2018 IEEE 5th International Conference on Data Science and Advanced Analytics

Recommendation of Points-of-Interest using Graph Embeddings

Giannis Christoforidis
Department of Informatics
Aristotle University
Thessaloniki, Greece
icchrist@csd.auth.gr

Pavlos Kefalas
Department of Informatics
Aristotle University
Thessaloniki, Greece
kefalasp@csd.auth.gr

Apostolos N. Papadopoulos
Department of Informatics
Aristotle University
Thessaloniki, Greece
papadopo@csd.auth.gr

Yannis Manolopoulos
Department of Informatics
Aristotle University
Thessaloniki, Greece
manolopo@csd.auth.gr

***Abstract*—The rapid growth of Location-based Social Networks (LBSNs) has lead to the generation of massive datasets which are collected in an exponential rate. The collected information may be used to facilitate users' needs with recommendations related to their past preferences. Many recommendation models were introduced in the literature, which learn by the history of users and provide recommendations for Points-of-Interest. Unfortunately, most of them ignore the relation existing among the temporal properties, the spatial attributes and the periodicity of the check-ins. In this work, we present a novel methodology, named $JLGE$,that combines all aforementioned factors into one unified approach which facilitates POI recommendations. In particular, the model jointly learns the embeddings of six informational graphs i.e., two unipartite (user-user and POI-POI) and four bipartite (user-location, user-time, location-user, and location-time) into the same latent space and personalize the recommendations based on these embeddings. We have experimentally evaluated the accuracy of our model using two real-world datasets in terms of the top-n POIs recommendations. The performance evaluation results indicate a significant improvement in accuracy, in comparison to another state-of-the-art graph-based approach.**

***Index Terms*—POI recommendation, second-order proximity, graph embeddings**

I. Introduction

The presence of Online Social Networks (OSNs) offers to users the opportunity to share their content and experiences with their friends. Within some time these networks incorporated geographical information to users content and triggered new functionalities for Location Based Social Networks (LBSNs). This way, they merged the digital with the physical layers and gave a clear view of how users behave on social networks while posting messages, photos, videos, or any other type of information related to a *Point of Interest* (POI). It is evident that these actions have set new goals for both industry and research. In this context, many LBSN providers aim to use all collected information towards supporting registered users with several types of *recommendations* towards: making new friends, buying new items or even attending new events. Those systems initiated a significant research direction involving the generation of numerous novel models aiming to personalize the information retrieved.

There is a plethora of different types of recommendations that could be used in this setting. In this work, we focus on techniques that provide *recommendations for Points-of-Interest* (POIs). However, most of the existing approaches [1]–[4] lack a comprehensive way of personalizing their recommendations since they ignore one or more of the following factors: i) the social influence, ii) the check-in periodicity, iii) the geographical proximity of the POIs, and iv) the preference dynamics. In particular, some models treat locations as regular nodes ignoring the spatial proximity between two locations [5]–[9]. Thus, if a user performs a check-in at a restaurant in *London*, she may be correlated with POIs located at a very long distance, as result of ignoring geographical proximity. On the contrary, other models that consider only geographical influence miss the periodicity of users' behavior [1], [3]. As an example, imagine another user who performs the majority of her check-ins during weekends in a specific region of *London*. In such a case, some models that ignore the temporal dimension may support that user with POI recommendations during week days.

Moreover, models that explore both spatial and temporal dimensions ignore users' preference dynamics which constantly evolve [10], [11]. Recent research [12]–[14], that focused on human behavior, indicates that users change their preferences during specific time periods. For example, one user may attend *Chelsea* on September since there are venues in that region while on October most of the check-ins are in the *Chelsfield* region, due to available discounts at the malls. This evolution indicates a dynamic behavior that should be considered during computation [15].

Motivated by the lack of a holistic approach that takes into account all of the aforementioned factors in a unified way, in this work we present a new model that provides POI recommendations. In particular, the model named *Jointly Learn the Graph Embeddings (JLGE)*, consists of a three step process that explores all these informational networks collectively. In the first step, it builds information networks describing the users' and the POI relations and then it weights the edges between the participating nodes of these graphs. In the second step, it uses these six informational graphs i.e., two unipartite (user-user and POI-POI) and four bipartite (user-location, user-time, location-user, and location-time) to jointly learn the embeddings of the users and the POIs into the same latent space. Finally, in the third step, the model personalizes the

978-1-5386-5090-5/18/$31.00 ©2018 IEEE
DOI 10.1109/DSAA.2018.00013

POI recommendations for each user dynamically by tuning the influence of the participation networks to the final suggestions for the target user.

The main idea behind the proposed approach is based on the following concepts:

- **Social influence:** Users are highly effected by their social network, thus, they tend to follow their friends not only on OSN but also in real life [12], [15], [16]. For example, it is quite possible for a user to perform a check-in in an unvisited location where her friends used to check-in in the past.
- **Geographical proximity influence:** Earlier studies [12], [14] indicate that people prefer to visit proximate POIs rather than the ones located far away. Hence, POIs that are close to the target user are more likely to be recommended than rest. According to this assumption we linked the proximate locations within a distance of 2.5 Km to create the POI-POI graph.
- **Periodicity influence:** Users' check-in behavior is periodical and highly affects the overall accuracy of the recommendations [12], [17], [18]. For example, users tend to check-in different POIs during weekends in contrast to weekdays. Thus, temporality is a crucial factor which we encountered into our model through User-Time period and POI-Time period graphs.
- **Preference dynamics:** Along with periodicity, users' preferences evolve as well dynamically [12], [13], [19]. For example, POIs visited multiple times indicate a user's preference tendency against other locations. Notice that this preference alters between time periods pointing the change of the user's taste over locations. In our model, we incorporate this preference dynamics though User-POI and POI-User graphs.

The most important contributions of our work as summarized below:

- We use the check-in history to build a heterogeneous graph of six information networks.
- Also, we present a novel model that learns the embeddings of the spatial, the temporal, and the social influences along with users' preference dynamics into the same latent space.
- The model use a novel probabilistic weighting strategy to further boost the accuracy of the predictions.
- We use a tuning parameter to balance the influence of the learned users' and POIs embeddings during recommendations.
- We evaluate the performance of our model in terms of accuracy for the top@n recommendations. Regarding this task, the experiments indicate that as we embed more dimensions into the same latent space, the personalization fits better to users' behavior since the overall performance is boosted. Also, our model outperforms the state-of-the-art models presented by [10].

The rest of the paper is organized as follows: In Section II we summarize related work, whereas Section III studies the proposed methodology in detail. Performance evaluation results based on real-world datasets are offered in Section IV. Finally, Section V concludes our work and discusses briefly future research directions.

II. Related Work

In this section we discuss the recent research conducted related to POI recommendation models. More, specifically we focus on related work that use one or more for the following factors: i) social, ii) geographical, iii) temporal, to analyze users' behavior and provide recommendations. In the following paragraphs we present the most important papers along with their weaknesses.

Recent studies indicate that users tend to be influenced by their friends and visit common locations [1], [20]. This is known in the literature as *social influence* and its quite important in collaborative filtering approaches were other users with similar behavior are used to provide recommendations to a target user. In this direction, Li et al. [2] distinguish three types of friendships, that are i) linked, ii) co-located, and iii) proximate friends and exploit their check-in records through a unified framework. First, this model learns the common POIs that the target user and all three types of friends check-in the past. Then, it uses matrix factorization models that minimize two loss functions over the learned POIs to personalize the recommendations. Unfortunately, this model ignores the periodical behavior and the preference dynamics.

Additionally, related work points that users are willing to go to proximate locations rather than to locations in long distance [3]–[6]. Thus, supporting user with POI recommendations in long distance is highly possible not to be attended. Thus, it is evident that spatial proximity is crucial during recommendations and it is widely known as *geographical influence*. Moreover, the influence is highly affected by the sequence of the check-ins performed. In this direction, Wang et al. [21] focused on the importance of sequential influence of spatial items, and proposed a system named SPORE. User preferences are fused with the sequential influence of spatial items into the same latent space over a probabilistic topic-region unified model. Extending previous research related to spatial items, they introduced another probabilistic model named LSARS [22] that jointly correlates the geographical influence, item attributes, and users' reviews. Both models support the claim that users are willing to interact with proximate items, thus they use geographical influence jointly with additional factors. Unfortunately, these approaches ignore the temporal evolution.

To overcome this issue some state-of-the-art methods [10]–[12] explore the temporal influence of the check-in history of users. These methods are described in the literature as *hybrid* because they combine multiple models to overcome drawbacks each one faces individually. In this direction, Kefalas et al. [12] proposed a novel hybrid model that combines the CF and CB in a unified way, exploring: i) the proximate users' preferences, ii) the textual influence alternation within time periods, and iii) the preference dynamics evolution. Results indicate an evolution of users' check-in behavior, since they are highly influenced by all factors combined. The results pointed an incremental robustness of the precision against models considering each factor separately. Similarly, Baral et al. [11] proposed a multi aspect personalized model (MAPS) for POI recommendation which also explores the category of the attended locations during particular times of the day. Thus, the category of the visited POIs in the past history at specific times of the day is crucial along with all other factors. This way the model refines the recommendations considering all four dimensions. In the same context, Xie et al. [10] with a model named GE that jointly learns POI embeddings by capturing i) the sequential effect, ii) the spatial influence, iii) the periodicity, and iv) the semantics into the same latent space. Unfortunately, these approaches ignore the preference dynamics.

In contrast to all aforementioned studies, we develop a novel unified model for POI recommendations that explores all factors described previously. In particular, it jointly learns the graph embeddings of users and POIs into the same latent space using LINE model instead of Metapath2vec [23]. The reason of choosing one over the other is that the later does not consider the relations between unipartite graphs and focuses only on bipartite ones. Our methodology consist of three steps. First, it builds information networks describing the users' and the POI relations and then applies a weighting strategy on them. Then, it uses these networks to jointly learn the embeddings of the users' and the POIs into the same latent space. Finally, it personalizes the POI recommendations dynamically by tuning the influence parameter that controls the network contribution to the final recommendations.

III. Methodology

In this section, we present our methodology which consists of three main steps. In the first step, we define the weighting strategy applied to prepare the graphs required. The second step focuses on learning the six graph embeddings. Finally, in the third step, we provide personalized POI recommendations based on the learned networks for each user separately. Table I presents the most frequently used symbols.

In Figure 1, we present a toy example of all the information graphs used by the proposed approach. In particular, the figure shows the three main layers comprising the unipartite and the bipartite relations between users, POIs and the respective time periods.

A. Weighing Strategy

Given the set of users of a social network and their check-in history we weigh the edges between the two nodes as follows:

Definition 1: **(User-User graph):** Given the social ties of the users we set $\mathcal{G}_{uu} = (U \cup V, \mathcal{W}_{uv})$ to be the graph presenting these relations. In this graph, U and V are the sets of users, and $\mathcal{W}_{uv}$ is the weight among them. The weights are computed as:

TABLE I
FREQUENTLY USED SYMBOLS.

Symbol	Description
U, L, T, C	Set of users $U = \{u_1, ..., u_n\}$, Set of locations $L = \{l_1, ..., l_m\}$, Set of Time intervals $T = \{t_1, ..., t_k\}$, Set of Check-ins $C = \{c_1, ..., c_o\}$
c_{u_i}	User's check-in
ΔT	Time interval
S, N	Number of samples, Number of negative samples
$e_{i,j}$	Edge between two nodes
E	Set of edges $e_{i,j}$ over each graph
$w_{i,j}$	Weighted edge $e_{i,j}$
W	Set of weights $w_{i,j}$ over each graph
$\mathcal{S}$	Sample an edge $e_{i,j}$ from W

$$\mathcal{W}_{uv} = \frac{1}{\sum_{i=1}^{n} |v_i|}$$

Definition 2: **(User-POI graph):** Represents the check-in activity of a user into multiple locations. We represent this graph as $\mathcal{G}_{ul} = (U \cup L, \mathcal{W}_{ul})$, where U is the set of users, L is the set of locations, and $\mathcal{W}_{ul}$ is the set of the weighted edges among them. The weights are computed as:

$$\mathcal{W}_{ul} = \frac{\sum_{\forall c_{u_i} \in l_i} c_{u_{i,l}}}{\sum_{\forall c_{u_i} \in L} c_{u_{i,L}}}$$

Definition 3: **(User-Time period graph):** Represents the temporal activity of users during different time periods. In particular, we denote this relation as $\mathcal{G}_{uu} = (U \cup T, \mathcal{W}_{ut})$, where U is the set of users, T is the set of time periods, and $\mathcal{W}_{ut}$ is the set of the weighted edges among them. The weights are computed as:

$$\mathcal{W}_{ut} = \frac{\sum_{\forall c_{u_i} \in t_i} c_{u_{i,t}}}{\sum_{\forall c_{u_i} \in T} c_{u_{i,T}}}$$

Definition 4: **(POI-POI graph):** The spatial relation between the locations is defined as $\mathcal{G}_{ll'} = (L \cup L, \mathcal{W}_{ll'})$, where L is the set of locations, and $\mathcal{W}_{ll'}$ is the set of the weighted edges between location nodes which point the geographical proximity between them. The weights are computed as:

$$\mathcal{W}_{ll'} = 1 - \frac{(geodist_{l,l'})}{Rg}$$

where $geodist_{l,l'}$ computes the geographical distance between two locations. Our model uses only the locations that are in close distance rather than computing the distance between all pairs. This way, it reduces the time and computation

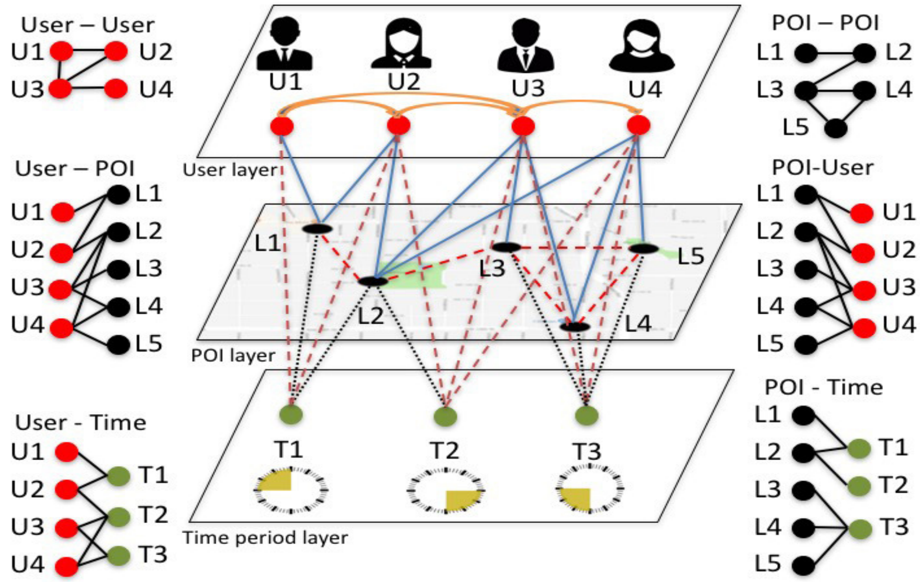


Fig. 1. A simple example of the unipartite and bipartite graphs described in the definitions.

complexity. In Algorithm 1, we present the optimized solution to extract the proximate locations for a target location. In particular, given the distance range, the target location, and the list of all locations in tuples of longitude and latitude, first we sort the list of locations in line 1. Then, starting from the target location we initialize the range variable in line 3 and for each location (line 2) in the sorted list we compute the distance between each pair (l, l') in line 5. If the range is within the given radius Rg (line 4), then we store this location in the $geodist_{l,l'}$ list.

Algorithm 1: OPTIMIZED SELECTION AMONG THE POI NODES

Input: Range Distance: Rg, Target location: l,
List of all locations L in tuples: $\langle long, lat\rangle$
Output: $geodist_{l,l'}$

```
1 S = sort_rows(L, long) // Sort rows of the locations
2 foreach l' ∈ S do      // For each l' in the sorted list
3     range ← 0 // Starting from target l coordinates
4     while range ≤ Rg do // While l' is within distance
5         range ← distance(⟨l_long, l_lat⟩ , ⟨l'_long, l'_lat⟩)
6         geodist_{l,l'}.insert(l, l') // Insert l' in the list
7 return geodist_{l,l'}
```

Definition 5: **(POI-User graph):** Similar to definition 2 this graph defines the attenuation of the users over different locations. We denote this graph as $\mathcal{G}_{lu} = (U \cup L, \mathcal{W}_{lu})$, where L is the set of locations, and U is the set of users, and $\mathcal{W}_{lu}$ is the set of the weighted edges among them. The weights are computed as:

$$\mathcal{W}_{lu} = \frac{\sum\limits_{\forall c_u \in l_i} c_u}{\sum\limits_{\forall c_U \in l_i} c_U}$$

Definition 6: **(POI-Time Period graph):** Refers to the time periods during which each location was attended by users. We represent this relation as $\mathcal{G}_{lt} = (L \cup T, \mathcal{W}_{lt})$, where L is the set of locations, T is the set of time periods, and $\mathcal{W}_{lt}$ is the set of the weighted edges among them. The weights are computed as:

$$\mathcal{W}_{lt} = \frac{\sum\limits_{\forall c_U \in t_i} |n_{lt}|}{\sum\limits_{\forall c_U \in T} |n_{lt}|}$$

Based on the the previous discussion, the problem we investigate is formally stated as follows:

Problem Definition: *"Given a user u at a location l at time t as a query $Q(u, l, t)$, and her/his check-in history, the goal is to predict the top-N unvisited proximate POIs to that target user."*

B. Learn embeddings in a bipartite graph

Nodes that share many connections but they are not directly related with an edge, they belong to the same neighborhood, thus, there are most likely to be similar to each other. For example, in Figure 2, U1 and U2 have checked-in many common locations but there is not friendship connection between them. These nodes consist the ***second-order proximity***, that is the similarity between two unlinked nodes according to their network structure. The second order proximity can be applied to any graph. To extract this kind of proximity on unipartite graphs, the $LINE$ model [24] learns the embeddings of large graphs into a low dimensional space. With this work, we extend this model to learn the embeddings over bipartite graph nodes. Moreover, our model can be applied into all kind of bipartite graphs i.e., directed/undirected, weighted/unweighted, or combinations of them.

The general notion is that given two disjoint sets $\mathcal{G} = (\mathcal{S}_A \cup \mathcal{S}_B, \mathcal{W})$, the nodes in set $\mathcal{S}_A$ that share many connections into set $\mathcal{S}_B$ but are not directly connected with an edge are most likely to have the same distributions. The conditional probability of one node $v_j \in \mathcal{S}_B$ is generated through node $v_i \in \mathcal{S}_A$ such as:

$$p(v_j|v_i) = \frac{exp(\vec{v}_j^T \cdot \vec{v}_i)}{\sum_{u_k \in \mathcal{S}_B} exp(\vec{v}_k^T \cdot \vec{v}_i)} \tag{1}$$

where the embeddings vectors of vertices v_i, and v_j are represented as $\vec{v}_i$ and $\vec{v}_j$, respectively. Thus, for each node $v_i \in \mathcal{S}_A$, Equation. (1) defines the conditional distribution $p(\cdot|u_i)$ to all the corresponding nodes in the set $\mathcal{S}_B$. Then, for each edge there is a weight which implies the strength of this tie. To retain the proximity of the unlinked nodes in $\mathcal{S}_A$, we make the described conditional distribution approximates the empirical distribution $\hat{p}(\cdot|v_i) = \frac{w_{i,j}}{\sum w_{i,k}}$ with the following objective function:

$$O = \sum_{u_i \in \mathcal{S}_A} \lambda_i d(\hat{p}(\cdot|v_i), p(\cdot|v_i)) \tag{2}$$

where $d(\cdot|\cdot)$ denotes the Kullback-Leibler divergence of the conditional and the empirical distributions, and λ_i is a regularization parameter to tune the significance of node v_i. For reasons of simplicity, we set this parameter equal to the out-degree of each node (number of links pointing away from the node). Thus, Equation (2) equals to the minimization of the following objective function:

$$O = -\sum_{e_{i,j} \in W} w_{i,j} \log p(v_j|v_i) \tag{3}$$

The $\{\vec{v}_i\}_{i=1..\mathcal{S}_A}$ and $\{\vec{v}_j\}_{j=1..\mathcal{S}_B}$ vectors that minimize Equation (3) are the low rank representation of each node in $\mathbb{R}^d$.

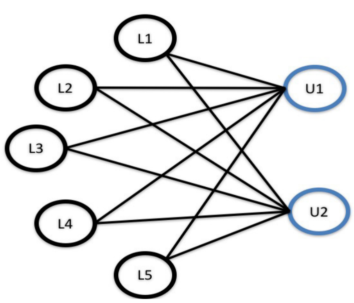


Fig. 2. A toy example illustrating the concept of Second Order Proximity.

C. Model optimization trough negative sampling

To avoid the calculation of the conditional probability $p(\cdot|v_i)$ which needs the summation of the entire set of nodes in Equation (3), we apply negative sampling (NEG) over each edge. In particular, we use the noisy distribution of each edge individually, to sample N negative edges as described in the following objective function:

$$\log \sigma(\vec{v}_j^T \cdot \vec{v}_i) + \sum_{h=1}^{N} W_{u_n \sim P_n(u)}[\log \sigma(-\vec{v}_n^T \cdot \vec{v}_i)] \tag{4}$$

where $\sigma(x) = 1/1 + exp(-x)$ is the sigmoid function with output values in $[0-1]$, and $P_n(v \propto d_u^{3/4})$ is the noise distribution in which the negative samples are chosen with a unigram distribution empirically tuned in [25]. The term of unigram posits that each node occurrence in the set, is independent to all other node occurrences. Thus, selecting a node as a negative sample is related to the out-degree in that power. To further improve the solution of Equation (4), we apply asynchronous stochastic gradient descent (ASGD). In particular, each time an edge $e_{i,j}$ is sampled, we calculate the gradient of node v_i over the corresponding embedding vector $\vec{v}_i$ as follows:

$$\frac{\partial O}{\partial \vec{v}_i} = w_{i,j} \cdot \frac{\partial \log p(v_j|v_i)}{\partial \vec{v}_i} \tag{5}$$

Notice that the gradient of node v_i is multiplied by the weight related to that edge. Thus, tuning the learning rate of the model, may cause problems due to the valiance of the weights. On the one hand, **'overfitting'** may occur to the gradients with large weights, if large learning rate is chosen according to edges multiplied with small weights values. On the other hand, **'underfitting'** may occur to the gradients with small weights, if small learning rate is chosen for edges multiplied with large weights values.

To balance the learning rate of our model, we adopt the sampling method proposed in [24]. In particular, we sample a random edge $e_{i,j} \in [0-W_{sum}]$, with W_{sum} denote the sum of all weights in the particular network, and then we examine the interval in which the particular sampled edge falls into, i.e., $[\sum_{j=0}^{i-1} w_j, \sum_{j=0}^{i} w_j]$. Finally, we draw a sampled edge using alias table which eventually reduces the complexity to $\mathcal{O}(1)$. Table II presents the over all complexity of optimization the edge sampling procedure.

TABLE II
COMPLEXITY ANALYSIS.

Sample edge from alias table	$\mathcal{O}(1)$
Negative sampling optimization	$\mathcal{O}(N+1)$
Overall complexity	$\mathcal{O}(N \cdot E)$

D. Dynamic Joint Embedding Learning in Heterogeneous Graphs

Given the six bipartite graphs describing users' preferences spatially and temporally as presented in Section III-A, the task is to integrate them into our model. On the one hand, the graphs correspond to the users' relations are: the User-User, the User-POI, and the User-Time Period. On the other hand, the graphs correspond to the location ties with other networks: the POI-POI, the POI-User, and the POI-Time period.

We collectively integrate the embeddings of these six graphs corresponding to users' and POIs ties, by minimizing the following objective function:

$$O = \underbrace{O_{uu} + O_{ut} + O_{ul}}_{\text{user networks}} + \underbrace{O_{ll} + O_{lt} + O_{lu}}_{\text{POI networks}} \tag{6}$$

where the above functions are computed as follows:

$$O_{uu} = -\sum_{e_{i,j} \in W_{uu}} w_{i,j} \log p(u_i|v_j) \tag{7}$$

$$O_{ut} = -\sum_{e_{i,j} \in W_{ut}} w_{i,j} \log p(u_i|t_j) \tag{8}$$

$$O_{ul} = -\sum_{e_{i,j} \in W_{ul}} w_{i,j} \log p(u_i|l_j) \tag{9}$$

$$O_{ll} = -\sum_{e_{i,j} \in W_{ll}} w_{i,j} \log p(l_i|l_j) \tag{10}$$

$$O_{lt} = -\sum_{e_{i,j} \in W_{lt}} w_{i,j} \log p(l_i|t_j) \tag{11}$$

$$O_{lu} = -\sum_{e_{i,j} \in W_{lu}} w_{i,j} \log p(l_i|u_j) \tag{12}$$

To minimize the objective function presented in Equation (6), first we merge together edges of all unipartite and bipartite graphs, and then, in each step we update the model by sampling a new edge. The probability of sampling an edge corresponds to the weight related to that edge. This way, our model walks through the heterogeneous bipartite graphs with respect to the inner and the outer vertices of the graphs and the weight influence. The training of all graphs into one unified model performed jointly as shown in Algorithm 2.

E. Recommendations using Joined Learning of Graph Embeddings

By the time all embeddings presented in the previous section are learned, and given a prediction request $Q(u, l, t)$, concerning a user u in a location l at a timestamp t, we project these values to the corresponding time period t, with geographical range distance less than 10 Km. We claim that one user is willing to attend proximate locations. Thus, given a recommendation beyond this range, the probability of attending is reduced significantly. Then, we rank a list with the top@n unvisited candidate POIs for that user in that distance. The prediction score for each of the unvisited locations is computed as follows:

Algorithm 2: DYNAMIC JOIN TRAINING OF THE EMBEDDINGS

Input: $\mathcal{G}_{uu}$,$\mathcal{G}_{ul}$, $\mathcal{G}_{ut}$, $\mathcal{G}_{ll}$, $\mathcal{G}_{lu}$, $\mathcal{G}_{lt}$, S, N
Output: $\vec{\mathbf{u}}$: User embeddings, $\vec{\mathbf{l}}$: POI embeddings, $\vec{\mathbf{t}}$: Time period embeddings

```
Initialize u⃗, l⃗, t⃗;
while (iteration ≤ N) do    // Until the desired number of samples reached
    At each iteration Draw K negative edges of the corresponding graph
    S(w_uu) ∈ W_uu, and update u⃗;
    S(w_ul) ∈ W_ul, and update u⃗ and l⃗;
    S(w_ll) ∈ W_ll, and update l⃗;
    S(w_lu) ∈ W_lu, and update l⃗ and u⃗;
    S(w_ut) ∈ W_ut, and update u⃗ and t⃗;
    S(w_lt) ∈ W_lt, and update l⃗ and t⃗;
return u⃗, l⃗, t⃗
```

$$Q(u, l, t) = \alpha \cdot (\vec{\mathbf{u}}^T \cdot \vec{\mathbf{l}}) + (1 - \alpha) \cdot (\vec{\mathbf{t}}^T \cdot \vec{\mathbf{l}}) \tag{13}$$

where $\vec{\mathbf{u}}$ is the embedding of user u, $\vec{\mathbf{l}}$ is the embedding of location l, and $\vec{\mathbf{t}}$ is the embeddings of the time period t in which the particular check-in was performed. Moreover, our model learns jointly the embeddings from different information networks in the same latent space, thus the learned POI embeddings $\vec{\mathbf{l}}$ captures the information of all participated networks presented in the previous sections, such as uu, ul, ll etc. This way, we aim to eliminate sparsity problem simply by using additional information networks. Finally, α, is a regularization parameter that defines the importance of each information network in our model.

IV. PERFORMANCE EVALUATION

In this section, we present the results of the performance evaluation. The source code we have developed is available at https://github.com/thedx4/JLGE.

A. Datasets

In our experiments, we have used two real-world datasets, Foursquare [10][1] and Weeplaces [11][2]. Their main characteristics are presented in Table III. In particular, both datasets contain users' check-in history with timestamp and geographical information during a time period spanning 5 and 91 months respectively. It is observed that the sparsity of the check-ins in the user-POI matrix is high for both datasets, i.e., 99.97% and 99.95% respectively.

In Figure 3 we present the distribution of the check-ins i) per user and ii) per location for both datasets. It is evident that the check-ins follow a power law distribution according to which, there are few users with many check-ins and many

[1] https://sites.google.com/site/dbhongzhi
[2] http://www.yongliu.org/datasets

TABLE III
DATASETS SPECIFICATIONS

	Users	POIs	Check-ins	Time span
Foursquare	114,508	62,462	1,434,668	Sep 2010 - Jan 2011
Weeplaces	15,799	971,309	7,658,368	Nov 2003 - Jun 2011

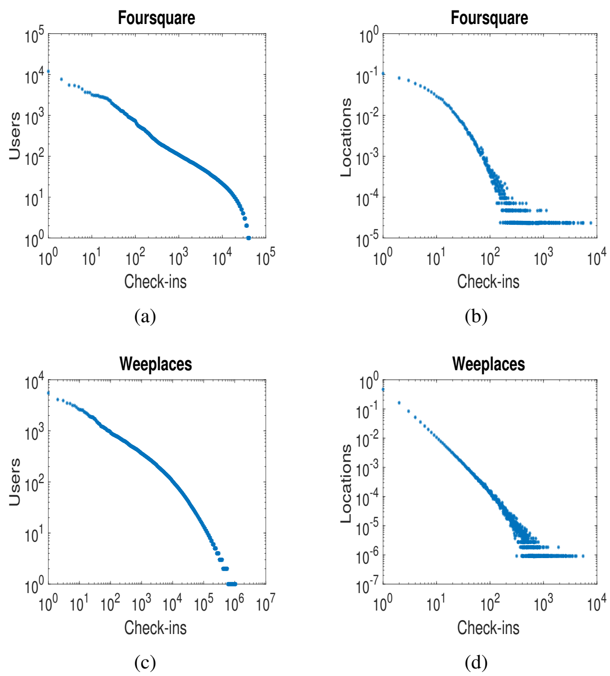


Fig. 3. Power Law distribution diagrams for Foursquare [(a) and (b)], and Weeplaces [(c) and (d)] datasets.

users with small check-in record. Similarly, the same principal stands for locations.

B. Evaluation Process

In this section, we describe the evaluation process followed for the POI recommendation task. We consider the partitioning of the dataset into three subsets:

- the training set $\mathcal{D}_C^T$, which covers the 80% of the total check-ins and is treated as known information,
- the probe set $\mathcal{D}_C^P$, that is 10% and is used for testing our model, and
- (iii) the validation set $\mathcal{D}_C^V$ is the rest 10% for tuning the hyper-parameters.

Therefore, for each target user we generate the recommendations based only on the POIs in $\mathcal{D}_C^T$.

To evaluate the recommended POIs we measure the Accuracy@n as proposed in [21], [26]. In particular, for each $l \in \mathcal{D}_C^P$ given as a query $Q(u, l, t)$, we compute the prediction score for that l along with all unvisited proximate POIs of the target user with Equation (13). We rank the predicted scores into a list and we select the top@n. If the ground truth $l \in \mathcal{D}_C^P$ appear in the top@n, then we have predicted correctly that location, otherwise we have a miss. To compute the overall accuracy of the top@n we average all predictions test cases such as:

$$Accuracy@n = \frac{\#\text{True Positive}@n}{\mathcal{D}_C^P} \tag{14}$$

Due to the fact that some users tend to visit the same locations, makes the evaluation very difficult task since we aim to recommend them new POIs. This problem is even harder if one user tend to check-in the same location/s. In that case, there will be no instances to validate in the probe set $\mathcal{D}_C^P$.

To eliminate this problem, we have applied a simple process of swapping the records between the training and the probe sets until: i) the records in the $\mathcal{D}_C^P \notin \mathcal{D}_C^T$, and ii) we get 10% in the probe set. In particular, for each user we sort her records based on the timestamp and then we split it into these sets. Please notice that we use the later records as probe information since we want to recommend new locations that the user will attend in the future. If one record exists in both sets, then we swap it with the previous record on the training set. Also, if the record history is too small, we split the list with respect to the percentage of the records existing in the probe set.

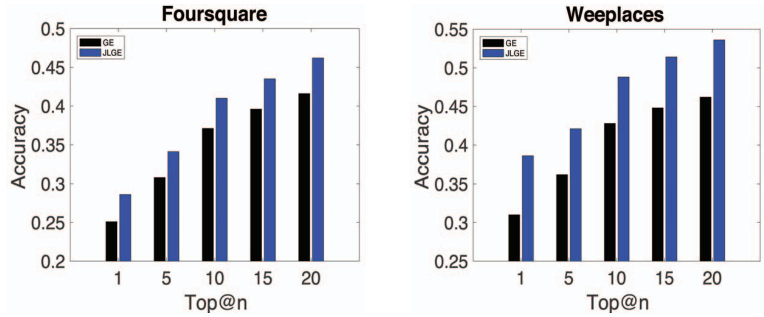
C. Comparison with GE

In this section, we present the results of our method against a state-of-the-art approach named GE [10]. In particular, we examine the overall accuracy for for the top-n predictions ($n = 1, 5, 10, 15, 20$) of both models against the 2 big datasets presented in previous section. To achieve a fair comparison we set the parameters of GE with the higher accuracy values.

The results presented in Figure 4 indicate the benefits of jointly learning the users' and POI embeddings into the same latent space, throughout the following influential factors examined: i) social, ii) geographical proximity, iii) periodicity, and iv) preference dynamics. Thus, the accuracy performance against GE is significant higher. This is due to the fact that GE uses the following three informational graphs that are the POI-POI, the POI-region, and the POI-time (the POI-word network is ignored when there are not topics available). In contrast, our model refines its results by exploring richer information through the six information networks used.

First we use the social influence of user's friends through the User-User graph. Also, we examine the geographical proximity of the locations taking advantage of the connections in POI-POI network. Additionally, our model explores the temporal influence of check-ins using both the User-Time period and POI-Time period graphs. This way, we determine when a user is more active and when a location is more popular during the time span. In addition, the preference dynamics are learned under both a user and a POI point of view. Thus, we examine whether a user likes most to check-in along with the POIs that are more popular. Furthermore, we tune the influence of the learned embeddings to the final recommendations through the tuning parameter α.

Another important difference between the two approaches is that GE builds the POI-POI graph by checking the visits



(a) (b)

Fig. 4. Accuracy against the top-n POI recommendations for (a) Foursquare, and (b) Weeplaces datasets.

of a user during a time period, while we use the equation presented in Definition (4) under the assumption that users are willing to check-in close locations. This way, we explore all proximate locations within a distance while GE explores only the locations visited during a time interval. Thus, for users with few check-ins this graph may contain few nodes which makes learning a difficult task and eventually tackles their accuracy. An example if this claim is shown at Figures 4 (a) and (b) where the deviation of the accuracy between the models is increasing.

D. Parameter Tuning

In this section, we present the results of tuning the parameters of our model described in Section III. In particular, we examine the impact of: i) the sample size S and the dimension size d, ii) the time interval ΔT and iii) the regularization parameter a.

1) Impact of Samples and Dimensions number:

Both parameters examined in this section describe the ability of the model to fit and well-describe the data. Thus we have to obtain the best values that tune the model. On the one hand, if the size of both parameters is low, this will lead to bad description of the dataset. On the other hand, if the size is high this will increase the computational cost and may effect the accuracy of the final recommendations. To this problem, here we examine the threshold that reach the best values per cost in terms of accuracy.

To conclude to these sizes gain the higher accuracy we have experimented with different values of the sample size S and the dimension size d. Table IV presents the model's accuracy for the top@10 predictions for different values of S and d. It is noticeable that as long as the sample size S increases, the accuracy increases as well until reaching convergence point. This means that the model is highly affected by the size of the samples used. Also, it is evident that for values reaching this threshold the improvements is not that significant. Therefore, we set the sample size equal to 150 and 200 for both datasets. It is evident that this process reduces the computational cost significantly since the scale is in millions.

TABLE IV
IMPACT OF S AND D PARAMETERS FOR TOP@10

(a) Foursquare

S(*mil) \ d	80	90	**100**	110	120
50	0.385	0.385	0.388	0.388	0.388
100	0.398	0.390	0.391	0.391	0.392
150	0.408	0.409	**0.410**	0.411	0.411
200	0.408	0.408	0.409	0.411	0.411

(b) Weeplaces

S(*mil) \ d	80	90	**100**	110	120
100	0.471	0.474	0.476	0.476	0.476
150	0.475	0.479	0.479	0.479	0.480
200	0.480	0.484	**0.488**	0.488	0.489
250	0.481	0.485	0.488	0.489	0.489

On the other hand, we observe that the dimension size d does not affect the accuracy significantly beyond the value of 100 for both datasets. In particular, the improvement is not significant, which means that with a lower dimensionality the data can be described as well as with a higher number of dimension. The main difference is that the computation cost is much smaller than in the later case. In Tables IV (a) and (b), we highlighted the default values used for the rest of the evaluation.

2) Impact of Time Period:

The impact of time period to our model is crucial since the data are not always dense. Therefore, in many cases there are users who are not active for many days or their check-ins are not enough to be able to make predictions. The same happens with POIs that are not so popular. In both cases the accuracy of the model may be affected. To tackle this issue we examine the size of the time period ΔT (in days) in order to find the threshold that provides enough data for training and leads to higher accuracy.

In the experimental results shown in Table V we present

TABLE V
IMPACT OF TIME PERIOD ΔT FOR THE TOP@n PREDICTIONS

(a) Foursquare

ΔT (days) \ Acc@n	1	5	10	15	20
10	0.261	0.325	0.397	0.403	0.446
15	0.273	0.328	0.402	0.425	0.453
20	**0.286**	**0.341**	**0.410**	**0.435**	**0.462**
25	0.273	0.335	0.402	0.420	0.453

(b) Weeplaces

ΔT (days) \ Acc@n	1	5	10	15	20
20	0.368	0.405	0.463	0.497	0.519
30	0.373	0.413	0.475	0.501	0.524
40	**0.386**	**0.421**	**0.488**	**0.514**	**0.536**
50	0.374	0.417	0.479	0.507	0.527

the accuracy gained for different values of time periods and top@n predictions. First, its noticeable that the accuracy increases along with the size of n predictions. Also, we observe that the accuracy is increasing to a point and then gradually drops. This is quite reasonable since for small size of time periods there are not enough data to train our model. On the other hand, for bigger size of time periods there is too much noise in the data used for training which drops the accuracy. Thus, we set the size of ΔT equal 20 and 40 for both the Foursquare and the Weeplaces datasets, as shown in Tables V (a) and (b). We observe that for the Weeplaces dataset which spans a period of 91 months, the value of ΔT is significantly larger than that of Foursquare.

3) Regularization Parameter α:

In Figure 5 we present the results of tuning regularization parameter α in terms of accuracy for both datasets. As described in equation (13), each parameter corresponds to the social influence and the geographical proximity. We evaluated the overall accuracy of the model by setting parameter α to a value between [0-1]. It is evident that there is a peak point where our model gets the higher value of accuracy that was used in all of our experiments. Moreover, the importance of each influential network is diversified trying to fit the model on users behavior. Thus, peak points denote which information network is vital for the recommendations.

Our goal is to investigate what is the reason that motivates users to visit a new location. Thus, the regularization parameter α defines the importance of either the social influence, or the geographical proximity. As shown in Figure 5, the influential parameter is not the same for both datasets. Moreover, the higher the value of the accuracy for particular values of the parameter α indicates the influence of the corresponding network to the final recommendations. Regarding the Foursquare dataset, we observe that both contribute the same since the

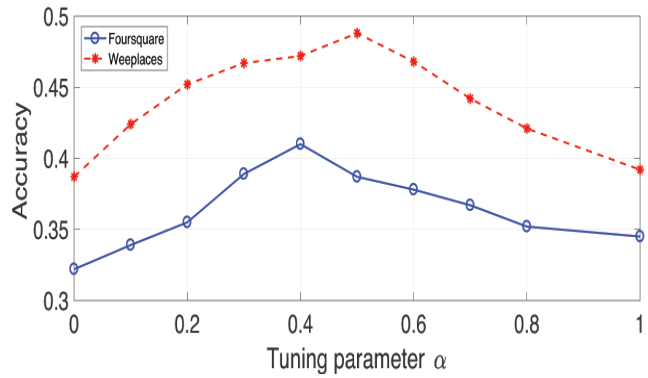


Fig. 5. The impact of the tuning parameter α on recommendation accuracy for both datasets.

higher value of accuracy is reached while $\alpha = 0.5$. On the other hand, the contribution of geographical influence is higher for Weeplaces where the higher accuracy gained for $\alpha = 0.4$.

V. CONCLUSIONS

The exponential growth of users' generated data in LBSNs led research and industry to the generation of models that aim to offer recommendations to users.One of the most popular topic is the Point-of-Interest recommendation, which refers to proposing POIs to visit. In this context, many models have been proposed in literature trying to analyze users' preferences and behavior in order to support them with that kind of recommendations.

Unfortunately, most of existing approaches lack a comprehensive way of personalizing their recommendations, since they ignore one or more of the following factors:

- the social influence,
- the check-in periodicity,
- the geographical proximity of POIs, and
- the preference dynamics.

For instance, some models, treat locations as regular nodes ignoring the spatial proximity between two locations, whereas models that considers geographical influence miss the periodicity of users' behavior. In addition, models exploring both the spatial and the temporal dimensions ignore users' preference dynamics which also evolve as time progresses. Recent research focused on humans behavior, indicates that users change their taste during time periods. This evolution indicates certain dynamics which should be considered during computation, towards increasing the accuracy of predictions.

To tackle the aforementioned issues, in this paper we propose a novel approach for POI recommendations that considers all factors in a unified way. The proposed model, jointly learns the embeddings of six weighted informational graphs, describing the users' and POI relations into the same latent space. Then, we personalize the final recommendations by tuning the regularization parameter which defines which influential network contributes the most. We have evaluated our method with two real life datasets in terms of accuracy for the top@n recommendations. Also, we have compared our method with the state-of-the-art approach named GE. Performance evaluation results indicate a significant improvement.

Below, we describe briefly some interesting future research directions:

- It is interesting to investigate the check-in sequences during specific time periods and extend the informational networks used for training the model.
- Unsupervised techniques that take into account the frequency of a sequence of POIs could also be applied to enhance the recommendation performance.
- Finally, stream-based recommendations is an important topic where the model parameters should be updated as time evolves, to reflect in a more aggressive way changes in user preferences. For example, sliding window semantics may be applied, where our predictions should be based on the recent activity of the users (defined by the sliding window).

References

[1] J.-D. Zhang and C.-Y. Chow, "iGSLR: Personalized geo-social location recommendation: A kernel density estimation approach," in *Proceedings of the 21st International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*. Orlando, Florida, 2013, pp. 334–343.

[2] H. Li, Y. Ge, R. Hong, and H. Zhu, "Point-of-interest recommendations: Learning potential check-ins from friends," in *Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, San Francisco, California, USA, 2016, pp. 975–984.

[3] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan, "Personalized ranking metric embedding for next new poi recommendation," in *Proceedings of the 24th International Conference on Artificial Intelligence (AAAI)*, Buenos Aires, Argentina, 2015, pp. 2069–2075.

[4] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *Proceedings of the 34th International Conference on Research and Development in Information Retrieval (SIGIR)*, Beijing, China, 2011, pp. 325–334.

[5] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: Successive point-of-interest recommendation," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (AAAI)*, Beijing, China, 2013, pp. 2605–2611.

[6] H. Gao, J. Tang, X. Hu, and H. Liu, "Content-aware point of interest recommendation on location-based social networks," in *Proceedings of the 29th Conference on Artificial Intelligence (AAAI)*, Austin, Texas, 2015, pp. 1721–1727.

[7] L. Liu, J. Xu, S. S. Liao, and H. Chen, "A real-time personalized route recommendation system for self-drive tourists based on vehicle to vehicle communication," *Expert Systems with Applications*, vol. 41, no. 7, pp. 3409–3417, 2014.

[8] X. Cao, G. Cong, and C. S. Jensen, "Mining significant semantic locations from GPS data," *Proceedings VLDB*, vol. 3, no. 1-2, pp. 1009–1020, 2010.

[9] P. Vansteenwegen, W. Souffriau, G. V. Berghe, and D. V. Oudheusden, "The city trip planner: An expert system for tourists," *Expert Systems with Applications*, vol. 38, no. 6, pp. 6540–6546, 2011.

[10] M. Xie, H. Yin, H. Wang, F. Xu, W. Chen, and S. Wang, "Learning graph-based poi embedding for location-based recommendation," in *Proceedings of the 25th International on Conference on Information and Knowledge Management (CIKM)*, Indianapolis, Indiana, USA, 2016, pp. 15–24.

[11] R. Baral and T. Li, "MAPS: A multi aspect personalized poi recommender system," in *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys)*, Boston, Massachusetts, USA, 2016, pp. 281–284.

[12] P. Kefalas and Y. Manolopoulos, "A time-aware spatio-textual recommender system," *Expert Systems with Applications*, vol. 78, pp. 396–406, 2017.

[13] Y. Koren, "Collaborative filtering with temporal dynamics," in *Proceedings 15th ACM International Conference on Knowledge Discovery & Data Mining (KDD)*, 2009, pp. 447–456.

[14] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann, "Time-aware point-of-interest recommendation," in *Proceedings 36th International ACM Conference on Research & Development in Information Retrieval (SIGIR)*, 2013, pp. 363–372.

[15] P. Kefalas, P. Symeonidis, and Y. Manolopoulos, "Recommendations based on a heterogeneous spatio-temporal social network," *World Wide Web*, vol. 21, no. 2, pp. 345–371, 2018.

[16] S. Krishnan, J. Patel, M. J. Franklin, and K. Goldberg, "A methodology for learning, analyzing, and mitigating social influence bias in recommender systems," *Proceedings of the 8th ACM Conference on Recommender systems (RecSys)*, pp. 137–144, 2014.

[17] G. Guo, F. Zhu, S. Qu, and X. Wang, "PCCF: periodic and continual temporal co-factorization for recommender systems," *Information Sciences*, vol. 436-437, pp. 56–73, 2018.

[18] J. He, X. Li, L. Liao, D. Song, and W. K. Cheung, "Inferring a personalized next point-of-interest recommendation model with latent behavior patterns," in *Proceedings of the Thirtieth Conference on Artificial Intelligence (AAAI)*, 2016, pp. 137–143.

[19] C. Zhang, K. Wang, H. Yu, J. Sun, and E. Lim, "Latent factor transition for dynamic collaborative filtering," in *Proceedings of the SIAM International Conference on Data Mining (SDM)*, 2014, pp. 452–460.

[20] J.-D. Zhang and C.-Y. Chow, "GeoSoCa: Exploiting geographical, social and categorical correlations for point-of-interest recommendations," in *Proceedings of the 38th International Conference on Research and Development in Information Retrieval (SIGIR)*, Santiago, Chile, 2015, pp. 443–452.

[21] W. Weiqing, Y. Hongzhi, W. S. Shazia, C. Ling, X. Min, and Z. Xiaofang, "SPORE: A sequential personalized spatial item recommender system," in *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, Helsinki, Finland, 2016, pp. 954–965.

[22] H. Wang, Y. Fu, Q. Wang, H. Yin, C. Du, and H. Xiong, "A location-sentiment-aware recommender system for both home-town and out-of-town users," *Proceedings of the 23rd International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 1135–1143, 2017.

[23] Y. Dong, N. V. Chawla, and A. Swami, "Metapath2vec: Scalable representation learning for heterogeneous networks," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '17, 2017, pp. 135–144.

[24] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web (WWW)*, Florence, Italy, 2015, pp. 1067–1077.

[25] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS)*, Lake Tahoe, Nevada, 2013, pp. 3111–3119.

[26] H. Yin, X. Zhou, Y. Shao, H. Wang, and S. Sadiq, "Joint modeling of user check-in behaviors for point-of-interest recommendation," in *Proceedings of the 24th International on Conference on Information and Knowledge Management (CIKM)*, Melbourne, Australia, 2015, pp. 1631–1640.