Recommendation System for Facebook Public Events based on Probabilistic Classification and Re-ranking

Dien L. Nguyen
Faculty of Information Technology
Ho Chi Minh University of Science
Ho Chi Minh City, Vietnam
Email: 1212084@student.hcmus.edu.vn

Tung M. Le
Faculty of Information Technology
Ho Chi Minh University of Science
Ho Chi Minh City, Vietnam
Email: lmtung@fit.hcmus.edu.vn

***Abstract*—The evolution of Facebook social network with event feature helps improve the quality of interaction among users in real life. Different from other objects such as movies and books, recommendation problem for events is inherently cold-start and affected by the natures of each social network platform (Facebook, Meetup, etc). In this paper, we discuss the character of Facebook social network and how the event recommendation problem in this case is different to others. We argue that rather than just trying to place as many as possible the most suitable events (often based on similarity measurement) on top recommendations, it's better to remove the unsuitable ones and reorder the remaining in a way that improves the user experience. From that, we propose a new method for event recommendation divided into two consecutive stages - classification and re-ranking. For the first phase, we use a blending model of probabilistic classifiers to predict positive and negative probabilities for each user-event pair, then evaluate on those results to eliminate all bad cases before passing the rest to the next phase. We also propose a new optimization procedure in this evaluation process. For the second phase, we treat the positive probability as a measurement of similarity and make some comparisons across several re-ranking techniques to choose the best based on the objective of improving quality of recommended lists. Experimental results on crawled Facebook public events show the effectiveness of the proposed method.**

I. Introduction

Facebook is the largest social network with 1.59 billion monthly active users [1]. On Facebook, the user can perform many actions: sharing news, ideas, photos, statuses; making discussions about particular topics; sending messages; etc. Facebook Event is a useful feature and used for various purposes besides the most usage for event announcement, such as making discussions about a particular topic. Although this utility has been developed with many powerful features, events are usually buried in tons of other information, so the users rarely notice about them. With the potential from the number of users, an application focusing only on events can solve this problem effectively. However, for the initial step, in this paper we focus on analyzing and constructing a recommendation system for Facebook public events.

Cold-start is a natural problem in event recommendation. For the objects like movies and books, the user usually cannot achieve a full review by himself in a short time, so a reasonable way is glimpsing some accessible comments or ratings of other users before making a decision. In those cases, collaborative filtering gets the job done. Events are short-term and have little or even no information about attendees. Therefore Collaborative filtering is inappropriate in event recommendation problem [2].

Facebook has some different characteristics from event-based social networks (EBSNs) such as Meetup. In Meetup, the user must belong to some groups, and actions on this site are all about events, from joining an existed to creating and promoting a new event. Meetup aims to create enduring like-minded communities and keep sustainable relationships through real life activities called meetups. The results of [3] show that the social context (derived from group membership) is the most important factor in making decisions on this site. Different from that, Facebook is designed mainly for sharing information. Although the user can be congregated by public fan page or group, they might be totally strangers in real life. Besides, the information about members and their profile are not public, so we suppose that the social context only makes a slight effect in event recommendation problem and overlook these information in this paper.

In most of previous studies, the solutions for event recommendation problem focus on providing the most suitable suggestions based on similarity ranking. The issues of this approach are it's easy to make users feel boring when they have to see many similar events repeatedly; and because the unsuitable ones aren't inspected, it's more risk when applying any techniques to improve the recommendation diversity. We suggest a different approach that firstly gets rid of unsuitable events, then we can apply any diversity improving techniques with less risky. From this and the above analyses, we propose a new method for event recommendation problem including two stages: classification and re-ranking. Independently for each stage, we exploit various techniques and choose the best. The two finest will be incorporated to make a complete recommendation system. Through the experimental results on crawled Facebook public events, we build the system step-by-

978-1-4673-8929-7/16/$31.00

step and show the effectiveness of the proposed method. The contributions of this work can be summarized as follows:

- We propose a different viewpoint on the event recommendation problem, that keeps the user away from the unsuitable events before making the most comfortable suggestions for them.
- We propose a new method for event recommendation divided into two consecutive stages: classification and re-ranking.
- We propose a new optimization procedure for probabilistic binary classification. This will be described in part B of section III.

The remainder of this paper is organized as follows. In section II we discuss some related works. In section III we present the proposed method divided into three subsections as feature construction, classification and re-ranking. In section IV we analyze the experimental results. We conclude the paper in section V and offers some directions for future work.

II. Related Works

In this section, we briefly describe some works relating to our proposed method.

A. Recommendation as Classification

For the purpose of recommending a list of items without exact rating points, a recommendation problem can be interpreted as a classification problem. Basu et al. [4] presented an approach which formalizes movies recommendation to a binary classification problem. The model takes a combination features from both ratings and content information of each user-movie pair as input, and produce a label as output indicating whether the movie is liked or disliked by the user. The recommended result for the user is an unordered list containing only the liked movies.

In event recommendation problem, unordered output is quite not useful. A user cannot perceive the whole list in just one time, so the first seen things might have higher selected probabilities, and the last are often overlooked when some prior were selected. Moreover, events are usually short-lived items and cannot be traced back when they expired, so users could miss the really interested events unless they had seen them earlier. From those points, in event recommendation problem, this approach should only be applied as a filter to eliminate negative items from the results passed to posterior processes.

B. Event recommendation from contextual information

Because of the cold-start problem, the interaction information between events and users are rarely available. Some approaches using contextual information were proposed to deal with this problem and improve the accuracy of recommended results. Macedo et al. [2] presented some evaluations about the effects of geographical and time information on decisions of users. [3] described a contextual learning approach and showed the effectiveness of using social, content, location and time information for mitigating the cold-start problem. Daly et al. [5] classified events based on social and location properties, and showed that these information might be used as a base to determine the appropriate ranking measurement.

C. Blending of models

In many classification problems, an individual model might have poor performance because of the invalidity with the data distribution or the type of problem. Combination techniques are applied to deal with this situation. Caruana et al. [6] presented a method to form an ensemble and showed the improvement in common metrics. Recently in Netflix Grand Prize [7], Sill et al. [8] presented Feature-Weighted Linear Stacking technique used in the submission of the runner-up. Moreover, ensemble method is also used in the solution of the winner with more sophisticated setup using multilevel blending of hundreds of different models [9], [10], [11]. These show the effectiveness and potential of ensemble method to event recommendation.

D. Improve recommendation diversity

Traditional approaches in recommendation problem were designed for optimizing the accuracy of recommended results. However, diversity is also an important quality, especially in the next generation of recommender [12]. In e-commerce, some studies insist that more diversity of recommended results will give more opportunities for customers to get various items, and thus increase benefits for both business and user [13], [14]. However, there is a trade-off between accuracy and diversity, and it should be considered when designing the system to match the goal.

The diversity of recommendation system can be assessed in two aspects: individual aspect, which considers diversity in each recommended list independently; and aggregate aspect, which tends to boost the total number of different items recommended in the whole system. Bradley et al. [12] presented some selection techniques to increase the diversity of individual based on relative benefit calculated from the similarity and dissimilarity between items. Adomavicius et al. [14] described some re-ranking methods to increase aggregate diversity. The relationship between the two aspects has no clear evidence. In this paper, we focus on the individual diversity for our system.

III. Event Recommendation System based on Probabilistic Classification and Re-ranking

From all available events in database, each user will be recommended a list of events which are the most comfortable for them. For a user, any event can be classified into one of two cases: interested or uninterested. The process to determine which case an event belongs to can be formalized as a binary classification problem. Besides that, users usually treat interested events not equally, and might want to discover more things he hasn't seen yet. Re-ranking technique plays an important role in improving the quality of recommended list and satisfying those behaviors of the users. All of these form a system for event recommendation problem. In the following, we start with constructing the feature set, then describe the two main stages of proposed recommendation method: classification and re-ranking.

A. Feature construction

The Feature set is built for each pair of user-event (*u*, *e*). As mentioned above, all information about friendship and membership are inaccessible, so we won't investigate anything related to those and focus on extracting features from available information of events. A public event on Facebook usually has: name, owner, occurring time, location, descriptions, list of going, interested and declined users. From these information, we construct a feature set about content, location, time, and owner of events. Because behaviors are indicated by historical decisions, these features are tied to the past of user. The history of each user is constructed from the attended, interested and declined users list of events (this process will be detailed in section IV - data setup).

1) Content features: Description is a promising information for users to perceive the majority of important properties of an event. Descriptions are presented in text data and need to be preprocessed before it can be used. We use the common bag-of-words model to transform each description into a TF-IDF vector. Although the descriptions of collected events in this work are in both English and Vietnamese, we don't do any translation because there is no tool to do this job effectively, and for somewhat it might diminish the attributes of language used in the events. The only additional action we did is segmenting Vietnamese descriptions into lexical units by vnTokenizer [15] tool, because Vietnamese has some distinct characteristics needing to be tackled.

For each pair of user-event (*u*, *e*) to be predicted, we separate historical events of *u* into two types: interested (attended or interested) and uninterested (declined or be invited but not going). For each type, we calculate the cosine value between *e* and every event of the current type, then choose the minimum, the maximum and the average values to form the content features. There are 6 features constructed in this step. Abbreviations of them are: min_int_sim (minimum similarity with interested events), max_int_sim (maximum similarity with interested events), avg_int_sim (average similarity with interested events), min_uint_sim (minimum similarity with uninterested events), max_uint_sim (maximum similarity with uninterested events), avg_uint_sim (average similarity with uninterested events).

2) Location features: We use the latitude-longitude representation of places to construct location features. Users concern the information about venue when they are interested in the event and want to go. The mobility patterns might exist but they are different among users. While some people tend to join the events near their home, some others prefer to join the events on suburban venues. The patterns often depend on many factors such as home location, workplace, environment, etc., but these information is inaccessible to collect in Facebook social network. However, when users attended in some events held at particular sites, they can also go to the future events held at the same location or nearby. Besides, a user interests in an event but decides not to go after seeing the venue mostly because it's held at the location that too far from the areas he can reach. Follow these insights, we calculate the geographical distances from e to the events that user u had attended in the past. Finally, we choose the maximum and the average values to form the location features. There are 2 more features added to the model after this step. Abbreviations of them are: max_dist (maximum distance), avg_dist (average distance).

3) Time features: Start time of event is a concerned factor for users to make decisions. Some users tend to join events at the weekend, while some other prefer weekday events. We suppose there exists the pattern of time when a user prefer to attend. We formalize this intuition by calculating the ratio of events occurring on the same day with e in historical interested events of user *u*. We also calculate the corresponded ratio in historical uninterested events. Then we use the two ratios as time features. Abbreviations of them are: dow_int (day of week from interested events), dow_uint (day of week from uninterested events).

4) Owner features: An owner of a public event on Facebook has a unique ID as a fan page or a user. It can be supposed that events hosted by the same owner are about a particular or some similar topics. Joining in events of the same owner for multiple times reflects not only the preferences of the user but also the reputation of the owner. We capture this idea in a similar way with time features. For each kind of interested and uninterested event, we calculate the ratio of number of events hosted by the owner of *e* in the past. If the owner of event *e* didn't host any historical interested or uninterested events of user *u*, the ratio value for corresponded kind of event is zero. After this step, we have two more features for the classification model. Abbreviations of them are: owner_int (owner from interested events), owner_uint (owner from uninterested events).

B. Classification

Classification model will calculate 2 probabilities for each test case. We use a fusion model made by Random Forest and Logistic Regression to mitigate overfitting because the collected events are in a small area (all of them are limited in Ho Chi Minh city).

For each pair of user-event (*u*, *e*), let $P_p(u,e)$ is the probability of event *e* is interested (positive), and $P_n(u,e)$ is the probability of *e* is uninterested (negative) by user *u*. There are 3 scenarios can be used to determine the class of each test case:

- **Scenario 1**: compare the two probabilities and choose the class corresponding to the greater.

$$Class(u,e) = \begin{cases} positive, & P_p(u,e) > P_n(u,e) \\ negative, & P_p(u,e) \leq P_n(u,e) \end{cases}$$

- **Scenario 2**: compare $P_p(u,e)$ with a constant $threshold_p$. If the probability is greater then the result is positive, otherwise negative.

$$Class(u,e) = \begin{cases} positive, & P_p(u,e) > threshold_p \\ negative, & P_p(u,e) \leq threshold_p \end{cases}$$

- **Scenario 3**: compare $P_n(u,e)$ with a constant $threshold_n$. If the probability is greater then the result is negative, otherwise positive.

$$Class(u,e) = \begin{cases} negative, & P_n(u,e) > threshold_n \\ positive, & P_n(u,e) \leq threshold_n \end{cases}$$

The effectiveness of each scenario will be evaluated by four indexes: error rate, precision, recall and specificity. Besides the accuracy of classification model, we care about increasing the specificity, because it has an important role in users' experience. The lower this index is, the higher probability that uninterested events will be recommended for user. When users have to see too many events they don't like, the reliability of the system will be depleted.

In both scenarios 2 and 3, there is a threshold constant need to be defined before applying the classification model on test data. We identify these constants through a tuning process, which is manipulated from the random search for hyper-parameter optimization method analyzed by Bergstra et al [16]. The searching procedure is performed in *n* epochs on the training set, then the results are averaged to obtain the final threshold. Each epoch is conducted as follows:

- Generate a random permutation of training data, then use the first 2/3 as train set and the last 1/3 as validation set.
- Use the train set for training model, then predict probabilities of positive and negative classes on the validation set.
- For each scenario, take the corresponded predicted probabilities and labels of the validation set to fit an SVM classifier. After that, calculate the average value of all support vectors (with only 1 dimension) and use this as optimized threshold.

The scenario which has the best performance on test data will be used in our system. After classified, all cases predicted as negative are eliminated and the rest will be passed to re-ranking phase.

C. Re-ranking

Although we have attempted to remove as many as possible uninterested events for each user in classification phase, some of them might be wrongly classified and passed to re-ranking phase. Therefore, we need a ranking algorithm pushing them to as last as possible positions of recommended list to minimize the risks of user experience. We treat the positive probability predicted in the prior phase as a measurement of similarity and calculate score of each event from this value, then sort all the events in descending order of score to make the final list for the user. Base on the idea about re-ranking methods from [14], we do some comparisons and choose the best among the following scoring techniques:

- **Similarity**: score of event e_i is its own positive probability:

$$score(u, e_i) = P_p(u, e_i)$$

- **Reverse Similarity**: score of event e_i is reverse value of positive probability:

$$score(u, e_i) = 1/P_p(u, e_i)$$

- **Serendipity**: score of event e_i is the product of similarity and (1 - similarity):

$$score(u, e_i) = P_p(u, e_i) \times (1 - P_p(u, e_i))$$

- **Popularity**: let $U(e_i)$ is the set of all users who interacted with e_i, and $U_p(e_i)$ is the set of the users who attend or interest in e_i ($U_p(e_i) \subseteq U(e_i)$). Score of event e_i is the number of users in $U_p(e_i)$:

$$score(u, e_i) = |U_p(e_i)|$$

- **Relative Popularity**: score of event e_i is the ratio of users in $U_p(e_i)$ among all users who interacted with e_i:

$$score(u, e_i) = |U_p(e_i)|/|U(e_i)|$$

To evaluate the effectiveness of the re-ranking algorithms, we use MAP@10 metric, which is the mean average precision of all users at top 10 events in ranked list. The average precision at n for a user is:

$$ap@n = \sum_{k=1}^{n} \frac{P(k)}{min(m, n)}$$

where P(*k*) is the precision at cut-off *k* in the event list, *m* is the number of ranked events. If *m* is zero, ${}^{P(k)}/_{min(m,n)}$ is set to zero. Then the mean average precision for *N* users at *n* is:

$$MAP@n = \sum_{i=1}^{N} \frac{ap@n_i}{N}$$

Beside the MAP score, we also consider the diversity of recommended lists. Because the users are treated independently with others, we evaluate the diversity on individual aspect by measuring the average dissimilarity of all pairs in the same list. The formula to calculate the dissimilarity of a pair is:

$$Dissimilarity(e_i, e_j) = 1 - cos(v_i, v_j)$$

where v_i, v_j are the corresponded TF-IDF vectors of events e_i and e_j. Finally, we average the results of all lists to obtain the system's diversity.

IV. Experimental Evaluation

In most of the previous works [3], [17], recommendation systems were built based on trying to emplace the attended events of user on top recommendations, and treat all the not attended events as wrong predictions. We found that this is a pessimistic treatment, because the not attended events also include many unseen ones, which we don't have any evidences to verify that they are suitable for the user or not, especially in the information social network platform like Facebook. In this study, we only evaluate the interacted events of the user and try to differentiate the suitable and unsuitable ones, so it would be inappropriate to do comparison between the approaches which have too different viewpoints.

Firstly in this section, we describe how the data is prepared. Then we assess the effects of the features on the classification model. Finally, we analyze the performances of the three scenarios (in classification phase) and the five scoring techniques (in re-ranking phase) mentioned above.

A. Data Setup

Data is crawled by Graph API including 1160 Facebook public events limited in Ho Chi Minh City. All of them are short-time which occurs within a day, and they were held in the period between 2013 and 2016. In table I, we summarize some information including the number of events, users and different locations in the whole dataset. The information of each event includes name, description, start time, location, owner, list of attendees, list of users who interested in event and list of users who had declined to go. From the three lists of users, we transform to a new key-value format, where each key is a user and the corresponded value is a three-list, which includes the list of attended, the list of interested and the list of declined events sorted in ascending order of start time. Then for each user, we take the first half of each of the three lists to make the history. The rest are used to make a database for generating training and testing data.

TABLE I: Statistics of the dataset

City	No. of events	No. of users	No. of locations
Ho Chi Minh	1160	13683	524

We conduct 10 epochs for both training and testing. In each epoch, we randomly generate a train set and a test set from database, fit a model from train data, then apply on the test set to obtain the results. For scenario 2 and 3, we do a hyper-parameter tuning process to calculate the corresponded threshold before fitting the model. After 10 epochs, we calculate the average values of all results and use them for analyzing.

B. Feature Analysis

Figure 1 shows the importance of features in the classification model. Location features (max_dist: 0.177, avg_dist: 0.176) are the most important. The content features also strongly affects (max_int_sim: 0.124, avg_int_sim: 0.114) to the classifier.

The results can be explained by the order of information seen by users. Description is usually the first thing users want to see, because it has more information and might cover all of the other parts of an event. After reviewing the description, a user can decide that whether he interests in this event or not. If not, he will overlook it right away and discover other events. But if he wants to go, he will be curious about the

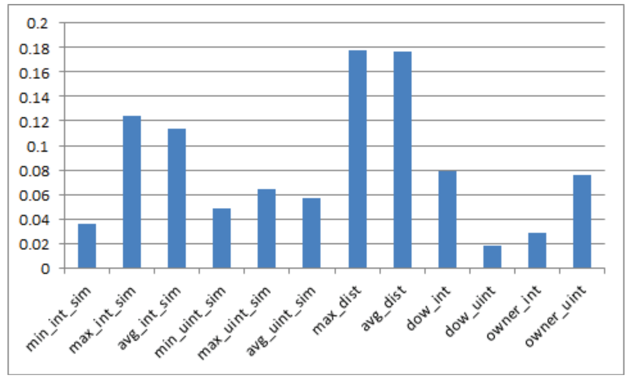

Fig. 1: Influence of features on classification model.

venue, and the decision will be made at this time. The other information will be assessed when the user wants to confirm or change his decision.

C. Classification Analysis

Figure 2 shows the performances of the three scenarios estimated by four indexes: error rate, precision, recall, specificity. In the first three indexes, there is no significant difference among scenarios, despite the fact that scenario 3 has a little poorer performance than others (the differences are less than 4%). However, at the specificity index, scenario 3 outperforms the others with approximately 12% higher.

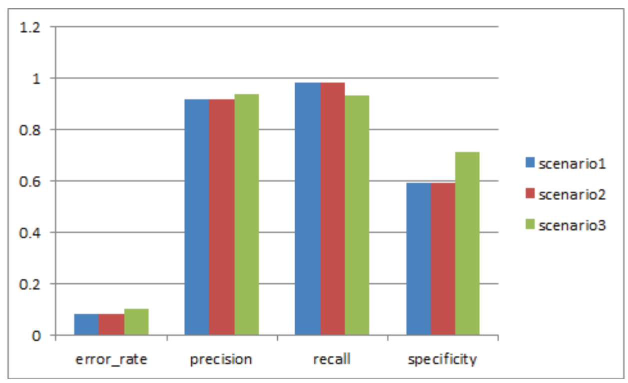

Fig. 2: Error rate, precision, recall, specificity of three scenarios.

The average optimized thresholds we obtain from validation process are 0.374 for $threshold_n$ and 0.626 for $theshold_p$. These two values show that the classifier tends to classify an event to negative class in scenarios 2 and 3. However, scenario 3 directly assess the negative probabilities and be optimized to classify uninterested events, so it outperforms the others at specificity index. Since the trade-off with other indexes is acceptable, we choose scenario 3 for our system and pass its results to the next phase.

D. Re-ranking Analysis

From the results of classification phase, we eliminate all negative events before conducting the re-ranking process.

Figure 3 shows the MAP@10 performances of the five scoring techniques. The differences among these techniques are very small (max: 0.879, min: 0.872), but ranking by relative popularity produces a little higher MAP score than the others.

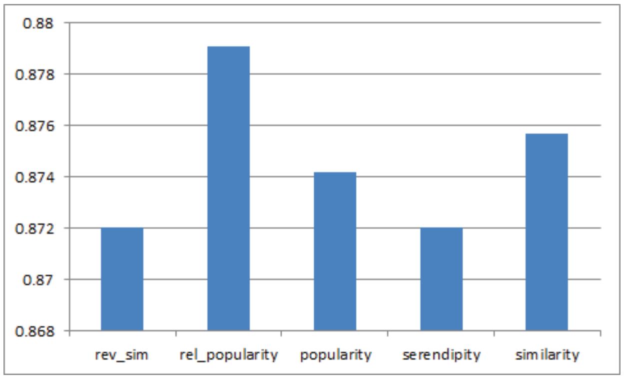

Fig. 3: MAP score of 5 re-ranking techniques.

Figure 4 presents the diversity scores of the five techniques. The differences are insignificant (max: 0.2424, min: 0.2421), but Reverse Similarity and Serendipity are better than the rest.

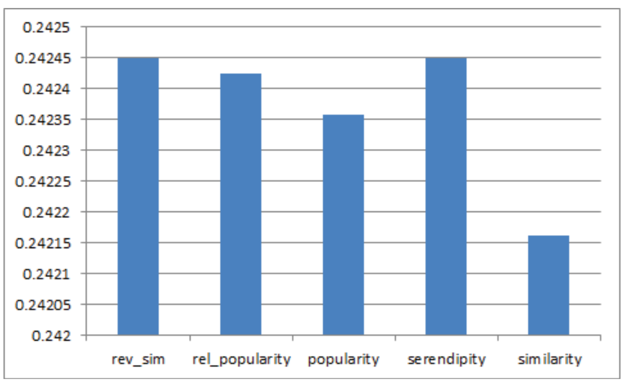

Fig. 4: Diversity score of 5 re-ranking techniques.

The results can be explained that the data is relatively narrow, since it only has explicit feedbacks from users (going, interested, declined) but has no information about users who saw the events but didn't react. Therefore in the data, the decisions of users were made deliberately which leads to the performances with high accuracy in the classification phase (the error rate doesn't exceed 11%). As a result, there is no significant differences between the scoring techniques in re-ranking phase. However, there is a trade-off between MAP and Diversity scores. We choose the Relative Popularity for our system, because its performance and trade-off are better than the others. The Relative Popularity technique is careless about the similarity of events, but based on the ratio of positive react from users, which indicates the reputation of events. Therefore, the ranking results will have more chances to be diverse and have better quality, which helps improve the user experience and the reliability of recommendation system.

V. Conclusions and Outlook

In this paper, we discussed the character of event recommendation problem in Facebook social network. We proposed a new method to solve this problem based on probabilistic classifiers including classification and re-ranking phases. In the former phase, most of the bad cases are eliminated by assessing directly on the negative probabilities by an optimized threshold. The latter phase uses Relative Popularity scoring technique that helps the correctly classified events hold higher positions than the misclassified ones while increasing the diversity of recommended lists. Experimental results on Facebook data, which includes 1160 public events limited in Ho Chi Minh City, show the effectiveness of the proposed method. In the future, we will expand the research for different cities and apply this work to constructing an event recommendation application for Facebook users.

References

[1] Statista, "Global social networks ranked by number of users 2016," 2016. [Online]. Available: http://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/
[2] A. Q. De Macedo and L. B. Marinho, "Event recommendation in event-based social networks." in *HT (Doctoral Consortium/Late-breaking Results/Workshops)*, 2014.
[3] A. Q. Macedo, L. B. Marinho, and R. L. Santos, "Context-aware event recommendation in event-based social networks," in *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM, 2015, pp. 123–130.
[4] C. Basu, H. Hirsh, W. Cohen *et al.*, "Recommendation as classification: Using social and content-based information in recommendation," in *AAAI/IAAI*, 1998, pp. 714–720.
[5] E. M. Daly and W. Geyer, "Effective event discovery: using location and social information for scoping event recommendations," in *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 2011, pp. 277–280.
[6] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes, "Ensemble selection from libraries of models," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 18.
[7] netflix, "Netflix prize homepage," 2016. [Online]. Available: http://www.netflixprize.com
[8] J. Sill, G. Takács, L. Mackey, and D. Lin, "Feature-weighted linear stacking," *arXiv preprint arXiv:0911.0460*, 2009.
[9] A. Töscher, M. Jahrer, and R. M. Bell, "The bigchaos solution to the netflix grand prize," *Netflix prize documentation*, pp. 1–52, 2009.
[10] Y. Koren, "The bellkor solution to the netflix grand prize," *Netflix prize documentation*, vol. 81, 2009.
[11] M. Piotte and M. Chabbert, "The pragmatic theory solution to the netflix grand prize," *Netflix prize documentation*, 2009.
[12] K. Bradley and B. Smyth, "Improving recommendation diversity," in *Proceedings of the Twelfth National Conference in Artificial Intelligence and Cognitive Science (AICS-01)*. Citeseer, 2001, pp. 75–84.
[13] D. Fleder and K. Hosanagar, "Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity," *Management science*, vol. 55, no. 5, pp. 697–712, 2009.
[14] G. Adomavicius and Y. Kwon, "Toward more diverse recommendations: Item re-ranking methods for recommender systems," in *Workshop on Information Technologies and Systems*, 2009.
[15] N. T. M. Lê Hng Phng, A. R. Huyn, and H. T. Vinh, "A hybrid approach to word segmentation of vietnamese texts," in *Proceedings of the 2nd International Conference on Language and Automata Theory and Applications*, 2008.
[16] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.
[17] Y. Zhang, H. Wu, V. S. Sorathia, and V. K. Prasanna, "Event recommendation in social networks with linked data enablement." in *ICEIS (2)*, 2013, pp. 371–379.