Collaborative Nowcasting for Contextual Recommendation

Yu Sun [†‡1], Nicholas Jing Yuan[*] [‡2], Xing Xie [‡3], Kieran McDonald [§4], Rui Zhang [†5]

[†] Department of Computing and Information Systems, University of Melbourne
{[1] sun.y, [5] rui.zhang}@unimelb.edu.au

[‡] Microsoft Research [§] Microsoft Corporation
{[2] nicholas.yuan, [3] xing.xie, [4] kieran.mcdonald}@microsoft.com

ABSTRACT

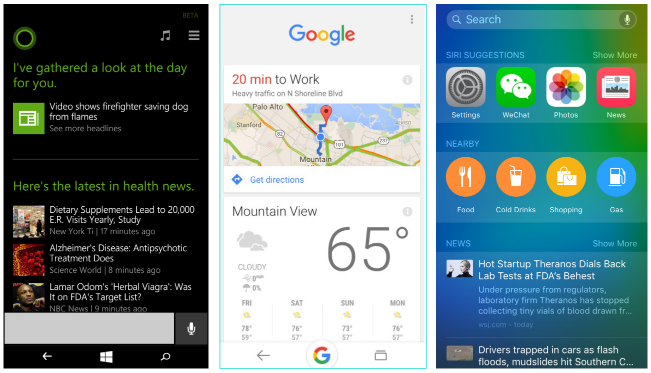
Mobile digital assistants such as Microsoft Cortana and Google Now currently offer appealing proactive experiences to users, which aim to deliver the right information at the right time. To achieve this goal, it is crucial to precisely predict users' real-time intent. Intent is closely related to context, which includes not only the spatial-temporal information but also users' current activities that can be sensed by mobile devices. The relationship between intent and context is highly dynamic and exhibits chaotic sequential correlation. The context itself is often sparse and heterogeneous. The dynamics and co-movement among contextual signals are also elusive and complicated. Traditional recommendation models cannot directly apply to proactive experiences because they fail to tackle the above challenges. Inspired by the nowcasting practice in meteorology and macroeconomics, we propose an innovative *collaborative nowcasting* model to effectively resolve these challenges. The proposed model successfully addresses sparsity and heterogeneity of contextual signals. It also effectively models the convoluted correlation within contextual signals and between context and intent. Specifically, the model first extracts *collaborative latent factors*, which summarize shared temporal structural patterns in contextual signals, and then exploits the collaborative Kalman Filter to generate serially correlated *personalized latent factors*, which are utilized to monitor each user's real-time intent. Extensive experiments with real-world data sets from a commercial digital assistant demonstrate the effectiveness of the collaborative nowcasting model. The studied problem and model provide inspiring implications for new paradigms of recommendations on mobile intelligent devices.

General Terms

Algorithms, Design, Experimentation

Keywords

Recommendation; Intent Monitoring; Nowcasting



I've gathered a look at the day for you.

Video shows firefighter saving dog from flames
See more headlines

Here's the latest in health news.

Dietary Supplements Lead to 20,000 E.R. Visits Yearly, Study
New York Ti | 17 minutes ago

Alzheimer's Disease: Antipsychotic Treatment Does
Science World | 8 minutes ago

Lamar Odom's 'Herbal Viagra': Was It on FDA's Target List?
NBC News | 9 minutes ago

20 min to Work
Heavy traffic on N Shoreline Blvd

Get directions

Mountain View

Hot Startup Theranos Dials Back Lab Tests at FDA's Behest
Under pressure from regulators, laboratory firm Theranos has stopped collecting tiny vials of blood drawn fr...
wsj.com - today

Drivers trapped in cars as flash floods, mudslides hit Southern C...

Figure 1: Proactive experience on digital assistants

1. INTRODUCTION

The emergence of mobile digital assistants offers a new paradigm of recommendations. Based on context, digital assistants aim to recommend "the right information at just the right time" [1] and help you "get things done" [2] even "before you ask" [3]. Figure 1 shows examples of such *proactive experiences* in different digital assistants: Microsoft Cortana, Google Now, and Apple's Siri. The recommended information includes videos, news, traffic conditions, weather, apps, places, and many other types such as calendars, stock prices, sports, events, etc. The different types of information are typically presented as *cards*. Due to the limited display size of mobile phones, only one or two cards can be effectively shown. Therefore, it is crucial to determine exactly which cards match a user's current interest or intent.

The intent of a user is closely related to the user's context, including both the external context, e.g., the location of a user, and the internal context such as the user's current physical activity or usage of an app. For example: i) When it is 6:00 p.m. and the user is in the office, she may intend to drive home. ii) When the user has just left a shopping center and is using Yelp, she may intend to find a restaurant. To recommend the right card at the right time, the first step is to precisely and continuously predict the user's intent based on context. We call this the *intent monitoring* problem. The correlation between the intent and context is chaotic. It exhibits tremendously dynamic temporal characteristics and strong sequential correlation. The intent and context may swiftly change in a very short time. The current intent may be influenced by a previous context, and conversely, the current context may result from the action triggered by a previous intent. Context itself is also heterogeneous and complicated. All contemporaneous information related to

[*] Corresponding author.

WWW 2016, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4143-1/16/04.
http://dx.doi.org/10.1145/2872427.2874812.

the intent is included in the context. Modeling the structure of the context and the relationship between the context and intent is a great challenge.

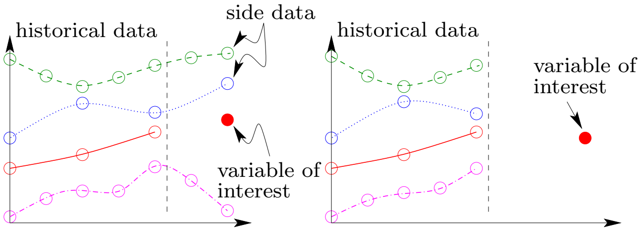
Existing recommendation models cannot effectively tackle the above challenge. Instead of monitoring users' intent, most existing recommendation algorithms deal with a particularly given intent, e.g., to find interesting movies, music tracks, or books, and try to recommend new items fulfilling the given intent. State-of-the-art recommendation models [8][17][36] that capture the evolving of user preferences and item attributes also fail to address the above challenge. This is because instead of evolving on a daily or monthly basis, the intent, together with the context, may change dramatically within a short time. Models [15][30] for short-term (e.g., next-basket) recommendations that depend on the similarity or co-occurring patterns between items cannot resolve the challenge either because they overlook the context. Although a few context-aware recommendation models [5][22] have looked at the context of a user, the considered context is often static and contains very few signals.

Inspired by models explaining the chaotic weather and dynamic economic variables, we propose resolving the intent monitoring problem with an innovative *collaborative nowcasting* model. Nowcasting is widely used in meteorology and macroeconomics. It is defined as: the prediction of the present and very near future (cf. Section 5.1 for more discussions). A main difference between nowcast and forecast is the effective exploitation of *side data*, which are quantities contemporaneous with the variable of interest. Utilizing the context as side data to intent, the collaborative nowcasting model effectively resolves the sparsity, heterogeneity, and dynamics of the context and intent. It also successfully models sequential patterns, co-movement, and correlation within the context and between context and intent.

In the collaborative nowcasting model, we treat the context as stochastic processes and represent the history of contextual information as correlated time series. To resolve the sparseness of such series, we make use of tensor decomposition techniques to obtain *collaborative latent factors*, which summarize the prevalent co-movement and temporal structure among all the series. Then, to effectively monitor the intent of each individual user and model the correlation of contextual information, we deploy the collaborative Kalman Filtering to generate *personalized latent factors*. Finally, we employ the personalized latent factors to nowcast/monitor the intent for each user. The contribution of this paper is summarized as follows:

- We identify the intent monitoring problem, which closely tracks the user's real-time intent and has wide applications in emerging proactive experiences.
- We propose a collaborative nowcasting model, which successfully models the dynamic characteristics, sequential patterns, and complex correlation between context and intent, and effectively solves the intent monitoring problem.
- We conduct extensive experiments with real-world data sets from a commercial digital assistant. The results confirm the superiority of the collaborative nowcasting model over various baselines.

The rest of the paper is organized as follows. Section 2 formally defines the studied problem and introduces the nowcasting concept. Section 3 discusses the collaborative nowcasting model. Section 4 presents the experiments. Section 5 summarizes related work and Section 6 concludes.



(a) Nowcast (b) Forecast

Figure 2: Difference between nowcast and forecast

2. PRELIMINARY

We first formally define the intent monitoring problem for contextual recommendations, and then introduce nowcasting and the existing nowcasting model.

2.1 Problem Formulation

The intent we consider can be any potential need of the user, for example, the need to read news, check the weather or traffic conditions, find nearby restaurants, check stock prices, install new apps etc. Within a time range t, the user u may have several types of intent. Let Γ_t^u be the intent set. Given a type of intent γ, we use $\mathcal{I}_{\Gamma_t^u}(\gamma)$ to indicate whether the user u has the intent γ within t, where

$$\mathcal{I}_{\Gamma_t^u}(\gamma) = \begin{cases} 1 & \text{if } \gamma \in \Gamma_t^u \\ 0 & \text{if } \gamma \notin \Gamma_t^u. \end{cases}$$

The context X_t^u of a user u can be any contemporaneous information relevant to the user's intent, such as the physical environment like the spatial and temporal information, the activities the user is performing (or has recently performed), keywords the user recently searched for in a search engine, etc. We formally define the intent monitoring problem as follows:

Definition 1 (Intent Monitoring). *Given a starting time t_0, a monitoring granularity Δ, a type of intent γ and the context X_t^u of user u, the intent monitoring problem is to predict the value of $\mathcal{I}_{\Gamma_t^u}(\gamma)$ with the context X_t^u for each time step t of length Δ starting from t_0.*

2.2 Nowcasting

To effectively utilize contemporaneous information relevant to the variable of interest, we need *nowcast* instead of forecast. Nowcast is defined as the prediction of the current value of a variable of interest or its value in the very near future, e.g., two hours (hence nowcast is sometimes also referred to as short-term forecast).

The main difference between forecast and nowcast lies in the availability of *side data*. As illustrated in Figure 2(a), side data, different from historical data, are quantities that are contemporaneous with, closely related to, and more frequently available than the variable of interest (e.g., the industrial output to the gross domestic product (GDP)). In nowcasting, we can infer the value of the variable of interest more accurately by utilizing both the historical and side data. When conducting a forecast, as shown in Figure 2(b), all the information we can exploit are historical data (relative to the variable of interest). In fact, for nowcasting, we tend to rely more on side data than historical data.

Table 1: Example of a panel

Time step	10 a.m.	11 a.m	12 p.m.	1 p.m.	Now
Facebook	306	0	915	32	257
Skype	0	1853	0	0	-
McDonald's	0	1256	652	0	0
IKEA	0	0	0	532	1247
Dist-to-Office	10.4	8.3	9.1	21.3	-
Day-of-Week	6	6	6	6	6
News Intent	0	0	1	1	?

To solve the intent monitoring problem, context is an important information source, which can be treated as side data to the intent. Therefore, the intent monitoring problem fits into the nowcasting scenario very well. A widely used nowcasting model in economics [11][12] first uses a few factors to describe the bulk movement of the time series of many macroeconomic variables, and then exploits the relationship between the factors and variable of interest for nowcasting. A direct application of this nowcasting model, however, is not sensible because: i) The nowcasting granularity of the above model is monthly or quarterly, which is quite different from the usually hourly granularity of the contextual recommendation scenario. ii) The macroeconomic variables, i.e., side data, in the above model are universal, while in the intent monitoring problem, the context is personalized for each individual user. iii) The time series of macroeconomic variables are not sparse. Each series has a non-zero value at plenty of (usually all) time steps. However, in the intent monitoring problem, as we will see, the contextual data are very sparse. Moreover, there are many implicit assumptions in the model to address, and to the best of our knowledge, such a nowcasting model has never been applied to a recommendation scenario. Nevertheless, inspired by the nowcasting scenario and above model, we develop our collaborative nowcasting model.

3. COLLABORATIVE NOWCASTING

We first introduce the model in Section 3.1 and then discuss the three steps for estimating the model parameters in Sections 3.2, 3.3, and 3.4, respectively.

3.1 Model Formulation

Following existing work on nowcasting [6][7][12], we model the contextual information as stochastic processes and represent the user's historical and side data as time series. Each type of contextual information is one stochastic process and produces one series. All the available series for a user u form a *panel* $\boldsymbol{X}^u$. Table 1 shows an example of a panel containing six series: two app series named Facebook and Skype, respectively; two venue series: McDonald's and IKEA; one spatial series: Dist-to-Office and one temporal series: Day-of-Week. The monitoring intent is to read news. The monitoring granularity (i.e., time step length) is one hour and the panel shows the user's historical and side data from 10 o'clock in the morning to *now*. We denote by $x_{i,t}^u$ the tth random variable of the ith process in panel $\boldsymbol{X}^u$, which is also referred to as the *contextual indicator*. The value of $x_{i,t}^u$ either indicates the length users use an app or visit a venue, or any other relevant quantities for the process such as the distance to the office. In the sequel, we use the two words process and series interchangeably when the context is clear. Note that in the last time step, the side data may not be available in a synchronous manner, which means we may have *missing values* (denoted by the symbol "$-$" in the above example) for real-time nowcasting. In practice, there can be hundreds of series in a panel and the monitoring granularity can range from minutes to hours depending on the application. Each user has data specific to herself and hence has a different number of series. We denote by N^u the number of series in $\boldsymbol{X}^u$, and by T the number of time steps. For expositional convenience, we will present the model using the panel of each individual user, and in the following part of this section we will drop the superscript u for notational simplicity.

To obtain a parsimonious model and hence retain the model's forecasting power, we assume that the dynamics of the panel are driven by a few latent factors. Let R denote the number of factors for $\boldsymbol{X}$. We assume that the contextual indicator $x_{i,t}$ in panel $\boldsymbol{X}$ has the following structure

$$x_{i,t} = \boldsymbol{\lambda}_i' \cdot \boldsymbol{f}_t + \xi_{i,t}, \quad 1 \le i \le N,\ 1 \le t \le T,$$

where $\boldsymbol{f}_t = (f_{1,t}, \ldots, f_{R,t})'$ contains the latent factors, $\boldsymbol{\lambda}_i = (\lambda_{i,1}, \ldots, \lambda_{i,R})'$ is called the factor *loading*, and $\xi_{i,t}$ is the random noise following a Gaussian distribution with zero mean and variance $\tilde{\psi}_{i,t}$. Note that the factor loading $\boldsymbol{\lambda}_i$ is only relevant to the ith series and the factor $\boldsymbol{f}_t$ is shared by all the series in the panel. Writing the above model in the matrix form, we have

$$\boldsymbol{x}_t = \boldsymbol{\Lambda} \boldsymbol{f}_t + \boldsymbol{\xi}_t \tag{1}$$

where $\boldsymbol{\Lambda} = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_N)'$, $\boldsymbol{x}_t = (x_{1,t}, \ldots, x_{N,t})'$, and $\boldsymbol{\xi}_t = (\xi_{1,t}, \ldots, \xi_{N,t})'$ are the factor loading matrix, the panel column vector, and noise vector at time step t, respectively. We also collect the factors in matrix $\boldsymbol{F} \in \mathbb{R}^{R \times T}$ and let $\boldsymbol{f}_t$ stand for the tth column of the factor matrix $\boldsymbol{F}$. For model simplicity, we assume that the noise components are orthogonal across series and time steps, i.e.,

$$E(\boldsymbol{\xi}_t \boldsymbol{\xi}_t') = \boldsymbol{\Psi}_t = \text{diag}(\tilde{\psi}_{1,t}, \ldots, \tilde{\psi}_{N,t})$$
$$E(\boldsymbol{\xi}_t \boldsymbol{\xi}_{t-\delta}') = 0, \text{for all } \delta > 0.$$

To handle the missing value at the last time step and simplify the model, we set

$$\tilde{\psi}_{i,t} = \begin{cases} \tilde{\psi}_{i,t} = \psi_i & \text{if } x_{i,t} \text{ is available} \\ \infty & \text{if } x_{i,t} \text{ is not available} \end{cases}$$

which means one series has the same noise variance across different time steps and the missing value is treated as noise with a very large variance.

To fully exploit the sequential pattern and co-movement of the latent factors, we assume that the dynamics and autocorrelation of the latent factors have the following structure

$$\boldsymbol{f}_t = \boldsymbol{A} \boldsymbol{f}_{t-1} + \boldsymbol{B} \boldsymbol{\omega}_t \tag{2}$$

where $\boldsymbol{A} \in \mathbb{R}^{R \times R}$ is the transition matrix, $\boldsymbol{B} \in \mathbb{R}^{R \times Q}$ is a matrix of full rank, and $\boldsymbol{\omega}_t$ is the white noise (i.e., $\boldsymbol{\omega}_t \sim \text{WN}(0, \boldsymbol{I}_Q)$).

The given type of intent is also modeled as a stochastic process, where the value of the produced time series indicates the likelihood of a user having the intent. When the likelihood is above a chosen threshold, we say that the user has such intent. Let $\hat{y}_t$ be the value of the nowcasted likelihood at time step t. Assuming that the intent likelihood and contextual indicators are jointly normal, we obtain that the likelihood is a linear function of the estimated latent factors $\hat{\boldsymbol{f}}_t$ [12], i.e.,

$$\hat{y}_t = \alpha + \boldsymbol{\beta}' \hat{\boldsymbol{f}}_t \quad \text{for } 1 \le t \le T \tag{3}$$

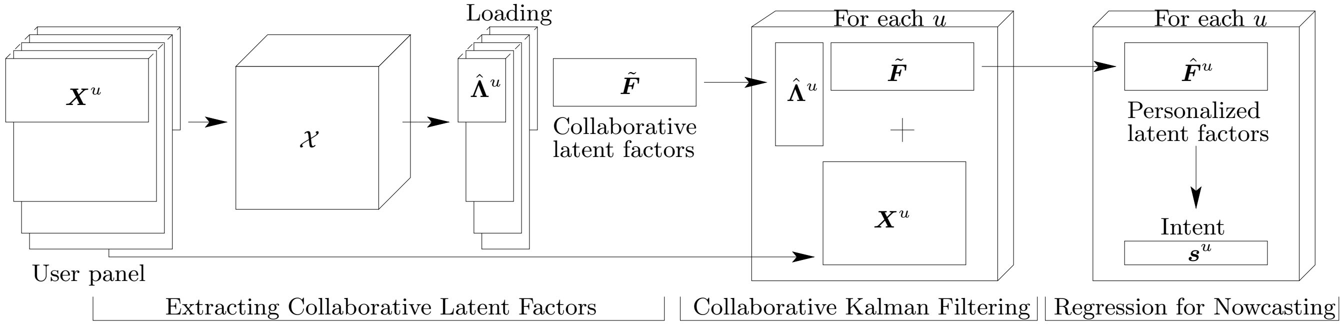

Loading
$\boldsymbol{X}^u$
$\mathcal{X}$
$\hat{\boldsymbol{\Lambda}}^u$
$\tilde{\boldsymbol{F}}$
Collaborative latent factors
For each u
$\hat{\boldsymbol{\Lambda}}^u$
$\tilde{\boldsymbol{F}}$
+
$\boldsymbol{X}^u$
For each u
$\hat{\boldsymbol{F}}^u$
Personalized latent factors
Intent
$\boldsymbol{s}^u$
User panel
Extracting Collaborative Latent Factors
Collaborative Kalman Filtering
Regression for Nowcasting

Figure 3: Collaborative nowcasting model

where α and $\boldsymbol{\beta}$ are coefficients. At this point, the model is fully established.

Discussion. The model described above is able to effectively handle the intent monitoring problem because: i) It models the context and intent as correlated time series and hence fully takes into account the temporal dynamics and sequential patterns in the series itself and across series. ii) Instead of estimating a full model which may introduce too much uncertainty due to a large number of parameters in the panel, it restricts the estimation to only a few latent factors which leads to a parsimonious model and retains the model's forecasting power. iii) It utilizes the real-time data flow reflected in the side data and is able to make reliable nowcast even if only a small amount of side data are available. The uncertainty of the nowcast is also expected to decrease with the arrival of new side data [12].

The remaining issue is estimating the parameters in the model. One challenge in the intent monitoring problem is that the panel (as illustrated in Table 1) is usually very sparse, and this will cause significant problems in estimating the model parameters, especially the latent factors. We propose solving this problem by employing the collaborative capabilities among users. In particular, as illustrated in Figure 3, we first i) collect the panels of all users and make these panels form a *tensor*, and then ii) use tensor decomposition techniques to extract *collaborative latent factors*, which are then iii) used in the collaborative Kalman Filtering step to obtain *personalized latent factors* and iv) finally we use the personalized factors in the nowcasting for each user.

3.2 Extracting Collaborative Latent Factors

To make use of the collaborative capabilities among users, we extract (i.e., estimate) latent factors by simultaneously utilizing the panels of all users via tensor decomposition. We call the obtained latent factors *collaborative latent factors*. Before discussing the methods of obtaining collaborative latent factors, we first introduce notation and some basics of tensor and tensor decomposition.

3.2.1 Tensor and Tensor Decomposition Preliminary

A tensor is a multi-way (i.e., multidimensional) array and the high-order generalization of vectors and matrices. As shown in Figure 4, the three-dimensional array, denoted by $\mathcal{X} \in \mathbb{R}^{N\times T\times M}$, is a three-way tensor. The way of a tensor is also known as modes or orders. In this paper, we will mainly focus on three-way, i.e., third-order, tensors. The general element of a three-way tensor $\mathcal{X}$ is denoted by x_{ntm}. Analogous to columns and rows in a matrix, the column, row and tube *fibers* of a tensor contain the elements of $x_{\cdot tm}$, $x_{n\cdot m}$ and $x_{nt\cdot}$, respectively, where the symbol "$\cdot$" means all values for that subscript. Similarly, the horizontal, lateral and frontal *slices* of a tensor consist of the elements of $x_{n\cdot\cdot}$, $x_{\cdot t\cdot}$ and $x_{\cdot\cdot m}$, respectively. For convenience, we also denote the uth frontal slice of $\mathcal{X}$ by $\boldsymbol{X}^u$.

Similar to matrix factorization, tensor decomposition decomposes a tensor into the sum of a few low-rank (in particular rank-one) tensors that best approximates the given tensor. Two common tensor decomposition techniques are the Tucker and CANDECOMP/PARAFAC (CP) decomposition. The CP decomposition can be treated as a special case of the Tucker decomposition. To avoid over parameterizing the model, we will mainly focus on the CP decomposition and its variants. For a given three-way tensor $\mathcal{X} \in R^{N\times T\times M}$, the CP decomposition is expressed as

$$\mathcal{X} \approx \sum_{r=1}^{R} \boldsymbol{u}_r \circ \boldsymbol{v}_r \circ \boldsymbol{w}_r$$

where $\boldsymbol{u}_r$, $\boldsymbol{v}_r$, $\boldsymbol{w}_r$ are vectors of size $N \times 1$, $T \times 1$, and $M \times 1$, respectively, and the symbol "$\circ$" stands for the outer product[1]. Figure 4 illustrates the CP decomposition.

To obtain the CP decomposition, the following optimization problem is to be solved:

$$\min \|\mathcal{X} - \hat{\mathcal{X}}\|, \text{ where } \hat{\mathcal{X}} = \sum_{r=1}^{R} \boldsymbol{u}_r \circ \boldsymbol{v}_r \circ \boldsymbol{w}_r$$

where the symbol "$-$" denotes the element-wise subtraction (which produces a tensor $\mathcal{Z}$ with $z_{ntm} = x_{ntm} - \hat{x}_{ntm}$) and "$\|\cdot\|$" denotes the tensor norm which is (similar to the matrix Frobenius norm) defined as

$$\|\mathcal{X}\| = \sqrt{\sum_{n=1}^{N}\sum_{t=1}^{T}\sum_{m=1}^{M} x_{ntm}^2} \quad .$$

For convenience, we collect the vectors $\boldsymbol{u}_r$, $\boldsymbol{v}_r$, and $\boldsymbol{w}_r$ in matrices $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{W}$ which are of sizes $N \times R$, $T \times R$ and $M \times R$, respectively. A common method to solve the above optimization problem is the alternating least square (ALS) algorithm. The ALS first initializes $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{W}$ with singular value decomposition (SVD), and then fixes $\boldsymbol{U}$ and $\boldsymbol{V}$ and solves for $\boldsymbol{W}$ (which reduces the problem to an ordinary least square problem), and then fixes $\boldsymbol{U}$ and $\boldsymbol{W}$ and solves for $\boldsymbol{V}$ and so forth, until some convergence condition such as little or no change in $\boldsymbol{U}$, $\boldsymbol{V}$, $\boldsymbol{W}$ is met.

[1] The outer product of two vectors $\boldsymbol{a} = (a_1, \ldots, a_m)'$ and $\boldsymbol{b} = (b_1, \ldots, b_n)'$ is a matrix $\boldsymbol{M}$ of size $m \times n$ with the general entry $\boldsymbol{M}_{ij} = a_i b_j$, and similarly the outer product of a vector and a matrix is a three-way tensor.

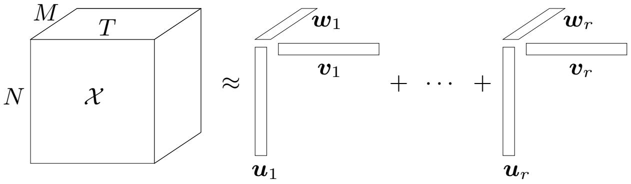


Figure 4: CP decomposition

3.2.2 First Approach: CP Decomposition

Next, we discuss the method of extracting collaborative latent factors from tensors. The simplest approach to forming the tensor is to use each of the contextual information (N), time (T) and users (M) as one mode (i.e., dimension), as illustrated in Figure 4. One difficulty, however, lies in forming the contextual information mode because each user u has different types of contextual information and hence a different panel size N^u.

It is not sensible to deploy a uniform contextual information mode (i.e., let each horizontal slice represent one type of contextual information) by pooling together all types of contextual information from each user. The reasons are: **i**) The types of contextual information for all users are numerous since there are, for instance, tens of thousands of different apps and hundreds of thousands of venues from all users, which will result in an unnecessarily large tensor. **ii**) For each individual user, she may experience only a small portion of the various types of context in the pool, which means the frontal slice for this user will include a large amount of row fibers containing only zeros, and this contradicts our goal of reducing sparsity. **iii**) Unlike the user-item matrix widely used in traditional recommendations that contains target variables (e.g., ratings) to be predicted, the contextual information is not to be completed like the user-item matrix, but to be exploited as side data to extract latent factors that summarize the temporal dynamics and sequential patterns. It is thus meaningless to incorporate all types of contextual information for a single user.

Therefore, as a first approach, we collect the individual panel of each user, assemble these panels together, and append series containing zeros to small panels to make the contextual mode uniform in size. Let M denote the number of users and

$$N = \max\{N^u | u = 1, \ldots, M\}$$

denote the number of series in the largest panel. As illustrated in Figure 4, we obtain the tensor $\mathcal{X} \in \mathbb{R}^{N\times T\times M}$, where the first, second, and third modes are the contextual information, time, and user dimensions, respectively.

After applying the CP decomposition to the obtained tensor $\mathcal{X}$, the panel of the uth user, i.e., the uth frontal slices of $\mathcal{X}$, is approximated by

$$\boldsymbol{X}^u \approx \boldsymbol{U}\boldsymbol{D}^{(u)}\boldsymbol{V}'$$

where $\boldsymbol{D}^{(u)} = \text{diag}(\boldsymbol{W}_{u,1}, \ldots, \boldsymbol{W}_{u,r})$, and $\boldsymbol{U} \in \mathbb{R}^{N\times R}$, $\boldsymbol{V} \in \mathbb{R}^{T\times R}$, $\boldsymbol{W} \in \mathbb{R}^{M\times R}$ are the matrices obtained in the CP decomposition. The matrix $\boldsymbol{V}$ contains the collaborative latent factors, i.e.,

$$\tilde{\boldsymbol{F}} = \boldsymbol{V}'.$$

At this point, the latent factor matrix for user u equals

$$\tilde{\boldsymbol{F}}^u = \tilde{\boldsymbol{F}}$$

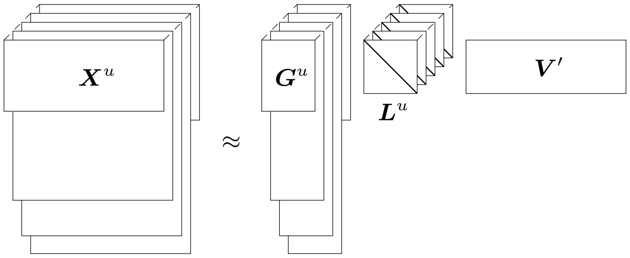


Figure 5: PARAFAC2 decomposition

and the factor loading matrix is computed by

$$\hat{\boldsymbol{\Lambda}}^u = \boldsymbol{U}^u\boldsymbol{D}^{(u)}$$

where $\boldsymbol{U}^u$ contains the first N^u rows of the matrix $\boldsymbol{U}$. The factor and loading matrices are then used in the following collaborative Kalman Filtering step.

The collaborative latent factors, different from those obtained from a single panel, contain prevalent features among a large number of users. They carry much more information on the common pattern and shared structure of the contextual data, which is not available from any single panel.

3.2.3 Second Approach: PARAFAC2 Decomposition

By making the contextual mode be of uniform size, the tensor $\mathcal{X}$ contains many manually-imposed zero elements, which bring noise into the parameter estimation procedure. To further reduce noise and data sparsity, in this method, we only assemble the panel of each user together, and make no modifications to any panel (i.e., equivalent to removing the appended zero-series from the tensor used in the CP decomposition). An example of the resulting tensor is shown in Figure 5. In this setting, the tensor is a “jagged” tensor that contains slices of various sizes in the contextual mode.

In order to obtain the collaborative latent factors, we use the PARAFAC2 [13] decomposition technique to perform tensor decomposition on the “jagged” tensor. PARAFAC2 is a variant of the CP decomposition that relaxes some constraints of the CP's. For a three-way tensor, the PARAFAC2 decomposition only requires two out of the three modes to have uniform sizes, which in our scenario are the time and user modes, while the third mode, i.e., the contextual mode, can be of various sizes. An illustration of the PARAFAC2 decomposition is shown in Figure 5. In our problem, the PARAFAC2 decomposition is equivalent to solving the optimization problem

$$\left(\tilde{\boldsymbol{F}}, \tilde{\boldsymbol{\Lambda}}^u\right)_{\{u=1,\ldots,M\}} = \min_{\boldsymbol{F},\boldsymbol{\Lambda}^u} \sum_{u=1}^{M} \|\boldsymbol{X}^u - \boldsymbol{\Lambda}^u\boldsymbol{F}\|_F^2$$

where F stands for the Frobenius norm.

After decomposition, the panel for the uth user is approximated by

$$\boldsymbol{X}^u \approx \boldsymbol{G}^u\boldsymbol{H}\boldsymbol{L}^u\boldsymbol{V}'$$

where $\boldsymbol{G}^u \in \mathbb{R}^{N^u\times R}$ is an orthonormal matrix, $\boldsymbol{H} \in \mathbb{R}^{R\times R}$ is a matrix invariant to u, $\boldsymbol{L}^u \in \mathbb{R}^{R\times R}$ is a diagonal matrix and $\boldsymbol{V} \in \mathbb{R}^{T\times R}$ is the matrix containing the collaborative latent factors. For the uth user, the initially estimated latent factors are

$$\tilde{\boldsymbol{F}}^u = \tilde{\boldsymbol{F}} = \boldsymbol{V}'$$

and the factor loading matrix is computed by

$$\hat{\boldsymbol{\Lambda}}^u = \boldsymbol{G}^u \boldsymbol{H} \boldsymbol{L}^u.$$

The PARAFAC2 decomposition is an effective approach because: i) The original structure of each panel is well approximated with no manually-imposed noise. ii) Since the temporal mode, i.e., time dimension, of the tensor is uniform across slices, PARAFAC2 is able to extract the shared temporal characteristics by utilizing such uniformity, which is vital in the intent monitoring problem. iii) The flexibility of PARAFAC2, i.e., allowing one mode to be of various sizes, is particularly suitable for the non-uniform contextual mode, which introduces no extra constraints and hence obtains more information than the CP decomposition. Extensive experiments (cf. Section 4) also validate the superiority of the PARAFAC2 decomposition. Therefore, we use this approach in the proposed model.

3.3 Collaborative Kalman Filtering

For the intent monitoring problem, it is not sufficient to utilize only the collaborative latent factors obtained from the tensor decomposition. The collaborative factors only reflect the static common structure of the historical and side data. The dynamics of the factors and hence the correlation and co-movement of time series, however, are not fully taken into consideration. Moreover, the factors are extracted from the data of all users and hence are the same for all users, which is not suitable for personalized intent monitoring. Therefore, for each user u, we apply the Kalman Filter to the collaborative factors $\tilde{\boldsymbol{F}}^u$ and the panel $\boldsymbol{X}^u$ to obtain the final estimation $\hat{\boldsymbol{F}}^u$ of the latent factors. The factors $\hat{\boldsymbol{F}}^u$ reflects both the collaborative and personalized patterns and the static and dynamic structures of all the available data. For notational simplicity, in the sequel, we will drop the superscript u as the following parameter estimation procedure is for each user.

3.3.1 Estimating Required Parameters

For the proposed model, we have obtained the estimations of the factors and loading matrix. To utilize the Kalman Filer for each user, we first estimate the remaining parameters of Eq. 1 and 2. By applying regression on the estimated factors, the estimations of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are computed by

$$\hat{\boldsymbol{A}} = \sum_{t=2}^{T} \tilde{\boldsymbol{f}}_t \tilde{\boldsymbol{f}}'_{t-1} \left(\sum_{t=2}^{T} \tilde{\boldsymbol{f}}_{t-1} \tilde{\boldsymbol{f}}'_{t-1} \right) \text{ and } \hat{\boldsymbol{B}} = \boldsymbol{C}\boldsymbol{E}^{\frac{1}{2}}$$

respectively, where $\boldsymbol{E} \in \mathbb{R}^{Q \times Q}$ is a diagonal matrix containing the largest Q eigenvalues of matrix $\boldsymbol{\Omega}$ (defined below), $\boldsymbol{C} \in \mathbb{R}^{R \times Q}$ is a matrix containing the corresponding eigenvectors, and

$$\boldsymbol{\Omega} = \frac{1}{T-1} \sum_{t=2}^{T} \tilde{\boldsymbol{f}}_t \tilde{\boldsymbol{f}}'_t - \hat{\boldsymbol{A}} \left(\frac{1}{T-1} \sum_{t=2}^{T} \tilde{\boldsymbol{f}}_{t-1} \tilde{\boldsymbol{f}}'_{t-1} \right) \hat{\boldsymbol{A}}'.$$

Let the sample covariance matrix $\boldsymbol{S}$ of the historical and side data (after standardized normalization) be

$$\boldsymbol{S} = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{x}_t \boldsymbol{x}'_t.$$

The covariance matrix $\boldsymbol{\Psi}$ in Eq. 1 is estimated by

$$\hat{\boldsymbol{\Psi}} = \operatorname{diag}(\boldsymbol{S} - \boldsymbol{P}\boldsymbol{\Sigma}\boldsymbol{P}')$$

where $\boldsymbol{\Sigma} \in \mathbb{R}^{R \times R}$ is a diagonal matrix containing the largest R eigenvalues of $\boldsymbol{S}$, $\boldsymbol{P} \in \mathbb{R}^{N \times R}$ is a matrix consisting of the corresponding eigenvectors with $\boldsymbol{P}'\boldsymbol{P} = \boldsymbol{I}$, and the "diag" means keeping only the elements at the main diagonal.

3.3.2 Correcting the Factors with Kalman Filter

With all required parameters at hand, we reestimate the factors by applying the Kalman Filter. Let the a priori and a posteriori factors and the corresponding measurement error covariance matrices at each time step be $\tilde{\boldsymbol{f}}_t$, $\hat{\boldsymbol{f}}_t$, $\tilde{\boldsymbol{P}}_t$ and $\hat{\boldsymbol{P}}_t$, respectively. By Eq. 2, in the time update (prediction) step, the a priori factors for the next time step are computed by

$$\tilde{\boldsymbol{f}}_t = \hat{\boldsymbol{A}} \hat{\boldsymbol{f}}_{t-1} + \hat{\boldsymbol{B}} \boldsymbol{\omega}_t$$

and the a priori error covariance is computed by

$$\tilde{\boldsymbol{P}}_t = \hat{\boldsymbol{A}} \hat{\boldsymbol{P}}_{t-1} \hat{\boldsymbol{A}}' + \hat{\boldsymbol{\Psi}}_t.$$

In the measurement update (correction) step, the Kalman gain $\boldsymbol{K}_t$ is obtained by considering the ratio of the measurement and transition error covariance and equals

$$\boldsymbol{K}_t = \tilde{\boldsymbol{P}}_t \hat{\boldsymbol{\Lambda}}' (\hat{\boldsymbol{\Lambda}} \tilde{\boldsymbol{P}}_t \hat{\boldsymbol{\Lambda}}' + \hat{\boldsymbol{\Psi}}_t)^{-1}.$$

With the Kalman gain, the a priori (collaborative) factors are corrected by utilizing the user's panel, and the corrected, i.e., personalized, factors are estimated by

$$\hat{\boldsymbol{f}}_t = \tilde{\boldsymbol{f}}_t + \boldsymbol{K}_t (\boldsymbol{x}_t - \hat{\boldsymbol{\Lambda}} \tilde{\boldsymbol{f}}_t).$$

The a posteriori covariance used for next time step is then computed by

$$\hat{\boldsymbol{P}}_t = (\boldsymbol{I} - \boldsymbol{K}_t \hat{\boldsymbol{\Lambda}}) \tilde{\boldsymbol{P}}_t.$$

The a posteriori factors $\hat{\boldsymbol{f}}_t$ are the estimated personalized latent factors we need for the next step. In practice, we can also apply the Kalman Smoother (RTS Smoother) to fully exploit all the available data. Following existing work [33], the above approach is referred to as the collaborative Kalman Filtering because it uses the same latent factors extracted from the data of all users.

3.4 Regression for Nowcasting

The final step is to establish the relationship between the personalized latent factors and the intent, i.e., to estimate the coefficients in Eq. 3. We use the ordinary least square (OLS) regression to estimate the coefficients α and $\boldsymbol{\beta}$. In particular, let τ be the last time step where the intent is available. Let matrix $\bar{\boldsymbol{F}} = (\hat{\boldsymbol{f}}_1, \ldots, \hat{\boldsymbol{f}}_\tau)$ contain the personalized latent factors until time step τ. Let the corresponding intent likelihood in the τ time steps be $\boldsymbol{y} = (y_1, \ldots, y_\tau)$. The coefficients α and $\boldsymbol{\beta}$ are then estimated by running OLS with $\bar{\boldsymbol{F}}$ and $\boldsymbol{y}$. The linear function of Eq. 3 is then used in the intent monitoring for following time steps. The threshold θ we use is the median of the fitted intent likelihood $\hat{y}_t$ for $1 \leq t \leq \tau$. If $\hat{y}_{\tau+\delta} > \theta$ for any $\delta > 0$, we say the user has the intent, i.e., $\mathcal{I}_{\Gamma_{\tau+\delta}}(\gamma) = 1$.

4. EXPERIMENTS

We use the contextual recommendation task in digital assistants to empirically evaluate the collaborative nowcasting model. The experiments are conducted on a 64-bit Windows computer with a 2.8GHz Intel(R) CPU and 24GB main memory. The algorithms are implemented with Matlab.

4.1 Data Preparation

The data sets we use are sampled from the recommendation log of a commercial digital assistant between 10 June 2015 and 9 July 2015. When a user uses the digital assistant, various types of cards carrying information such as news, weather, jokes, etc. are recommended. If the user is interested in a card, she may click the card for more information. We use such click as an indicator of the intent. Different types of cards indicate different types of intent. We pick out eight types of intent that are commonly monitored in most digital assistant applications. The eight types cover the aspects of news, events, weather, places, finance, calendar, traffic and sports, respectively. The sampled data sets for these types of intent in total contain $20,807$ anonymous users. For each type of intent and each user, we collect the user's intent-related context, particularly the apps used and the venues visited by the user. To further protect the user's privacy, we use an anonymous identifier for each app and venue, and remove the latitude and longitude of the venue.

4.2 Evaluation Criteria

We use the macro and micro F-measures on the predicted intent to evaluate the model performance. Let ρ be the number of testing time steps. We denote by $\boldsymbol{s}^u = (s_1^u, \ldots, s_\rho^u)'$ the true intent of user u, where $s_t^u = 1$ means the user has the given intent (i.e., clicks the corresponding card) and $s_t^u = 0$ means no such intent at time step t. Let $\hat{\boldsymbol{s}}^u = (\hat{s}_1^u, \ldots \hat{s}_\rho^u)'$, $\hat{s}_t^u \in \{0, 1\}$ be the predicted intent. The precision and recall for user u are computed by

$$\text{Prec}^u = \frac{\boldsymbol{s}^{u\prime}\hat{\boldsymbol{s}}^u}{\boldsymbol{1}'\hat{\boldsymbol{s}}^u} \quad \text{and} \quad \text{Rec}^u = \frac{\boldsymbol{s}^{u\prime}\hat{\boldsymbol{s}}^u}{\boldsymbol{1}'\boldsymbol{s}^u},$$

respectively. Let $\overline{\text{Prec}}$ and $\overline{\text{Rec}}$ be the average precision and recall among all users, respectively. The macro F-measure equals

$$\text{Macro F-measure} = 2 \times \frac{\overline{\text{Prec}} \times \overline{\text{Rec}}}{\overline{\text{Prec}} + \overline{\text{Rec}}}.$$

The precision and recall considering all testing instances are computed by

$$\text{Prec} = \frac{\sum_u \boldsymbol{s}^{u\prime}\hat{\boldsymbol{s}}^u}{\sum_u \boldsymbol{1}'\hat{\boldsymbol{s}}^u} \quad \text{and} \quad \text{Rec} = \frac{\sum_u \boldsymbol{s}^{u\prime}\hat{\boldsymbol{s}}^u}{\sum_u \boldsymbol{1}'\boldsymbol{s}^u},$$

respectively, and the micro F-measure equals

$$\text{Micro F-measure} = 2 \times \frac{\text{Prec} \times \text{Rec}}{\text{Prec} + \text{Rec}}.$$

The macro F-measure reflects the average performance among all users by weighting equally the precision and recall of each user. The micro F-measure evaluates the performance of the model per recommendation instance, which has a bias towards the users who have more intent records.

4.3 Methods to Compare

The methods we compare with the collaborative nowcasting model **CNowcast** include

- **BoostedTree**. BoostedTree [35] is an ensemble of regression trees (decision trees). It is used in existing contextual ranking models [32] and gives the best performance on the intent monitoring problem among several classic algorithms we have tried including linear regression, SVM, etc.

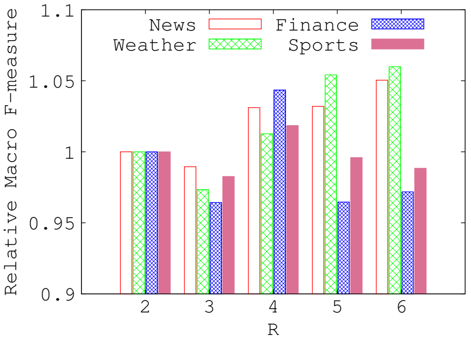


Figure 6: Relative performance of the collaborative nowcasting model to $R = 2$ when R is varied from 2 to 6 for four selected types of intent.

- **FM**. Factorization machine (FM) [29] is a state-of-the-art method for next-basket recommendations [30], which recommend the items that will be in the user's shopping cart during the next time step. It also effectively performs many other recommendation tasks.

- **NowcastIndi**. This is the nowcasting model [12] introduced in Section 2.2. In this method, the model is applied to the panel of each individual user.

- **CNowcastCP**. In this method, we use the CP tensor decomposition to obtain the collaborative latent factors, which is introduced in Section 3.2.2.

The temporal features are implicitly modeled by the nowcasting related methods. To help the BoostedTree and FM models utilize the temporal features, we also add the time of day and day of week as additional features. We use the first three quarters of the data sets to train the model and the remaining for testing. Unless otherwise specified, we parameterize the collaborative nowcasting model with four factors and two transition noise: $R = 4$, $Q = 2$, and use default parameter values for all other models.

4.4 Results

4.4.1 Effect of Parameters R and Q

Effect of R. We first study the effect of the number of factors R (i.e., dimension of $\boldsymbol{f}_t$) by varying R from 2 to 6. Figure 6 shows the relative performance of the collaborative nowcasting model for four types of intent: news, weather, finance, and sports. We can see that the performance, measured by the macro F-measure, of the model first decreases and then increases when R varies from 2 to 4. This is because when $R = 2$, the fundamental structure and movement of the context can already be effectively captured (this is consistent with the findings in [11] that many macroeconomic variables can be captured by two factors). When R increases to 3, the increased uncertainty brought by estimating more parameters outruns the marginal benefits from capturing moderately more dynamics of the context. However, this situation is reversed when R increases to 4. When we further increase R to 5 and 6, the performance of the model keeps increasing moderately for news and weather types of intent, but decreases for finance and sports types of intent. The reason for the increase is the same as before. The decrease is because 5 or 6 factors make the model overfit for these two types of intent. We will discuss in detail the difference between different types of intent in Section 4.4.3.

Model	News	Events	Weather	Places	Finance	Calendar	Traffic	Sports
BoostedTree	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
FM	0.738	0.747	0.922	1.791	2.770	5.788	0.192	0.699
NowcastIndi	2.586	3.720	5.806	24.26	14.28	14.61	2.387	5.181
CNowcastCP	2.625	3.845	5.796	25.54	13.66	17.27	2.940	5.533
CNowcast	**3.024**	**4.410**	**6.479**	**28.23**	**16.50**	**18.13**	**3.068**	**6.426**

Table 2: The macro F-measure of each model relative to BoostedTree when $\Delta = 1$ hour

Model	News	Events	Weather	Places	Finance	Calendar	Traffic	Sports
BoostedTree	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
FM	1.327	1.618	2.207	10.32	0.767	5.502	0.255	1.085
NowcastIndi	1.832	2.282	2.870	20.02	1.742	7.756	0.860	1.352
CNowcastCP	1.994	2.437	3.014	21.85	1.731	9.251	1.098	1.540
CNowcast	**2.130**	**2.688**	**3.155**	**23.19**	**1.910**	**9.441**	**1.116**	**1.669**

Table 3: The micro F-measure of each model relative to BoostedTree when $\Delta = 1$ hour

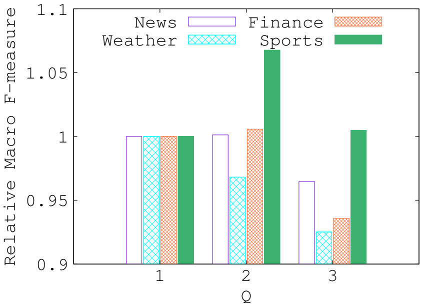


Figure 7: Relative performance of the collaborative nowcasting model to $Q = 1$ when Q is varied from 1 to 3 for four selected types of intent.

From the figure we can also see that the performance variance of the proposed model is very small. In most cases, the variance is less than 5%. This indicates that the proposed model is robust to the choice of the number of factors. The relative performance measured by the micro F-measure is similar and hence omitted.

Effect of Q. Figure 7 shows the relative macro F-measure of the model when Q is changed from 1 to 3. We can see that when $Q = 2$, the performance of the model slightly increases (except for weather). When Q increases to 3, the performance drops. This indicates that a two dimensional white noise can effectively model the other aspects in the dynamic transition between factors. The relative micro F-measure is similar and thus omitted.

4.4.2 Comparison across Models

Tables 2 and 3 respectively present the macro and micro F-measures of each method on the eight types of intent when the monitoring granularity is one hour (i.e., $\Delta = 1$ hour). To protect the usage statistics of the proprietary digital assistant and for expositional convenience, we report each method's relative F-measure to the BoostedTree model.

Cnowcast vs. BoostedTree. From the two tables we can see that the Cnowcast method consistently outperforms the BoostedTree method, and the performance advantage is up to 28 times. This demonstrates that the proposed model is able to effectively utilize the user's real-time context, while the BoostedTree, although providing strong performance in many other problems, fails to capture the structure and dynamics of the context and intent. We can also see that the superiority of Cnowcast over BoostedTree is larger on the macro F-measure than micro F-measure. This indicates that the proposed model is able to monitor the intent of much more users effectively than the BoostedTree method. Therefore, the proposed model is more suitable for real-world applications where there are a large number of users and every user counts.

Cnowcast vs. FM. The Cnowcast method also consistently outperforms the FM method, with a performance advantage of up to 16 times (for places and traffic columns in Table 2). This shows that although the FM method provides state-of-the-art performance on the short-term next-basket recommendation problem, it is unable to make effective contemporaneous recommendations in a highly dynamic scenario like the intent monitoring problem. From Table 2, we can see that for many types of intent, FM also has a much lower macro F-measure than the Cnowcast method. This again supports that the proposed method is able to provide effective recommendations for more users and is more appropriate for real-world applications.

Cnowcast vs. NowcastIndi. From the two tables, we can see that the collaborative nowcasting model consistently and greatly outperforms the individual nowcasting model, in terms of both macro and micro F-measures. This confirms that by exploiting the panels of all users simultaneously, the proposed model is able to obtain the collaborative latent factors that capture the common characteristics for the intent-related context, and hence utilizes the collaborative capabilities of all users. This also validates that the proposed model can effectively address the data sparsity and personalized nowcast problem encountered by the nowcasting model when it is applied to the intent monitoring problem.

Cnowcast vs. CnowcastCP. We can see from Tables 2 and 3 that, across all types of intent, the proposed model significantly outperforms the CnowcastCP model (which appends zero-series to obtain the collaborative latent factors) in terms of both macro and micro F-measures. This validates that by keeping the panels in their original forms, the proposed model avoids the manually-imposing noise, which gives the model a significant advantage in effectively modeling the swiftly changing context and intent.

4.4.3 Comparison across Intent Types

From Tables 2 and 3, we can observe that the performance

Model	News	Events	Weather	Places	Finance	Calendar	Traffic	Sports
BoostedTree	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
FM	0.877	1.102	1.459	3.465	1.263	9.179	1.332	1.395
NowcastIndi	1.746	2.643	4.403	12.70	3.788	14.92	5.800	4.221
CNowcastCP	1.766	2.513	4.329	12.16	3.412	15.33	5.483	4.195
CNowcast	**1.963**	**2.950**	**4.904**	**14.13**	**4.680**	**16.95**	**7.377**	**5.264**

Table 4: The macro F-measure of each model relative to BoostedTree when $\Delta = 4$ hours

Model	News	Events	Weather	Places	Finance	Calendar	Traffic	Sports
BoostedTree	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
FM	1.040	1.280	1.497	4.951	0.932	7.231	1.114	1.276
NowcastIndi	1.365	1.733	2.223	8.073	1.526	8.019	1.997	1.625
CNowcastCP	1.422	1.686	2.301	7.893	1.427	8.447	2.048	1.636
CNowcast	**1.513**	**1.927**	**2.432**	**9.026**	**1.822**	**8.888**	**2.572**	**2.037**

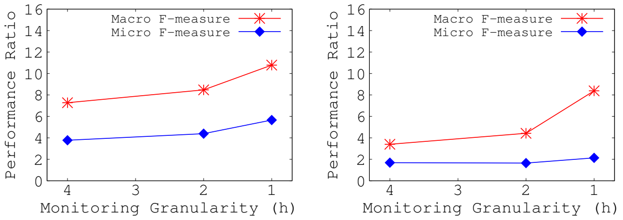
Table 5: The micro F-measure of each model relative to BoostedTree when $\Delta = 4$ hours

of different models varies greatly across different types of intent. **i)** For the places intent, the proposed model outperforms the BoostedTree and FM methods significantly more than other types, in terms of both macro and micro F-measures. This is because a place's type of intent depends on a more complex context than other types. The BoostedTree and FM methods are unable to effectively model the context and the extra complexity makes it more difficult for them to produce effective recommendations. **ii)** From Table 3, we can see that for the finance and traffic types of intent, FM performs worse than the BoostedTree method, and for sports, its performance is very close to that of the BoostedTree. In addition, for these three types of intent, the advantage of the proposed model over the BoostedTree method is also lower than other types (less than two times). These phenomenon are due to that the three types of intent are related to a relatively less complicated and less dynamic context. The modeling of such context can be to some extent narrowed down by the time of day and day of week features (e.g., users often check stock prices during the exchange time on weekdays). Nevertheless, for any type of intent, the related-context consists of much more information than only the time-related features. The best performance of the proposed method demonstrates that it can effectively model the structure of the context and the dynamic correlation between the context and intent, regardless of the intent type and complexity of the context.

4.4.4 *Comparison across Monitoring Granularity*

Tables 4 and 5 present the macro and micro F-measures of each method when the monitoring granularity is four hours, respectively. With the decrease of granularity (from 1 hour to 4 hours), the user's panel becomes less sparse, which gives the BoostedTree, FM and NowcastIndi methods an opportunity to outperform the proposed model if data sparsity is the main impediment. From these tables, we can see that the proposed model still consistently performs the best, and outperforms the other methods significantly. This indicates that the worse performance of the other methods is not mainly due to data sparsity, but because they fail to capture the structure and dynamics of the context and intent.

Figure 8 presents the average performance ratio of the proposed model (across all types of intent) to the BoostedTree and FM methods when the monitoring granularity Δ is varied from four hours to one hour, respectively. From the figures we can see that with the increase of the monitoring granularity (i.e., from 4 hours to 1 hour) the advantage of the proposed model over the BoostedTree and FM models also becomes increasingly larger. With the increase of granularity, the intent is closer to the present, i.e, "now". The increasing advantage indicates that the proposed model is particularly suitable for the nowcasting scenario where the user's real-time intent is closely tracked.



(a) Ratio to BoostedTree (b) Ratio to FM

Figure 8: Average performance ratio of the collaborative nowcasting model to BoostedTree and FM across all types of intent when Δ varies from 4 to 1.

From these results, we can see that, under various scenarios and in terms of both macro and micro F-measures, the proposed collaborative nowcasting model consistently performs best and outperforms state-of-the-art methods by a significant margin. The effectiveness and superiority of the proposed collaborative nowcasting model for the intent monitoring problem is thereby empirically confirmed.

5. RELATED WORK

5.1 Nowcasting Models

The term nowcasting is first used in meteorology, which refers to: monitoring the current weather condition and forecasting the weather within the next three (or six) hours [4]. The current weather condition for a certain area can be highly dynamic and may not be directly observable by a limited number of observation stations. The side data that can be used in weather nowcast are radar reflectivity and satellite imagery [9][34]. With the exponential increase of real-time surface observations, more and more side data are available for weather nowcast such as the vertical atmospheric conditions provided by commercial aircraft during ascent and descent [27], water vapor distributions provided by ground-based GPS receivers [23], and large amounts of social media data from Facebook, Twitter etc. [25][26]. The model used, for instance in thunderstorm nowcasting [9], mainly uses a linear regression model with double exponential smoothing to effectively identify and track the storm and other physical atmospheric conditions. The model used in inclement weather nowcasting [21] with *tweets* (posts on Twitter) as side data uses the sum aggregate of weather related tweets within a certain spatio-temporal range followed by a linear regression to predict the impact of inclement weather. The variable of interest and side data that these models focus on are of quite different nature than the intent monitoring problem, and hence are inapplicable.

Nowcasting is then used in macroeconomics [12] to monitor the contemporaneous value of a variable of interest that is officially published with a significant lag such as the GDP. The side data used in such nowcast are macroeconomic figures that are released much more frequently than the variable of interest, which for instance in GDP nowcast includes: personal consumption, industrial production, surveys, financial variables (e.g., interest rates, stock prices, consumer price index (CPI)), Google Trend data [31] etc. A widely used nowcasting model is proposed in the seminal paper of Giannone etc. [12], which is now applied in GDP nowcasting by many agencies [7] including the Federal Reserve Board and European Central Bank.

Recently, nowcasting is studied in data mining to obtain real-time information describing real-world phenomena such as the levels of rainfall, regional influenza-like illness rates [19], or the mood of the nation on some on-going events [20]. The side data currently exploited include search engine query log (e.g., Google Trend data) [10], posts in social media [19] like Twitter, etc. The model in [19] uses tweets and the sparse learning method Bootstrapped Least Absolute Shrinkage and Selection Operator to select a consistent subset of textual features from the n-grams of web encyclopedias, and then regression is applied on the selected features and variable of interest. This model cannot apply to intent monitoring because it cannot address the personalized scenario. A non-trivial task is to first build from a high-quality textual corpus an initial set of good candidate textual features related to the personalized intent.

5.2 Contextual Recommendations

Collaborative Filtering (CF). CF is a technique widely used in traditional recommendation systems. The essential idea of CF is to make use of the data from other (in particular similar) users or items. Two common CF techniques are matrix factorization (MF) and neighborhood methods [18]. In the MF approach, the user-item matrix, containing the ratings of each user to each item, is factorized into the product of two low-rank matrices. In the neighborhood approach, recommendations are based on similar items or users. One problem in CF is that the user's context is not considered, which makes it inapplicable to intent monitoring.

Time-aware recommendations. By additionally considering the gradual evolving of user preferences and item attributes, there are several time-aware recommendation models. The timeSVD++ model [17] augments the MF approach with gradually changing user preferences. The model includes in the MF a time-related preference bias, which is based on the mean date during the period a user rates the items. The dynamic Poisson factorization [8] extends the timeSVD++ model by further allowing for progressively evolving item attributes. The auto-regressive moving average model in [36] applies on the daily time series of token features extracted from product reviews and recommends the items expected to be popular in the future. These approaches cannot apply to intent monitoring because the context they consider is only time, and the gradually evolving preferences or attributes are quite different from the frequently varied intent.

Context-aware recommendations. Besides time, context-aware recommendation models [5][22] try to incorporate more evidence of a specific situation such as the location, device, purchasing purpose, etc. to model the user preferences on unseen items. Assuming that there are static latent contextual factors that influence the user preferences, these factors can be learned with the probabilistic latent semantic analysis (PLSA) [14] or hierarchical linear models (HLMs) [28]. The PLSA and HLM models, however, cannot apply to the intent monitoring problem because the contextual factors are required to be static while in our problem the latent factors are highly dynamic and have strong serial and cross-sectional correlation. The model in [24] considers the dynamic contextual factors over the course of an interaction, e.g., conversation, with the user. However, in the proactive experiences where we monitor the intent, there is no interaction with the user. The multiverse recommendation [16] uses a multidimensional tensor: user-item-context, to model user preferences. This model cannot apply to intent monitoring either because the tensor in our problem is not to be completed, but to be utilized to continuously nowcast the intent at the last time step. The model in [32] addresses the proactive experiences in search engines and digital assistants. Unlike monitoring intent, it uses the reactive search history to re-rank a given list of cards. Therefore, the model cannot apply to intent monitoring.

6. CONCLUSIONS

Nowcasting the user's real-time intent is required by emerging proactive experiences in digital assistants. It is an essential step for recommending the right content that fulfills the user's contemporaneous need. The problem has many new characteristics that traditional recommendations lack, which requires the deployment of new models. The proposed collaborative nowcasting model utilizes the collaborative capabilities among users and successfully addresses the sparsity problem. It generates the collaborative latent factors that summarize the co-movement and temporal structures shared by all panels. It also generates the personalized latent factors that effectively model the dynamics and correlation of the contextual data and are suitable for the task of personalized intent monitoring. The collaborative nowcasting model thus successfully resolves the new characteristics and is able to effectively nowcast the intent. Extensive experiments with real-world data sets have demonstrated that the collaborative nowcasting model outperforms various baselines by a significant margin. We hope that the studied problem and model can draw more attention to new paradigms of recommendations on mobile intelligent devices.

Acknowledgments. Yu Sun is partially supported by the National Natural Science Foundation of China under Grant No. 61402155.

7. REFERENCES

[1] http://www.google.com/landing/now/.

[2] http://dev.windows.com/en-us/cortana.

[3] http://www.apple.com/ios/whats-new/.

[4] http://glossary.ametsoc.org/wiki/Nowcast.

[5] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer, 2011.

[6] M. Banbura, D. Giannone, M. Modugno, and L. Reichlin. Now-casting and the real-time data flow. *Handbook of Economic Forecasting*, 2013.

[7] M. M. Bańbura, D. Giannone, and L. Reichlin. Nowcasting. *The Oxford Handbook of Economic Forecasting*, 2012.

[8] L. Charlin, R. Ranganath, J. McInerney, and D. M. Blei. Dynamic poisson factorization. In *RecSys*, pages 155–162, 2015.

[9] M. Dixon and G. Wiener. Titan: Thunderstorm identification, tracking, analysis, and nowcasting-a radar-based methodology. *Journal of Atmospheric and Oceanic Technology*, 10(6):785–797, 1993.

[10] B. Duncan and C. Elkan. Nowcasting with numerous candidate predictors. In *ECML PKDD*, pages 370–385. 2014.

[11] D. Giannone, L. Reichlin, and L. Sala. Monetary policy in real time. In *NBER Macroeconomics Annual 2004, Volume 19*, pages 161–224. 2005.

[12] D. Giannone, L. Reichlin, and D. Small. Nowcasting: The real-time informational content of macroeconomic data. *Journal of Monetary Economics*, 55(4):665–676, 2008.

[13] R. A. Harshman. Parafac2: Mathematical and technical notes. *UCLA Working Papers in Phonetics*, 22(3044):122215, 1972.

[14] T. Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *SIGIR*, pages 259–266, 2003.

[15] D. Jannach, L. Lerche, and M. Jugovac. Adaptation and evaluation of recommendations for short-term shopping goals. In *RecSys*, pages 211–218, 2015.

[16] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In *RecSys*, pages 79–86, 2010.

[17] Y. Koren. Collaborative filtering with temporal dynamics. In *KDD*, pages 447–456, 2009.

[18] Y. Koren and R. Bell. Advances in collaborative filtering. In *Recommender systems handbook*, pages 145–186. 2011.

[19] V. Lampos and N. Cristianini. Nowcasting events from the social web with statistical learning. *TIST*, 3(4):72, 2012.

[20] T. Lansdall-Welfare, V. Lampos, and N. Cristianini. Nowcasting the mood of the nation. *Significance*, 9(4):26–28, 2012.

[21] L. Lin, M. Ni, Q. He, J. Gao, A. W. Sadek, and T. I. T. I. Director. Modeling the impacts of inclement weather on freeway traffic speed: An exploratory study utilizing social media data. In *Transportation Research Board 94th Annual Meeting*, 2015.

[22] Q. Liu, H. Ma, E. Chen, and H. Xiong. A survey of context-aware mobile recommendations. *International Journal of Information Technology & Decision Making*, 12(01):139–172, 2013.

[23] A. E. MacDonald, Y. Xie, and R. H. Ware. Diagnosis of three-dimensional water vapor using a gps network. *Monthly Weather Review*, 130(2):386–397, 2002.

[24] T. Mahmood, F. Ricci, and A. Venturini. Improving recommendation effectiveness: Adapting a dialogue strategy in online travel planning. *Information Technology & Tourism*, 11(4):285–302, 2009.

[25] C. Mass. Nowcasting: The promise of new technologies of communication, modeling, and observation. *Bulletin of the American Meteorological Society*, 93(6):797–809, 2012.

[26] C. Mass and C. F. Mass. Nowcasting: The next revolution in weather prediction. *Bulletin of the American Meteorological Society*, 2011.

[27] W. R. Moninger, S. G. Benjamin, B. D. Jamison, T. W. Schlatter, T. L. Smith, and E. J. Szoke. Evaluation of regional aircraft observations using tamdar. *Weather and Forecasting*, 25(2):627–645, 2010.

[28] S. W. Raudenbush and A. S. Bryk. *Hierarchical linear models: Applications and data analysis methods*, volume 1. 2002.

[29] S. Rendle. Factorization machines with libFM. *TIST*, 3(3):1–22, 2012.

[30] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *WWW*, pages 811–820, 2010.

[31] S. L. Scott and H. R. Varian. Predicting the present with bayesian structural time series. *International Journal of Mathematical Modelling and Numerical Optimisation*, 5(1-2):4–23, 2014.

[32] M. Shokouhi and Q. Guo. From queries to cards: Re-ranking proactive card recommendations based on reactive search history. In *SIGIR*, pages 695–704, 2015.

[33] J. Z. Sun, D. Parthasarathy, and K. R. Varshney. Collaborative kalman filtering for dynamic matrix factorization. *Transactions on Signal Processing*, 62(14):3499–3509, 2014.

[34] J. W. Wilson, N. A. Crook, C. K. Mueller, J. Sun, and M. Dixon. Nowcasting thunderstorms: A status report. *Bulletin of the American Meteorological Society*, 79(10):2079–2099, 1998.

[35] Q. Wu, C. J. Burges, K. M. Svore, and J. Gao. Adapting boosting for information retrieval measures. *Information Retrieval*, 13(3):254–270, 2010.

[36] Y. Zhang, M. Zhang, Y. Zhang, G. Lai, Y. Liu, H. Zhang, and S. Ma. Daily-aware personalized recommendation based on feature-level time series analysis. In *WWW*, pages 1373–1383, 2015.