2017 16th IEEE International Conference on Machine Learning and Applications

Time-Sensitive Behavior Prediction in a Health Social Network

Amnay Amimeur[1], NhatHai Phan[2], Dejing Dou[1], David Kil[3], Brigitte Piniewski[4]
{aamimeur,dou}@cs.uoregon.edu, Phat@njit.edu, david.kil@healthmantic.com, BPiniewski@peacehealthlabs.org
[1] University of Oregon, USA; [2] New Jersey Institute of Technology, USA; [3] Health Mantic Inc., USA; [4] PeaceHealth Laboratories, USA

***Abstract*— Human behavior prediction is critical in understanding and addressing large scale health and social issues in online communities. Specifically, predicting when in the future a user will engage in a behavior as opposed to whether a user will behave at a particular time is a less studied subproblem of behavior prediction. Further lacking is exploration of how social context affects personal behavior and the exploitation of network structure information in behavior and time prediction. To address these problems we propose a novel semi-supervised deep learning model for prediction of return time to personal behavior. A carefully designed objective function ensures the model learns good social context embeddings and historical behavior embeddings in order to capture the effects of social influence on personal behavior. Our model is validated on a unique health social network dataset by predicting when users will engage in physical activities. We show our model outperforms relevant time prediction baselines.**

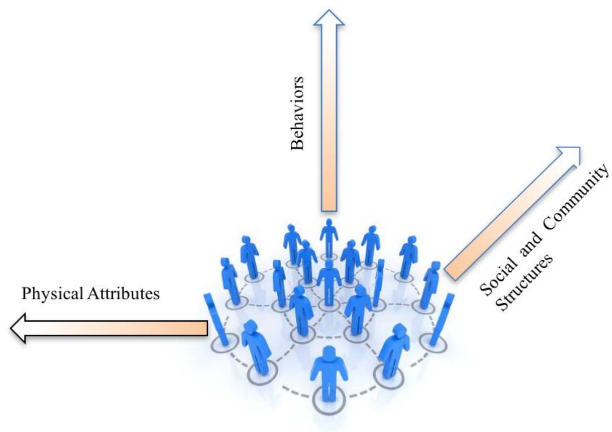

Fig. 1. Model Framework

I. Introduction

Social media is strongly influencing human social interaction [1] and has been shown to have a considerable impact on user behavior both online and in the real world [2]. Effectively predicting future behaviors of users on online social media platforms has attracted interest from research communities and industry alike. It is an important step in personalized intervention regiments for patients with mental illness, or it can be used by services looking to improve customer satisfaction and retention.

However there should be a distinction between predicting how a user will behave at a specified time and when a user will engage in a behavior again. Return time to an activity is a significant metric towards understanding barriers to personal engagement as it provides insight into the psychological states dictating behavior. It may therefore be more useful to predict not whether a user will engage in a behavior at a particular time but when the user will engage in such a behavior again. We call the problem of predicting how a user will behave at specified time *behavior prediction* and the problem of predicting when a user will perform a behavior again *time-sensitive behavior prediction*.

Time-sensitive behavior prediction becomes a challenging problem as social networks grow more intertwined in our daily lives. Modern social networks integrate information across multiple dimensions (Figure 1) such as social structure, online behavior, real world behavior, etc. All features play important roles in determining how and when people engage in activities. It is therefore necessary to consider how these dimensions interact as the underpinnings of behavioral influence in modern social networks.

The problem of behavior prediction has been well studied, especially within the context of health informatics [3], [4]. Early works in behavior prediction rely on information cascade models [5] which struggle when networks grow and evolve quickly. More recent works introduce deep learning approaches for behavior prediction in high dimensional networks in order to learn implicit social influence features from user attributes [3], but these methods measure pairwise user similarity as a function of attribute vectors, capturing first-order proximity in the network [6]. However, because many legitimate links may be missing from the network, first-order proximity can be a weak notion of user similarity. Higher order measures and embeddings can better capture latent social structures within the network topology that are useful for social network analysis tasks. Furthermore, state-of-the-art human behavior prediction models are not optimized for predicting the return time to a behavior.

Problems similar to time-sensitive behavior prediction have been studied within the context of recommender systems [7] and web service user retention [8]. In these contexts, the major goal is to learn low-rank representations that capture an affinity between a user and web service [7]. Matrix factorization based collaborative filtering methods have had great success on this task but are typically suited for static user-item rankings and do not handle the temporal dynamics of user beh

0-7695-6321-X/17/31.00 ©2017 IEEE
DOI 10.1109/ICMLA.2017.000-4

that end there has been a flurry of recent work in modeling how user interest in online activities evolves over time [9], [10]. However, these works only consider how a user interacts with an online service or a set of online services, effectively treating a user's social context as independent of their online behaviors.

We are interested in how to predict return time to personal behavior by effectively integrating multiple dimensions of data from complex and rich social networks. In this work we present a semi-supervised deep learning model to solve our time-sensitive behavior prediction problem by jointly learning embeddings for two dimensions of a complex social network: users' historical behaviors and their social contexts. We name our model the Context Aware Behavior Embedding Model (CABE). We employ a combination of both recurrent and autoencoder neural network architectures to accomplish this task. Unlike previous works we carefully design an objective function which ensures that our model is able to learn good social embeddings by considering global network structure and user-user interaction dynamics. Furthermore, we are able to learn the nonlinear hidden correlations between social context and user behavior.

We validate our model on a complex health social network which integrates online social behaviors, physical activity records, and physical attributes like BMI and blood pressure measurements. In fact, utilizing mobile sensor technologies, our recent study was conducted in 2010-2011 as a collaboration between several laboratories, Telecom Corporations, and Universities to record daily physical activities, social activities (i.e., text messages, social games, events, competitions, etc.), biomarkers, and biometric measures (i.e., cholesterol, BMI, etc.) for a group of 500 individuals. Physical activities were reported via a mobile device carried by each user. All users enrolled an online social network allowing them to make friends and communicate with each other. Users' biomarkers and biometric measures were recorded via monthly medical tests performed at our laboratories. Specifically, we show that latent community features in the social network structure play a role in behavioral outcomes, and we perform time-sensitive behavior prediction, showing our model outperforms relevant baselines in our health social network.

The rest of the article is organized as follows. In Section 2 we review related work on human behavior prediction and time-sensitive recommendation. Section 3 provides background on semi-supervised embedding techniques. In Section 4 we describe our time-sensitive behavior prediction model. Section 5 gives an experimental comparison between our model and existing approaches, and Section 6 concludes.

II. Related Work

Behavior Prediction. In earlier work on behavior prediction for high-dimensional social networks, Shen et al [4] extend the Gaussian Process Model to capture latent social correlations in social networks. Termed the Socialized Gaussian Process (SGP), their model shows improved performance on physical activity prediction. Recent work by [3] present deep learning based behavior prediction models which are able to learn hidden features for social influence. Amimeur et al. [11] propose a graph embedding method based on user-user interaction data for explicitly capturing social influence and show their model improves accuracy on a behavior prediction task. However these models are not designed to predict when in the future a user will engage in a behavior.

Time Prediction. The first work on time prediction for recommendation was [7], a tensor based extension of previous work on matrix factorization for collaborative filtering. This work was extended in [9] to provide context-aware recommendations while considering temporal dynamics. Context in [9] is defined as user-specific attributes like gender, age, etc and not social context. Recently, Du et al. [10] treat user engagement as a temporal point process. Doing so, however, assumes a particular parameterization of the distribution of behavior events over time. This led to a recent paper by the same group [12] proposing an RNN based temporal point process which does not encode any prior knowledge about the underlying generative process of user-item interaction dynamics. Wang et al. [13] extend [10] to model the coevolution of user latent features and item latent features. The most relevant work to ours, by Dai et al. [14], is an extension of [13] and [12] which proposes a recurrent feature embedding process to learn the correlation between user latent feature evolution and item latent feature evolution for time-sensitive item recommendation. Kapoor et al. [15] fit a hidden markov model (STIC) to the gaps in time between events of interest. Our work improves upon these time prediction methods by considering how social influence plays a role in user behavior.

III. Semi-Supervised Embedding

The modularity of deep neural networks and the flexibility of unsupervised objectives make semi-supervised deep embedding a natural framework for modeling the various components of social networks and their highly nonlinear dependencies. Semi-supervised embedding via deep learning [16] is a set of latent feature learning techniques which take advantage of a small amount of labeled in a large dataset under the premise of the smoothness assumption, that data points near one another likely share a label. Most methods optimize two objectives, a supervised objective and an unsupervised regularization. There are a number of ways to introduce an unsupervised regularization constraint into a deep neural network, and below, the two most common approaches are described.

Transductive semi-supervised learning is a classic formulation that cannot generalize to unobserved data instances, though for the purposes of our social network application that is irrelevant. The goal in this formulation is to regularize the supervised loss function at the output layer.

The loss function for transductive learning can be defined as follows.

$$-\frac{1}{N}\sum_{i}^{N} l(f(x_i, y_i) + \sum_{i}^{N+U} L(f(x_i), f(x_j), W_{ij})) \qquad (1)$$

The first term of the loss function of the class label prediction, the supervised task, with x_i the input to the model and y_i the instance label. $f(x_i$ is the model output for input x_i with $L(f(x_i), f(x_j), W_{ij})$ being an unsupervised loss function. The second interesting mode for semi-supervised deep learning is one which regularizes one or more individual hidden layers within the overall model architecture.

This sort of mode would have the following loss function.

$$-\frac{1}{N}\sum_{i}^{N} l(f(x_i,y_i) + \sum_{i}^{N+U} L(h^k(x_i),h^k(x_j),W_{ij})) \qquad (2)$$

Here $h^k(x_i)$ denotes the output of the k-th layer of a DNN with input x_i.

The formulations above fit quite nicely with problems like behavior prediction, and its variants, given increasingly complex and high dimensional networked datasets. One can train a model to perform a behavior prediction (supervised) task using features from a variety of social and behavioral dimensions while preserving important properties of those features through unsupervised regularizations.

IV. Context Aware Behavior Embedding Model

A. Overview

In this section we present our semi-supervised model for jointly learning embeddings for users in a social network $\mathcal{G} = (U,E,B)$, where U is the set of all users in the network, E the set of all friend connections, and B the set of behaviors of all users over all time. The model, shown in Figure 2, consists of two modules, an autoencoder tasked with reconstructing the weighted adjacency matrix of $\mathcal{G}$ and a recurrent neural network tasked with predicting behavior return time.

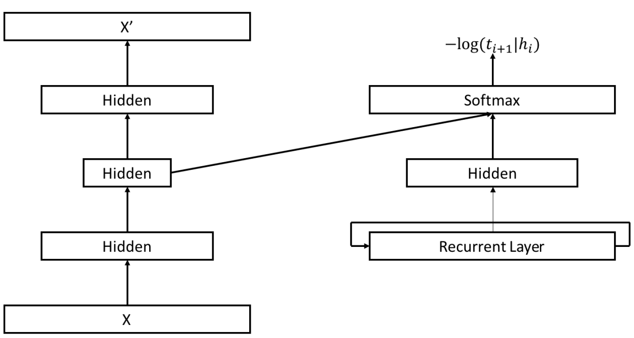

Fig. 2. Model Framework

B. Historical Behavior Embedding

We begin our model discussion by detailing our method for embedding user specific historical behaviors. Recurrent neural networks (RNN) offer a nice framework to capture historical dependencies between sequence events. We employ a particular variant of the RNN, the LSTM, which handles the vanishing/exploding gradient problem associated with vanilla RNNs.

An LSTM unit contains an input gate, a memory cell, an output gate, and a forget gate. The input gate controls how the input to the unit affects the state of the memory cell while the output gate controls how the cell state of the unit affects other LSTM down the chain. The forget gate controls the self-recurrent connection of the memory cell. These behaviors are summarized by the gate and cell activation equations below. $g^i_{t,j}$ denotes the output of the input gate, $g^f_{t,j}$ the output of the forget gate, C_t the cell state at time t, $o_{t,j}$ the output of the output gate, and h_t the new hidden state of the LSTM unit. W_j and U_j are weight matrices and b_j is a bias.

$$g^i_{t,j} = a(W^i_j t_i, U^i_j h_{t-1}, b^i_j) \qquad (3)$$

$$g^f_{t,j} = a(W^f_j t_i, U^f_j h_{t-1}, b^f_j) \qquad (4)$$

$$C_t = g^i_{t,j} \times \tilde{C}_t + g^f_{t,j} \times C_{t-1} \qquad (5)$$

$$o_{t,j} = a(W^o_j t_i, U^o_j h_{t-1}, b^f_o) \qquad (6)$$

$$h_t = o_{t,j}(tanh(C_t)) \qquad (7)$$

For a given event $(t_i,u_i) \in B$ with timing vector t_i corresponding to a behavior by user u_i, the model aims to learn an embedding h_i unique to u_i given a past embedding h_{i-1} and t_i. The timing vector contains all timing features associated with event (t_i,u_i). Features may for example include the day, month, time of day, an indicator for weekday or weekend, inter-event duration, etc. Following the LSTM gate formulations above, we define our model as follows.

$$g^i_{t,j} = \sigma(W^i_j t_i + U^i_j h_{t-1} + b^i_j) \qquad (8)$$

$$g^f_{t,j} = \sigma(W^f_j t_i + U^f_j h_{t-1} + b^f_j) \qquad (9)$$

$$C_t = g^i_{t,j} \times \tilde{C}_t + g^f_{t,j} \times C_{t-1} \qquad (10)$$

$$o_{t,j} = \sigma(W^o_j t_i + U^o_j h_{t-1} + b^f_o) \qquad (11)$$

$$h_b = o_{t,j}(tanh(C_t)) \qquad (12)$$

We tend to unroll our recurrent network for only short sequences for more efficient optimization given that behavioral sequences typically favor short term dependencies.

C. Deep Autoencoder

We now discuss our method for embedding community and social influences in social networks. Deep autoencoders have been effectively applied towards a number of various tasks from object recognition and image segmentation [17] to word embedding and machine translation [18].

A stacked autoencoder is an unsupervised deep neural network configuration which learns a parameterization of a composite of nonlinear transformations in order to embed and reconstruct a feature vector. Stacked autoencoders are particularly interesting because they tend to capture a part-whole grouping of input features, a property which is nice when trying to learn hidden social and community structures. Network embeddings can be learned by treating the adjacency list of a node as the input x_i to an autoencoder. However, it is unusual for all relationships within a network to be equally influential. As noted in [11], it is beneficial to consider user-user interactions when learning network embeddings, so we choose to weigh the adjacency list according to the frequency of interactions between friends in the network.

The loss function of our autoencoder is a distance between the weighted adjacency list x_i and its reconstruction $\hat{x}_i$. We choose euclidean error as our distance function resulting in the following loss.

$$\mathcal{L}_1 = \sum_{j=1}^{|x_i|} \left\| \hat{\mathbf{x}}_{\mathbf{ij}} - \mathbf{x}_{\mathbf{ij}} \right\|_2 \tag{13}$$

D. Model Objective

Consider the unsupervised regularization task of our model. Our objective function for learning social context embeddings is shown below.

$$\mathcal{L}_1 = \sum_{i=1}^{n} \|\hat{\mathbf{x}}_{\mathbf{i}} - \mathbf{x}_{\mathbf{i}}\|^2 \tag{14}$$

Our second task of course is to predict the return time of a user to a behavior. We treat this as an epoch based classification problem where we predict the next epoch of time during which a behavior may be performed. This offers a bit of flexibility with respect to the granularity of our time prediction. Du et al. [12] offer a way of predicting the exact time, although in most applications an exact time prediction is unnecessary. Nevertheless, it is easy to use our hidden features in their conditional intensity formulation for an exact time prediction. Our softmax output layer predicts the probability of a behavioral event occurring during an epoch of time given the joint hidden representation of the social context and historical behavior of a user.

$$y_i = \frac{e^{s_i}}{\sum_{c \in Epoch} e^{s_c}} \tag{15}$$

where

$$s_i = \sigma(W_s h_s + W_b h_b + b_s) \tag{16}$$

Here, h_s holds the social context embeddings and h_b the behavioral embeddings corresponding to input i. $Epoch$ corresponds to the set of dates from which the model predicts the next activity will occur. The appropriate loss function for the prediction task is the cross-entropy error shown below.

$$\mathcal{L}_2 = -\sum_i \hat{y}_i (y_i - log \sum_j e^{y_i}) \tag{17}$$

This brings us to our final objective which is to jointly minimize the semi-supervised loss function shown below.

$$\mathcal{L} = -\sum_i \hat{y}_i (y_i - log \sum_j e^{y_i}) - \alpha \sum_{i=1}^{n} \|\hat{\mathbf{x}}_{\mathbf{i}} - \mathbf{x}_{\mathbf{i}}\|^2 \tag{18}$$

$\hat{y}_i$ is the label of the next epoch during which activity occurs and y_i the softmax output.

The historical behavioral embeddings and social context embeddings can be learned via backpropagation through time and regular backpropagation of the error gradient, respectively.

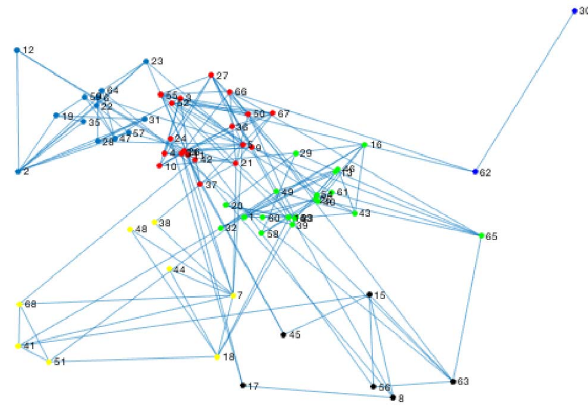

Fig. 3. Visualization of clustered context embeddings

V. Experiments

A. Social and Physical Activity Dataset

Our dataset was collected via a study which aimed to record the effects of online social behavior with real world physical activity in order to learn how best to stage digital interventions to promote favorable health outcomes. The study was a collaboration among several health laboratories and universities to help people maintain active lifestyles and lose weight. The dataset collected records from 254 users stratified into four dimensions: personal information, social network activities, biometric and biomarker measurements, and their daily physical activities over ten months. The entire dataset can be considered an extremely rich social network, capturing a number of dimensions of varying complexity. Overall, it is a interface between online social interaction and real world behavior and physical attributes, making for a unique dataset from which actionable insights can be mined.

The study created a private health social network serving as a platform to encourage study participants to engage with one another with regard to their physical performance. Forms of social interaction include peer-to-peer private messages, public board postings, and online competitions through which subsets of users could share physical activity performance with one another.

The initial physical activity data, collected from each user via special wearable electronic equipment, records information such as the number of walking and running steps. Each entry corresponding to an activity event has a date and start time as well as a distance parameter and a speed parameter which we can use to define a threshold for intense physical activity to more effectively filter low effort activity that could be considered noise in our context. After preprocessing, there are approximately 50,000 activity events across 120 users over the nine month duration of the study. Note that, given a week if a user does exercise more than the previous week, he/she is considered increasing exercise in that week. Otherwise, the user will be considered decreasing exercise.

B. Community Effects on Physical Activity

One the driving claims in this work is that hidden community and social structures play a significant role in behavioral outcomes of users in social networks, be they real world behaviors or online behaviors. It is therefore useful to investigate how well our model can capture latent social structures from networked data and explore how users within communities behave over time. We do this by clustering users using the set of social embeddings learned by our model and looking at variational patterns of return time to physical activity as users interact with their community and global network members.

Similar to spectral clustering based methodologies, we can apply standard clustering algorithms to detect communities from high order social embeddings. In this application, the number of communities is not known a priori so DBSCAN is chosen to perform community detection. Figure 3 shows the color-coded clustering results and network embeddings in two dimensions. Euclidean distance is the measure of social similarity.

The thirty-eight week study is broken down into windows of three weeks. For each cluster, we calculate the standard deviation of the users' average return time to moderate or strenuous physical activity (in days) within every three week window of the study. We define moderate physical activity as at least a light jog, active at a pace four miles per hour or greater. The results are shown in Figure 4. In general, among users within a cluster, variance among return times to physical activity is much lower compared to the control group consisting of users who were physically active but were not active in the social network, i.e. they did not send a message to anyone. Furthermore, there is a slight but noticeable trend among active users. The standard deviation within clusters trended downwards throughout the study. This suggests that those who interact within the network frequently tend to behave more similarly over time and that there is a correlation between behaviors within a social context, showing that it is important to account for community dynamics when predicting how and when social network users will behave.

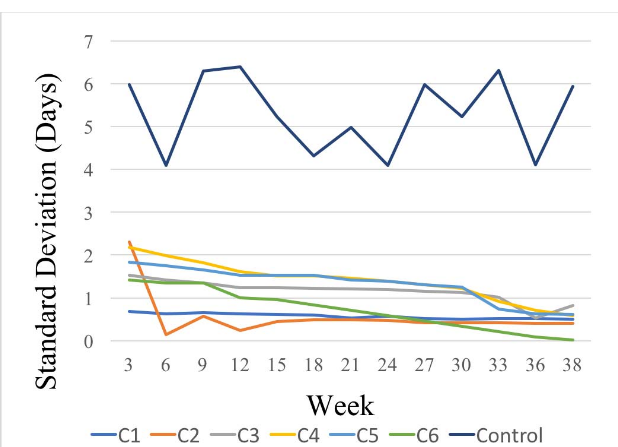

Fig. 4. Community effects on behavior homogeneity over time

C. Time Sensitive Behavior Prediction

Problem Definition. Given the social network $\mathcal{G}$ and individuals' past behaviors until day t, $\mathcal{X}_{1..t} = (X^1_{1..t}, X^2_{1..t}, ..., X^N_{1..t})$, where $X^i_{1..t} = (x^i_1, x^i_2, ...x^i_t)$ with x^i_t being the timestamp of a physical activity event. N is the number of users in the social network. The time-sensitive behavior prediction problem is to predict the time of the individual's next behavior x^i_{t+1}, i.e., $\mathcal{X}_{t+1}$.

Dataset and Experiment Configurations. We only consider users who interact with others in the network (i.e., users must send (resp., receive) messages to (resp., from) other users). We have 123 users with 2,766 inbox messages for our experiments.

We consider only users who interact in the network because they are most likely to influence users or be influenced by other users to be physically active. This also gives us the ability to weigh the transition matrix serving as input to our model by incorporating the frequency of interactions between users and to include additional temporal information into our model.

To validate our time prediction model, we predict the next epoch during which a user will be physically active. This, for example, may be the next date within a window of time that physical activity takes place. In total, the dataset spans thirty-eight weeks of activity. We choose to predict the next date users will be physically active within a three week window. We choose this window size because all users that are socially active in the network perform moderate physical activity at least once every three weeks. We unroll the recurrent layer of the network by three steps. That is, we consider the duration between the last three physical activity events to predict the time of the next physical activity event.

Competitive Prediction Models. We compare our behavior embeddings with the embeddings learned with the Recurrent Marked Temporal Point Process Model which we consider state of the art for deep learning based time sensitive prediction models. The RMTPP model learns embeddings of historical behavior via a vanilla recurrent neural network which are used to parameterize a point process for time prediction. In our health social network application, there is no marker input to the RMTPP so we elect to train without that portion of the model.

Our second competitive baseline is the STIC model [15] which aims to model two psychological latent states with respect to behavioral streams, bored and engaged. The model fits a hidden semi-Markov model for the gaps between user behavioral events.

Experimental Results. We report the performance of our model (CABE) versus the RMTPP and STIC models. We show results for our model in three different modes corresponding to different values of the parameter λ controlling the weight of the reconstruction regularization versus other components in the model's loss function. Values of 0.5, 1, 2, and 5 are chosen to demonstrate model performance given relative importance of social context embedding.

Results are shown for eight separate windows of three weeks of physical activity. Accuracy is used as the evaluation metric. For each window beginning at week N all weeks of activity are leveraged for prediction between weeks N through $N+2$. That

TABLE I

PREDICTION ACCURACY COMPARISON WITH DIFFERENT MODELS(T15-T38)

Weeks	STIC	RMTPP	CABE $\lambda = 0.5$	CABE $\lambda = 1$	CABE $\lambda = 2$	CABE $\lambda = 5$
T15-T17	0.1243	**0.1534**	0.1491	0.1521	0.1529	0.1518
T18-T20	0.1256	0.1492	0.1523	**0.1579**	0.1567	0.1571
T21-T23	0.1275	0.1559	0.1578	0.1634	**0.1669**	0.1653
T24-T26	0.1348	0.1553	0.1683	0.1779	**0.1794**	0.1784
T27-T29	0.1362	0.1586	0.1648	**0.1752**	0.1735	0.1742
T30-T32	0.1378	0.1685	0.1723	0.1851	**0.1896**	0.1782
T33-T35	0.1457	0.1883	0.1837	01848	**0.1887**	0.1835
T36-T38	0.1472	0.1861	0.1855	0.1905	**0.1926**	0.1892

is, the softmax output of our model produces a distribution across the twenty one days within each window, predicting the next date during which the user will perform physical activity. The results in Table 1 show that our model with values of 1 and 2 for λ outperform both the RMTPP and STIC models during every time window while accuracy suffers with too low and too high of values for λ. It is important to consider the results of our community effects experiment above together with the time prediction experiment in order to see why our model outperforms the RMTPP and STIC baselines. The subtle correlations between social context and frequency of physical activity can be learned by our model and are therefore useful for time sensitive behavior prediction.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a novel time-sensitive behavior prediction model which can integrate multiple dimensions of complex social networks via a semi-supervised deep learning framework in order to perform the time prediction task. An experiment conducted on a health social network showed the effects of latent community structures on the frequency of behavior of users in social networks, and we showed our model outperforms relevant baselines on a time prediction task. In future works, we can extend our model to capture coevolving historical behaviors more explicitly while still remaining aware of social context. We also plan on exploring how to effectively incorporate physical attributes, like height, weight, and blood pressure, into our problem, for example to predict when a user may reach a particular weight goal.

ACKNOWLEDGMENT

This work is supported by the NIH grant R01GM103309 to the SMASH project. We thank Xiao Xiao and Rebeca Sacks for their contributions.

REFERENCES

[1] D. Centola, "The spread of behavior in an online social network experiment," *science*, vol. 329, no. 5996, pp. 1194–1197, 2010.

[2] N. A. Christakis and J. H. Fowler, "The spread of obesity in a large social network over 32 years," *New England journal of medicine*, vol. 357, no. 4, pp. 370–379, 2007.

[3] N. Phan, D. Dou, H. Wang, D. Kil, and B. Piniewski, "Ontology-based deep learning for human behavior prediction in health social networks," in *Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics*. ACM, 2015, pp. 433–442.

[4] Y. Shen, N. Phan, X. Xiao, R. Jin, J. Sun, B. Piniewski, D. Kil, and D. Dou, "Dynamic socialized gaussian process models for human behavior prediction in a health social network," *Knowledge and information systems*, vol. 49, no. 2, pp. 455–479, 2016.

[5] K. Saito, R. Nakano, and M. Kimura, "Prediction of information diffusion probabilities for independent cascade model," in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, 2008, pp. 67–75.

[6] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: large-scale information network embedding," *CoRR*, vol. abs/1503.03578, 2015. [Online]. Available: http://arxiv.org/abs/1503.03578

[7] Y. Koren, "Collaborative filtering with temporal dynamics," *Communications of the ACM*, vol. 53, no. 4, pp. 89–97, 2010.

[8] K. Kapoor, M. Sun, J. Srivastava, and T. Ye, "A hazard based approach to user return time prediction," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 1719–1728.

[9] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering," in *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010, pp. 79–86.

[10] N. Du, Y. Wang, N. He, J. Sun, and L. Song, "Time-sensitive recommendation from recurrent user activities," in *Advances in Neural Information Processing Systems*, 2015, pp. 3492–3500.

[11] A. Amimeur, N. Phan, D. Dou, D. Kil, and B. Piniewski, "Interaction network representations for human behavior prediction," in *Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on*. IEEE, 2016, pp. 87–93.

[12] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, "Recurrent marked temporal point processes: Embedding event history to vector," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1555–1564.

[13] Y. Wang, N. Du, R. Trivedi, and L. Song, "Coevolutionary latent feature processes for continuous-time user-item interactions," in *Advances in Neural Information Processing Systems*, 2016, pp. 4547–4555.

[14] H. Dai, Y. Wang, R. Trivedi, and L. Song, "Recurrent coevolutionary feature embedding processes for recommendation," *CoRR*, vol. abs/1609.03675, 2016. [Online]. Available: http://arxiv.org/abs/1609.03675

[15] K. Kapoor, K. Subbian, J. Srivastava, and P. Schrater, "Just in time recommendations: Modeling the dynamics of boredom in activity streams," in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. ACM, 2015, pp. 233–242.

[16] J. Weston, F. Ratle, H. Mobahi, and R. Collobert, "Deep learning via semi-supervised embedding," in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 639–655.

[17] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[18] S. C. AP, S. Lauly, H. Larochelle, M. Khapra, B. Ravindran, V. C. Raykar, and A. Saha, "An autoencoder approach to learning bilingual word representations," in *Advances in Neural Information Processing Systems*, 2014, pp. 1853–1861.