Second Screen User Profiling and Multi-level Smart Recommendations in the context of Social TVs

Angelos Valsamis*, Alexandros Psychas, Fotis Aisopos, Andreas Menychtas and Theodora Varvarigou

Distributed, Knowledge and Media Systems Group, National Technical University of Athens, Greece

angval@central.ntua.gr, {alps, fotais, ameny} @mail.ntua.gr, dora@telecom.ntua.gr

Abstract. In the context of Social TV, the increasing popularity of first and second screen users, interacting and posting content online, illustrates new business opportunities and related technical challenges, in order to enrich user experience on such environments. SAM (Socializing Around Media) project uses Social Media-connected infrastructure to deal with the aforementioned challenges, providing intelligent user context management models and mechanisms capturing social patterns, to apply collaborative filtering techniques and personalized recommendations towards this direction. This paper presents the Context Management mechanism of SAM, running in a Social TV environment to provide smart recommendations for first and second screen content. Work presented is evaluated using real movie rating dataset found online, to validate the SAM's approach in terms of effectiveness as well as efficiency.

Keywords: Second Screen, Social TV, Context Management, Recommendations

1 Introduction

The usage of mobile devices has become one of the leading daily activities. This phenomenon extends to the usage of those devices in parallel with other devices also. SAM project [1] aims at exploiting, researching and creating the appropriate technologies that revolve around the usage of mobile devices simultaneously with TV, the so called 2nd screen phenomenon[1]. The software created for the purposes of SAM revolves around the creation of a complete experience for the user delivered in to his mobile device during a TV program. In a very simple way users get multimedia content (in the form of widgets) about the TV program they are watching in to their mobile devices. This Content varies from simple information about the characters of the TV program to social media content about the program.

---

[1] Second Screen Society: http://www.2ndscreensociety.com/

Initially, SAM is planned to use a set of educative-oriented videos, which are created by SAM's Media Providers and selected by didactic teams of schools for educational purposes. Contents linked to such videos will be editorially curated, and its didactic value reviewed by the didactic team of each school. The most common learning scenarios, for example, include short documentaries or nutrition-related video reports, where users (students in this case) will use their mobile devices and computers to have an interactive social experience.

However, delivering multimedia content to user mobile devices poses a variety of challenges and requirements. The creation of a mechanism that delivers personalized content as well as the contextualization of this content are the requirements that drove the research and developments of this paper. The work is summarized into two main objectives:

- Creating sophisticated contextualization of the information and content
- Create recommendation system for personalized content delivery to enrich user experience

There is a vast variety of recommendation algorithms to implement in a system. In this paper the research focus not on creating new algorithms but creating sophisticated data to use as input in these algorithms. More specifically representing relations between users and multimedia content in a more refined way and further more adapt this data to be used as input to the well-established and tested algorithms for recommendation.

The rest of the document is organized as follows: The current section introduces the Social TV Context Management concept. Section 2 presents the related work regarding 2nd screen Context Management and graph Recommendations, while Section 3 analyzes the SAM model and approach. In Section 4, experiments' configuration as well as their results is provided and finally Section 5 concludes the current paper and discusses the work to be done in the future.

2 Related Work

Context management. Social TV and 2nd screen are now some of the most emerging technologies, used for eLearning, Political Surveys or just Social Networking purposes. Cezar et al. [2] analyzed the usages of the 2nd screen in an Interactive Television Environment, to control, enrich, share, and transfer TV content. This work provided an initial market assessment in the areas of media creation and distribution and subjected its prototype implementation to test by a dozen groups of users in a social setting. Giglietto & Selva [3] applied a content analysis to a big dataset of 2nd screen tweets during an entire TV season, in order to clarify the relationship between TV political talk shows and related comments on social media. This study points out the effects of celebritization of politics and confirms the coexistence of different and interlinked forms of participation (with political prevailing on audience participation). Elaborating on personalized experiences, Geerts et al. [4] investigated a 2nd screen application, stimulating social interaction in the living room, offering more insight into how viewers experience second such applications, and contrasted this with the perspective of producers and actual usage data.

Graph analysis and recommendations. Although much work has been carried out concerning movie/TV programs, Second screen and Social TV recommendations are quite immature. Context-based recommendations using graphs are evidently the more efficient, as SQL databases are now obsolete for big data analytics. Demovic et al. [5] presented a suchlike approach, saving movie data in a graph and using Graph Traversal Algorithms to efficiently address user preferences. This work uses explicit user "likes" for movies or genres, but does not collect any contextual or social data. When it comes to Social TV Platforms, authors in [6] and [7] highlight the concept of context management and analysis in the frame of social enabled content delivery to 2nd screen devices. These papers present a novel solution for media context management in a Neo4j graph database, and provide the baseline context of the current work.

k-NN and Collaborative Filtering. Collaborative filtering techniques are commonly used for TV program recommendations. Authors in [8] use collaborative filtering for such recommendations, enhanced with singular value decomposition resulting into a low-dimension item-based filtering with promising accuracy. Andrade and Almeida [9] adopt the k Nearest Neighbors (k-NN) algorithm to implement a hybrid strategy that combines collaborative filtering with a content-based method for delivering TV recommendations to individual users. K-NN is also employed in [10] and [11] to implement personalized popular program recommendation systems for digital TV data clustered by k-means. Both works generate datasets of user profiles, to examine resulting recommendations in terms of accuracy as well as computation time.

3 SAM Context Management and recommendations

3.1 Context Management Database

The Context Manager is connected with various key components of the SAM system. In order to contextualize process and create the recommendations, Context Controller needs to be connected with data listeners to gather data from SAM 1st and 2nd screen. After processing the collected data, Context Controller stores the data to the SAM Cloud storage. SAM "Syndicator" component, which is responsible for orchestrating the 2nd screen experience, as well as the 2nd screen component itself are also connected with the Context Manager to retrieve the information needed in order to deliver the recommendations. In the following paragraphs a more in depth analysis of the storage and the recommendation techniques used will be presented.

As mentioned, graph databases are the leading solution for the data analysis and recommendation systems. All data created or imported in SAM are stored in a Neo4j Graph database in order to be further analyzed and used for recommendation purposes.

The structure of the graph database is a very important factor for the optimization of the recommendation algorithms. Graph databases contain two types of data in general, the nodes and the edges. Nodes represent entities such as persons and multimedia Assets, edges represent relationships between the entities (nodes). In SAM there are three types of nodes:

- **Assets**: which represent any type of multimedia content
- **Persons**: which represent every user that interacts with the Assets
- **Keywords**: which represent words that describe Assets

There are several types of edges that describe the relationships between nodes:

- Has-keyword is the relationship that connects the Assets with the Keywords that most appropriately describes them.
- Consume, like, dislike, comment, full-screen, dismiss and show-more are relationships that describe interactions between a user (Person) and an Asset.
- Is-root-of describes the relationship between a root asset and the assets (widgets) that are appear in the second screen when it is being consumed.

The following paragraphs explain how entities and interactions are used to estimate relevance with an asset and produce recommendations.

3.2 User interactions and analysis

SAM first and second screen listeners collect various user actions and store them in order to be able to later recommend videos and/or widgets. In particular, actions concerning videos include:

1. Comment a movie: a comment in Twitter/Facebook/SAM Dynamic Communities that is being processed by SAM's sentiment analysis service.
2. Consume a movie: action of selecting a movie and start watching.
3. Initiate Full screen: action of pressing the enlargement button in order to see the movie in full screen mode.
4. Like/Dislike a movie: action of pressing the like/dislike button under a movie.

Actions concerning widgets appearing in 2nd screen while watching a video are:

1. Like/Dislike a widget: The actions of pressing the like/dislike button under each widget.
2. Dismiss a widget: The action of pressing the dismiss button in order to hide a widget.
3. Show more: The action of pressing the show-more button, located under each widget that enlarge the widget’s size and adds more info.

SAM's sentiment analysis service [12],[13], which is used to identify sentiment on widget and movie comments, is also able to perceive the sentiment polarity of a comment in regards to the movie's keywords. For example, if the user commented "That movie was awesome. Jennifer Lawrence's acting was spot on!" the sentiment analysis service will generate a positive number for the comment in regard to the movie, and also a positive number for the movie's keyword "Lawrence". Thus, we also identify an (indirect) action: *"Comment on keyword*".

A basic part of the analysis of the graph is to apply some kind of "weights" (i.e. relevance scores) to the lines connecting users and assets. Setting +1 and -1 as absolute values of relevancy and irrelevancy respectively, we apply those values to user-asset relations that explicitly show such a rating ("like" weights for +1, "dislike" weights for -1). On the other hand, comments on assets are saved along with their sentiment polarity and intensity (percentage of positivity or negativity), thus we apply for positive comments a decimal weight, ranging from (0, +1] and for negative comments from [-1, 0). Zero value obviously expresses neutrality.

However, consuming or pressing 'Full Screen' on a video also indicates interest by the user. The same applies for pressing 'show more" on a specific widget in 2nd screen, while dismissing it before it automatically closes indicates lack of interest. To capture those implicit patterns, we need to make sure that they will not totally overlap the explicit ones already mentioned. For example, if a user has "liked" an asset, but on the other hand dismissed it early on, this implies a weaker "like" or "interest" relation. The approach that we follow to make sure the overall relevance score (sum of weights) is mainly defined by "likes" / "dislikes" and only partly affected by other interactions is to apply to the latest a weight of

$$w_i = \frac{p_i}{t+1}$$

where p = polarity indication (+1,-1) and t = number of implicit interactions existing for asset type (movie or widget). Following this approach, if an explicit interaction weight w_e contradicts to all implicit weights w_i, the overall weight

$$W = w_e + \sum w_i$$

will still bare the (now normalized) "polarity" of w_e.

Given the aforementioned list of interactions that we collect from SAM Dashboard, the weights of different asset interactions are summarized in Table 1:

Table 1. Polarity contribution of the various user interactions.

Interactions	**Movie**	**Widget**	**Keyword**
		Explicit weights w_e	
Comment	(-1,1)		(-1,1)
Like	+1	+1	
Dislike	-1	-1	
		Implicit weights w_i	
Full screen	+⅓		
Consume	+⅓		

Dismiss		-⅓
Show More		+⅓
#Implicit_interations t	2	2

Taking into account the above, the overall direct weight of a user and an asset is:

$$W_d = \sum w_e + \sum w_i$$

In the example given above, where a user "liked" an asset but dismissed it early on, the overall weight is now: W = +1 -1/3 = +2/3, which is still a positive score.

3.3 Analysis of indirect relations

When analyzing interactions, to estimate assets' relevance we identify two cases:

1. An interaction of a user with a widget/keyword of a movie. (Fig 1.)
2. An interaction of a user with a movie, with common widgets to another. (Fig 2.)

To estimate user relevance to a movie, we can also use interactions with widgets and keywords related to it and other movies already consumed. Also, we can use interactions with movies that share widgets and keywords with this movie.

For example, if a user has "liked" or commented positively for all widgets or keywords of a root asset (case 1), a strong indication of relevance to this root asset also exists. Similarly to the previous logic, we need to make sure that indirect relations to assets will not overlap a direct weight to it. Thus, for every rating to a connected widget/keyword we apply a weight of

$$W_w = \frac{r_w}{a + k + 1}$$

where r_w = rating of neighboring node, a = number of neighboring assets and k = number of keywords connected to the "under investigation" asset.

In cases, where a user has "liked" a root asset (case 2), which shares keywords or widgets with the asset under investigation, a weaker indication of relevance has to be taken into account. Thus, for every rating to a movie connected with shared keyword/widget we apply a weight of

$$W_m = \frac{r_m}{(m + 1)2}$$

where r_m = rating of neighboring node, m = number of movies related to widget.

Therefore, the indirect relevance weight of a user for an unconsumed movie is:

$$W_{ind} = W_w + W_m = \sum \frac{r_x}{a + k + 1} + \sum \frac{r_m}{(m + 1)2}$$

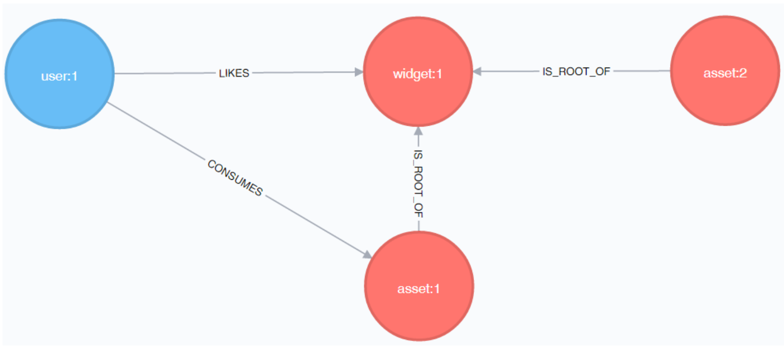

Figure 1. User interacts with widget.

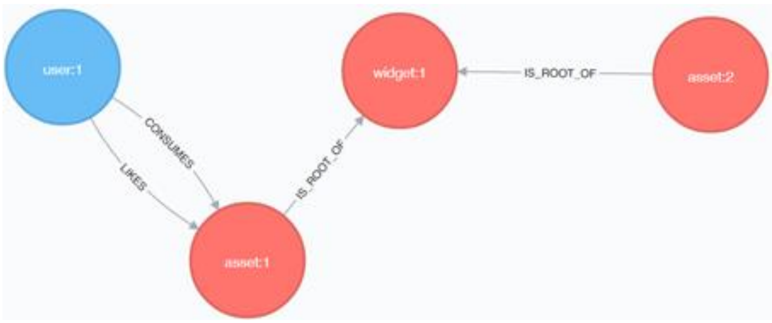

Figure 2. User interacts with movie.

When it comes to widgets of a movie, we use interactions with other widgets/ keywords belonging to that movie (case 1) and interactions with movies that share the widget in question (case 2). Similarly to the previous logic, for every rating to a root asset connected to the under investigation widget, we apply a weight of

$$W_r = \frac{r_r}{m+1}$$

and for every rating to a keyword/ widget connected to the root asset of the widget:

$$W_k = \frac{r_k}{(a+k+1)\,2}$$

where r_k = rating of neighboring node, a = number of neighboring to the widget assets and k = number of keywords connected to the root asset.

Therefore, the overall relevance weight of a person for a widget of an unconsumed asset becomes:

$$W_{ind} = W_r + W_k = \sum \frac{r_r}{m+1} + \sum \frac{r_k}{(a+k+1)\,2}$$

As a result of the analysis above, the overall relevance estimation of an asset is:

$$W = W_d + W_{ind} = \begin{cases} \sum w_e + \sum \frac{p_i}{t+1} + \sum \frac{r_w}{a+k+1} + \sum \frac{r_m}{(m+1)2}, for\ movies \\ \sum w_e + \sum \frac{p_i}{t+1} + \sum \frac{r_r}{m+1} + \sum \frac{r_k}{(a+k+1)\ 2}\ , for\ widgets \end{cases}$$

3.4 Collaborative filtering analysis

The technique described cannot provide rich results for assets that the user has not interacted with or with their neighbours ("isolated" assets). Thus, as a supporting solution, collaborative filtering is applied among users, to estimate their relevance with such assets, based on correlation with other users. A common approach for collaborative filtering on a dataset of simple numeric ratings [14], as in our dataset, is using the Pearson Correlation Coefficient. Its equation is the following:

$$c_{au} = \frac{\sum_{i=1}^{h} (r_{ai} - \bar{r}_a) \times (r_{ui} - \bar{r}_u)}{\sqrt{\sum_{i=1}^{h} (r_{ai} - \bar{r}_a)^2 \times \sum_{i=1}^{h} (r_{ui} - \bar{r}_u)^2}}$$

for users *a* and *u*, where in our case $h = |I_{au}|$ is the amount of assets rated by both users, r_{ai} is user *a*'s weight for asset *i* and $\bar{r}_a = average(r_{a1}, r_{a2}, \dots, r_{ah})$.

After calculating the correlation coefficients of a user with other users, Pearson collaborative filtering can provide a prediction, estimating her relevance with an asset j, based on other users' relevance for the specific asset and their correlation:

$$p_{aj} = \bar{r}_a + \frac{\sum_{u=1}^{g} (r_{uj} - \bar{r}_u) \times c_{au}}{\sum_{u=1}^{g} |c_{au}|}$$

where *g* is the number of users that consumed j and p_{ai} is the predicted rating of relevance for user *a*.

3.5 Asset recommendations

The analysis of the graph, together with the Collaborative Filtering method described above, provide relevance estimation for every user and asset. Relevance scores per asset allow for the implementation of the following recommendations:

- Recommendation of a ranked list of root assets, based on their relevance to users.
- Recommendation of a ranked list of widgets, while users watch their root videos.

These recommendation methods are used for the personalization of the 1st and 2nd screen environment, based on each user context. Using those, SAM is able to recommend relevant videos in 1st screen, as well as prioritize widgets when watching a video, hiding the irrelevant ones or highlighting the most interesting.

4 Experiments

4.1 Dataset and Configuration

Finding a dataset that contains user interactions, and in general two-level (implicit and explicit) data, proved challenging. For the evaluation of the current solution, we used a well known movie rating dataset found online [15], comprising a huge database of movies and user ratings, as well as keywords linked with those movies.

The dataset imported was interpreted into the SAM logic, generating SAM users, assets and keywords. We analyzed the rating values in order to simulate like/dislike actions based on those values (explicit information), and we were also able to use the implicit information of connected movies to the same keywords, thus having the two-level information that is necessary for our algorithm to display its full potential.

To limit our scale and make a meaningful analysis, we selected a random sample of available movies along with all the associated ratings and keywords. The overall numbers and statistics of the dataset imported can be found in the following table:

Table 2 – Sample retrieved from MovieLens dataset clusters correlation coefficient

	Users	**Movies**	**Ratings**	**Keywords**
Overall metrics	656	1032	9902	495

	Mean Rating	**Mean Rating per Movie**	**#Keywords per movie**	**#Movies rated by a user**
Average metrics	3.48	9.6	0.64	15.09

The dataset was split in a 70/30 ratio into training and a testing set respectively. Note that for k-NN, Pearson as well as SAM algorithm, no training step is really required, although the training set is used in the preparatory step of calculating coefficients (in standalone Pearson's filtering) and clusters (in k-NN algorithm). In case of SAM, training set is used as prior knowledge since interactions of users have to be saved in the graph in order to analyze and recommend new assets/widgets.

Experiments were performed on a desktop machine with an Intel Core TM i5-3400 Processor, 2.80 GHz, 12GB of RAM memory, running 64-bit Windows 10 Pro N.

4.2 Experiment Results

The graph analysis, supported by the Pearson Collaborative filtering, presented above, was applied and compared with the stand-alone implementation of Pearson Collaborative filtering as well as an implementation of k-NN algorithm run over Neo4j. K-NN is one of the most popular clustering approaches for recommendations using graphs. In our case, we followed the user-based algorithm approach with adjusted cosine similarity function[2]. As evaluation metric we used the variation between the ground truth rating and the relevance computed by the different algorithms. In Table 3 we present the errors of each method using root mean squared error (RMSE), mean absolute error (MAE) and mean percentage error (MPE).

Table 3 - Results of different algorithms run over Neo4j database.

Algorithm	Mean absolute error	Root mean squared error	Mean percentage error	Average Response time (ms)
K-NN algorithm	0.3415	0.4242	17.08%	6910
Stand-alone Pearson Collaborative filtering	0.2809	0.3190	14.04%	5882
SAM algorithm sup. by Pearson Collaborative filtering	**0.2584**	**0.2878**	**12.92%**	**5529**

Based on measured errors, it is evident that the graph analysis is superior to the collaborative filtering approach when there are adequate user interactions. In case where there are not enough user interactions we fall back to Pearson's filtering. In addition, as can be observed, the SAM algorithm outperforms K-NN accuracy-wise, one of the most popular clustering approaches for recommendations using graphs.

Apart from the accuracy experiments, we also measured the response time of the three different algorithms (SAM, Collaborative filtering, K-NN) exposed by SAM's Context Management component as web services. Last column of Table 3 presents the average response time in ms for each algorithm after 1000 requests on each. SAM algorithm's locality search seems to outperform other approaches.

5 Conclusions and Future work

This work has been focused on an efficient Context Management and Personalized Recommendation system for Social TV First and 2nd screen. 2nd screen Content Listening and related Recommendations is a new promising area that has yet to be explored by the research community. To this end, the authors have proposed an innovative and adaptive model, using social media and user context information, and applied over SAM's Content Syndication and Social learning environment. This model was supported by a collaborative filtering mechanism and evaluated over real-world dataset found online.

In the future, SAM will be piloted to elementary schools as well as high schools as an eLearning and Media Delivery application, in order to test its functionalities and acquire real datasets of user interactions. Those interactions will constitute a more concrete dataset, to be used in order to evaluate the current representation model and the resulting recommendation system's effectiveness.

[2] https://neo4j.com/graphgist/8173017/

6 Acknowledgment

This work has been supported by the SAM project and funded from the European Union's 7th Framework Programme for research, technological development and demonstration under grant agreement no 611312.

References

[1] Socialising Around Media (SAM) Project: Dynamic Social and Media Content Syndication for 2nd Screen, http://samproject.net/

[2] Cesar, Pablo, Dick CA Bulterman, and A. J. Jansen. "Usages of the secondary screen in an interactive television environment: Control, enrich, share, and transfer television content." *European Conference on Interactive Television*. Springer Berlin Heidelberg, 2008.

[3] Giglietto, Fabio, and Donatella Selva. "Second screen and participation: A content analysis on a full season dataset of tweets." *Journal of Communication* 64.2 (2014): 260-277.

[4] Geerts, David, et al. "In front of and behind the second screen: viewer and producer perspectives on a companion app." *Proceedings of the 2014 ACM international conference on Interactive experiences for TV and online video*. ACM, 2014.

[5] Demovic, Lubos, et al. "Movie recommendation based on graph traversal algorithms." Database and Expert Systems Applications (DEXA), 2013 24th International Workshop on. IEEE, 2013.

[6] A. Menychtas, D. Tomás, M. Tiemann, C. Santzaridou, A. Psychas, D. Kyriazis, J.V. Vidagany, S. Campbell, "Dynamic Social and Media Content Syndication for Second Screen", International Journal of Virtual Communities and Social Networking (IJVCSN), 2015

[7] C. Santzaridou, A. Menychtas, A. Psychas and T. Varvarigou, "Context Management and Analysis for Social TV Platforms", eChallenges e-2015, 2015

[8] Barragáns-Martínez, Ana Belén, et al. "A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition." Information Sciences 180.22 (2010): 4290-4311.

[9] Andrade, Maria Teresa, and Fábio Almeida. "Novel hybrid approach to content recommendation based on predicted profiles." Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC). IEEE, 2013.

[10] Lai, Chin-Feng, et al. "CPRS: A cloud-based program recommendation system for digital TV platforms." Future Generation Computer Systems 27.6 (2011): 823-835.

[11] Chang, Jui-Hung, et al. "3PRS: a personalized popular program recommendation system for digital TV for P2P social networks." *Multimedia Tools and Applications* 47.1 (2010): 31-48.

[12] Pedro Aniel Sánchez-Mirabal, Yarelis Ruano Torres, Suilen Hernández Alvarado, Yoan Gutiérrez, Andrés Montoyo and Rafael Muñoz "UMCC_DLSI: Sentiment Analysis in Twitter using Polarity Lexicons and Tweet Similarity". In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland. Page 727 - 731.

[13] Gutierrez, Yoan, David Tomas, and Javi Fernandez. "Benefits of using ranking skip-gram techniques for opinion mining approaches." eChallenges e-2015 Conference. IEEE, 2015.

[14] Konstantinos Tserpes, Fotis Aisopos, Dimosthenis Kyriazis, and Theodora Varvarigou. 2010. Service selection decision support in the internet of services. In Proceedings of the 7th international conference on Economics of grids, clouds, systems, and services (GECON'10), Jörn Altmann and Omer F. Rana (Eds.). Springer-Verlag, Berlin, Heidelberg, 16-33.

[15] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages.