A Location- and Diversity-aware News Feed System for Mobile Users

Wenjian Xu, Chi-Yin Chow, *Member, IEEE*

Abstract—A location-aware news feed (LANF) system generates news feeds for a mobile user based on her spatial preference (i.e., her current location and future locations) and non-spatial preference (i.e., her interest). Existing LANF systems simply send the most *relevant* geo-tagged messages to their users. Unfortunately, the major limitation of such an existing approach is that, a news feed may contain messages related to the same location (i.e., point-of-interest) or the same category of locations (e.g., food, entertainment or sport). We argue that *diversity* is a very important feature for location-aware news feeds because it helps users discover new places and activities. In this paper, we propose D-MobiFeed; a new LANF system enables a user to specify the minimum number of message categories (l) for the messages in a news feed. In D-MobiFeed, our objective is to efficiently schedule news feeds for a mobile user at her current and predicted locations, such that (i) each news feed contains messages belonging to at least l different categories, and (ii) their total relevance to the user is maximized. To achieve this objective, we formulate the problem into two parts, namely, a decision problem and an optimization problem. For the decision problem, we provide an exact solution by modeling it as a maximum flow problem and proving its correctness. The optimization problem is solved by our proposed three-stage heuristic algorithm. We evaluate the performance of D-MobiFeed using a real data set crawled from Foursquare. Experimental results show that our proposed three-stage heuristic scheduling algorithm outperforms the brute-force optimal algorithm by at least an order of magnitude in terms of running time and the relative error incurred by the heuristic algorithm is below 1%. D-MobiFeed with the location prediction method effectively improves the relevance, diversity, and efficiency of news feeds.

Index Terms—Location-aware news feeds, diversity constraint, online scheduling, location-based services, user mobility

1 Introduction

With the advance of wireless communications and the ubiquity of GPS-equipped smartphones, social network applications have become more prevalent and location-aware, as widely known as location-based social networks (LBSNs) (e.g., Facebook Places [17] and Foursquare [19]). A *news feed* is a common functionality of existing LBSNs. It enables mobile users to post geo-tagged messages and receive nearby user-generated messages as news feeds at anytime, anywhere. For example, "*Bob can receive a news feed with 3 messages that are most relevant to him among the messages within 1 km from his location every 10 seconds*". Figure 1a depicts an application scenario. The geo-location of a message could be a point (e.g., m_4), a circular region (e.g., m_5), or the spatial region of a venue (e.g., m_6 and m_7 are spatially associated with restaurant R_1). Besides, geo-tagged messages can be categorized by their underlying venues; for instance, m_6 and m_7 are posted from users at restaurant R_1, so they are intuitively categorized to a "restaurant" category.

Our previous work developed MobiFeed [39]; the state-of-the-art location-aware news feed system schedules news feeds for mobile users. In MobiFeed, the relevance of a message m to Bob is measured by both the content similarity between m and Bob's submitted messages (i.e., a non-spatial factor) and the distance between m and Bob (i.e., a spatial factor). MobiFeed is motivated by the fact that, if the news feeds are only computed based on a user's location at the query time (i.e., it does not consider the user's future locations, e.g., GeoFeed [7]), the overall relevance of news feeds is not optimized. For example, in Fig. 1a, there are 11 messages (i.e., m_1 to m_{11}) with their geo-location intersecting Bob's query regions (i.e., circular regions in Fig. 1a) at time t_0, t_1, and/or t_2. Assume m_i is more relevant to Bob than m_j if $i < j$, and the number of messages per news feed (i.e., k) is 3. GeoFeed returns (m_1, m_2, m_3) at t_0, (m_4, m_6, m_7) at t_1, and (m_5) at t_2. To improve the relevance of news feeds, given Bob's current location at t_0, MobiFeed predicts two future locations for him at t_1 and t_2, and *schedules* news feeds by considering all three query regions at the same time, which results in a better solution with (m_1, m_2, m_3), (m_4, m_8, m_9), and (m_5, m_6, m_7) at t_0, t_1 and t_2, respectively. In summary, MobiFeed aims at maximizing the *total relevance* of generated news feeds by utilizing *location prediction* techniques.

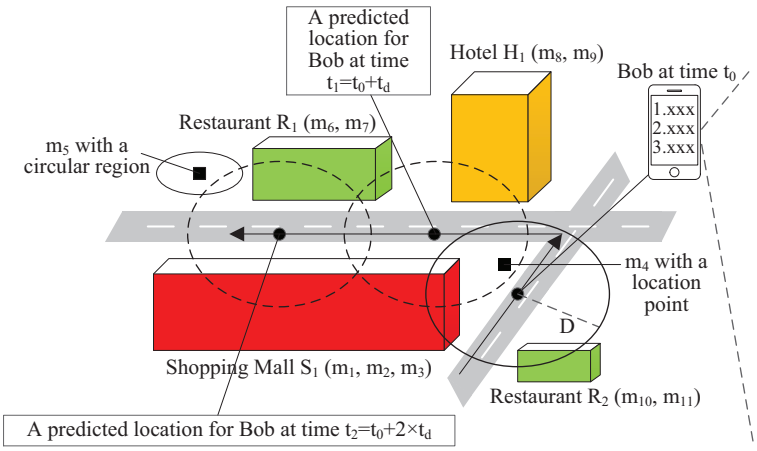
Unfortunately, relevance alone is unable to capture the broader aspects of user satisfaction. Although users expect to receive messages that are highly relevant to their interests, they may prefer a location-aware news feed with a certain level of diversity (i.e., the messages in a news feed belong to a certain number of categories). In conventional web search or recommender systems, topic diversification is a key method to improve user satisfaction [2], [3], [37], [42]. This work considers a mobile environment that makes our location- and diversity-aware news feed system unique and more challenging. With the geographical distance between a message and a mobile user in a relevance measure model, the relevance of a message to a mobile user is changing as the user is moving. Such a dynamic environment gives us an opportunity to employ location prediction technique to improve the quality of news feeds and the system efficiency.

To show the limitation in our previous model MobiFeed, we have conducted experiments to investigate the diversity of its news feeds generated for mobile users. Experimental results show that,

• *W. Xu and C.-Y. Chow are with the Department of Computer Science, City University of Hong Kong, Hong Kong.*
E-mail: wenjianxu2-c@my.cityu.edu.hk, chiychow@cityu.edu.hk

Category information

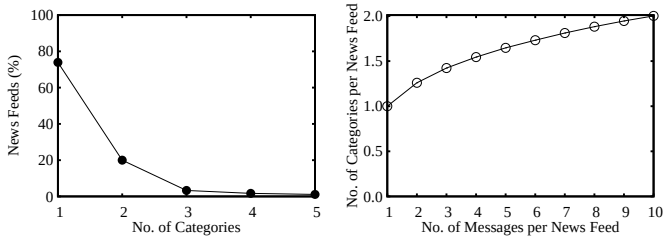
Message	Category
m_1	Shopping
m_2	Shopping
m_3	Shopping
m_4	Cinema
m_5	Museum
m_6	Restaurant
m_7	Restaurant
m_8	Hotel
m_9	Hotel
m_{10}	Restaurant
m_{11}	Restaurant



Message	Category	Message content
m_1	Shopping	Blanches, the cafe on the top floor, is great for lunch or a quick snack while shopping
m_2	Shopping	I just love their sales. I did my Xmas shopping there & only spent $50.
m_3	Shopping	This is one of our favorite shopping destinations in NYC. They also have our favorite windows during the holidays. And they have the nicest public restrooms.

(a) (b)

Fig. 1: (a) An application scenario. (b) The news feed at t_0 generated by MobiFeed.



(a) Distribution of news feeds with different numbers of categories (k = 5)

(b) The number of categories in news feeds with various k

Fig. 2: Diversity of news feeds generated by MobiFeed.

when $k = 5$, over 75% news feeds contain messages belonging to one category and about 20% of news feeds are related to two categories; over 95% of news feeds are related to only one or two categories, as depicted in Figure 2a. Even if k is increased to 10, the diversity of the news feeds generated by MobiFeed is still very low (i.e., the average number of categories is two), as shown in Figure 2b.

To this end, we propose D-MobiFeed, a framework that takes both the relevance and diversity of news feeds into account when scheduling news feeds for moving users. In particular, we add an l-*diversity* constraint into MobiFeed; this constraint requires that messages in a news feed belong to at least l distinct message categories. This l-diversity constraint brings a brand new challenge to D-MobiFeed, i.e., *the trade-off between relevance and diversity* of news feeds. Consider the same example in Fig. 1a, the category information table depicts the category of each message (e.g., m_1, m_2, and m_3 belong to the "shopping" category and m_4 belongs to the "cinema" category). With a 3-diversity constraint, D-MobiFeed returns a solution with (m_1, m_4, m_{10}), (m_2, m_6, m_8), and (m_3, m_5, m_7) at t_0, t_1 and t_2, respectively. Compared to MobiFeed, the relevance of the news feeds generated by D-MobiFeed is slightly degraded. Despite that, users may still prefer the news feeds with a higher level of diversity, as shown in the result of user studies performed by Ziegler et al. [42]. As a result, the challenge of D-MobiFeed is: *how to effectively schedule location- and diversity-aware news feeds for mobile users while maintaining their high relevance to the users at the same time?*

In this paper, we aim at designing D-MobiFeed by further considering the diversity in news feeds to address this challenge. Specifically, D-MobiFeed is composed of four key functions: *location prediction*, *relevance measure*, l-*diversity constraint checker* and *news feed scheduler*. As shown in Figure 1a, given a user u's location at current time t_0, minimum message display time t_d, range distance D, the requested number of messages per news feed k, the minimum number of categories l, and a system-specified look-ahead steps n, the *location prediction* function estimates n future locations for u at times $t_1 = t_0 + 1 \times t_d$, $t_2 = t_0 + 2 \times t_d$, ..., and $t_n = t_0 + n \times t_d$, the *relevance measure* function computes the relevance score of each candidate message with a geo-location intersecting any u's query region (i.e., circular regions in Figure 1a). After that, the l-*diversity constraint checker* decides whether the system could compute news feeds from the candidate messages for u's query regions at t_0, t_1, ..., t_n, such that messages in each news feed belong to at least l categories. This l-*diversity* constraint checking is referred to as a *decision problem*. Finally, the *news feed scheduler* generates $n + 1$ news feeds satisfying the l-diversity constraint, and their total relevance score is maximized. The problem of maximizing the total relevance score is referred to as an *optimization problem*. The computed $n + 1$ news feeds are sent to u. u's mobile device immediately displays the first news feed (i.e., the one generated for the query region at t_0), and then displays each of the remaining news feeds one by one for every t_d.

To the best of our knowledge, this is the *first study* to incorporate both relevance and diversity for scheduling location-aware news feeds for mobile users in LBSNs. In general, the key contributions of this work can be summarized as follows:

- We extend our previous model MobiFeed (i.e., the state-of-the-art location-aware news feed system) to consider both the relevance and diversity of news feeds when generating news feeds for mobile users.
- We model the *decision problem* as a maximum flow problem to find the minimum total diversity of a set of $n + 1$ news feeds for a user based on the user-specified l-diversity constraint. (Section 5)
- We propose a three-stage heuristic approach to solve the *optimization problem*. The first stage solves a minimum cost flow problem to guarantee the minimum total diversity in a set of $n + 1$ news feeds. The second stage addresses a replenish-up-to-k problem to maximize the total relevance of these news feeds. The last stage simply sorts the

messages in each news feed. (Section 6)
- We conduct extensive experiments to evaluate the performance of D-MobiFeed using a real LBSN data set crawled from Foursquare in terms of relevance, diversity, and efficiency. (Section 7)

The rest of the paper is organized as follows. We highlight related work in Section 2. We describe the system model of D-MobiFeed in Section 3. Section 4 gives an overview of D-MobiFeed. In Sections 5 and 6, we present our solutions for the decision problem and the optimization problem, respectively. Section 7 evaluates the performance of D-MobiFeed through extensive experiments. Finally, we conclude this paper in Section 8.

2 Related Work

In this section, we highlight the state-of-the-art techniques in location-aware news feed systems and existing diversity models in recommender systems and web search systems.

Location-aware news feed systems. Most existing news feed systems only provide publish/subscribe services that simply forward messages to subscribed users [10], [35]. Bao et al. [7] injected the location-awareness into a news feed system, which enables a message to be associated with a spatial extent to control where users can receive it. We proposed a framework MobiFeed [39] that is designed to schedule news feeds for mobile users. MobiFeed takes the limitations of mobile devices and the user's preferences into account, and schedules the most *relevant* geo-tagged messages to mobile users. Unfortunately, MobiFeed has a major limitation that only considers the relevance of messages to users, so a news feed may contain messages related to the same category; and thus it would impede users to discover new places and activities. In conventional web search/recommender systems, topic diversification is a key method to improve user satisfaction [2], [3], [37], [42]. To address this limitation, our D-MobiFeed framework allows users to specify their required levels of diversity of news feeds in terms of the number of message categories (i.e., the l-diversity constraint). D-MobiFeed aims at maximizing the total relevance of news feeds and satisfying the condition that each news feed contains messages belonging to at least l categories.

l-diversity principle for privacy-preserving data publishing. The l-diversity principle [28] is proposed for privacy-preserving data publishing. Basically, this principle is used to generalize non-sensitive attributes (e.g., zip codes 13053 and 13068 are generalized to "130**" and ages 28, 29, and 21 are generalized to "< 30") in a class of records such that the sensitive attribute achieves the l-diversity constraint, in order to protect the privacy of published data. The entropy l-diversity is further used to defend against the homogeneity problem without considering the role of background knowledge, i.e., entropy increases as frequencies of sensitive attributes become more uniform. In this work, we focus on a different problem because D-MobiFeed aims to maximize the relevance of news feeds for mobile users while news feeds satisfy the l-diversity constraint (i.e., the messages in each news feed belong to at least l categories).

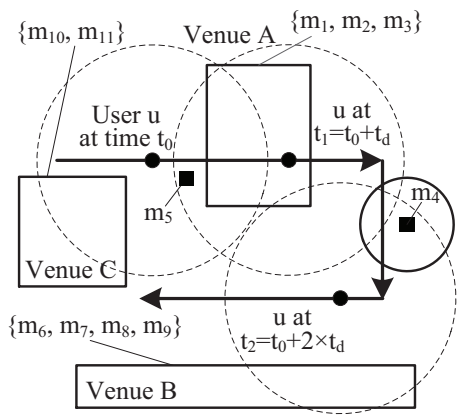
Diversity-aware recommender systems. In MobiFeed [39], the only metric used to evaluate its quality as a recommender system is the relevance of messages to users (i.e., accuracy). However, it is argued in [30] that, developing recommender systems with accuracy as the single goal has many drawbacks, and the recommender community should move beyond the conventional accuracy metrics. One promising direction that has drawn recent interest is to diversify the recommendation lists [4]. Ziegler et al. [42] proposed an *intra-list similarity* metric to measure the overall diversity of a recommendation list, where the similarity between products is derived from their taxonomy-based categorization. The authors employed a heuristic algorithm to increase the diversity of a recommendation list, and their user study results show that in spite of the loss in accuracy, users still prefer the recommended items with larger extent of diversity. Zhang et al. [41] addressed the diversification problem as the joint optimization of two objective functions (i.e., the relevance and diversity of a recommendation list), which is solved by using binary quadratic programming algorithms.

Diversity-aware web search systems. The process of web search systems differs from that of recommender systems since it involves an explicit user query (i.e., keywords). The query, however, is also ambiguous and has more than one interpretation [34]. One possible way to address this problem is to produce a set of diversified results that cover different interpretations of the target query. Specifically, the search result diversification approaches in the literature can be classified as either *implicit* or *explicit*. Implicit approaches [8], [40] assume that similar documents will cover similar aspects of a query. Their basic idea is to iteratively select documents which are similar to the query but different to the already selected ones in terms of vocabulary [8] or divergence in language models [40]. Explicit approaches [5], [9], on the other hand, model aspects of a query in an explicit approach. For example, Agrawal et al. [5] assumed that there exists a classification taxonomy over queries and documents to represent user intentions, and they proposed a diversification function that maximizes the probability of finding at least one relevant document in the top-k positions. Similarly, Carterette and Chandar [9] modeled the aspects of a query as topics extracted from the top ranked documents, and they designed a probabilistic method to maximize the coverage of the retrieved documents.

The above-mentioned diversity-aware recommender systems and web search systems focus on retrieving an individual list of items with a certain level of diversity, in order to improve user satisfaction. In this work, we focus on a mobile environment, where mobile users are moving in a road network. Our problem is unique and more challenging as D-MobiFeed considers the geographical distance factor between messages and mobile users in the relevance measure model, and thus, the relevance of messages to users could be changing as they are moving. In addition, D-MobiFeed has an opportunity to employ a location prediction technique to improve the quality of news feeds by scheduling multiple (i.e., $n + 1$, where n is a look-ahead step) location- and diversity-aware news feeds for mobile users simultaneously. The main reason is that computing each news feed individually as in the web search or recommender systems will not maximize the total relevance of news feeds for a user. In our experimental results, as depicted in Section 7, D-MobiFeed with $n = 0$ generating a news feed at a time performs worse than D-MobiFeed with $n > 0$ computing a set of n news feeds simultaneously, in terms of relevance, diversity, and efficiency.

3 System Model

In this section, we present the system model of D-MobiFeed.



(a)

Table I. Candidate messages at t_0

Message	Category	Relevance score to u
m_1	Restaurant	0.58
m_2	Restaurant	0.58
m_3	Restaurant	0.33
m_5	Theater	0.53
m_{10}	Museum	0.55
m_{11}	Museum	0.5

Table II. Candidate messages at t_1

Message	Category	Relevance score to u
m_1	Restaurant	0.68
m_2	Restaurant	0.68
m_3	Restaurant	0.43
m_4	Stadium	0.4
m_5	Theater	0.15

Table III. Candidate messages at t_2

Message	Category	Relevance score to u
m_1	Restaurant	0.35
m_2	Restaurant	0.35
m_3	Restaurant	0.1
m_4	Stadium	0.53
m_6	Shopping	0.3
m_7	Shopping	0.4
m_8	Shopping	0.5
m_9	Shopping	0.5

(b)

Fig. 3: (a) An example. (b) Three sets of candidate messages associated with their category and relevance score to u.

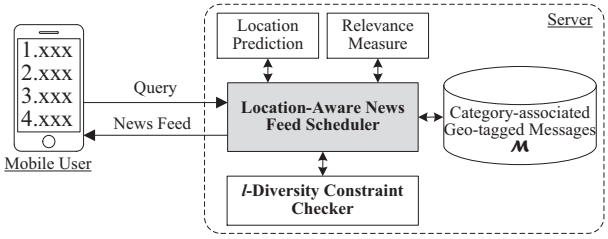


Fig. 4: System architecture of D-MobiFeed.

3.1 System Architecture

Figure 4 depicts the system architecture of D-MobiFeed, which is designed based on the framework in [39]. D-MobiFeed consists of two major entities.

Category-associated geo-tagged messages. We use $\mathcal{M}$ to denote the message collection in D-MobiFeed. Each message $m_j \in \mathcal{M}$ is defined as a tuple (*MessageID*, *SenderID*, *Content*, *Timestamp*, *Spatial*, *Category*), where *MessageID* is a message identifier, *SenderID* is its sender's identifier, *Content* is its content, *Timestamp* is its post time, *Spatial* is its spatial extent, and *Category* is its category. D-MobiFeed supports three types of spatial extent, namely, points, regions, and venues. As our running example depicted in Figure 3a, m_4 is geo-tagged by a circular region, m_5 is geo-tagged by a point location, and $\{m_1, m_2, m_3\}$, $\{m_6, m_7, m_8, m_9\}$, and $\{m_{10}, m_{11}\}$ are geo-tagged with venues A, B, and C (represented by rectangles), respectively. In D-MobiFeed, each message is associated with exactly *one* category, and set $\mathcal{C} = \{c_1, c_2, \ldots, c_h\}$ denotes all categories. If the explicit category of messages is not available, clustering methods can be applied to assign a category for each message [38]. Figure 3b shows the category for each message in our running example.

System users. In D-MobiFeed, a mobile user u equipped with a GPS-enabled mobile device can post a new message tagged with a spatial extent. A location- and diversity-aware news feed query consists of four parameters: (1) the number of messages in a news feed (k), (2) the minimum number of categories for the messages in a news feed (l), (3) the message display time for a news feed (t_d), and (4) a query range distance (D). The user is able to specify these four query parameters based on his/her preferences. In practice, the system could provide default values for these query parameters. For example, the simplest way is to set these parameters to the most common values or the average values. In other words, the user is able to receive at most $u.k$ messages within her specified range distance $u.D$ (i.e., the query region of a news feed) as a news feed. D-MobiFeed computes a news feed for u by selecting messages based on their category, their relevance to u and u's movement. Since the user needs some time to read the messages, each news feed will be displayed on u's mobile device for a time period $u.t_d$. Note that each message can be displayed to a user only once. Assume the look-ahead step is n, u reports its location to the server at every time period $(n+1) \times u.t_d$. After receiving u's location update, $n+1$ news feeds are computed for u. u's mobile device immediately displays the first news feed, and then displays each of the remaining news feeds one by one for every $u.t_d$.

3.2 Location Prediction and Relevance Measure

In this subsection, we describe the *location prediction* and *relevance measure* functions in D-MobiFeed.

3.2.1 Location Prediction

The *location prediction* function can employ any existing location prediction algorithm if it can predict a user's location at a specified future time in a road network. In the experiments (Section 7), we incorporate the path prediction algorithm [26] into D-MobiFeed. Given a user u's current location, u's historical trajectories, the road map, and a future time t, the path prediction algorithm estimates u's location at t. For the technical detail of the prediction algorithm employed in D-MobiFeed, please refer to [26].

3.2.2 Message Relevance Measure

D-MobiFeed only requires the *relevance measure* function to return a score to indicate the relevance of a message m_j to a user u_i, i.e., *relevanceScore*(u_i, m_j). We combine the following non-spatial and spatial factors to implement the *relevance measure* function.

Message contents. The user may be more interested in messages that are similar to his or her submitted ones (e.g., a user's common keywords reflect his or her interests [33]). For example, a user issued a message "I like spicy food" would be happy to receive messages about Thai restaurants. Therefore, we use vector

space model [29] to measure the relevance of a message to a particular user in terms of content similarity. Specifically, let $\mathcal{T}$ denotes the term set of the message set $\mathcal{M}$, and each message m_j is represented as a vector of weights of all terms in $\mathcal{T}$, i.e., $m_j.V = \langle w_{j1}, w_{j2}, \ldots, w_{j|\mathcal{T}|} \rangle$. In general, a term $T_k \in \mathcal{T}$ should be weighted higher for m_j if T_k occurs more frequently in m_j and occurs rarely in other messages in $\mathcal{M}$. The weight can be computed by the $TF \times IDF$ scheme: $w_{jk} = tf_{jk} \cdot \log \frac{|\mathcal{M}|}{df_k}$, where tf_{jk} is the *term frequency* of T_k in m_j and df_k is the *document frequency* of T_k in $\mathcal{M}$. To incorporate the vector space model into D-MobiFeed, we maintain a query vector for each user based on her submitted messages. Given a querying user u_i with a query vector $u_i.V$ and a message m_j, we use the cosine similarity to compute *contentScore*$(u_i, m_j) = \frac{\sum_{k=1}^{|\mathcal{T}|} w_{ik} \cdot w_{jk}}{\sqrt{\sum_{k=1}^{|\mathcal{T}|} w_{ik}^2} \cdot \sqrt{\sum_{k=1}^{|\mathcal{T}|} w_{jk}^2}}$, where $w_{ik} \in u_i.V$ and $w_{jk} \in m_j.V$.

Distance. We argue that the geographical proximities between users and messages have a significant influence on the relevance measure. In our scenario, the relevance of a message m_j to a user u_i can be measured by their distance, i.e., *Dist*(u_i, m_j). If m_j is associated with a circular region or a venue, *Dist*(u_i, m_j) returns the *minimum* distance between u_i and the spatial extent of m_j. To accommodate the difference in the value ranges of $Dist(u_i, m_j)$ and other relevance measures, we normalize $Dist(u_i, m_j)$ to be from zero to one, i.e., *NDist*$(u_i, m_j) = 1 - \frac{Dist(u_i, m_j)}{u.D}$, where $u.D$ is the query range distance of a news feed.

Relevance measure function. We employ a linear combination method to integrate the aforementioned two factors into the *relevance measure* function [11]. Specifically, we have:

$$\begin{aligned} \textit{relevanceScore}(u_i, m_j) =& \textit{contentScore}(u_i, m_j) \times (1 - \beta) \\ &+ \textit{NDist}(u_i, m_j) \times \beta \end{aligned} \quad (1)$$

where $0 \leq \beta \leq 1$ and β is a parameter that gives a weight for the importance of the distance factor with respect to the content similarity factor. Thus, we ensure that *relevanceScore*(u_i, m_j) is between zero and one. In practice, as the D-MobiFeed model depicted in Figure 4, the relevance measure function is separated from the core location-aware news feed scheduler. Any relevance measure function which returns a numeric score as an output can be used in D-MobiFeed.

Figure 3b directly gives the computed relevance score of each message for the user. The detailed score computation of this example can be found in [39]. Note that since we consider the distance between a message and a user as one of the factors in the relevance measure, the relevance of a message to a user could vary at different locations. In our example, *relevanceScore*$(u, m_1) = 0.58$ at t_0 while *relevanceScore*$(u, m_1) = 0.68$ at t_1.

4 System Overview

In this section, we give a system overview of D-MobiFeed by defining key problems in each of its three major steps, namely, *candidate message step*, *decision step*, and *scheduling step*, as depicted in Figure 5, and showing how these steps interact with the four key functions, i.e., *location prediction*, *relevance measure*, *l-diversity constraint checker*, and *news feed scheduler*.

Candidate message step. Given a user *u*'s location at current time t_0, *u*'s specified minimum message display time t_d, *u*'s required range distance D, *u*'s requested number of messages per news feed k, *u*'s specified minimum number of categories l, the *location prediction* function (see Section 3.2.1) returns n future locations at times t_1, t_2, ...t_n for u, where $t_i = t_0 + i \times t_d$. Then the *news feed scheduler* generates $n+1$ query regions centered at each location (i.e., one reported location and *n* predicted locations). For each query region at t_i, a range query is issued to retrieve a set of candidate messages *CandidateMsg*$_i$ with their spatial extent intersecting the query region. After that, the *relevance measure* function (see Section 3.2.2) calculates a relevance score for each message in *CandidateMsg*$_i$ to u.

Decision step. After the *candidate message step*, we have $n+1$ sets of candidate messages associated with their category and relevance score to u (e.g., Figure 3b for our running example). The *l-diversity constraint checker* now decides whether we could generate $n+1$ news feeds from the candidate message sets, under the constraint that messages in each news feed belong to at least l categories. This decision problem is defined as follows:

Definition 1. *l-Diversity Constraint Checking (DCC) Problem: Given a user u's news feed query and a look-ahead step n, D-MobiFeed decides whether it could schedule at most k messages for each of $n+1$ news feeds, such that messages in each news feed belong to at least l categories.*

The details of how to address this decision problem will be described in Section 5. If the result of DCC is positive (i.e., each news feed of a set of $n+1$ news feeds can satisfy the l-diversity constraint), D-MobiFeed proceeds to perform the scheduling step. Otherwise, the *l-diversity constraint checker* returns a value of γ for the *news feed scheduler*, where γ is the minimum total diversity of a set of $n+1$ news feeds.

Definition 2. *Minimum Total Diversity (γ): Given a set of $n+1$ news feeds and an l-diversity constraint, the minimum total diversity (γ) is the sum of (1) the maximum number of categories (which is less than l) of a news feed that cannot satisfy the l-diversity constraint and (2) the minimum number of categories (which is equal to l) of a news feed that can satisfy the l-diversity constraint. If every news feed can satisfy the l-diversity constraint, $\gamma = l \times (n+1)$. However, if at least one news feed cannot satisfy the l-diversity constraint, $\gamma < l \times (n+1)$.*

Scheduling step. The *news feed scheduler* computes $n+1$ news feeds that satisfy the *minimum total diversity* (γ) and have the *maximum total relevance score*. D-MobiFeed supports different weights for different positions in a news feed result list, i.e., a higher weight is given to a message displayed at a higher position because it would be easier to draw a user's attention [39]. Specifically, given a result list with k positions, the weight of the first position is k, the weight of the second position is $k-1$, and so on. In general, the weight of a message m_j at the j-th position ($1 \leq j \leq k$) is *displayWeight*$(j, k) = k - (j-1)$. Thus, the relevance score of a news feed f with k messages m_1, m_2, ..., m_k is calculated as:

$$\begin{aligned} \textit{relevanceScore}(f) =& \sum_{j=1}^{k} \textit{relevanceScore}(u, m_j) \\ &\times \textit{displayWeight}(j, k), \end{aligned} \quad (2)$$

where the sum of the relevance scores of $n+1$ news feeds is the total relevance score of a query answer. Maximizing such a total relevance score is an optimization problem that is defined as follows:

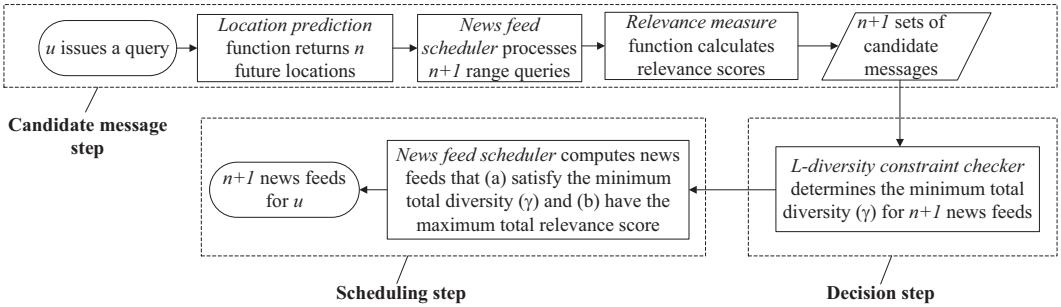


Fig. 5: The work flow of D-MobiFeed.

Definition 3. *l-Diversity Constrained News Feed Scheduling (DCS) Problem: Given a user u's news feed query and a look-ahead step n, D-MobiFeed schedules at most k messages for each of $n+1$ news feeds, such that ($\mathcal{O}1$) messages in each news feed belong to at least l categories, and ($\mathcal{O}2$) the total relevance score of the generated news feeds is maximized.*

The details of this problem will be discussed in Section 6. Consider our running example in Figure 3a, where $n = 2$, $k = 4$, and $l = 3$. Note that when we discuss these two problems (Definitions 1 and 3) using this example in the next two sections, we assume that we have already finished the *candidate message step* and got three sets of candidate messages along with their category and relevance score to the user (Figure 3b).

Finally, we discuss how the *news feed scheduler* deals with an inaccurate predicted location. A tolerance threshold θ is defined for the difference between a user's actual location and its corresponding predicted location for a query region. If the location deviation is larger than θ, a prediction error occurs, and u reports its actual location to the server to retrieve a new set of news feeds. For example, for a trajectory with a user's locations at times t_0, $t_1 = t_0 + t_d$, $t_2 = t_0 + 2 \times t_d$, ..., where t_d is the minimum display time, a first call of 2-look-ahead scheduling at t_0 generates news feeds for t_0, t_1, and t_2. If no prediction error occurs from t_0 to t_2, a second call of scheduling will take place at t_3. In contrast, if a prediction error occurs at t_1, our schedule will perform re-scheduling at t_1 and re-generate news feeds for t_1, t_2, and t_3. To reduce communication overhead, we determine (i) a set of new messages that are in a newly computed set of news feeds, but not in a previous set of news feeds, and (ii) positive or negative updates indicate that a certain message should be added to or removed from the previous result lists, respectively [31]. D-MobiFeed only sends the set of new messages, positive and negative updates to the user. In practice, θ can be set to the accuracy of the underlying positioning technique and device. For example, if assisted-GPS is used, θ is set to 50 meters [15].

5 *l*-Diversity Constraint Checking

The l-Diversity Constraint Checking (DCC) problem is non-trivial. Using a brute-force method to find an exact solution has to try all possible combinations of news feeds for $n + 1$ news feeds. Such a brute-force method is too costly for our online scheduling problem. To this end, we model the DCC problem as a maximum flow problem and prove the exactness and correctness of the model. Finally, the DCC problem returns the minimum total diversity γ (Definition 2) as an input for the scheduling step (see Fig. 5). In the next section (i.e., Section 6), we will describe how the scheduling step computes a set of news feeds that satisfy the minimum total diversity and have the maximum total relevance score.

5.1 Maximum Flow Model

The DCC problem is equivalent to the following reduced problem in terms of the result (i.e., given the same input, if the original problem has a positive (respectively, negative) answer, the reduced problem also has a positive (respectively, negative) answer, and vice versa):

Definition 4. *Reduced l-Diversity Constraint Checking (RDCC) Problem: Given $n + 1$ sets of candidate messages for u's query regions at t_0, t_1, t_2, ..., and t_n (i.e., CandidateMsg$_i$, where $0 \leq i \leq n$), along with their category, D-MobiFeed decides whether it could schedule $n + 1$ news feeds such that each news feed contains exactly l messages, and each message belongs to a distinct category.*

The RDCC problem can be modeled as the *maximum flow problem* [6]. Consider our example in Figure 3, where $n = 2$, $l = 3$, and three sets of candidate messages are available for u's query region at t_0, t_1, and t_2. This RDCC problem is represented by a *flow graph* depicted in Figure 6. The flow graph consists of three sets of nodes:

- N_r represents $n + 1$ news feeds, where n is the number of look-ahead steps. In our example, $N_r = \{r_0, r_1, r_2\}$.
- N_c is divided into $n + 1$ groups, and each group is composed of all the categories for messages in corresponding candidate message set. In our example, '*Restaurant*' (Re), '*Theater*' (Th), and '*Museum* (Mu)' are three nodes in the first group of N_c since they are the categories for messages in *CandidateMsg*$_0$.
- N_m stands for the set of *distinct* messages in all *CandidateMsg*$_i$, where $0 \leq i \leq n$. There are eleven nodes (m_1 to m_{11}) belonging to N_m in our example.

Furthermore, we add two extra nodes, namely, the source s and the sink t, in the flow graph. Let V be the set of nodes in the graph, i.e., $V = N_r \cup N_c \cup N_m \cup \{s, t\}$. Each node $v \in V$ has a *balance* $b(v)$ [6]. For every $r_i \in N_r$, $c_j \in N_c$ and $m_k \in N_m$, the balance is set to 0. For s and t, $b(s) = \gamma$ and $b(t) = -\gamma$, where γ is the *value of flow* we want to maximize. In our example, the balances are shown on the top of each set of nodes in Figure 6. Let E be the set of edges in the flow graph. Each edge $e(v_i, v_j) \in E$ is associated with a *capacity* $c(v_i, v_j)$. The set of edges E comprises: (1) an edge $e(s, r_i)$ for every news feed $r_i \in N_r$, with capacity l (e.g., $l = 3$ in our example), (2) an edge $e(r_i, c_j)$ between every news feed $r_i \in N_r$ and the categories of messages

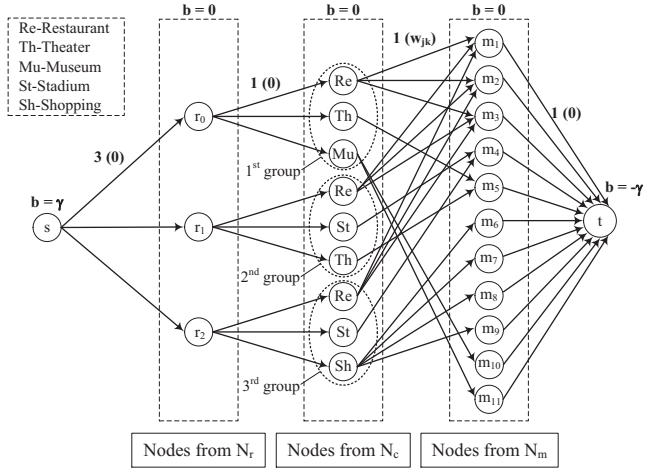


Fig. 6: Flow graph representation of the RDCC problem (Definition 4) and the γ-Selection problem (Definition 5). All the edges in the graph have capacity 1 except the edges between source node s and nodes from N_r. The value in parentheses indicates the cost of edges for the representation of the γ-Selection problem.

in *CandidateMsg*$_i$ in N_c, with capacity 1 (e.g., in our example, r_0 links to the nodes representing '*Restaurant*', '*Theater*', and '*Museum*' in the first group of N_c), (3) an edge $e(c_j, m_k)$ for every category $c_j \in N_c$ (c_j is assumed in $(i+1)$-th group of N_c) linking to message $m_k \in$ *CandidateMsg*$_i$ which belongs to category c_j, with capacity 1 (e.g., in Figure 6, for the category '*Restaurant*' in the first group of N_c, it connects to m_1, m_2, and m_3), and (4) an edge $e(m_k, t)$ for every message $m_k \in N_m$, with capacity 1. In Figure 6, the label on the top of each set of edges indicates their edge capacity.

Given the above graph, the *maximum flow problem* is to assign an integer *flow value* $x(v_i, v_j) \in [0, c(v_i, v_j)]$ for each edge $e(v_i, v_j) \in E$ such that for every node $v \in V$, the following *flow conservation property* holds:

$$\sum_{e(v,v_i)\in E} x(v, v_i) - \sum_{e(v_i,v)\in E} x(v_i, v) = b(v), \qquad (3)$$

and the value of the flow (i.e., γ) is maximized, that is, we want to route as much flow as possible from s to t.

Solution. The maximum flow problem and its applications are extensively discussed in [18] and [16]. Apart from the classical algorithms, many others, such as binary blocking flow algorithm [22] and push-relabel method [23], have been proposed to efficiently solve the maximum flow problem. We employ the relabel-to-front algorithm [12], which is a particular implementation of the push-relabel method. This algorithm runs in time $O(V^3)$, where V is the number of nodes in the flow graph.

5.2 Correctness Proof

After solving the above maximum flow problem, we obtain the *optimal* value of flow (i.e., γ). We interpret this value as follows[1]:

Theorem 1. *(i) If $\gamma = (n+1) \times l$, l-diversity constraint can be satisfied for a news feed query.*
(ii) If $\gamma < (n+1) \times l$, l-diversity constraint cannot be satisfied for a news feed query. However, γ indicates the minimum total diversity for these $n+1$ news feeds, as defined in Definition 2.

Proof. Given the graph modelling of the RDCC problem (see Figure 6), since the capacity $c(m_k, t)$ for every $m_k \in N_m$ is set to 1, it ensures that every message can be scheduled at most once. With the edges between nodes in N_c and nodes in N_m, every message is associated with its corresponding category. Besides, since the capacity $c(r_i, c_j)$ with $r_i \in N_r$ and $c_j \in N_c$ is set to 1, it guarantees that for each news feed, D-MobiFeed can select at most one message for each category. Finally, because we set the capacity of source edges (i.e., $c(s, r_i)$ for every $r_i \in N_r$) as l, it ensures that when a news feed already contains messages with l distinct categories, scheduling more messages with new categories does not contribute to the optimal value of the flow (i.e., γ).

For the case (i), if the *optimal* value of flow (i.e., γ) equals $(n+1) \times l$, all the source edges are saturated (i.e., $x(s, r_i) = l$ for each edge (s, r_i)). It means that for each of $n+1$ news feeds, D-MobiFeed could schedule exactly l messages, with each one belonging to a distinct category. Therefore, l-diversity constraint can be satisfied for current news feed query.

For the case (ii), if $\gamma < (n+1) \times l$, one or more source edges are not saturated; it corresponds to the fact that at least one news feed cannot satisfy the *l-diversity* constraint. However, as we *exactly* solve the maximum flow problem formulated in Section 5.1, the value of γ is maximized. That is, γ is the largest value of the minimum total diversity for the $n+1$ news feeds (Definition 2). □

For the example in Figure 3, when we apply the relabel-to-front algorithm to the flow graph in Figure 6, the optimal value of flow γ is $7 < (n+1) \times l = 9$, which means D-MobiFeed cannot schedule three news feeds such that each of them satisfies the 3-*diversity* constraint. However, we ensure that the minimum total diversity for these three news feeds is 7.

5.3 Discussion

Here we give some discussions about the relationship between the decision step and the scheduling step. Note that if $\gamma < (n+1) \times l$, it is not meaningful to output l as the diversity constraint to the scheduling step, since l-diversity can never be satisfied. Fortunately, in either case of Theorem 1, the value of γ reflects the minimum total diversity we can satisfy for current news feed query. To this end, we output γ as the new diversity constraint to the scheduling step, in which we aim at maximizing the total relevance of $n+1$ news feeds under that diversity constraint.

6 *l*-Diversity Constrained News Feed Scheduling

The scheduling step receives the minimum total diversity (γ) from the decision step. In this section, we redefine the requirement $\mathcal{O}1$ of the l-Diversity Constrained Scheduling (DCS) Problem (see Definition 3) as: ($\mathcal{O}1$') *messages in $n+1$ news feeds fulfill the minimum total diversity γ*. In the DCS problem with the redefined requirement, our objective is to compute a set of $n+1$ news feeds that satisfy γ and have the maximum total relevance score. Similar to the DCC problem, a brute-force method is very costly for the DCS problem. To this end, in this section we propose a three-stage heuristic algorithm to solve the DCS problem, and then analyze the quality of its scheduled news feeds.

1. The value of γ cannot be larger than $(n+1) \times l$ according to the modelling of the flow graph in Section 5.1.

6.1 A Three-Stage Heuristic Algorithm

In this section, we solve the DCS Problem with a three-stage heuristic algorithm. In the first two stages, we neglect the factor of display weight (see Equation 2), only aiming at selecting $k \times (n+1)$ messages to maximize the *unweighted* sum of their relevance scores. In the third stage, we rank the selected messages in each news feed according to their relevance score.

6.1.1 Stage One: Satisfying l-Diversity Constraint

In this stage, we want to select γ messages for $n+1$ news feeds, such that in each news feed, there are at most l messages associating with distinct categories. After selecting such γ messages, we guarantee that $n+1$ news feeds satisfy the minimum total diversity γ (see Definition 2). Since we neglect the display weight of positions in this stage, we focus on maximizing the sum of relevance scores of target γ messages. We define this stage as the γ-*Selection Problem*.

Definition 5. *γ-Selection Problem: Given $n+1$ sets of candidate messages for u's query regions at t_0, t_1, t_2, ..., and t_n, along with their category and relevance score to u, D-MobiFeed selects γ messages for $n+1$ news feeds, such that ($\mathcal{O}1$) each news feed contains at most l messages with distinct categories, and ($\mathcal{O}2$) the sum of relevance scores of these messages is maximized.*

Modeling. The maximum flow model in Section 5 cannot be directly used here since it does not consider the message's relevance score. We translate the γ-Selection Problem into a *minimum cost flow problem* [6]. Consider the example in Figure 3, the flow graph representation of the γ-Selection problem (Figure 6) is almost the same as that of the RDCC Problem in Section 5.1, except the following two differences: (i) Each edge $e(v_i, v_j) \in E$ has not only a *capacity* $c(v_i, v_j)$ but also a *cost* $w(v_i, v_j)$ (the value in parentheses in Figure 6). For each edge $e(c_j, m_k)$ between nodes in N_c (assume c_j is in $(i+1)$-th group of N_c) and nodes in N_m, we set its cost $w(c_j, m_k) = 1 - \textit{relevanceScore}(u, m_k)$, where *relevanceScore*$(u, m_k)$ is the relevance score of m_k to *u* at t_i. The cost of all the other edges in E is 0. In our example, since the relevance score of m_1 to *u* at t_0 is 0.58 (Table I in Figure 3b), the cost of edge connecting node '*Restaurant*' (in the first group of N_c) and $m_1 \in N_m$ is $1 - 0.58 = 0.42$. (ii) For the *balance* of node *s* and *t* (i.e., $b(s)$ and $b(t)$), they are no longer variables. Instead, they are set with fixed value, i.e., $b(s) = \gamma$ and $b(t) = -\gamma$, where γ is the *minimum total diversity* derived from the maximum flow problem in Section 5.1, and γ is set to 7 in our example.

Given the above flow graph, the modeled *minimum cost flow problem* is to assign an integer *flow value* $x(v_i, v_j) \in [0, c(v_i, v_j)]$ for each edge $e(v_i, v_j) \in E$ such that for every node $v \in V$ the flow conservation property (Equation 3) holds, and the following objective function is minimized:

$$\mathcal{G}(x) = \sum_{e(v_i,v_j)\in E} w(v_i, v_j) \times x(v_i, v_j). \quad (4)$$

After solving the minimum cost flow problem formulated above, for any edge (c_j, m_k) with $x(c_j, m_k) = 1$ where c_j is in $(i+1)$-th group of N_c, it means that D-MobiFeed assigns m_k to the news feed for *u*'s query region at t_i.

Proof of correctness. According to the correctness proof in Section 5.2, this minimum cost flow model can select γ messages which satisfy the requirement $\mathcal{O}1$ of the γ-Selection Problem (Definition 5). Furthermore, in the assignment result, every edge $e(c_j, m_k)$ with $x(c_j, m_k) = 1$ incurs cost $w(c_j, m_k) = 1 - \textit{relevanceScore}(u, m_k)$, and there are exactly γ such edges (determined by the balance of source node s in Figure 6). Therefore, we have:

$$\begin{aligned}\mathcal{G}(x) &= \sum_{e(v_i,v_j)\in E} w(v_i, v_j) \times x(v_i, v_j) \\ &= \sum_{x(c_j,m_k)=1} w(c_j, m_k) \\ &= \sum_{x(c_j,m_k)=1} (1 - \textit{relevanceScore}(u, m_k)) \\ &= \gamma - \sum_{0\le i\le n} \textit{unweightedRelevanceScore}(f_i), \end{aligned} \quad (5)$$

where *unweightedRelevanceScore*(f_i) is the sum of relevance scores for messages in a news feed f_i, regardless of the display weight. Since the modeled minimum cost flow problem aims to minimize $\mathcal{G}(x)$, the selected γ messages have the maximum sum of relevance scores (Equation 5), which satisfies the requirement $\mathcal{O}2$ in Definition 5.

Solution. In the literature, there are several algorithms proposed to solve the minimum cost flow problem, including network simplex method [32], cost scaling [21], minimum mean cycle canceling [24], and successive shortest path algorithm (SSPA) [14]. We employ the SSPA algorithm, which is the most general approach with the lowest complexity $O(\gamma \cdot (|E| + |V| \cdot log|V|))$ [36], where γ is the minimum total diversity in our scenario.

In our example, after solving the minimum cost flow problem using the SSPA algorithm, $\{m_2, m_5, m_{10}\}$, $\{m_1\}$ and $\{m_3, m_4, m_8\}$ are assigned to the news feeds at t_0, t_1, and t_2, respectively (Figure 7a).

	Message	Category	Relevance score to u
news feed at t_0	m_2	Restaurant	0.58
	m_5	Theater	0.53
	m_{10}	Museum	0.55
news feed at t_1	m_1	Restaurant	0.68
news feed at t_2	m_3	Restaurant	0.1
	m_4	Stadium	0.53
	m_8	Shopping	0.5

(a)

	Message	Category	Relevance score to u
news feed at t_0	m_2	Restaurant	0.58
	m_{10}	Museum	0.55
	m_5	Theater	0.53
	m_{11}	Museum	0.5
news feed at t_1	m_1	Restaurant	0.68
news feed at t_2	m_4	Stadium	0.53
	m_8	Shopping	0.5
	m_9	Shopping	0.5
	m_3	Restaurant	0.1

(b)

Fig. 7: (a) The assignment results of news feeds at t_0, t_1, and t_2 after Stage One. (b) The assignment results of news feeds at t_0, t_1, and t_2 after Stages Two and Three.

6.1.2 Stage Two: Scheduling Remaining Messages

In Stage One, we have scheduled γ messages satisfying the minimum total diversity (γ). However, some news feeds may not be full, and some candidate messages remain unscheduled. In our example, we can schedule one more message for the news feed at t_0 after Stage One (Figure 7a). Therefore, in this stage, we select messages from remaining candidate messages, such that each of $n+1$ news feeds accommodates (at most) k messages. Keeping the objective of the DCS Problem (Definition 3) in mind, we also maximize the sum of relevance scores of newly selected messages in this stage.

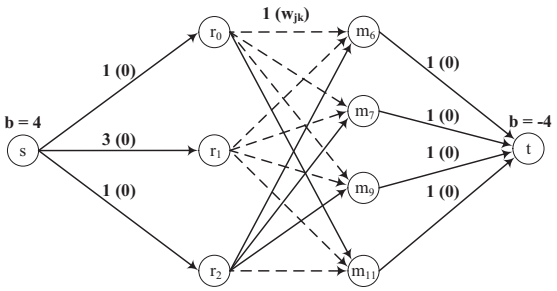


Fig. 8: Flow graph representation of the Replenish-Up-To-k problem.

Definition 6. *Replenish-Up-To-k Problem: Given the results of the γ-Selection Problem, D-MobiFeed selects messages from remaining candidate messages sets and adds them to $n+1$ news feeds, such that ($\mathcal{O}1$) in each news feed, there are at most k messages, and ($\mathcal{O}2$) the sum of relevance scores of newly selected messages is maximized.*

Modeling. We also translate the *Replenish-Up-To-k Problem* into a *minimum cost flow problem*. For our running example, its flow graph representation is shown in Figure 8. Compared to the representation of the γ-Selection Problem (Figure 6), the flow graph does not include the node set N_c (i.e., categories) any more. Also, N_m only contains remaining candidate messages after Stage One. In our example, according to the assignment results after Stage One (Figure 7a), there are four messages m_6, m_7, m_9, and m_{11} can still be selected from N_m. In the flow graph the edge set E consists of three types of edges: (i) An edge $e(s, r_i)$ for every news feed $r_i \in N_r$ has cost 0 and capacity r_i.*free*, where r_i.*free* indicates the number of unoccupied positions for the news feed r_i after performing the message assignment in Stage One. In our example, r_0.*free* $= 1$, r_1.*free* $= 3$, and r_2.*free* $= 1$. (ii) An edge $e(r_i, m_k)$ for every pair of $r_i \in N_r$ and $m_k \in N_m$ has capacity 1. Its cost $w(r_i, m_k) = 1 -$ *relevanceScore*(u, m_k), where *relevanceScore*(u, m_k) is the relevance score of m_k to *u* at t_i. If $m_k \notin$ *CandidateMsg*$_i$, i.e., m_k is not the candidate message at t_i, we set *relevanceScore*$(u, m_k) = 0$ (i.e., the dashed edges in Figure 8). (iii) An edge $e(m_k, t)$ for every $m_k \in N_m$ has cost 0 and capacity 1. Finally, we set the *balances* $b(s) = \gamma'$ and $b(t) = -\gamma'$, where the *required flow* γ' equals $\min\{|N_m|, \sum_{r_i \in N_r} r_i.\textit{free}\}$. In our example, $\gamma' = \min\{4, 5\} = 4$. The *balances* of other nodes are still set to 0.

The same as the definition of the *minimum cost flow problem* described in Section 6.1.1, our task is to assign an integer *flow value* $x(v_i, v_j) \in [0, c(v_i, v_j)]$ for each edge $e(v_i, v_j) \in E$, such that for every node $v \in V$ the flow conservation property (Equation 3) holds, and the objective function (Equation 4) is minimized.

The Replenish-Up-To-k Problem is addressed by solving the minimum cost flow problem formulated above. Given the results of the flow value, for any pair (r_i, m_k) with $x(r_i, m_k) = 1$ and $w(r_i, m_k) < 1$ (i.e., $m_k \in$ *CandidateMsg*$_i$), it means that D-MobiFeed adds m_k to the news feed for *u*'s query region at t_i.

Proof of correctness. Since we set the capacity for each edge $e(s, r_i)$ as the number of unoccupied positions in news feed r_i after Stage One, the requirement $\mathcal{O}1$ is guaranteed. Moreover, similar to Equation 5, the sum of relevance score for each message selected in this stage is negative linear to the total cost of the result flow. Therefore, the selected messages have maximum sum of relevance scores, which satisfies the requirement $\mathcal{O}2$.

Solution. As in Section 6.1.1, we also employ the SSPA algorithm to solve the minimum cost flow problem. In our example (Figure 8), m_{11} and m_9 are added to the news feeds at t_0 and t_2, respectively.

6.1.3 Stage Three: Sorting

After performing the first two stages, we have scheduled a set of messages for each news feed. In this stage we consider the display weight of different positions in the news feed. Intuitively, we display a message with higher relevance score in a higher position (i.e., with a larger display weight). Therefore, for each of $n+1$ news feeds, D-MobiFeed sorts the messages by their relevance scores in non-increasing order as its final result. Figure 7b depicts the final results of three news feeds for our running example.

6.2 Scheduling Quality Analysis

In this section we analyze the scheduling quality of the three-stage heuristic algorithm. After applying the three-stage heuristic algorithm to solve the DCS Problem, news feeds f_0, f_1, ..., and f_n are generated, which have the property:

Theorem 2. *The $n+1$ news feeds generated by the three-stage heuristic algorithm have the maximum $\sum_{0 \le i \le n}$ unweightedRelevanceScore(f_i) among all possible scheduling schemes satisfying the minimum total diversity γ.*

Proof. Suppose one unchosen message m' from candidate messages for new feed f_i ($0 \le i \le n$) can replace one scheduled message $m \in f_i$, so we could increase *unweightedRelevanceScore*(f_i)[2] under the condition that these $n+1$ news feeds still satisfy the minimum total diversity γ. Obviously we have *relevanceScore*$(u, m) <$ *relevanceScore*(u, m').

According to the description of the three-stage heuristic algorithm in Section 6.1, we can divide messages in f_i (before replacement) into two sets, i.e., M_D and M_R, where M_D contains the messages scheduled in the Stage One and M_R stands for the messages assigned in the Stage Two. (1) If $m \in M_R$, after replacing m with m' we increase the sum of relevance scores for messages in M_R, Which is a contradiction according to the definition of M_R (see Definition 6). (2) If $m \in M_D$, there are two cases. Assume that in f_i (before replacement), the set of categories for messages belonging to M_D is C_D; besides, message m and m' are associated with category c_m and $c_{m'}$, respectively. (a) In the first case, $c_{m'} \notin C_D - \{c_m\}$, which means after replacing m with m', $n+1$ news feeds still satisfy the minimum total diversity γ. Thus, we increase the sum of relevance scores for messages in M_D, leading to a contradiction according to Definition 5. (b) In the second case, $c_{m'} \in C_D - \{c_m\}$. In order to guarantee that $n+1$ news feeds still satisfy the minimum total diversity γ after replacement, there must be another scheduled message $m'' \in M_R$ with its category $c_{m''} \notin C_D - \{c_m\}$. Since m'' belongs to M_R instead of M_D in the original f_i, we have *relevanceScore*$(u, m'') <$ *relevanceScore*$(u, m) <$ *relevanceScore*(u, m'), i.e., m' is better than m''. It contradicts with the fact that for the original f_i, the three-stage heuristic algorithm schedules m'' instead of m' in M_R. □

2. I.e., the sum of relevance scores for messages in a news feed f_i, regardless of the display weight.

7 Experiment Results

In this section, we first present experimental setting, and then evaluate the performance of D-MobiFeed through experiments.

Experiment settings. We implemented the experiments in C++ using a real location-aware social network data set in New York City (NYC), USA, which was crawled from Foursquare [20] from December 15 to 25, 2012. The data set contains 417,898 geo-tagged messages (belong to 420 categories in total[3]) and 99,292 users. We extracted the road map of NYC from USA Census TIGER/Line Shapefiles [1] and randomly generated 1,000 30-minute trajectories using A* algorithm [25]. Unless mentioned otherwise, all users move with a constant speed of 40km/h, the look-ahead steps (n) is 5, the minimum number of message categories per news feed (l) is 3, the minimum message display time (t_d) is 10 seconds, the number of messages per news feed (k) is 5, and the query range distance (D) is 600 meters. Based on our experimental results, the average number of messages in a candidate message set is 109. All experiments were run on a Ubuntu 11.10 machine with a 3.4GHz Intel Core i7 processor and 16GB RAM.

Evaluated algorithms. In this section, we compare D-MobiFeed with two baseline approaches.

- n-LA-no-diversity: This algorithm is our previous work [39] (i.e., MobiFeed). n-LA-no-diversity does NOT consider any diversity in news feeds. Specifically, it looks ahead n steps and schedules messages among $n+1$ news feeds using a greedy manner (i.e., a higher priority is given to a message with a larger relevance value computed by Equation 2).
- zero-LA: This algorithm is based on our D-MobiFeed, but its look-ahead step (n) is set to zero. In other words, zero-LA generates news feeds considering both diversity and relevance, but it does not perform any location prediction and processes a news feed request at a time. This baseline is designed to evaluate the effectiveness of the look-ahead technique used by D-MobiFeed.
- n-LA: This is our proposed D-MobiFeed with the look-ahead technique (i.e., $n > 0$).

Performance metrics. We evaluate the performance of evaluated algorithms in terms of three performance metrics.

- *The number of categories:* This metric indicates the quality of news feeds, as it measures the average number of categories in a news feed.
- *Relevance score:* This metric shows the quality of news feeds by measuring the average total relevance score of a news feed.
- *Running time:* This metric indicates the efficiency of an evaluated algorithm. The total running time is split into three parts, namely, scheduling step, decision step, and candidate message step.

7.1 The Summary of Experimental Results

In this section, we first summarize the key findings in our experimental results.

Efficiency. Our proposed three-stage heuristic scheduling algorithm performs much better than the brute-force optimal algorithm by at least an order of magnitude in terms of running time (Figure 9b). Our D-MobiFeed with the n-look-ahead method ($n > 0$) (i.e., n-LA) is more efficient than D-MobiFeed with the zero-look-ahead method ($n = 0$) (i.e., zero-LA) with respect to various levels of diversity requirements, n look-ahead steps, requested numbers of messages in a news feed, and query distance ranges, as depicted in Figures 10c, 11c, 12c, and 13c, respectively. This is because the n-look-ahead method can effectively share execution among multiple news feeds simultaneously.

Diversity. n-LA outperforms our previous work n-LA-no-diversity (i.e., MobiFeed) in terms of diversity. In particular, n-LA-no-diversity cannot satisfy the default l-diversity requirement ($l = 3$) in all the experiments, while n-LA can always satisfy the default value of l ($l = 3$) and the required value of l (up to $l = 7$ in Figure 10a), as depicted in Figures 10a to 13a.

Relevance. Since n-LA considers both relevance and diversity, it slightly reduces the relevance of news feeds compared to the n-LA-no-diversity (i.e., MobiFeed), as depicted in Figures 10b to 13b. However, the n-look-ahead method ($n > 0$) effectively improves the relevance of news feeds compared to zero-LA (Figures 10b to 13b).

7.2 Comparison with the Optimal Solution

In this subsection, we mathematically formulate the DCS Problem in the scheduling step as an Integer Linear Programming (ILP) problem, which aids in better understanding of the problem. Also, we obtain the optimal solution using an ILP solver [27] and compare it with our three-stage heuristic algorithm.

ILP formulation of DCS problem. Without loss of generality, given a news feed query with a look-ahead steps n, we assume that all $n+1$ news feeds could satisfy the l-diversity constraint. For this query, there are N_m *distinct* candidate messages belonging to N_c categories, as well as N_p positions in the news feed result lists, where $N_p = k \times (n+1)$. We first define an $N_p \times N_m$ matrix X as boolean variables to indicate the scheduling results. Specifically, each variable $x_{ij} = 1$ means that we assign a candidate message m_j to position p_i, where $i = k \times a + b$, and p_i stands for the b-th position in the news feed result list at t_a; $x_{ij} = 0$ otherwise. Moreover, we have an $N_p \times N_m$ *weight matrix* W, where each element w_{ij} indicates the score we could obtain for scheduling m_j to position p_i, i.e., *relevanceScore*$(u, m_j) \times$ *displayWeight*(b, k). We also have an $N_m \times N_c$ *category matrix* Z, where each element $z_{jh} = 1$ if m_j belongs to category c_h and $z_{jh} = 0$ otherwise. The ILP formulation can be written as:

$$\textit{Maximize} \sum_{i=1}^{N_p} \sum_{j=1}^{N_m} x_{ij} \times w_{ij}, \tag{6}$$

$$\textit{Subject to: } x_{ij} = \{0, 1\} \quad \text{for all } i, j, \tag{7}$$

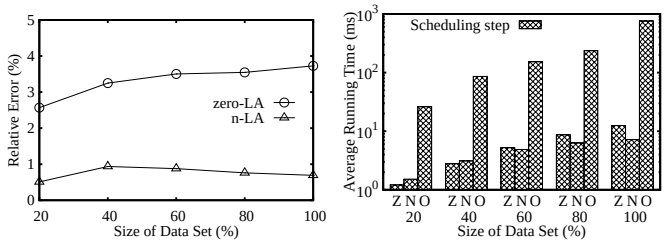
$$\sum_{j=1}^{N_m} x_{ij} \leq 1 \quad \text{for all } i, \tag{8}$$

$$\sum_{i=1}^{N_p} x_{ij} \leq 1 \quad \text{for all } j, \text{ and} \tag{9}$$

$$\textit{non\_zero\_element}(E_v X Z) \geq l \quad \text{for all } v. \tag{10}$$

Constraints 7-9 guarantee that each position accommodates at most one message and each message could be assigned for at most once. Constraint 10 corresponds to the l-diversity constraint. Specifically, E_v ($0 \leq v \leq n$) is an N_p-dimension row vector, where the $(v \cdot k + 1)$-th to $(v \cdot k + k)$-th elements are set to

3. In Foursquare, we define the category of a message as the *primary category* of its associated venue.



(a) Relevance score error (b) Running time (in log scale)

Fig. 9: Comparison with the optimal solution.

one and others are set to zero. Each element of E_vX indicates whether the corresponding message is scheduled in the news feed at t_v. By multiplying E_vX with Z we get an N_c-dimension row vector E_vXZ, where each element shows whether the news feed at t_v contains the corresponding category. The function *non_zero_element* counts the number of non-zero elements in E_vXZ, which is the number of categories in the news feed at t_v. Constraint 10 is non-linear, but it can be reformulated as multiple linear constraints by using an auxiliary vector variable [13]. Thus, all Constraints 7-10 become linear and in the standard ILP form.

Results. We use an ILP solver to get the optimal solution, and compare it with our three-stage heuristic algorithm, as depicted in Fig. 9. In this experiment, we calculate the average relative error of our scheme by: *Relative Error* $= \frac{1}{N} \times \frac{OptimalScore - HeuristicScore}{OptimalScore}$, where N is the number of news feeds, *OptimalScore* and *HeuristicScore* are the total relevance scores of the news feeds computed by the optimal solution and our heuristic algorithm, respectively. Since the optimal solution is computationally expensive, we set the look-ahead steps as two and vary the size of data set from 20% to 100%. Fig. 9a shows that the n-LA is very close to the optimal solution (less than 1% error) in terms of relevance score. Furthermore, n-LA generates news feeds with higher score than zero-LA (will be further described in Section 7.4). To compare the efficiency of the optimal solution and three-stage heuristic algorithm, we show the running time of scheduling step (Section 4) in Fig. 9b. Here, 'Z', 'N', 'O' refer to zero-LA, n-LA, and the optimal solution, respectively. The figure shows that our heuristic algorithm is much more efficient and scalable than the optimal solution.

Since the optimal solution cannot scale up to a large number of messages, we will not test it in other experiments. Next we will evaluate the performance of D-MobiFeed by varying different parameters. To show the benefit of look-ahead technique in D-MobiFeed equipped with n-LA, we employ the zero-LA as a baseline approach. Moreover, to show the loss of relevance for the sake of diversity in D-MobiFeed, we implemented the scheduling algorithm in MobiFeed [39] (termed n-LA-no-diversity) as another baseline. n-LA-no-diversity does not consider diversity of news feeds. It looks ahead n steps and schedules messages among $n+1$ news feeds in a greedy manner (i.e., giving a higher priority to a message with a larger value computed by Equation 2 in Section 4).

7.3 Effect of the l-diversity Constraints

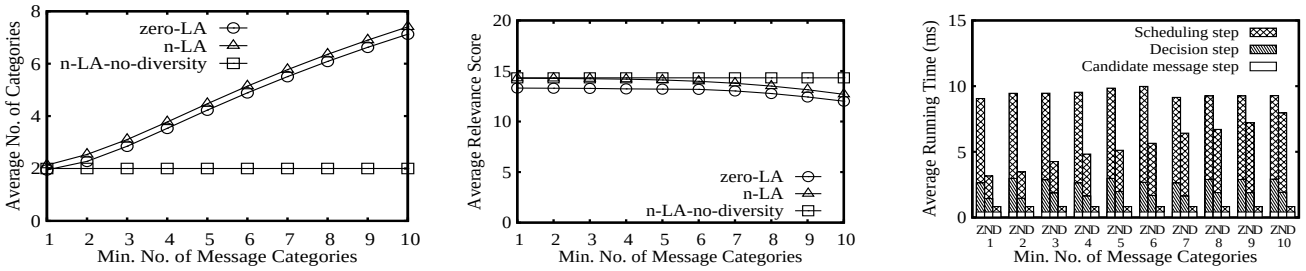
Fig. 10 depicts the performance of n-LA and the two baseline algorithms with respect to the increase of the minimum number of message categories per news feed (i.e., l) from 1 to 10. In this experiment we set $k = 10$ to guarantee that $k \geq l$. The performance of n-LA-no-diversity is not affected by the value of l. Fig. 10a shows that n-LA-no-diversity generates news feeds with only roughly two categories, while n-LA provides more diverse news feeds with larger l (Fig. 10a). Thus, D-MobiFeed provides more satisfactory news feeds compared to MobiFeed. Although n-LA generates news feeds with the best diversity, some of its news feeds may not be able to satisfy the l-diversity constraint, as we discussed in Section 5; and thus, the average number of categories per news feed could be smaller than l. zero-LA performs worse than n-LA because it only tries to find diverse messages for a single news feed. However, n-LA schedules messages with different categories among multiple news feeds, so it has a chance to balance between the diversity and the relevance of news feeds (Fig. 10b). Fig. 10c depicts that n-LA-no-diversity (abbreviated as 'D') is the most efficient due to its greedy characteristic [39]. The running time of our n-LA increases as l gets larger. This is because, in the Stage One of our three-stage heuristic algorithm in the scheduling step, the value of γ gets larger with the increase of *l*, thus increasing the complexity of the SSPA algorithm (see Section 6.1.1). D-MobiFeed is scalable with the increase of value l. Furthermore, zero-LA incurs the largest computational overhead, since it has to perform decision and scheduling steps more times than n-LA.

7.4 Effect of Look-ahead Steps

In this experiment, we evaluate the effect of look-ahead steps (i.e., n, varying from 1 to 10, on the performance of D-MobiFeed. As depicted in Fig. 11a, although the average number of categories per news feed of both n-LA and n-LA-no-diversity is less than the default value of $l = 3$, n-LA doubles the number of categories per news feed as in n-LA-no-diversity. Besides, when n gets larger, n-LA could generate news feeds with more categories. The main reason is that, by looking ahead more steps, n-LA has a higher chance to assign a message with a certain category to a better news feed within its lifetime, such that the diversity constraint could be satisfied to a larger extent. Fig. 11b shows that, compared to n-LA-no-diversity, there is a slight loss (i.e., less than 3%) of relevance for our n-LA. As n gets larger, the improvement of generated news feeds' relevance for n-LA is larger than that of zero-LA (Fig. 11b), since more query regions are considered at one time, thus making a message scheduled to a better news feed to maximize the total relevance score. In terms of the efficiency, when n becomes larger, the running time of n-LA initially decreases (Fig. 11c). However, after n is larger than five, its running time increases slightly. The main reason is that, when n is small, n-LA benefits from sharing the computation of scheduling $n+1$ news feeds for a user's location update; as n gets larger, this benefit is offset by the increase of overhead for our three-stage heuristic algorithm in the scheduling step, since there are more candidate messages within $n+1$ query regions, leading to more computation time spent on a flow graph with more nodes. D-MobiFeed efficiently provides news feeds with higher levels of diversity while preserving their high quality in terms of relevance.

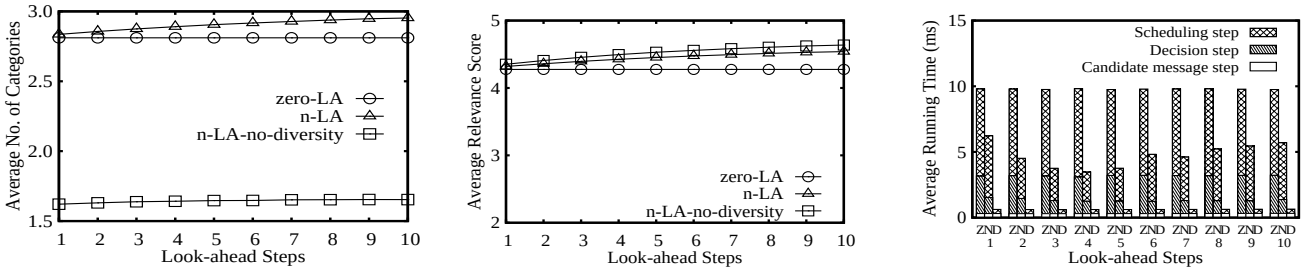
7.5 Effect of News Feed Size

This experiment investigates the effect of user-requested news feed size (i.e., k) on the performance of all the algorithms by increasing k from 3 to 10. The value of k starts from 3 because we need to ensure that k is equal to or larger than the default l-diversity



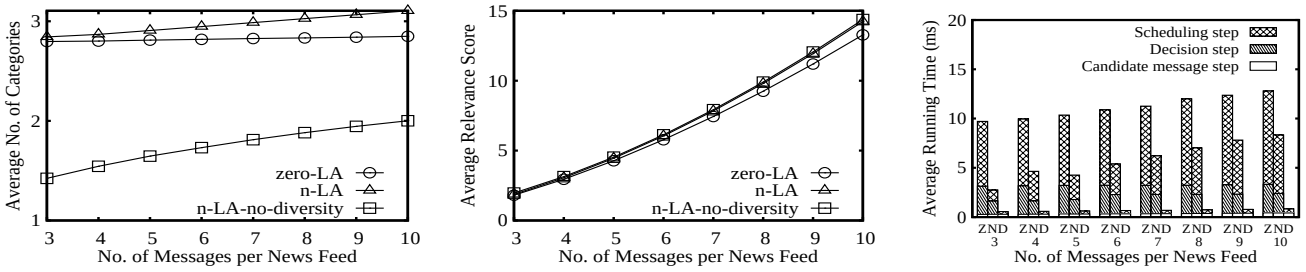
(a) Number of categories (b) Relevance score (c) Running time (in log scale)

Fig. 10: Effect of the minimum number of message categories per news feed.



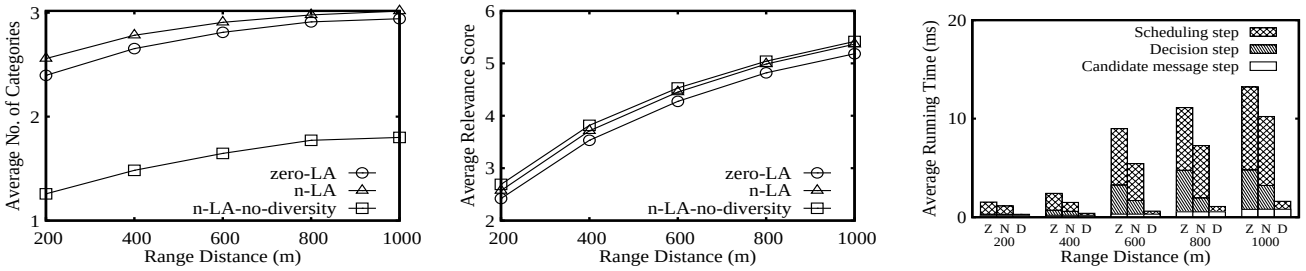
(a) Number of categories (b) Relevance score (c) Running time (in log scale)

Fig. 11: Effect of look-ahead steps.



(a) Number of categories (b) Relevance score (c) Running time (in log scale)

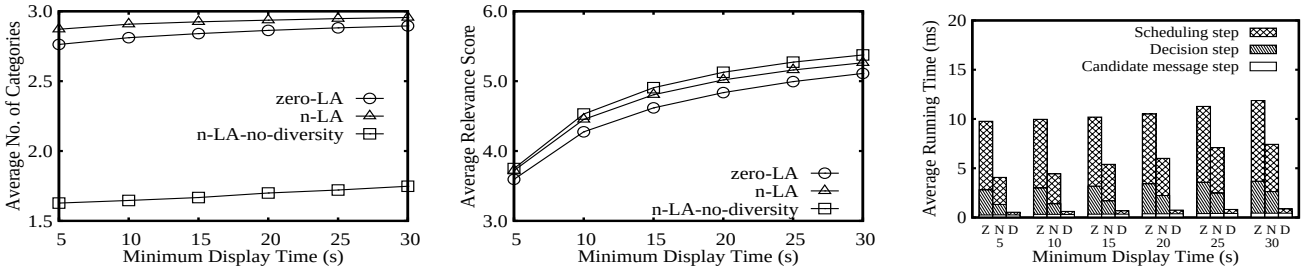
Fig. 12: Effect of news feed size.



(a) Number of categories (b) Relevance score (c) Running time (in log scale)

Fig. 13: Effect of query range distance.

constraint (i.e., $l = 3$). Fig. 12a shows that, on the one hand, n-LA-no-diversity cannot generate news feeds with more than two categories even if we increase the value of k to 10; on the other hand, D-MobiFeed could generate news feeds whose number of categories is close to the required diversity constraint (i.e., $l = 3$). The reason is that, the Stage One of our three-stage heuristic algorithm generates news feeds with the largest extent of diversity. It is also expected that the number of categories in a news feed increases when k gets larger (Fig. 12a). Fig. 12b depicts that the loss of relevance for our scheme compared to n-LA-no-diversity is minor. Furthermore, the increase of k leads to more messages in a news feed, so the total relevance score of news feed gets higher. However, when k gets larger, we need to replenish more messages in the Stage Two of our three-stage heuristic algorithm, so its running time gets longer (Fig. 12c).



(a) Number of categories (b) Relevance score (c) Running time (in log scale)

Fig. 14: Effect of minimum display time.

7.6 Effect of Query Range Distance

This experiment studies the effect of user-specified query range distance (i.e., D) on the performance of all the algorithms with various D from 200 to 1,000 meters, as shown is Fig. 13. It is expected that larger D leads to more candidate messages for a query region. Thus, all the algorithms find more diverse news feeds (Fig. 13a) and more relevant news feeds (Fig. 13b). However, Fig. 13c shows that the running time of all the algorithms gets larger when D increases, as they need to retrieve more candidate messages and process them to generate news feeds.

7.7 Effect of Minimum Display Time

Fig. 14 depicts the performance of zero-LA, n-LA, and n-LA-no-diversity with respect to the increase of the minimum display time (i.e., t_d) from 5 to 30 seconds. Fig. 14a shows that, compared to n-LA-no-diversity, n-LA provides messages associated with more categories due to the diversity constraint. Furthermore, as t_d gets larger, there are more categories in each news feed. The main reason is that, the overlapping area between two consecutive query regions becomes smaller when t_d increases, so there are more distinct candidate messages (thus belonging to more categories) in each query region. As a result, there is a higher chance for our scheduler to generate more diverse news feeds. When we have more distinct candidate messages in each query region with larger t_d, our news feed scheduler has a higher chance to generate news feeds with better relevance (Fig. 14b). In general, D-MobiFeed provides a better trade-off between the diversity and relevance of messages compared to MobiFeed. However, t_d should not be too large because mobile users may miss nearby relevant messages. The efficiency of all the algorithms becomes worse when t_d gets larger, since the scheduler has to process more distinct messages (Fig. 14c).

7.8 Effect of Location Prediction Errors

This section evaluates the effect of errors of the location prediction function (Section 3.2.1). We randomly generate 10,000 trajectories on the road map with speeds randomly selected from 20km/h to users' maximum movement speed (40 km/h by default). 5,000 trajectories are randomly selected as a training set for the location prediction function. We use the remaining 5,000 trajectories as a test set to evaluate the effect of the location prediction algorithm. In the experiments, a prediction error occurs if the difference between a user's actual location and predicted location is larger than the tolerance threshold θ that is set to 50 meters [15]. When a prediction error takes place, the scheduler needs to re-schedule news feeds based on a user's location update. We measure the prediction error as the ratio of the number of prediction errors to the number of location predictions.

TABLE 1: Prediction error (look-ahead steps).

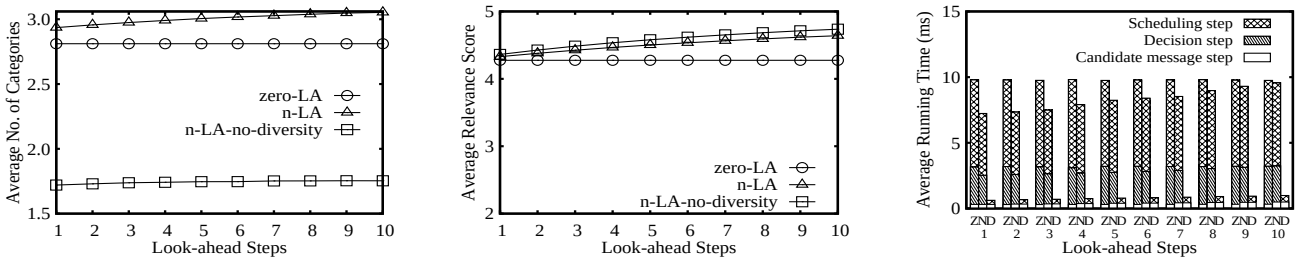
n	1	2	3	4	5	6	7	8	9	10
Error (%)	1.4	14.7	16.9	18.6	19.4	20.0	21.1	21.3	21.4	21.5

Fig. 15 and Table 1 depict the experimental results with an increase of the look-ahead steps (n) from 1 to 10. Fig. 15a and b exhibit the similar trend of diversity and relevance compared to their counterparts without prediction errors, as depicted in Fig. 11a and b, respectively. Fig. 15c shows that, a larger n increases the running times of our schedulers with errors in location prediction function. This is because as n gets larger, D-MobiFeed needs to predict longer path, and there is a higher chance for a prediction error to occur (Table 1). When more prediction errors take place, the scheduler has to re-schedule more news feeds based on updated locations, thus incurring higher computational overhead.

8 Conclusion

In this paper, we design D-MobiFeed; a location-aware news feed framework takes the relevance and diversity of news feeds into account when scheduling news feeds for moving users. D-MobiFeed users can specify the minimum number of categories in a news feed as an l-diversity constraint, and it aims at maximizing the total relevance of generated news feeds and satisfying the l-diversity constraint. We focus on two key problems in D-MobiFeed, namely, decision and optimization problems. For the decision problem, we model it as a maximum flow problem and enable D-MobiFeed to decide whether it can fulfill the l-diversity constraint for a news feed. For the optimization problem, we design an efficient three-stage heuristic algorithm to maximize the total relevance of news feeds under the l-diversity constraint. We evaluate the performance of D-MobiFeed using a real social network data set crawled from Foursquare and a real road network. Experimental results show that D-MobiFeed can efficiently provide location- and diversity-aware news feeds when maintaining their high quality in terms of relevance.

Our future direction is to measure the dissimilarity of pair-wise messages in terms of their category information and study a new multi-objective optimization problem of finding a set of news feeds, in which each news feed satisfies the l-diversity constraint and the dissimilarity of the messages in each news feed is maximized while maximizing the total relevance of a set of $n+1$ news feeds for mobile users (where n is the look-ahead step).



(a) Number of categories (b) Relevance score (c) Running time (in log scale)

Fig. 15: Effect of look-ahead steps (with prediction errors).

REFERENCES

[1] 2010 Census TIGER/Line Shapefiles. http://www.census.gov/geo/www/tiger/tgrshp2010/tgrshp2010.html.

[2] Z. Abbassi, V. S. Mirrokni, and M. Thakur. Diversity maximization under matroid constraints. In *ACM KDD*, 2013.

[3] G. Adomavicius and Y. Kwon. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE TKDE*, 24(5):896–911, 2012.

[4] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE TKDE*, 17(6):734–749, 2005.

[5] R. Agrawal, S. Gollapudi, A. Halverson, and S. Leong. Diversifying search results. In *ACM WSDM*, 2009.

[6] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[7] J. Bao, M. F. Mokbel, and C.-Y. Chow. GeoFeed: A location-aware news feed system. In *IEEE ICDE*, 2012.

[8] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *ACM SIGIR*, 1998.

[9] B. Carterette and P. Chandar. Probabilistic models of novel document rankings for faceted topic retrieval. In *ACM CIKM*, 2009.

[10] B. Chandramouli, J. Yang, P. K. Agarwal, A. Yu, and Y. Zheng. ProSem: Scalable wide-area publish/subscribe. In *ACM SIGMOD*, 2008.

[11] C.-Y. Chow, J. Bao, and M. F. Mokbel. Towards location-based social networking services. In *ACM SIGSPATIAL LBSN*, 2010.

[12] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (Third Edition)*. The MIT Press, 2009.

[13] G. B. Dantzig. *Linear Programming and Extensions*. Princeton university press, 1965.

[14] U. Derigs. A shortest augmenting path method for solving minimal perfect matching problems. *Networks*, 11(4):379–390, 1981.

[15] G. M. Djuknic and R. E. Richton. Geolocation and Assisted GPS. *IEEE Computer*, 34(2):123–125, 2001.

[16] J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, 1972.

[17] Facebook. http://www.facebook.com/about/location.

[18] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[19] Foursquare. http://www.foursquare.com.

[20] Foursquare Developer API. https://developer.foursquare.com/.

[21] A. V. Goldberg and R. Kennedy. An efficient cost scaling algorithm for the assignment problem. *Mathematical Programming*, 71:153–177, 1995.

[22] A. V. Goldberg and S. Rao. Beyond the flow decomposition barrier. *J. ACM*, 45(5):783–797, 1998.

[23] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum flow problem. In *ACM STOC*, 1986.

[24] A. V. Goldberg and R. E. Tarjan. Finding minimum-cost circulations by canceling negative cycles. *Journal of the ACM*, 36(4):873–886, 1989.

[25] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE TSSC*, 4(2):100–107, 1968.

[26] H. Jeung, M. L. Yiu, X. Zhou, and C. S. Jensen. Path prediction and predictive range querying in road network databases. *VLDB Journal*, 19(4):585–602, 2010.

[27] lp_solve 5.5.5.0. http://lpsolve.sourceforge.net/5.5/.

[28] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM TKDD*, 1(1):3, 2007.

[29] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[30] S. M. McNee, J. Riedl, and J. A. Konstan. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *ACM Conference on Human Factors in Computing Systems*, 2006.

[31] M. F. Mokbel, X. Xiong, and W. G. Aref. SINA: Scalable incremental processing of continuous queries in spatio-temporal databases. In *ACM SIGMOD*, 2004.

[32] J. B. Orlin. A polynomial time primal network simplex algorithm for minimum cost flows. *Mathematical Programming*, 78:109–129, 1997.

[33] O. Phelan, K. McCarthy, and B. Smyth. Using twitter to recommend real-time topical news. In *ACM RecSys*, 2009.

[34] F. Radlinski and S. T. Dumais. Improving personalized web search using result diversification. In *ACM SIGIR*, 2006.

[35] A. Silberstein, J. Terrace, B. F. Cooper, and R. Ramakrishnan. Feeding Frenzy: Selectively materializing user's event feed. In *ACM SIGMOD*, 2010.

[36] L. H. U, K. Mouratidis, M. L. Yiu, and N. Mamoulis. Optimal matching between spatial datasets under capacity constraints. *ACM TODS*, 35(2):9:1–9:44, 2010.

[37] S. Vargas and P. Castells. Rank and relevance in novelty and diversity metrics for recommender systems. In *ACM RecSys*, 2011.

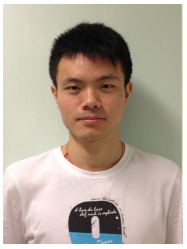
[38] L. Wang, Y. Jia, and W. Han. Instant message clustering based on extended vector space model. In *ISICA*, 2007.

[39] W. Xu, C.-Y. Chow, M. L. Yiu, Q. Li, and C. K. Poon. MobiFeed: A location-aware news feed system for mobile users. In *ACM SIGSPATIAL GIS*, 2012.

[40] C. X. Zhai, W. W. Cohen, and J. Lafferty. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In *ACM SIGIR*, 2003.

[41] M. Zhang and N. Hurley. Avoiding monotony: Improving the diversity of recommendation lists. In *ACM RecSys*, 2008.

[42] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW*, 2005.



Wenjian Xu received the bachelor's degree in software engineering from Northwestern Polytechnical University, Xi'an, China, in 2010. He is pursuing his M.Phil. degree at the City University of Hong Kong. His research interests span query processing, query optimization and big data analytics.

Chi-Yin Chow received the M.S. and Ph.D. degrees from the University of Minnesota-Twin Cities in 2008 and 2010, respectively. He is currently an assistant professor in Department of Computer Science, City University of Hong Kong. His research interests include spatio-temporal data management and analysis, GIS, mobile computing, and location-based services. He is the co-founder and co-organizer of ACM SIGSPATIAL MobiGIS 2012, 2013, and 2014.