Seminar paper at the Faculty of Science

Data science seminar, Fall semester 2017-18

Prof. Mourad Khayati

CONTEXT-AWARE ADVERTISEMENT

RECOMMENDATION FOR

HIGH-SPEED SOCIAL NEWS FEEDING

Mohammad ALTAHAN

10-01-2018

# Introduction

All social media users have encountered advertisements, which sometimes can be interesting, meaningful, relevant to our needs and often annoying. In this paper I give you an overview of a 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 to develop a new technique. This technique can make user recommendations over social media more efficient, real time and willing to make the user hit the advertisement icons, which satisfy his needs. The proposed model 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, the model will not be accurate, since the user also has dynamic interests, which are changing with the news feeds, that he gets from friends. This new information could change the interests in a way or another. The challenge is to combine the static interests and the dynamic interests into one model, that can recommend the most relevant advertisements, which meet the user’s interests.

# Related work

Advertisements are the major revenue source for social media platforms like Facebook or Twitter. To deliver ads to a potential interested user, Facebook and Twitter must 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. A hybrid model has been proposed, to 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 detect if the top ads has been changed. For example, when a friend shows the status in hospital, displaying gift delivery ads is a good choice.

# Pub/Sub System

The pub/sub 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, so the user will end up with many ads. The first difference is, 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. As these contents are continuously changing, this kind of solution can’t be applied.

# Global immutable region

The GIR indicates all the possible weight settings for which the current top-k recommendation holds. 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, a series of techniques has been designed, to quickly compute a subspace of GIR, so that the maintenance cost is greatly reduced.

# Construct the hybrid Model equations and algorithms

Let’s 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. The model can classify the ads into multi-dimensional topic vector (T). The relevance between the static user interests and an ad, can be measure according to some previous works as:

|  |  |
| --- | --- |
|  | **(1)** |

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. The context aware is also considering the dynamic news feed, when the recommend ads for a given user. The model have used a sliding window to store m most recent posts, to serve as a dynamic context for ad recommendation, so it applies 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. This equation is the result for that:

|  |  |
| --- | --- |
|  | (2) |

rel(a,w) is the relation between an ad(a) and a topic(w), where is the relation between dynamic user profile and ads. You can imagine the overall system as shown in the figure below.A close up of a map

Description generated with high confidence

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. Then, the posts from friends are retrieved and sorted chronologically and a sliding window containing m recent unread posts are returned. Finally, this equation has summed up these two equations into one which is presented by this linear equation [1]:

α ∈ [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.

# 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, the hybrid model must include the dynamic context in the recommendation calculations. Therefore, the model will 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”. If the model wants to retrieve the most relevant ads to a given user, the algorithm must 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, it will scan the whole ads database, to find the most relevant ads with the highest score. This would be computationally expensive. To handle this problem effectively, the equation (3) reconstructed like:

Qu(w)

=

Qu(w) is the aggregated relevance between user u and topic w. The 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. The Qu(w) will become constant, if the (u, a) is determined. It will not affect the ordering of the rel(a,w).

The online retrieval algorithm is working like this:

1. For each user u in U(space of all users)
2. For each Topic w in T(space of all topics)
   1. Maintain T inverted lists
   2. sort them by rel(a,w)
3. if read operation is triggered then
   1. retrieve the sorted lists
   2. apply the TA (the threshold algorithm)

the TA algorithm is working as follows:

1. for all T sorted lists
2. get the next object in sorted order
3. for each object S returned
   1. retrieve all fields X1, X2, …., Xm
   2. compute T(X1, X2,….,Xm)
   3. if one of the top-k (most relevant ad) has been found, store it.
4. Let xi be bottom value seen in the Li
   1. Define the threshold value t to be T (X1, X2, …, Xm)
5. Stop when found k objects with T value ≥ t
   1. Return top k object

Example: let’s assume, that we have one user (u = 1) and we want to recommend one ad(k=1) for him.

1. Let the weighting parameter α = 0.5, number of posts m = 3, and the number of topics |T| = 3. The static user interest is Hu = (0.3,0.5,0.2) and the topic distribution for these three posts are (0.3,0.7,0.0), (0.4,0.5,0.1), (0.2,0.6,0.2).
2. When u triggers a read operation, the context-aware query vector Qu is calculated using the previous equation like this: Qu= 0.5 · (0.3, 0.5,0.2) + 1−0.5/3 [(0.3, 0.7,0.0) + (0.4, 0.5,0.1) + (0.2, 0.5,0.1)] = (0.41,0.47,0.13)
3. Suppose Qu is used to query an ad database with four tuples {a1 = (0.2, 0.3, 0.6), a2 = (0.5, 0.6, 0.1), a3 = (0.7, 0.1, 0.2) and a4 = (0.4, 0.5, 0.8)} now we compute the relevance between an ad and a user, by multiply the Qu with the relevance between an ad and a topic rel(a,w). For the relevance between the first ad (a1) and a user it will be like this.

(0.41, 0.47,0.13). 0.2 = 0.2 for the first post, (0.41, 0.47,0.13). 0.3 = 0.3 for the second post

(0.41, 0.47,0.13). 0.6 = 0.6 for the third post. We do the same calculations for the rest of the ads (a2, a3, a4) and sort them.

1. Now we have three inverted list lw1, lw2, lw3. lw1 = {(a3,0.7), (a2,0.5), (a4,0.4), (a1, 0.2)}, lw2 = {(a2,0.6), (a4,0.5), (a1,0.3), (a3,0.1)} and lw3 = {(a4,0.8), (a1,0.6), (a3,0.2),

|  |
| --- |
| The threshold is the sum of these values which is t = 2.1. |

(a2,0.1)} now we apply the TA algorithm.

Now we calculate the score of a3, a2, a4. The Score of a3 = 0.7+0.1+0.2 = 1.0, a2 = 1.2, a4=1.7, so all of them is below the threshold.

|  |  |  |
| --- | --- | --- |
| Lw1 | Lw2 | Lw3 |
| a3 0.7 | a2 0.6 | a4 0.8 |
| a2 0.5 | a4 0.5 | a1 0.6 |
| a4 0.4 | a1 0.3 | a3 0.2 |
| a1 0.2 | a3 0.1 | a2 0.1 |

Now we go down to the second line, therefore the new threshold is t = 1.6. The score of a2 is precomputed which is 1.2, a1= 1.1 and a4 = 1.7. now we have a4 which have the score

t value > threshold. Therefore, a4 will be recommend to the user. In case more than two k needs to be recommended, then we go to the second line and a2 will be the recommended as well with a4.

# Safe region algorithm

According to some studies, 90 % of the social media users are readers (content viewers), 9 % are editors and 1 % are publishers. 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 a waste of CPU resources to retrieve the same set of ads. a safe region has been introduced to handle this challenge. It can examine, if the top relevant ads have been changed since the last read operation or not. the algorithm implemented effectively by implementing a safe region for each user. If 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.

# Safe Region Construction

a rectangle has been constructed in the high-dimensional topic space. Whenever new posts are in rectangle boundaries, the top ad will not change. We call the high-dimensional rectangle a safe region, denoted by S = (Qulb, Quub), where Qulb stores the lower bound of coordinates in all the dimensions and Quub stores the upper bound. A greedy safe algorithm has been proposed (GSR), to incrementally build the safe region. The algorithm works as follows:

1. R = Use online retrieval algorithm to compute the relevant ads and store them in R.
2. While true do
   1. Choose the most promising topics to expand the safe region
   3. For all ads(a) in R do
      1. where MinS is the minimum relevant between an ad and the safe region
   4. For all ad(a) in A \ R do
      1. where MaxS is the maximum relevant between an ad and the safe region.
   5. If Su ≥ Si then
   6. Else
      1. Return ()

An overview of this algorithm:

1. store the set of most relevant ads for the current news feed in R.
2. choose the most promising topics to expand the safe region.
3. For each of these topics we calculate the distance between Qu and the lower, upper bound. Then choose the topic with the minimum distance to these boundaries.
4. We must be aware, that this explanation is safe, by implanting an expansion unit , which is the maximum allowed change for a given query topic Qu(w).
5. The last condition indicates, that if the minimum relevance for a query topic Qu(w) to the top K ads in R (Su)is bigger or equal to 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.

Example: let us suppose, that we have a safe region and we want to expand it. Let’s take the query vector from the previous example Qu (0.41,0.47,0.13). let’s suppose that 0.41 is the topic distribution for economy, 0.47 sport and 0.13 science. We have recommended for this user (a4 and a2), they will be stored in R, while (a1, a3) will be in A \ R and we have m = 3 and = 0.5.

1. Our promising topics are (economy, sport and science).
2. Now we calculate the minimum relevance between the ads in R and a safe region, which is . For the a4 MinS will be the multiplication between

∑ (0.4,0.5,0.8). (0.41,0.47, 0.13) = (0.697,0.799,0.221). 0.697 is for economy, sport and science respectively, a2 it will be like this ∑ (0.5,0.6,0.1). (0.41,0.47,0.13) = (0.492,0.564,0.156).

1. Now we calculate the distance between the topics and the boundaries of the safe region for this topic. Let’s assume that the boundaries are and . this indicates that (0.1,0.7) is the boundaries for economy and so on.

for the economy it will be like this (0.697 – 0.2) + (0.7 – 0.697) = 0.5. we do the same for sport it will be 0.5 and finally for the Science it will 0.43, the total distances vector for a4 is (0.5,0.5,0.43). now we do the same for a2 and we get the distances as follows for economy = 0.5, sport = 0.5and science = 0.43. the vector will be like this (0.5,0.5,0.43).

1. We have the vector for the minimum distances between a4 and a2 is (0.5,0.5,0.43), We choose now the topic with the minimum distance to the boundaries to expand the safe region, which is the Science Su = 0.43.
2. We examine whether it is safe to expand the safe region, which is the science topic by the expansion unit .

Is 0.43 – ≥ , 0.43 – ≥ 0.17 which is true.

Is 0.43 – ≥ , 0.43 + ≤ 0.6 which is also true. the expansion is safe.

1. We compute now the max relevance between the ads, which are not in R, in our example (a1, a3). We compute the MaxS in the same way as MinS. We will get the following for (a1) MaxS = (0.451,0.517,0.143), a3(0.41,0.47,0.13).
2. Now we calculate the distance and take the max distance in the same way used before to compute MinS. We will get the following results, for a1(0.5,0.5,0.43), a3(0.5,0.5,0.43).
3. Now we have the two vectors for a1, a3. we choose now the minimum vector, which is (0.5,0.5,0.43). we choose the max now in science topic which is SI= 0.43.
4. If we check now we see that Su ≥ SI (0.43 ≥ 0.43). So, we can extend the safe region with the science topic. So, whenever the user still interested in science, the same set of ads will be recommended for him, so we don’t have to recalculated it again. Even if his interests change a by we still can recommend them for him.
5. If we continue we will see, it also safe to expand the safe region with economy and sport. In other words, whenever the user requests for his news feed, if Qu still located in S= () (his still interested in these topics), which is true in our example, (a4 and a2) will be recommend for him, without needing to recalculate the top-k again (using online retrieval algorithm).

# Optimizations

There are two possibilities for optimization, the first optimization is to reduce the number of MaxS computing times. Therefore, an upper bound has been developed (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. When the Qu moves out of the safe region of a given user (u), the top-k should be recalculated 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 Qu, 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.

# 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 the model uses the online strategy, otherwise used the safe region strategy.

# **Variance of topic distributions**

To measure the topic distribution a series of equations has been introduced as follows:

Xw,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, Mv,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. Because we have variance of topics, the equation will be defined is like this:

After making some derivation processes the final to calculate the variance of a topic w in a user v’s news feed, is as follows:

**7.2 Hybrid Retrieval Strategy**

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

There is a small gap as this equation only considers the topic distribution for the write operations, while ignoring the read frequency which is important. the last equation has been adopted, to fill up this gap by introducing a read frequency ηv. Here, λn measures the probability of selecting a post from a neighbour n for user v. The final equation for the hybrid model is as follows:

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

The hybrid algorithm is simple:

1. For each user v in U (Space for all users) do
   1. (threshold) t = ρmax
   2. Compute P\*(v)
   3. If P\*(v) < t
      1. Use safe region to calculate top-k for this user
   4. Else
      1. Use the online retrieval algorithm to calculate the top-k

Example: let’s suppose, that we have two topics t = 2, user static interest Hu(0.5,0.7), α = 0.4, m = 3, pmax= 0.5, N= 2(two neighbours), the possibilities, that the user will read a post from his first neighbours λ = 0.7 and λ = 0.5 for the second, = 0.8. Now we want to choose a strategy to calculate and recommend the top-k for a given user, either online retrieval algorithm or safe region algorithm.

* + 1. calculate p\*(v), we calculate the first part of p\*(v) which is , which is equal to (0.7 + 0.5)/0.8 = 1.5. now we calculate the second part P(v), for simplify the calculations we will assume var[Xw,v] = 0.9 and E[Xw,v] = 0.3 for the first topic, then P(v) = = 0.7. for the second topic we assume var[Xw,v] = 0.4 and E[Xw,v] = 0.5, then P(v) = 0.33. we choose now the max, which is =0.7 and multiply it with 1.5. then P\*(v) will be 1.05.
    2. We see that P\*(v) is greater than the threshold, as a result, the model will choose the online retrieval algorithm to calculate the top-k for the user v. in other words the query vector now out of the boundaries of the safe region, we need to recalculated using the online retrieval algorithm.

# Experimental study

A series of experiments on real social network datasets with billions of edges such as Twitter and News and Advertisement Datasets such as Amazon Products and AOL has been done. The 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.

**Measurements**

The hybrid method can outperform GSR by up to 30x speedups and online retrieval by up to 11x speedups in the experiments. 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.

We can see in the graphs below, that the hybrid model strategy still out performs the online retrieval and the GSR model, when the k-increases (the number of ads to be embedded in the news feed), vary of α, vary read/write ratio and vary number of topics.

A close up of a map

Description generated with very high confidenceA close up of a map

Description generated with high confidenceA picture containing text, map

Description generated with very high confidenceA screenshot of a cell phone

Description generated with very high confidence

**Drawbacks**

The value of p(the threshold) have been tunned, where the hybrid model gives it best. However, if the p has been tuuned to be large, the news feed of users will be mostly depending on the static user interests. Therefore the model will use the GSR to ad recommendation the and their may be frequent update of their safe regions, incurring high CPU cost. The same case will be if p is small, then the news feed will be depending on the dynamic user interests, the online retrival algorithm will be chosen to ad recommendation, CPU resources will be wasted if the read frequency is high. In these cases, we will not benefit from the hybrid model, because it will behave as one of the other algorithms (either online algorithm or GSR) and the other algorithm will not be considered.

# Conclusion

In the studied paper an online retrieval strategy. It retrieves a user’s news feed and computes the top-k based on the TA algorithm. Then we 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, we have produced the hybrid model. By tuning the value of the threshold, where the hybrid model can use both of algorithms and according to many experiments on huge datasets, the hybrid model has proved to be efficient and powerful.