval strategy, which is able to find the most relevant ads matching the dynamic context when a read operation is triggered, and the safe region method which has been developed, to avoid the frequent computations, when the context varies a little and to detected if the top ads has been changed. This hybrid model has been tested on multiple social media, and it has proven, that it is efficient and robust. In this paper it will be described how they could achieve that. Before of going into the hybrid model details, I would like to give a look on the related works, that the researches introduced, they have studied these works, analysed it, discovered what are the advantages and the disadvantages, to improve the quality of their model. Let us start with the Publish/Subscribe System.

2.1 Pub/Sub System

A publish/subscribe system is a middleware for matching events [2], which are generated by data sources (publishers), to subscriptions, which specify the interests of users (subscribers). Traditional publish/subscribe systems only support stateless subscriptions, defined as filters over the contents of individual events (e.g. stock quotes) against a set of subscriptions (e.g. trader profiles specifying quotes of interest). There are two major differences between this system and the context aware system, since the pub/sub is using Boolean expression matching, which means an event either matches a subscription or it does not, for instance, a stock quote will either match or not match a trader profile. A problem could be that there are a lot of events, which are matching the user subscription [2], so the user will end up with many ads. This will most probably make him annoyed. Firstly, in the context aware ad recommendation only the most relevant ads in the user news feeds will be displayed. The second difference is that the subscription has been built based on the static interests of the user. However, in the context aware the recommendation has been built based on the combination of the static interests and the dynamic interests.

2.2 Top-K Aggregation Query

This approach considers that each object attribute has its own score. In order to calculate the total score for an object, they are using a monotonic aggregation function [1]. After they are using algorithms such as the threshold (TA, CA) to obtain the most relevant ads for a user.

2.3 Local immutable region

The local immutable region determines immutable regions on individual decision factors [3]. An immutable region takes the form of a validity interval for an isolated query weight, assuming that all the other weights are kept constant. An interval is defined for each decision factor.

However, due to the local nature of the LIRs, it cannot support simultaneous readjustments to multiple weights.

2.4 Global immutable region

The GIR indicates all the possible weight settings for which the current top-k recommendation holds [3]. For the common case of linear scoring functions, the GIR is a convex polytope in query space, wherein the query vector may freely shift without inducing any changes in the result. 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, the researches designed a series of techniques to quickly compute a subspace of GIR, so that the maintenance cost is greatly reduced.

3. Construct the hybrid Model equations and algorithms

Let us assume that we have an advertisement database A. The goal is to recommend the most relevant ad from this database, when a user requests for his news feed. They can classify the ads into multi-dimensional topic vector (T). They have studied previous works to measure the relevance between static user interests (profiles) and an ad and they obtain the following equation [1]:

$$Øs(u,a) = \sum_{w \in T} rel(u,w) \cdot rel(a,w) \qquad (1)$$

$Øs(u,a)$ is the relation between static user profile and ads, rel(u, w) ∈ [0, 1] denotes the relevance between a user u and a topic (w) in T and rel(a, w) denotes the relations between an ad and a topic (w) in T [1]. Their context aware is also considering the dynamic news feed, when they recommend ads for a given user. They have used a sliding window to store m most recent posts, to serve as a dynamic context for ad recommendation, so they apply the same topic modelling technique to project each post in the window to the latent topic space and use rel(d,w) ∈ [0, 1] to measure the relevance between a post and a topic. They came up with the following equation [1]:

$$Ød(u,a) = \frac{\mathbf{1}}{\boldsymbol{m}} \sum_{d \in Wu} \sum_{w \in T} rel(d,w) \cdot rel(a,w) \qquad (2)$$

rel(a,w) is the relation between an ad(a) and a topic(w), where $\emptyset d(u, a)$ is the relation between dynamic user profile and ads. You can imagine the overall system as shown in the figure below [1].

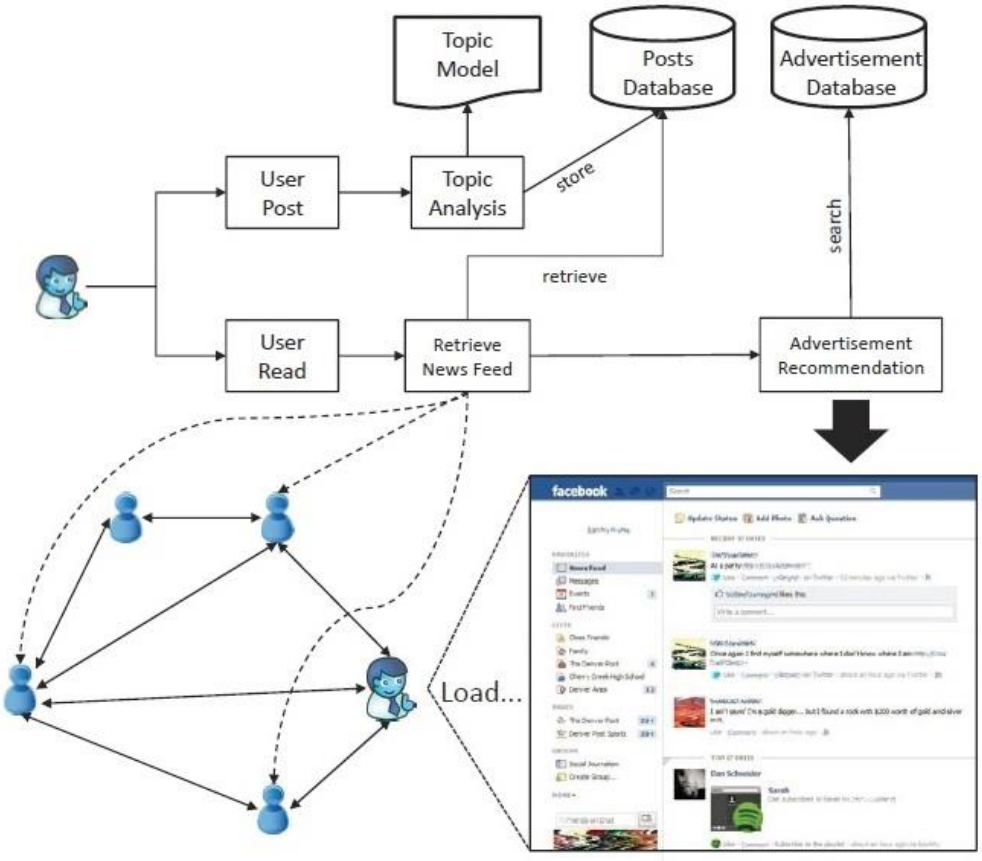

Fig. 1: System Overview of Context-Aware Advertisement Recommendation in Social Networks

Each user in social media, is either publisher or subscriber. When the user composes, shares or likes a post, we say the user, as a publisher, triggers a write operation. His post is saved in the database and may later be retrieved to appear in his friend's news feeds. When a user login or refresh his news feed, we say the user, as a subscriber, triggers a read operation. Finally, they have summed up these two equations into one which is presented by this linear equation [1]:

$$\emptyset (u, a) = \alpha \cdot \emptyset s(u, a) + (1 - \alpha) \cdot \emptyset d(u, a) \qquad (3)$$

$\alpha \in [0, 1]$ is a system parameter to balance the importance between personal interests and dynamic context and can be set based on the application requirements. When α is close to 1, the ads recommendation will be based mainly on the static user profile, when it is 0 then the recommendation will be based on the dynamic context. Then they have defined their problems as follows [1]:

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 Ø (u, a) ≥ Ø (u, a') ∀a ∈ R ∧ ∀a' ∈ A \R. In the equation (3) they have aggregate the dynamic news feed with the static personal profile, to query the ad database, they have called the aggregated vector context-aware query vector, denoted by Q_u.

4. Online retrieval algorithm

There are models in social media, that can calculate the top ad for personal interests offline, since the user profiles are static. They return it together with the news feed, when the user requests for his news feed. However, they must include the dynamic context in the recommendation calculations. Therefore, they are not able to do the calculation offline, because each write operation, will cause the news feeds for all the user's friends to vary, which is computationally expensive. The online retrieval algorithm will bring the top k "on the fly" [1]. If they want to retrieve the most relevant ads to a given user, they have to construct a query vector. It consists of the distribution of user static profile and dynamic context, which consist of the most recent, unread posts from his friends. Then it scans it against the ads database, this would be computationally expensive. To handle this problem effectively, they reconstruct the equation (3) to be like this [1]:

$$Ø\,(u, a) = \alpha \cdot Øs(u, a) + (1 - \alpha) \cdot Ød(u, a) =$$

$$\sum_{w \in T} \Big[\underbrace{\alpha \cdot \mathrm{rel}(u, w) + \frac{1-\alpha}{m} \sum_{d \in Wu} \mathrm{rel}(d, w)}_{\mathbf{Q_u(w)}} \Big] . rel(a, w)$$

Qu(w) is the aggregated relevance between user u and topic w. Their ranking function consists of two terms(Qu(w) and rel(a,w)). Since rel(a,w) is independent of the dynamic context, it could be computed and sorted offline. They Qu(w) will become constant, if the Ø (u, a) is determined. It will not affect the ordering of the rel(a,w). Therefore, we could establish |T| inverted lists, sorted by rel(a,w) for each user. When a read operation is triggered, they can we can retrieve the sorted lists and directly apply standard top-k aggregation techniques such as Threshold Algorithm(TA).

Example 1: Let the window size m = 3, the weighting parameter α = 0.25 and the number of topics |T| = 2. Given a user u, let Hu = (0.4, 0.6) be the topic distributions of his 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 Qu is calculated as Qu = 0.25 · (0.4, 0.6) + 1−0.25 3 [(0.2, 0.8) + (0.1, 0.9) + (1.0, 0)] = (0.55, 0.45)

= (0.425, 0.575). Suppose Qu is used to query an ad database with four tuples {a1 = (0.3, 0.9), a2 = (0.4, 0.7), a3 = (0.5, 0.8) and a4 = (1.0, 0)}. To support top-k aggregation, we pre-compute two inverted lists lw1 and lw2 for the topics and get lw1 = {(a4, 1.0), (a3, 0.5), (a2, 0.4), (a1, 0.3)} and lw2 = {(a1, 0.9), (a3, 0.8), (a2, 0.7), (a1, 0.0)}. By calling the TA algorithm presented above, a3 will be returned as the most relevant ad if k is set to 1.

5. Safe region algorithm

According to some studies, 90 % of the social media users are readers (content viewers), 9 % are editors and 1 % are publishers [1]. If you only do a read operation, then the online retrieval algorithm will not be convenient for you. Because if the content varies only a little bit in a short period, then the algorithms will be recomputed again. This is computationally expensive, and it is a waste of CPU resources to retrieve the same set of ads. They have introduced a safe region algorithm to handle this challenge. It can examine, if the top relevant ads have been changed since the last read operation or not. They have done this effectively, by implementing 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.

5.1 Safe Region Construction

They have constructed a rectangle in the high-dimensional topic space. Whenever new posts are located in rectangle boundaries, the top ad will not change. They call the high-dimensional rectangle a safe region, denoted by S = (Q^{ulb}, Q^{ulb}), where Q^{ulb} stores the lower bound of coordinates in all the dimensions and Qulb stores the upper bound. They have proposed a Greedy Safe Region (GSR), to incrementally build the safe region. The algorithm is as follows [1]:

Algorithm 1: GSR(User u)

$R \leftarrow$ Use TA to compute the relevant ads against Q_u
$Q_u^{lb} \leftarrow Q_u, Q_u^{ub} \leftarrow Q_u$
while *True* **do**
  $w \leftarrow$ **DimensionSelect**(v)
  $\vec{\delta} \leftarrow \frac{1-\alpha}{m} \vec{e_w}$
  for $a \in R$ **do**
    $\phi(a) =$ **MinS**$(a, Q_u^{lb} - \vec{\delta}, Q_u^{ub} + \vec{\delta})$
  $S_u \leftarrow \min\{\phi(a) | a \in R\}$
  for $a \in A \setminus R$ **do**
    $\phi(a) =$ **MaxS**$(a, Q_u^{lb} - \vec{\delta}, Q_u^{ub} + \vec{\delta})$
  $S_l \leftarrow \max\{\phi(a) | a \in A \setminus R\}$
  if $S_u \geq S_l$ **then**
    $Q_u^{lb} \leftarrow Q_u^{lb} - \vec{\delta}$
    $Q_u^{ub} \leftarrow Q_u^{ub} + \vec{\delta}$
  else
    return (Q_u^{lb}, Q_u^{ub})

An overview of this algorithm:

1) They store the set of most relevant ads for the current news feed in R.
2) They initialize the algorithm to be context aware.
3) They choose the most promising topics to expand the safe region.
4) For each of these topics they calculate the Q_u and then calculate the distance between it and the lower and upper bound. After they calculate the topic with the minimum distance to these boundaries.
5) They have to be aware, that this explanation is safe, by implanting an expansion unit $\frac{\mathbf{1}-\alpha}{\boldsymbol{m}}$, which is the maximum allowed change for a given query topic $Q_u(w)$.
6) The last condition indicates that if the minimum relevance for a query topic $Q_u(w)$ to the top K ads in R (Su)is bigger than the maximum relevance to the ads, that are not in R (SI), then the expansion is safe. Otherwise the algorithm terminates and returns a safe region with partial topic expansion.

Theorem 1: For a query vector Q_u with its bound vectors Q^{lb}_u and Q^{ub}_u returned by algorithm 1, whenever Q^{lb}_u (w) ≥x(w) ≥ Q^{ub}_u (w)∀w ∈ T, it corresponds to the same set of top-k ads as Q_u. [1]

5.2 Safe Region Based Query Processing

They have discovered with a simple check like $Q \geq Q^{lb}_u \wedge Q \leq Q^{ub}_u$, that if two processes are in the same scalar, maybe one of them will be in the safe region and the other not. For example, if they have two queries, which share the same set of ads such as Q = (0.3.0.5) and $Q*$ = (0.15, 0.25), maybe they will not be in the same safe region if for instance the Q^{lb}_u = (0.1, 0.2), Q^{ub}_u = (0.2, 0.4). Thus, too many calculations will be recomputed for the new safe region. They created a new flexible rule to check, whether a query is in the safe region. In Lemma1 they measured the intersections between a Q and safe region sphere using the equation 5 with replacing a with Q_u, by computing the angel between two vectors with at most T dimension (topic) [1].

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

6. Optimizations

There are two possibilities for optimization, the first one is to efficiently evaluate Sl and Su. To evaluate one of them, you need to scan all the ads database, which will be computationally expensive. The second one is to avoid reconstructing the safe region as much as possible,

when a Q_u is no longer in it. For the first optimization they want to reduce the number of MaxS computing times. Therefore, they have developed an upper bound (bw) for the inverted list of each topic (w). The maximum MaxS score of unvisited ads can be bounded by computing MaxS for b = (b1, .., 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 Sl [1].

When the Q_u moves out of the safe region of a given user (u), we have to recompute the top ads using the online retrieval algorithm. To avoid that and bring the results as fast as possible, the second optimization is to search into his friend's safe region. If we find a safe region from a user (v) that contains the new Q_u, we can assign the safe region from user (V) directly to the user (U). In this case we ensure that they have the same set of top relevant ads and safe the cost of precomputing the online retrieval algorithm [1].

7. Hybrid algorithm

The hybrid model has been introduced, to combine the advantages of the online retrieval and the safe region. The model measures the topic distribution in a user news feed. If it is varying much, then they have adopted the online strategy, otherwise they have used the safe region strategy.

7.1 Variance of topic distributions

To measure the topic distribution, they have introduced a series of equations as follows [1]:

$$\mathbf{X}_{\mathrm{w,v}} = \sum_{\mathrm{n} \in \mathrm{N(v)}} \sum_{1 \le \mathrm{i} \le \mathrm{Mv,n}} \mathrm{Dw,n(Fn)}$$

$X_{w,v}$ is a random variable describing the topics (w) weight in a user's (v) post, N(v) is the number of all his neighbours, $M_{v,n}$ is a random variable describing how many posts are selected from a neighbour n to form the news feed window of a post for user v and Fn is discrete uniform distribution. The variance if the topic will be defined is like this [1]:

$$\boldsymbol{var}[\mathbf{X}_{\mathrm{w}}, \mathrm{v}] = \mathrm{var}[\sum_{\mathrm{n} \in \mathrm{N(v)}} \sum_{1 \le \mathrm{i} \le \mathrm{Mv,n}} \mathrm{Dw,n(Fn)}]$$

After making some derivation processes they had the final equation to calculate the variance of a topic w in a user v's news feed. It is as follows [1]:

$$\boldsymbol{var}[\mathbf{X}_{\mathrm{w}}, \mathrm{v}] = \frac{-\mathbf{1}}{\mathbf{4}} \sum_{d \in Wu} \mathrm{m}\lambda_{a,v}^2 (\mathrm{f_a} + 1)^2 p_{w,a}^2$$

7.2 Hybrid Retrieval Strategy

Since the var[xw,v] only captures the variance of topic distributions in the news feed, they need to combine it with the static user interests, so it becomes like this [1]:

$$p(v) = \max_{w \in T} \frac{\frac{1-\propto}{m} \sqrt{var[X_{w,v}]}}{\propto . rel[u,w] + \frac{1-\propto}{m} . E[X_{w,v}]}$$

There is a small gap as this equation only considers the topic distribution for the write operations, while ignoring the read frequency. They have adopted the last equation to fill up this gap by introducing a read frequency η_v. The final equation for the hybrid model is as follows [1]:

$$p^*(v) = \frac{\sum_{n \in N(v)} \lambda_n}{\eta_v} . P(v)$$

In this case we can use p*(v) to decide about the retrieval strategy, that we should use. If P*(v) is greater than a given threshold, then we use the safe region strategy, otherwise we use the online retrieval strategy.

8. Experimental study

The same researchers did experiments on real social network datasets with billions of edges such as Twitter and AOL [1]. Their target is to guarantee the real-time delivery of relevant ads. They are interested to measure the average elapsed time in retrieving the top-k ads for each read operation.

Varying α

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 online retrieval by up to 11x speedups in their experiments [1]. This is because the hybrid model can automatically select a retrieval strategy for each user based on the 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 their proposed hybrid model. However, the hybrid model strategy still out performs the online retrieval and the GSR model, when the k-increases, vary read/write ratio and vary number of topics [1].

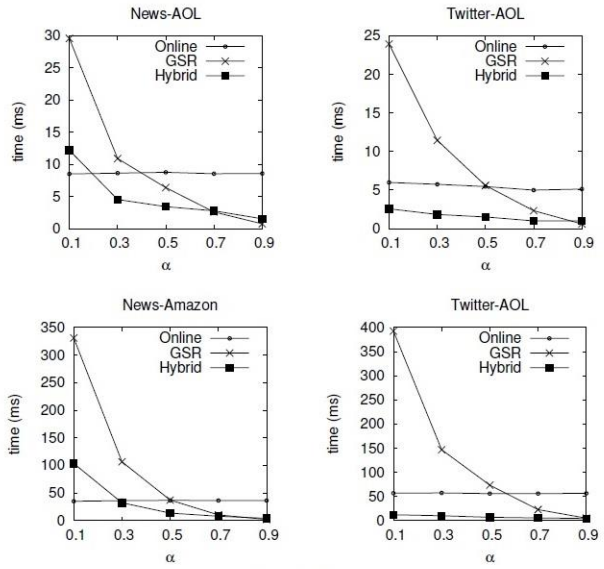

Fig. 7: Vary α

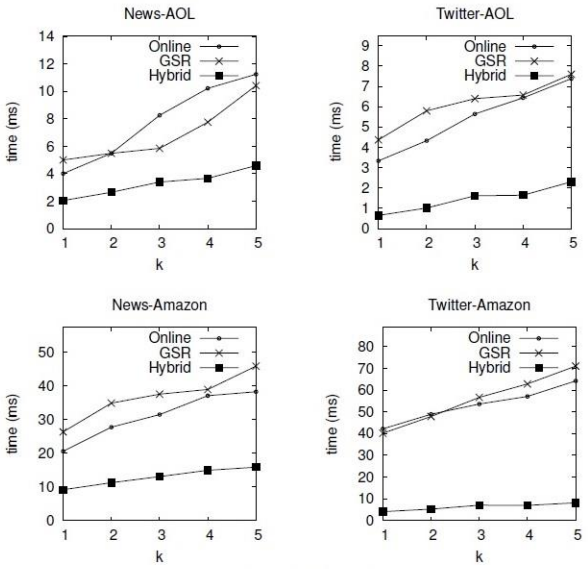

Fig. 8: Vary k

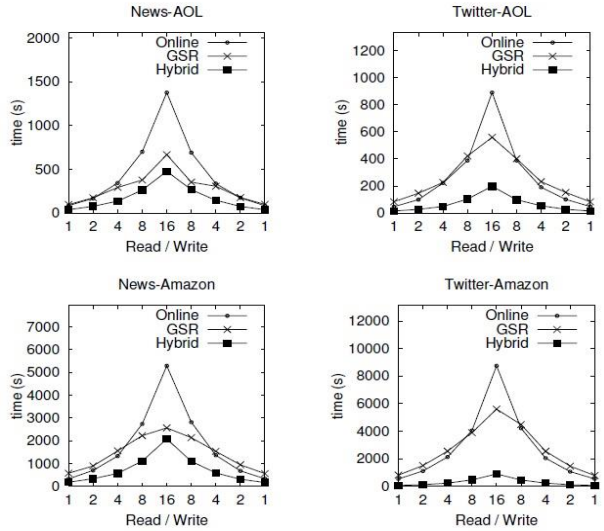

Fig. 9: Vary Read Write Ratio

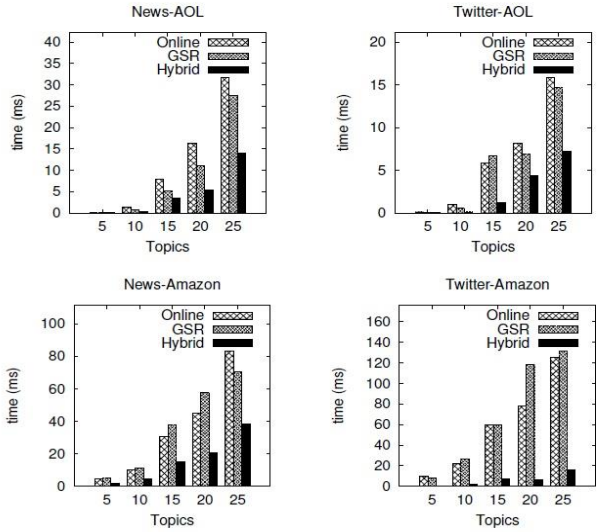

Fig. 10: Vary Number of Topics

9. Conclusion

In the studied paper they have introduced an online retrieval strategy. It retrieves a user's news feed and computes the top-k based on the TA algorithm. Then they created the GSR model, which maintains a safe region and only recomputes the recommended ads whenever the safe region is found invalid against an updated news feed. Finally, they have produced the hybrid model. It combines the two metrices of the online retrieval and the GSR model, to determine the suitable model to retrieve the top-k ads for a given user and to speed up the recommendation process. According to many experiments on huge datasets, the hybrid model has proved to be efficient and robust.