A Neighbourhood-based Location- and Time-aware Recommender System

Len Feremans[1], Robin Verachtert[2] and Bart Goethals[1]

[1]University of Antwerp, Belgium
[2]Froomle N.V., Belgium

Abstract
We address the problem of location- and time-aware recommender systems where users with dynamically changing locations are interested in trending and volatile items. Unlike existing work, we do not assume a known static location of each user and derive user-locational preferences from their long-term history of implicit feedback. We propose a recommendation model that accounts for spatial, temporal, popularity and social influences, thereby assuming items tagged with a location, i.e. geotag, city or country. Key ingredients of our online method include: (1) deriving location preferences from the history, (2) learning relevant nearby locations, (3) accounting for recency and popularity jointly, and (4) combining location- and time-aware recommendations with collaborative filtering. Supported by realistic offline and online experiments on a large dataset collected from a popular newspaper, and public datasets, we find that the proposed recommender outperforms content-based and time-aware collaborative filtering approaches.

Keywords
Context-aware recommender systems, Location-based news recommendations, Collaborative filtering

1. Introduction

Every day, users consume items tagged with locations, e.g. local news articles from a particular city or Twitter tags trending in a specific region. Recommendations are essential to tackle information overload and filter relevant items from a huge set of available articles. In this context, the first law of geography posed by Tobler [25] is crucial: "*everything is related to everything else, but near things are more related than distant things*".

A common strategy for context-aware recommendations [1] is pre-filtering, i.e. we collect the location of each user and rank geotagged items nearby. However, this strategy is problematic. Firstly, many users have *multiple and dynamic* regions of preference, i.e. they might be interested in items near their home, work or recent vacation stay. Secondly, even after filtering on a preferred location, there are some biases resulting from *population density*, i.e. items geotagged with a big city will likely be more numerous and popular, which is likely not relevant for all users near that city (vice versa, rural locations might lack recent item interactions). Additionally, we find that the intrinsic *popularity bias* is detrimental to inferring regional preferences, i.e. an item might be relevant to many users regardless of the associated geotag such as a news item related to an international sports event or a celebrity tweet. Finally, we have to consider the *relationship between locations*, i.e. in some applications physical distances are more important (i.e. point-of-interest recommendations in Facebook places), while in other applications (i.e. Twitter tags) location similarity is higher when many users consecutively prefer both locations [2].

In this work, we investigate the problem of location- and time-aware recommender systems (LTARS) and ranking highly volatile geotagged items by considering both spatio-temporal and user activity trends. We study factors correlated with item relevance: (1) geographic factors, such as the *geodesic distance* between the geotag of an item and the inferred regional preferences of a user, and (2) *user-item preference*, i.e. the item-neighbourhood based relevancy, and (3) the *recency* and *popularity* of an item. This problem has been studied in the context of *time-aware recommendations* [8, 3], *location-aware recommendations* [5, 21, 24, 7, 19], *context-aware recommendations* [1, 9, 16] and *location-based social networks* [2, 28, 12]. A key difference with closely related research by Pálovics et al. [21] is that we assume a user is interested in multiple locations and these locational preferences are unknown and dynamic. A second key difference we consider is the volatility of items, which is crucial in certain domains such as news recommendations [15] and often less in other domains such as point-of-interest recommendations [2]. A third difference is that we account for naturally occurring biases in the data such as an imbalance in the popularity and location distributions that hinder recommendations based on (context-aware) collaborative filtering [1].

We propose an LTARS that is orthogonal to existing

ORSUM@ACM RecSys 2022: 5th Workshop on Online Recommender Systems and User Modeling, jointly with the 16th ACM Conference on Recommender Systems, September 23rd, 2022, Seattle, WA, USA
len.feremans@uantwerpen.be (L. Feremans); robin.verachtert@froomle.com (R. Verachtert); bart.goethals@uantwerpen.be (B. Goethals)

location-aware and context-aware recommender systems that assume known contextual features or static locational preferences [21, 1, 24] and make the following key contributions:

- We propose novel techniques to (1) extract location preferences from users based on their frequency in the long-term history of geotagged items; (2) identify and rank relevant neighbour locations based on collaborative filtering or geographical distance; (3) filter and rank items jointly on recency and popularity; (4) combine location- and time-aware recommendations with collaborative filtering.
- The proposed method is straightforward to implement and publicly available[1]. It is also highly efficient supporting frequent or online updates and cold-start item recommendations.
- Motivated by the recent criticism of unrealistic offline evaluation protocols [3, 13, 23], we adopt an evaluation protocol based on a *sliding window protocol* and create subsets of interactions for training and testing during consecutive periods where we filter candidate items on publication date (or first interaction time).
- We find that the proposed method and hybrids thereof outperform popularity, content-based and (time-ware) collaborative filtering-based recommender systems on offline and online experiments for *regional new recommendations*.

2. Related work

In most related work on *location-aware recommendations*, general news [24] or Twitter tags [21] are recommended. In contrast to [24] and extended work [20, 4] we assume user locational preferences are non-stationary. Pálovics et al. propose an online model to recommend volatile items, thereby assuming non-stationary locational preferences [21]. Our work is complementary by inferring regional preferences for each user. Pálovics et al. also propose a hierarchically organized geolocation structure, while we propose an alternative neighbourhood-based method. We argue that focusing on neighbouring locations is essential in applications with a high cardinality set of locations, such as regional news recommendations. Finally, we use experimental validation in a different domain, i.e. regional news recommendations instead of geotagged tweets. We find that in both domains, location-aware methods outperform popularity, content-based and collaborative filtering-based approaches.

In *time-aware* and *news recommendations*, most algorithms are content-based, based on collaborative filtering or hybrids thereof [15, 22]. Das et al. propose time-aware news recommendations that combine item-based collaborative filtering, pre-filtering news items on recency, and age-based discounting to account for the bias towards recent items [8]. Similarly, we consider the short lifetime of items, online scalability issues and realistic offline evaluation protocols [3]. A key difference is that we account for location preferences and tackle cold-start item recommendations. We compare with time-aware collaborative filtering in Section 4.

In *location-based social networks* point-of-interests items are tagged geographically by users interacting on a social network such as Facebook places or Foursquare [28, 2, 12]. Related to our work, Ye et al. propose a hybrid recommendation system that models user-item preferences using a combination of collaborative filtering, social influence, and geographic modelling where location preferences are proportional to the inverse squared geographic distance [28, 18]. However, their approach is specific to point-of-interest applications where items such as restaurants are rated after users physically check in at a specific address with longitude and latitude coordinates. Location-based social network recommender systems typically also model social influence, assume explicit feedback and a long life-time of items.

3. A neighbourhood-based location- and time-aware recommender system

The proposed method is a combination of different steps and components that take different signals into account, i.e. spatial, temporal, popularity and social influence, as shown in Figure 1.

3.1. Task definition

Let $U = \{u_1, u_2, \dots u_n\}$ be the set of users and $I = \{i_1, i_2, \dots, i_m\}$ the set of items. We consider *implicit feedback* where a user interacts with a certain item at a certain timestamp, i.e. $\mathscr{D} = \{\langle u, i, t \rangle \mid u \in U \wedge i \in I\}$ and denote the user history using $I_{u,t}$, i.e. all interacted items up to timestamp t. Each item has one or more *locations* or geotags L_i where $L_i \subseteq L = \{l_1, l_2, \dots, l_k\}$, e.g. a specific address, city or country. Each item is available starting from a certain time t_i, i.e. the *publication timestamp*. In case this is not available, we compute $t_i = min(\{t \mid \langle u, j, t \rangle \in \mathscr{D} : i = j\})$. For time-aware recommendations we imitate the online setting as close as possible and evaluate offline based on a *sliding window protocol* where we partition $\mathscr{D}$ in time given parameters $\triangle t_{train}$ and $\triangle t_{test}$. That is, we train the model at timestamp t using $\mathscr{D}_{train} = \{\langle u_k, i_k, t_k \rangle \mid t - \triangle t_{train} < t_k \leq t\}$ and predict inter-

[1] https://bitbucket.org/len_feremans/regio-reco/

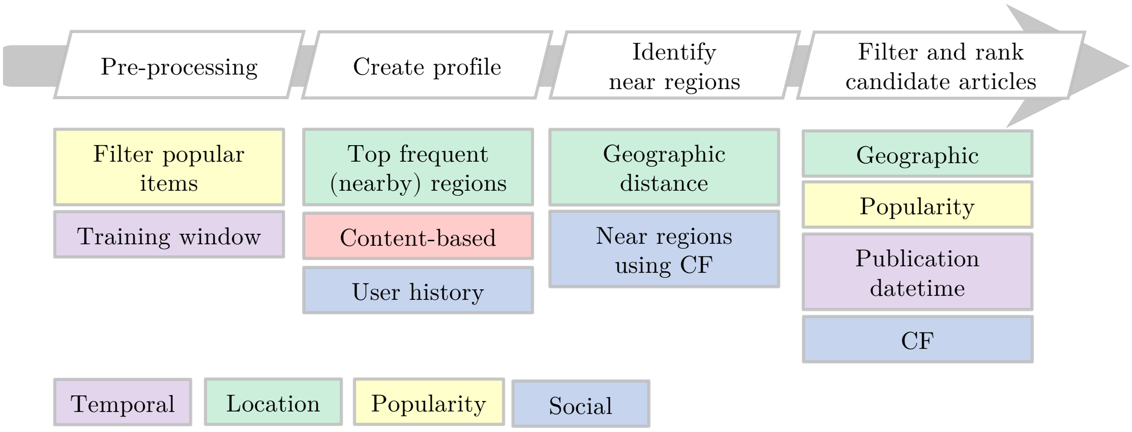

Figure 1: Overview of proposed LTARS recommender system consisting of 4 main steps. At each step there are different factors we account for: spatial influence, temporal influence, popularity influence and social influence

actions in $\mathscr{D}_{test} = \{\langle u_k, i_k, t_k\rangle \mid t < t_k < t + \triangle t_{test}\}$. Additionally, we define the set of *impressionable items* at timestamp t using $C_{impr} = \{i \mid i \in I \,:\, t - \triangle t_{train} < t_i < t + \triangle t_{test}\}$. The interaction data is represented using a user-item-location matrix as shown in Figure 2. The notations and definitions used in this paper are summarised in Table 1.

The goal is to generate a set of top-n personalised items most relevant to each user u given their history I_u at timepoint t.

3.2. Identifying regional preferences and neighbouring locations

We assume each user has preferences for multiple locations of interest, i.e. their home address, work address or recent vacation stay. First, we propose a straightforward technique for determining location preferences using the user's history based on *frequency*. Next, we use location-location and user-location (or collaborative) dependencies for determining nearby user-preferred locations.

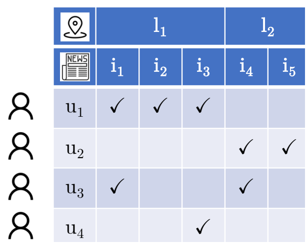

Figure 2: Illustration of the user-item-location matrix. Based on this example, we infer that user u_1 is primarily interested in items from location l_1, i.e. $sup(u_1, l_1, t) = 1$ and user u_3 in both location l_1 and l_2, where $sup(u_3, l_1, t) = sup(u_3, l_2, t) = 0.5$. For simplicity, we assume a single geotag per item in the illustration.

3.2.1. Top frequent locations

Given a user u and history $I_{u,t}$ at timestamp t (i.e. all interactions in $\mathscr{D}_{train}$) we count the top-k most frequent geotags using:

$$P^k(u,t) = \{l \mid \exists i \in I_{u,t} \,:\, l \in L_i \wedge rank(sup(u,l,t)) \leq k\} \quad \text{where}$$

$$sup(u,l,t) = \frac{|\{i \mid i \in I_{u,t} \,:\, l \in L_i\}|}{|I_{u,t}|}.$$

The frequency, or support, is a measure of the *user-location preference*, i.e. the ratio of location-specific views versus all views for that user. We remark that in our experiments, recommendations based on the top frequent locations improve substantially using a longer history (e.g. a full month instead of the last days) and after removing the 5% most popular items as very popular items are detrimental when inferring locational preferences.

3.2.2. Geographical neighbour locations

We define for each location l_i the top-k nearest locations using

$$N^k_{GEO}(l_i) = \{l_j \mid l_j \in L \wedge rank(dist(l_i, l_j)) \leq k\}$$

where we use *dist* to denote the geodesic distance between two geotags in kilometres. In practice, we use the set $L_{train} \subseteq L$ containing any location associated with any item in the training data since for many (smaller) locations we have no items published in the past period, i.e. *cold-start locations* and a lower likelihood for future recommendations.

Symbol	Description
U	set of users
I	set of items
L	set of locations
u	a user, $u \in U$
i	an item, $i \in I$
l	a location, $l \in L$
L_i or l_i	set of locations $L_i \subseteq L$ or a single location l_i association with item i
t	a timestamp, often representing current time
Δt	an interval in time
t_i	publication (or first interaction) timestamp of item i
$\mathscr{D}$	set of interactions, i.e. tuples $\langle u, i, t\rangle$
$\mathscr{D}_{train}$	set of interactions for training in $[t-\Delta_{train}, t]$
$\mathscr{D}_{test}$	set of interactions for evaluation in $]t, t+\Delta_{test}]$
C_{impr}	set of impressionable candidate items at t, i.e. published before $t+\Delta_{test}$
$I_{u,t}$	set of items viewed by user u before timestamp t
$sup(u,l,t)$	frequency, or support, of location l in $I_{u,t}$
$P^k(u,t)$	user-location preferences, i.e. top-k locations with highest support in $I_{u,t}$
$P'(u,t)$	top-k most frequent locations and nearby locations from either N_{GEO} or N_{CF}
$N^k_{GEO}(l)$	top-k geographically nearest locations to l
$P(l_j \mid l_i)$	conditional probability of visiting location l_j given l_i is visited before
$N^k_{CF}(l,t)$	top-k nearest locations to l based on collaborative filtering
$C_{LOC}(u,t)$	candidate items matching user-location preferences
$C_{POP}(t)$	candidate items matching recency and popularity constraints
$C(u,t)$	candidate items matching user-location preference, recency and popularity
$rank_{LOC}$	score i based on user-location preference
$rank_{POP+REC}$	score i based on recency and popularity
$rank_{LTARS}$	score i based on user-location preference, recency and popularity
$rank_{CF}$	score i based on collaborative filtering

Table 1
Overview of notation and definitions

3.2.3. Collaborative filtering-based neighbour locations

An alternative technique is to learn nearby locations based on the geotagged item histories of all users, resulting in *because you interacted with location l_i we recommend location l_j* type of inference. Hence, we compute the co-visitations for each location similar to item-based collaborative filtering but counting co-occurrences of the location of items instead of the items themselves. We store co-occurrence counts in a matrix $M^{|L|\times|L|}$. We use the notation $P(l_j \mid l_i)$ to denote the conditional probability between two locations, e.g. 30% of users who view items from l_i also view items from l_j, defined as:

$$P(l_j \mid l_i) = \frac{M_{i,j}}{M_{i,i}} \quad \text{where}$$

$$M_{i,j} = \sum_{u \in U} \min\left(\sum_{a \in I_{u,t}} \mathbb{1}\,(l_i \in L_a)\,, \sum_{b \in I_{u,t}} \mathbb{1}\left(l_j \in L_b\right)\right).$$

We remark that locations can re-occur in the history for each user and is represented by a multiset, henceforth the number of co-occurrences is the minimum of the total number of occurrences of each pair of locations aggregated over all users. We define for each location l_i the top-k nearest locations using:

$$N^k_{CF}(l_i,t) = \left\{l_j \mid l_j \in L \wedge rank\left(P(l_j|l_i)\right) \leq k\right\}$$

3.2.4. Recommending items based on neighbouring locations

We *extend* regional preferences $P^k(u,t)$ to near locations using:

$$P'(u,t) = P^{k_1}(u,t) \cup \{l_i \mid \exists\, l \in P^{k_1}(u,t) \;:\; l_i \in N^{k_2}_{\alpha}(l,t)\}$$

where N_α is either N_{GEO} or N_{CF} and k_1 and k_2 are hyperparameters. We remark that if the profile $P^k(u,t)$ only contains low-density locations, we might have fewer than top-n item recommendations after filtering. Additionally, locations in $P^k(u,t)$ are based only on the history of the current users, while we use N_{GEO} if users are interested in locations near frequently visited past locations or N_{CF} if users are interested in locations frequently co-visited by all users.

Next, we filter impressionable candidate items in C_{impr} matching location regional preferences. Given a user u and regional preferences $P'(u,t)$ we define the set of filtered candidate items at timestamp t:

$$C_{LOC}(u,t) = \{i \mid i \in C_{impr} \;:\; L_i \cap P'(u,t) \neq \emptyset\}.$$

We rank items in $C_{LOC}(u,t)$ based on the user-location preference score using:

$$rank_{LOC}(u,i,t) = \begin{cases} sup(i,l_i,t) & \text{if } l_i \in P^k(u,t) \\ sup(u,l_j,t)\cdot\left(1 - \frac{dist(l_i,l_j)}{\max(\{dist(l_j,l_k) \mid l_k \in N^k_{GEO}(l_j)\})}\right) & \\ \quad \text{if } l_i \in P'(u,t) \wedge N_\alpha = N_{GEO} & \\ sup(u,l_j,t)\cdot P(l_i|l_j) & \\ \quad \text{if } l_i \in P'(u,t) \wedge N_\alpha = N_{CF}\,. & \end{cases}$$

In case a candidate item i is tagged with multiple locations L_i, we use the location having the maximal score w.r.t. to the user for estimating relevance, i.e. $\underset{l_j \in L_i}{argmax}\; sup(u,l_j,t)$.

For example, assume u_1 has interacted with 100 items of which 50 are tagged with l_1 and 20 with l_2 during the training period $[t - \triangle t_{train}, t[$. Assume that 30% of all users who interacted with l_1 also interacted with l_3. The user-location preferences are then $rank_{LOC}(u_1, l_1, t) = 0.5$, $rank_{LOC}(u_1, l_2, t) = 0.2$ and $rank_{LOC}(u_1, l_3, t) = 0.5 \times 0.3$.

3.3. Ranking items on popularity and recency

This section considers strategies to filter and rank items on popularity and recency. A *popularity filter* keeps candidate items above a certain popularity threshold. A *recency filter* keeps candidate items published before a specific timestamp. Both approaches have their drawbacks, i.e. a popularity filter removes cold-start (or very recent) items, while a recency filter removes slightly aged yet popular items. We define the set of candidate items filtered on both recency and popularity at timestamp t as:

$$C_{POP}(t) = \{i \mid i \in C_{impr} : (t - \triangle t_1 < t_i) \wedge (t - \triangle t_2 < t_i \vee pop(i, t, \triangle t_{pop}) > \epsilon) \quad \text{where}$$

$$pop(i, t, \triangle t_{pop}) = |\{\langle u_k, i_k, t_k \rangle \mid \langle u_k, i_k, t_k \rangle \in \mathcal{D}_{train} : i_k = i \wedge t_k > t - \triangle t_{pop}\}|.$$

Here, $\triangle t_1$, $\triangle t_2$, $\triangle t_{pop}$ and ϵ are hyper-parameters. For instance, by selecting $\triangle t_1 = 4d$, $\triangle t_2 = 2d$, $\triangle t_{pop} = 4d$, and $\epsilon = 1$ we exclude items that are more than 4 days old or between 2 to 4 days old and have fewer than 1 interactions during the last 4 days. For brevity of this paper we omit detailed experiments on the effect of selecting candidate using C_{POP}. But we remark that on the regional news dataset, we have a recall@10 of 0.221 when using the default set of impressionable items C_{impr}, which increases to 0.231 (+4.5%) by filtering articles older than two days and to 0.239 (+8.1%) using C_{POP} w.r.t. the previous parameters.

Traditionally, we rank candidate items on recency, i.e. publication timestamp or popularity. However, both approaches have their disadvantages. We propose to rank candidate items i on recency and popularity *jointly* using:

$$rank_{POP+REC}(i, t) = \frac{pop(i, t, \triangle t_{pop}) + c}{t - t_i}$$

where we normalise the popularity by the number of hours since the publication and where c represents a bias term for cold-start items. For instance, given item i_1 that was published 1 hour ago with 100 views and an item i_2 published 5 hours ago with 200 views. We rank i_1 before i_2 since on average more users have viewed i_1 items in a single hour. As a second example, assume $b = 10$ and i_1 was published 15 minutes ago with 0 views and i_2 was published 5 hours ago with 100 views. Again we rank i_1 before i_2 since $rank_{POP}(i_1, t) = \frac{0+10}{0.25} = 40$ and $rank_{POP}(i_2, t) = \frac{100+10}{5} = 22$.

3.4. Combining location- and time-aware recommendations with collaborative filtering

This section proposes combinations of location-, time-aware and collaborative filtering-based recommendations.

3.4.1. Location- and time-aware recommendations

First, we propose to combine scores to recommend items having relevant geotags and are trending. We define the set of candidate items by filtering on both regional preferences, popularity and recency: $C(u, t) = C_{LOC}(u, t) \cap C_{POP}(t)$. For a user u the candidate items $i \in C(u, t)$ are ranked using:

$$rank_{LTARS}(u, i, t) = \alpha \cdot rank_{LOC}(u, i, t) + (1 - \alpha) \cdot \overline{rank}_{POP+REC}(i, t)$$

$$\text{where}$$

$$\overline{rank}_{POP+REC}(i, t) = \frac{rank_{POP+REC}(i, t)}{\max(\{rank_{POP+REC}(j, t) \mid j \in C(u, t)\})}$$

where α is a hyper-parameter to control the relative weight of a user's regional preferences versus the popularity of an item normalised over its age.

3.4.2. Online computation of location- and time-aware recommendations

An advantage of the proposed approach for location- and time-aware recommendation is that it can be computed *online*. For updating $P^k(u, t)$ for each user we store the frequency of each location in memory and update counts when new interactions arrive. Since, N_{GEO} is time-independent, we precompute the pairwise distances once offline having a complexity of $O(|L|^2)$ and store the resulting matrix. For computing neighbouring locations based on collaborative filtering, the complexity is $O(|\mathcal{D}| + |L|^2)$. Algorithms exist to update the item similarities incrementally [14], which in principle could be adopted for updating location similarities online. However, since location neighbourhoods are typically less dynamic, the need for frequent model updates is less important. Therefore, we choose to re-compute the co-visitation matrix regularly, thereby adopting *sparse optimisation* techniques that make this computation extremely efficient, even on large datasets [11]. Finally, we update the popularity counts of each item online. We remark that online model training is essential for both computational efficiency and accuracy, since in many domains, such as social media, news or auction websites recent items are the most relevant.

3.4.3. Hybrid recommendations

A limitation of the previous approach is that two users with the same regional preferences $P'(u,t)$ receive the same recommendations at timestamp t. A natural extension is to adopt existing time-aware collaborative filtering methods [8] and have a hybrid solution where we also account for user-item preferences. Therefore, we adopt item-based collaborative filtering as a strong baseline [10, 6]. We compute conditional probabilities for each item pair based on co-visitations and apply exponential age-based discounting to give more weight to recent interactions [8, 17]. We define a weighted co-visitation matrix $F^{|I| \times |I|}$ using:

$$F_{i,j} = \sum_{u \in U} w^t_{u,i} \cdot w^t_{u,j} \qquad \text{where}$$

$$w^t_{u,k} = \begin{Bmatrix} a \cdot (1-b)^{t-t_{u,k}} & \text{if } i_k \in I_{u,t} \\ 0 & \text{otherwise} \end{Bmatrix}.$$

Here, $t - t_{u,k}$ denotes the difference in hours between the current time t and the time of interaction $t_{u,k}$ transformed using an exponential time-decay function parameterised by a and b. For collaborative filtering-based recommendation we compute a score:

$$rank_{CF}(u,i,t) = \sum_{j \in I_{u,t}} P(i \mid j) = \sum_{j \in I_{u,t}} \frac{F_{i,j}}{F_{j,j}+1}$$

where $P(i|j)$ denotes the conditional probability between two items, e.g. 30% of users who (recently) viewed article j also (recently) viewed article i.

Finally, we recommend items based on both location- and time-aware preferences and collaborative filtering, i.e. given a user u and a candidate item $i \in C(u,i)$, we define:

$$rank_{LTARS+CF}(u,i,t) = \beta \cdot rank_{LTARS}(u,i,t) + (1-\beta) \cdot rank_{CF}(u,i,t).$$

4. Experimental setup and results

4.1. Dataset and offline evaluation

4.1.1. Regional news

We collect data from a prominent regional newspaper in Belgium. In the current digital age more and more users look for information on newspaper websites which brings several challenges for the recommendation system. Regional news is different from general news in that there is usually more of it where many regions and towns have multiple articles published every day.

We load all interactions and article metadata during a 40-day period (from 1st July until 11th August 2021) and exclude all articles containing general news and sport. Next, we filter items and users having fewer than five interactions and items that are viewed more than once by the same user. By default, we remove candidate items that are more than 4 days old. We remove the overall top 1% most popular items to overcome that the recommendation model is biased towards predicting only popular items. After pre-processing, we have 7.6 million interactions, 458 755 users and 9 493 items. Next, we perform an offline simulation where we evaluate recommendations using a sliding window of 2 hours ($\triangle t_{test} = 2h$) in the last week of data since the popularity distribution (and results) vary substantially in time [23]. At each two-hourly interval, we train a model based on interactions in the past $\triangle t_{train}$ days and report *recall@10* and *ndcg@10* during the entire test week for users with interactions in both the train and test set.

4.1.2. Public datasets

We also use two public location-based social network datasets to promote reproducibility. The datasets contain 227,428 and 573,703 check-ins collected for 10 months from Foursquare in New York City and Tokyo [27]. Each check-in is associated with a venue (or item), timestamp, GPS coordinates and category, which we ignore. Since the dataset is relatively small, we use a single time-based split and use the last month for evaluation. We pre-process the dataset as before and filter items and users with fewer than five interactions and items viewed more than once. Additionally, we round GPS coordinates to 2 decimals to create location tags, thereby considering all coordinates within 1.11 kilometres identical. The main properties of each dataset are shown in Table 2.

4.2. Comparing regional preferences and neighbouring locations determined using geodistance and collaborative filtering

In this first experiment, we investigate if users are more interested in geotagged items that are nearby geographically or from similar locations based on collaborative filtering. We investigate the following methods on the regional news dataset:

1. Using the top-k most frequent regions, i.e. $P^k(u,t)$ *without* nearby regions.
2. Add geographically near locations from N_{GEO}.
3. Add near locations based on collaborative filtering from N_{CF}.

Dataset	#users	#items	#locations	#interactions
Regional news	458 755	9 493	329	7 649 178
Foursquare TKY	2 292	7 057	732	128 555
Foursquare NYC	1 083	3 908	527	40 935

Table 2
Properties of datasets after preprocessing.

For extending the profile $P'(u,t)$ we select hyperparameters $k_1 = 3$ and $k_2 = 3$, use $\Delta_{train} = 30d$ and filter the top 5% most popular items before learning regional preferences and the location similarity matrix. We rank candidate items matching regional preferences on recency.

In Figure 3, we show each method result's for ndcg@10 and recall@10. The mean ndcg@10 is 0.184 when using collaborative filtering, 0.222 when using geographically nearby locations and 0.203 when only using the top-k frequent locations from the history. Therefore, we observe a relative increase of 8.5% using nearby regions determined using geodesic distance. A similar trend holds for recall@10. We observe that adding nearby locations using collaborative filtering does not perform well in this dataset. However, we argue that this variant has potential in other applications, such as Twitter tag predictions [21], where locations are more distant and international.

4.3. Comparing ranking methods

Intuitively purely ranking on recency as we do in the previous experiment does not result in the best top-*n* recommendations. In this experiment, we compare the following ranking methods on the regional news dataset:

1. On recency
2. On popularity
3. On $rank_{POP+REC}$
4. On $rank_{LTARS}$

We filter candidate items using the top-3 most frequent locations for each user and set hyperparameters $\triangle t_{pop} = 12h$ for the popularity window, $c = 0$ for $rank_{POP+REC}$ and $\alpha = 0.5$ for $rank_{LTARS}$ giving equal weight to the user-location preference score $rank_{LOC}$ and $rank_{POP+REC}$.

In Figure 4, we show the results for ndcg@10 and recall@10. The mean ndcg@10 is 0.205 by ranking on popularity, 0.204 using recency and 0.218 using $rank_{POP+REC}$ (+5.9%). If we rank using $rank_{LTARS}$ the ndcg@10 increases to 0.226 (+9.3%). We remark that by ignoring recency and ranking on user-location preference score only the ndcg@10 decreases to 0.122. The recall@10 is respectively 0.285, 0.296, 0.300 and 0.309 by ranking on popularity, recency, $rank_{POP+REC}$ and $rank_{LTARS}$. We conclude that ranking items on user-location preference and popularity/age outperforms baseline ranking functions by a wide margin.

4.4. Comparing popularity, time-aware collaborative filtering, content-based filtering and location- and time-aware recommendation systems

In this experiment, we compare the following recommendation systems on the proprietary regional news dataset and two public datasets:

1. A *popularity baseline* ranking the most trending items.
2. *Content-based filtering* ranking the most similar items using soft-cosine based on a pre-trained word2vec embedding [26].
3. *Item-based collaborative filtering* with age-based discounting [8, 17].
4. *LTARS* with geographically near locations and ranking jointly on user-location preference and recency and popularity.
5. A *hybrid* recommender where we combine LTARS with item-based collaborative filtering.

4.4.1. Regional news dataset

We select hyperparameters $\triangle t_{pop} = 12h$ for popularity, $\triangle t_{train} = 3d$ for collaborative filtering and $\triangle t_{train} = 30d$ for LTARS. For collaborative filtering we set the weight-decay parameters to $a = 1$ and $b = 0.1$. For LTARS we use the extended profile $P'(u,t)$ where we use the top-3 most frequent regions ($k_1 = 3$) and top-3 geographically nearest neighbours ($k_2 = 3$) and for ranking we set α to 0.25 thereby giving more relatively more weight to the user-location preference score. For the hybrid recommender we set β to 0.5.

The resulting ndcg@10 and recall@10 values over a week are shown in Figure 5 and the average values in Table 3. Concerning ndcg@10 the hybrid method works best, i.e. with an ndcg@10 of 0.270 we observe a 6.2% increase over item-based collaborative filtering, a 13.7% increase over LTARS, and a 50% increase over popularity. We omit the results from the plot for the content-based recommender: with an ndcg@10 of only 0.019 it performs poorly. With a recall@10 of 0.301 the proposed LTARS method has the highest recall@10 and we observe a 4.3% increase compared to item-based collaborative filtering, a 24.9% increase compared to the popularity baseline, and a small 1.3% increase compared to the hybrid recommender.

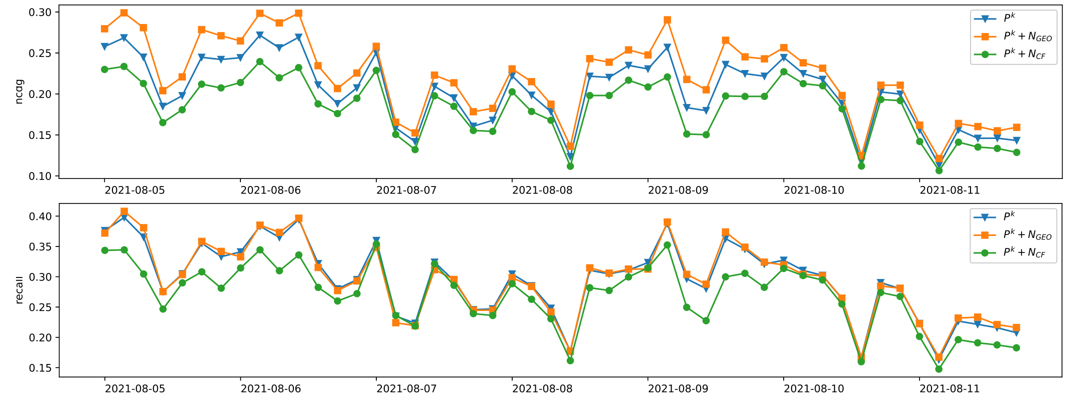

Figure 3: Ndcg@10 and recall@10 over one week for different methods for determining regional preferences and neighbouring locations on the regional news dataset. Using the top-3 frequent location the mean ndcg@10 is 0.203, by adding near location from N_{GEO} the ndcg@10 increases to 0.222 (+8.5%), and by adding nearby locations from N_{CF} the ndcg@10 decreases to 0.184 (-9.3%).

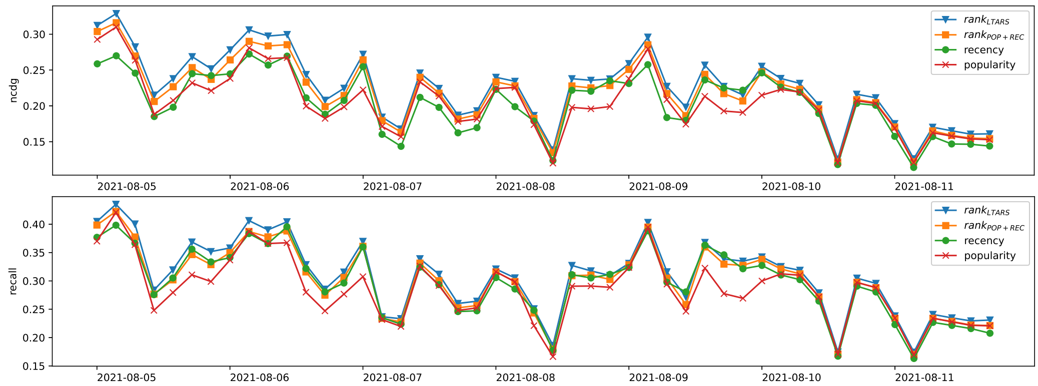

Figure 4: Ndcg@10 and recall@10 over one week for different methods for ranking candidate items on the regional news dataset. By ranking on popularity results the a mean ndcg@10 is 0.205 and on recency the mean ndcg@10 is 0.204. By ranking on $rank_{POP+REC}$ the ndgc@10 increases to 0.218 (+5.9%). By ranking on $rank_{LTARS}$ the ndcg@10 increases to 0.226 (+9.3%).

Interestingly, there is no clear winner over the entire week, i.e. on the last day we observe a severe *drift* in popularity bias better captured by collaborative filtering since LTARS filters out (popular) items not matching regional preferences. However, the accuracy of LTARS validates the premise that local items are often more relevant.

4.4.2. Public datasets

We repeat the previous experiment using two public location-based social network datasets. We remark that in both datasets, there is no significant preference towards more recent venues. We set the popularity and training window to use all available data, i.e. we select hyperparameters $\triangle t_{pop} = \triangle t_{train} = 9m$ for collaborative filtering and LTARS and do not use weight-decay. For LTARS we use the extended profile $P^k(u,t)$ where we use the top-20 most frequent regions and set α to 0.75. For the hybrid recommender we set β to 0.5.

We show the results in Table 3. We find that, concerning ndg@10 the hybrid method performs best, followed by LTARS with a large margin. Concerning recall@10 LTARS performs worse than the popularity baseline. We remark that by ignoring ranking on item recency would further improve results. We conclude that our method outperforms the popularity and item-based collaborative filtering methods by a large margin concerning ndcg@10.

4.4.3. Execution time

In Table 4 we show the total execution time in seconds for training the model and computing predictions. Runtimes are measured on a laptop with an 2,3 GHz 8-Core Intel

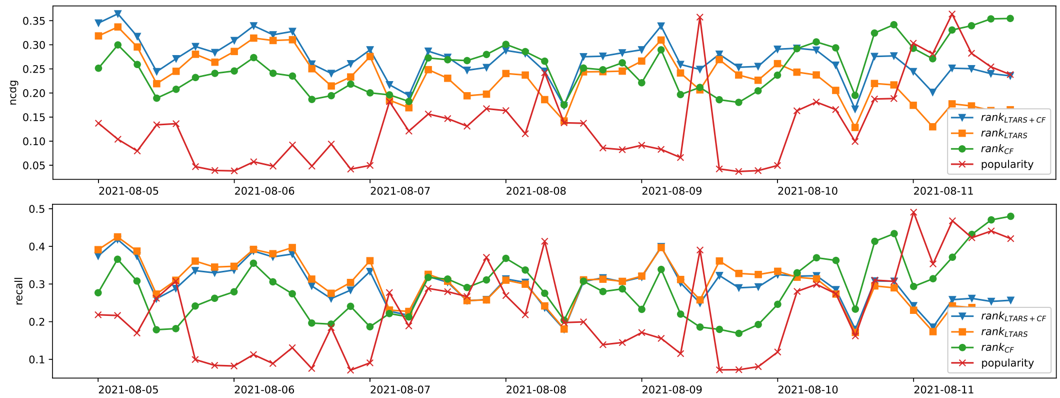

Figure 5: Ndcg@10 and recall@10 for different recommendations techniques over a week on the regional news dataset. The mean ndcg@10 is 0.135 for the popularity baseline, 0.253 for item-based collaborative filtering, 0.233 for LTARS and 0.270 for the hybrid method. The mean recall@10 is 0.226 for the popularity baseline, 0.288 for item-based collaborative filtering, 0.301 for LTARS and 0.297 for the hybrid method. We omit the results for the content-based recommender having only a mean ndcg@10 of 0.019.

Dataset	**Popularity**	**ItemKNN**	**LTARS**	**Hybrid**
<u>ndcg@10:</u>				
Regional news	0.135	<u>0.253</u>	0.233	**0.270**
Foursquare TKY	0.034	0.077	<u>0.095</u>	**0.129**
Foursquare NYC	0.031	0.052	<u>0.084</u>	**0.106**
<u>recall@10:</u>				
Regional news	0.226	0.288	**0.301**	<u>0.297</u>
Foursquare TKY	<u>0.042</u>	**0.042**	0.036	0.040
Foursquare NYC	**0.043**	<u>0.041</u>	0.028	0.033

Table 3
Comparing top-*n* recommender systems on private and public datasets. The best result on each dataset are in bold (best) or underlined (second best).

Core i9 and 16 GB of RAM. The publicly available implementation is in Python. We remark, that concerning complexity, model training is $O(|I|^2)$ for item-based collaborative filtering and $O(|L|^2)$ for LTARS with geodesic nearby regions. At test time methods have a comparable cost, i.e. for LTARS we filter items on regional preference and compute the user-location preference and popularity-based scores for each item, while for item-based collaborative filtering we compute the dot product between the history and the (time-weighted) item-item similarity matrix. In the Regional news datasets we have 6 624 156 interactions, 456 578 users and 8 462 items in the first window, yet total training time is only 27.3*s*. For making predictions we require 14.1s for 22 921 test users, which is less then 1 *millisecond* per user on average. We conclude that LTARS is highly efficient and scalable to large datasets.

4.5. Online evaluation

To validate the findings of the proposed approach we performed an *online A/B trial* on two regional news websites in Belgium. The goal of the recommendations is to surface relevant regional articles for each user. Users have the option to explicitly specify one or more locations they are interested in, however only 1 in 4 users provide this preference. Finding the right regions to recommend articles from, therefore is an important problem for these websites.

During the trials users were randomly assigned to either the control or treatment group during the test period of 9 days. Both groups received an equal amount of users. The *control algorithm* recommends the most recent items from the explicit interest locations if available and otherwise recommends articles from each user's most read region. In the *experimental group* a user profile $P^k(u,t)$ is constructed with $k = 3$ following the LTARS method described in this work, ignoring the 1% most popular

Dataset	Popularity	ItemKNN	LTARS	Hybrid
runtime (s)				
Regional news	3.6s	141.7s	34.7s	182.4s
Foursquare TKY	0.4s	4.8s	2.4s	8.4s
Foursquare NYC	0.8s	14.0s	7.5s	25.3s

Table 4
Comparing the execution time of top-n recommender systems on private and public datasets.

items. If a user has given an explicit regional preference, these regions are always included in their regional profile. So a user that indicated explicit interest in two regions, will have a third region deduced from their historical interactions. For this group items are filtered on regional preferences and ranked according to recency.

A total of 1.7 million boxes were requested for 235k users on the first newspaper and 2.5 million boxes for 375k users on the second. We find that the experimental group has a 5.1% (relative) increase in click-through-rate on the first newspaper and an 11.4% increase on the second. Both results were statistically significant at the 99% confidence level. In addition to the CTR results, the regional profiles also cover more of the user's regions of interest, recommending them articles from more diverse regions.

4.6. Sensitivity of hyperparameters

In Table 5 we summarise the hyperparameters introduced by the proposed LTARS. We investigate the sensitivity of our model with respect to the most important parameters, α and β for giving more weight to location, recency or collaborative filtering. To clearly show the influence of these parameters, we report recall@10 with different parameter settings on two datasets.

In Figure 6 we plot the recall@10 for varying values of α and β between 0.0 and 1.0 while keeping other parameters fixed as discussed before (i.e. $\Delta t_{train} = 30d$, $\Delta t_{pop} = 12h$, $k_1 = 3$ and $c = 0$ for Regional news). On the Regional news dataset we find that the hand-picked values for α and β in the last experiment are sub-optimal. With a value of α set to 0.75 the recall@10 is 0.361 which increases to 0.378 (+ 4.6%) using the optimal value of $\alpha = 0.3$ and with a value of $\beta = 0.5$ the recall@10 is 0.348 which increases to 0.360 (+ 3.4%) using $\beta = 0.9$. A similar trend is visible in Foursquare TKY where also a local maximum is found with values of α and β in between. This suggests that we should optimise hyperparameters to further increase accuracy. We remark that in a dynamic environment where there is a potential drift in popularity bias, it make sense to adjust hyperparameters periodically, i.e. using the last batch (or window) of interactions for tuning. Note that optimising α and β is computationally inexpensive since we only have

Parameter	Description
Δt_{train}	training window
Δt_{test}	test window
Δt_{pop}	training window for computing popularity
k_1	top-k_1 locations with highest frequency
k_2	top-k_2 nearby locations in N_{GEO} or N_{CF}
c	bias term in $rank_{POP+REC}(i, t)$
ϵ	threshold when filtering candidate items on popularity
a, b	weights for exponential decay in time-aware item-based collaborative filtering
α	relative weight for ranking on regional preferences versus popularity and recency
β	relative weight for ranking on location and time versus collaborative filtering

Table 5
Overview of hyperparameters

to evaluate the weighted sum w.r.t. α and β using the cached partial scores for location, recency and collaborative filtering-based recommendations for each user, item pair.

5. Conclusion

In this paper, we tackle the important problem of optimising location- and time-aware recommendations. We propose techniques for determining regional preferences and neighbouring locations of interest. Additionally, we consider ranking functions that consider spatial, temporal and behavioural factors. We performed an extensive comparison offline using a realistically time-aware protocol based on sliding windows. Experiments show that the neighbourhood-based location- and time-aware recommendation system and hybrids thereof outperform popularity, content-based and time-aware collaborative filtering-based methods on a large regional news dataset and two public location-based social network datasets. Additionally, we performed an online A/B trial showing a clear increase in click-through-rate.

A limitation of our work is that the proposed model is straightforward and many of the proposed components consist of heuristics. We motivate this by the fact that for

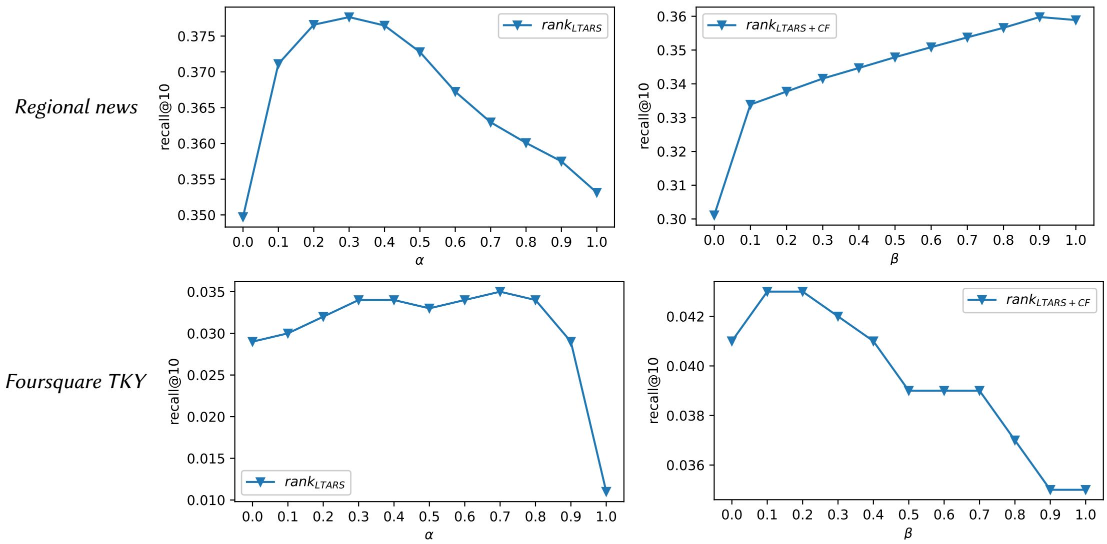

Figure 6: Recall@10 on the Regional news (up) and Foursquare Tokyo (bottom) datasets for varying hyperparameter. In the first experiment (left) we vary α between 0.0 (ranking only on $rank_{POP+REC}$) and 1.0 (ranking only on $rank_{LOC}$). In both datasets we observe a local maximum for α in between, advocating that ranking on both popularity/recency and user-location preferences is important. In the second experiment (right) we varying β between 0.0 (ranking only on collaborative filtering) and 1.0 (only on $rank_{LTARS}$). Again, we observe a local maximum for β in between, advocating that the hybrid method outperforms LTARS and CF-based recommendations.

location- and time-aware recommendation, the implicit interaction data is biased and extremely *sparse* where we have relatively few interactions specific to one period and location. Moreover, Dacrema et al. have recently shown that well-tuned simple baselines, such as ItemKNN, are difficult to beat when using more realistic evaluation strategies [6]. A second limitation is that the method is specific to applications where items are tagged with one or more locations and the volatility of items is crucial.

We find that the intrinsic simplicity and heuristic nature make our model efficient to compute online and the capacity to predict fresh and cold-start items. We conclude that the proposed algorithm is useful as an efficient and robust baseline when comparing future research in online location- and time-aware recommender systems. For future research, it would be of interest to update parameters dynamically, i.e. by selecting the best value of β and α based on the evaluation of those parameters on the previous period and adapt to the current temporal context, i.e. drift in the popularity distribution.

Acknowledgements

The authors would like to thank the VLAIO project on qualitative evaluation for online recommender systems for funding this research.

References

[1] Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer, 2011.

[2] Jie Bao, Yu Zheng, David Wilkie, and Mohamed Mokbel. Recommendations in location-based social networks: a survey. *GeoInformatica*, 19(3):525–565, 2015.

[3] Pedro G Campos, Fernando Díez, and Iván Cantador. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction*, 24(1):67–119, 2014.

[4] Cheng Chen, Xiangwu Meng, Zhenghua Xu, and Thomas Lukasiewicz. Location-aware personalized news recommendation with deep semantic analysis. *IEEE Access*, 5:1624–1638, 2017.

[5] Lisi Chen, Shuo Shang, Zhiwei Zhang, Xin Cao, Christian S Jensen, and Panos Kalnis. Location-aware top-k term publish/subscribe. In *2018 IEEE 34th international conference on data engineering (ICDE)*, pages 749–760. IEEE, 2018.

[6] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM conference on recommender systems*, pages 101–109, 2019.

[7] Shaojie Dai, Yanwei Yu, Hao Fan, and Junyu Dong. Spatio-temporal representation learning with social tie for personalized poi recommendation. *Data Science and Engineering*, 7(1):44–56, 2022.

[8] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*, pages 271–280, 2007.

[9] María del Carmen Rodríguez-Hernández and Sergio Ilarri. Ai-based mobile context-aware recommender systems from an information management perspective: Progress and directions. *Knowledge-Based Systems*, 215:106740, 2021.

[10] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1): 143–177, 2004.

[11] Len Feremans, Boris Cule, Celine Vens, and Bart Goethals. Combining instance and feature neighbours for extreme multi-label classification. *International Journal of Data Science and Analytics*, 10 (3):215–231, 2020.

[12] Huiji Gao, Jiliang Tang, and Huan Liu. Addressing the cold-start problem in location recommendation using geo-social correlations. *Data Mining and Knowledge Discovery*, 29(2):299–323, 2015.

[13] Florent Garcin, Boi Faltings, Olivier Donatsch, Ayar Alazzawi, Christophe Bruttin, and Amr Huber. Offline and online evaluation of news recommender systems at swissinfo. ch. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 169–176, 2014.

[14] Olivier Jeunen, Koen Verstrepen, and Bart Goethals. Efficient similarity computation for collaborative filtering in dynamic environments. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 251–259, 2019.

[15] Mozhgan Karimi, Dietmar Jannach, and Michael Jugovac. News recommender systems–survey and roads ahead. *Information Processing & Management*, 54(6):1203–1227, 2018.

[16] Sonal Linda and KK Bharadwaj. A genetic algorithm approach to context-aware recommendations based on spatio-temporal aspects. In *Integrated Intelligent Computing, Communication and Security*, pages 59–70. Springer, 2019.

[17] Nathan N Liu, Min Zhao, Evan Xiang, and Qiang Yang. Online evolutionary collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 95–102, 2010.

[18] Yiding Liu, Tuan-Anh Nguyen Pham, Gao Cong, and Quan Yuan. An experimental evaluation of point-of-interest recommendation in location-based social networks. *Proceedings of the VLDB Endowment*, 10(10):1010–1021, 2017.

[19] Lorenzo Massai, Paolo Nesi, and Gianni Pantaleo. Paval: A location-aware virtual personal assistant for retrieving geolocated points of interest and location-based services. *Engineering Applications of Artificial Intelligence*, 77:70–85, 2019.

[20] Yunseok Noh, Yong-Hwan Oh, and Seong-Bae Park. A location-based personalized news recommendation. In *2014 International Conference on Big Data and Smart Computing (BIGCOMP)*, pages 99–104. IEEE, 2014.

[21] Róbert Pálovics, Péter Szalai, Júlia Pap, Erzsébet Frigó, Levente Kocsis, and András A Benczúr. Location-aware online learning for top-k recommendation. *Pervasive and mobile computing*, 38: 490–504, 2017.

[22] Shaina Raza and Chen Ding. News recommender system: a review of recent progress, challenges, and opportunities. *Artificial Intelligence Review*, pages 1–52, 2021.

[23] Teresa Scheidt and Joeran Beel. Time-dependent evaluation of recommender systems. In *Perspectives@ RecSys*, 2021.

[24] Jeong-Woo Son, A-Yeong Kim, and Seong-Bae Park. A location-based news article recommendation with explicit localized semantic analysis. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 293–302, 2013.

[25] Waldo R Tobler. A computer movie simulating urban growth in the detroit region. *Economic geography*, 46(sup1):234–240, 1970.

[26] Stephan Tulkens, Chris Emmery, and Walter Daelemans. Evaluating unsupervised dutch word embeddings as a linguistic resource. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016. European Language Resources Association (ELRA). ISBN 978-2-9517408-9-1.

[27] Dingqi Yang, Daqing Zhang, Vincent W Zheng, and Zhiyong Yu. Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(1):129–142, 2014.

[28] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 325–334, 2011.