CatchTartan: Representing and Summarizing Dynamic Multicontextual Behaviors

Meng Jiang[1], Christos Faloutsos[2], Jiawei Han[1]
[1]Department of Computer Science, University of Illinois Urbana-Champaign, IL, USA
[2]Computer Science Department, Carnegie Mellon University, PA, USA
mjiang89@illinois.edu, christos@cs.cmu.edu, hanj@illinois.edu

ABSTRACT

Representing and summarizing human behaviors with rich contexts facilitates behavioral sciences and user-oriented services. Traditional behavioral modeling represents a behavior as a tuple in which each element is one contextual factor of one type, and the tensor-based summaries look for high-order dense blocks by clustering the values (including timestamps) in each dimension. However, the human behaviors are *multicontextual* and *dynamic*: (1) each behavior takes place within multiple contexts in a few dimensions, which requires the representation to enable non-value and set-values for each dimension; (2) many behavior collections, such as tweets or papers, evolve over time. In this paper, we represent the behavioral data as a two-level matrix (temporal-behaviors by dimensional-values) and propose a novel representation for behavioral summary called Tartan that includes a set of dimensions, the values in each dimension, a list of consecutive time slices and the behaviors in each slice. We further develop a propagation method CATCHTARTAN to catch the dynamic multicontextual patterns from the temporal multidimensional data in a *principled* and *scalable* way: it determines the meaningfulness of updating every element in the Tartan by minimizing the encoding cost in a compression manner. CATCHTARTAN outperforms the baselines on both the accuracy and speed. We apply CATCHTARTAN to four Twitter datasets up to 10 million tweets and the DBLP data, providing comprehensive summaries for the events, human life and scientific development.

Categories and Subject Descriptors

H.3.5 [**Information Systems**]: Information Storage and Retrieval - On-line Information Services; J.4 [**Computer Applications**]: Social and Behavioral Sciences

Keywords

Behavior Representation; Behavior Summarization; Minimum Description Length

1. INTRODUCTION

Behavioral representation and summarization is a fundamental component of behavioral scientific discovery: it supports the systematic analysis and investigation of human behaviors. It is also a fundamental problem in many user-oriented applications for a better understanding of the event from news, human life from tweets, and the scientific development from publications. However, it is rather challenging for the following two characteristics of the behaviors [10, 4]. (Terms and their definitions are given in Table 1.)

First, human behaviors are *multicontextual*: a behavior consists of one or multiple types (i.e., dimensions) of contextual factors [12], and it has one or multiple values in each dimension. Take the "Super Bowl" tweet in Figure 1a as an example: it has several dimensions such as the user, phrase, hashtag and shorten-URL, and this behavior has one user, one hashtag, *several* phrases and *no* URL. The publishing-paper behavior also has multiple values in the author, keyword and cited-paper dimensions. The representation should enable different combinations of the dimensions and a non-value/set-value setting of the dimensional values.

Second, human behaviors are *dynamic*. They naturally evolve with the changing of personality, physical environment and so-

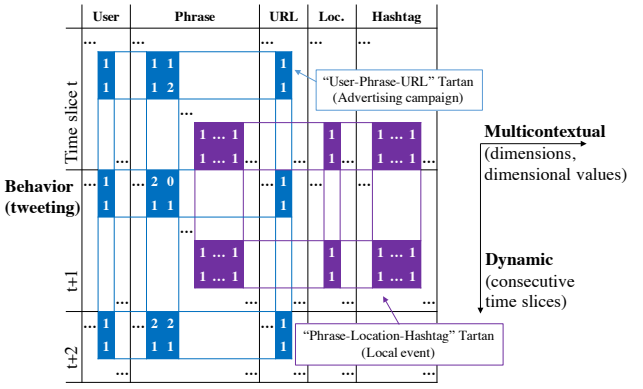
Tweeting behavior (Twitter) 20:03:09 @ebekahwsm : this better be the best halftime show ever in the history of halftimes shows. ever. #SuperBowl

Time slice	User	Location	Phrase	Hashtag	URL
20:00-20:30	@ebekahwsm	∅	{best halftime show, in the history, halftimes shows}	{#SuperBowl}	∅

Publishing-paper behavior (DBLP) SIGKDD 2009 "Sentiment analysis of blogs by combining lexical knowledge with text classification"

Time slice	Author	Venue	Keyword	Cited papers
2009	{P. Melville, W. Gryc, R. Lawrence}	SIGKDD	{sentiment analysis, lexical knowledge, text classification}	{p81623, p84395, p95393, p95409, p99073, p116349 …}

(a) Representing a behavior with dimensional values including non-value and set-values, instead of one guaranteed value.



(b) Tartans in a "two-level matrix": dimensions and values on the columns, time slices and behaviors on the rows.

Figure 1: The representation and summarization of dynamic multicontextual behaviors: it takes every behavior while the tensor fails.

KDD '16, August 13-17, 2016, San Francisco, CA, USA

DOI: http://dx.doi.org/10.1145/2939672.2939749

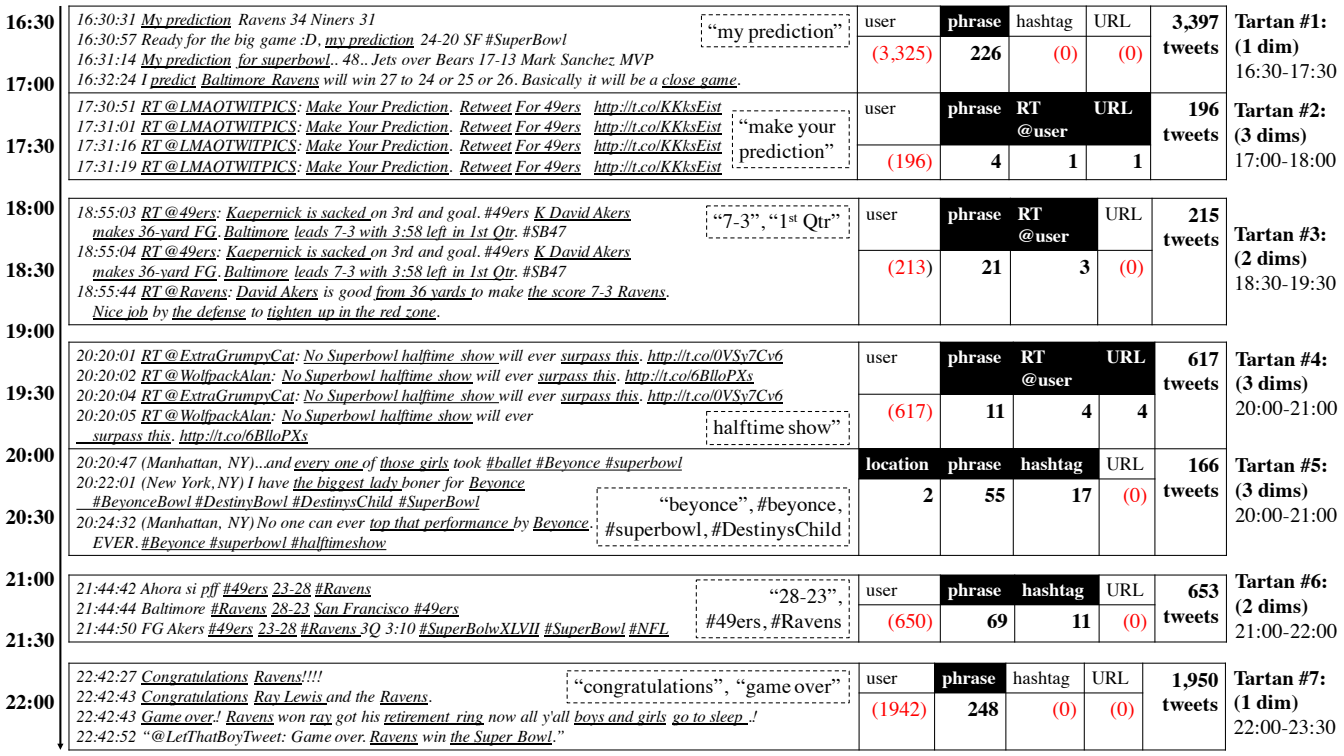


Figure 2: CATCHTARTAN captures multiple phases (e.g., score prediction, half-time show) in the "Super Bowl 2013" event by representing the dynamic and multicontextual patterns with "Tartans" (consecutive time slices, a set of dimensions and dimensional value sets).

Term	Definition
Dimension	The type of a contextual factor (e.g., location, phrase; author, keyword)
(Dimensional) value	The contextual factor in the dimension
Time slice	The period for consecutive behaviors
Behavior	A set of dimensions, a set of values in each dimension, a time slice for the timestamp

Table 1: Terms used throughout the paper and their definitions.

cial interaction [34]. For example, the crowds predicted the score before the Super Bowl, commented the singers and bands at the half-time show, and expressed their happiness or sadness after the match. Therefore, the representation should make the behaviors sortable by their time dimensional values (i.e., timestamps), while the other dimensions are not required to be compared.

Traditional behavior modeling used the "tensor" [15, 30, 11, 9] to represent the multidimensional behaviors and proposed a great line of block detection methods [5, 17, 24] to capture the dense blocks as interesting patterns.

Why not Tensor? FEMA [11] and CROSSSPOT [9] represented the tweets as (user, phrase, hashtag, URL) tuples and used the 4-mode tensor to define the tweet data. However, when the tweet has *neither* a hashtag *nor* a URL, it either has to be moved out or individually creates a 2-mode dense block. Therefore, the tensor representation either loses a large amount of such information or overweights the meaningfulness of the tweet.

Why not Block Detection? SVD and tensor decompositions have been widely used for multidimensional clustering, subgraph mining and community/block detection [17, 24, 9]. However, they mix all the values into one dimension even including the timestamp values. Their blocks cannot select the meaningful dimensions; the grouped timestamps cannot capture the dynamic patterns.

In this paper, we propose novel representations for the behaviors and summaries (see Figure 1b): a "*two-level matrix*" for the behaviors in which the columns are the dimensional values of the contextual types, and the rows are the behaviors in the time slices; a "*Tartan*" for the behavioral summary that includes (1) a set of meaningful dimensions, the meaningful values in each dimension to define the multicontextual patterns; and (2) a list of consecutive time slices and the representative behaviors in each slice to define the dynamic patterns. To address the problem of catching the Tartans (i.e., summarizing the behavioral data), we propose a propagation method called CATCHTARTAN that defines the meaningfulness metric of including or excluding a value, a dimension, a behavior and a time slice by leveraging the Minimum Description Length (MDL) principle. The general philosophy is that saving more bits in compression indicates a more important element in the Tartan. Moreover, CATCHTARTAN is carefully developed with several desired properties: it requires no user-defined parameters, runs in parallel and adapts to the dynamic environment.

Figure 2 shows seven of the Tartans that CATCHTARTAN catches from tweets about the "Super Bowl 2013" event. They summarize its five phases such as the score prediction, first half, half-time show, second half and sentiments after a win/loss. The Tartans consist of different numbers of dimensions from 1 ("Phrase") to 3 ("Location-Phrase-Hashtag","Phrase-RT@User-URL") and different consecutive time slices from 5pm, 8pm to 10pm, indicating the advertising campaigns, local trends and topical discussions.

It is worthwhile to highlight our contributions as follows.

- **The Tartan concept:** we propose a novel representation for behavioral summary to capture the dynamic and multicontextual patterns. It enables the non-value/set-values, the temporal ordering of behaviors and the dimension selectivity.
- **Scalable, parameter-free algorithm:** we propose a scalable and parameter-free method CATCHTARTAN for behavioral summarization, iteratively updating the Tartans with an

	FSG [20]	GRAPH-CUBE [35]	EVENT-CUBE [30]	MDC [22]	BOW [7]	FEMA [11]	COM2 [2]	CROSS-SPOT [9]	GRAPH-SCOPE [28]	VOG [18]	TIME-CRUNCH [27]	**CATCH-TARTAN**
Principled scoring								√	√	√	√	√
Parameter-free	√				√		√	√	√	√	√	√
Multidimensional		√	√	√	√	√	√	√				√
Multicontextual												√
Timestamp value			√	√		√	√	√	√		√	√
Dynamics							√		√		√	√

Table 2: Feature-based comparison of CATCHTARTAN with alternative approaches: it gives a straight line of checks (blanks for "×").

Dataset	**#Tweet**	**#User**	**#Loc**	**#Phrase**	**#Hashtag**	**#URL**	**# RT @User**	**#@User**	**Time Period**
NYC14	10,111,725	329,779	690	1,082,463	587,527	2,766,557	24,439	955,764	113 days
LA14	402,036	14,949	55	257,301	24,711	76,950	795	42,951	113 days
SPB13	2,072,402	1,456,992	9,306	416,461	105,473	140,874	284,647	223,261	25 half-hours
GRM13	2,606,933	1,457,664	5,750	433,548	81,582	334,707	235,097	160,184	52 half-hours
Dataset	**#Paper**	**#Author**	**#Venue**	**#Keyword**	**#Cited paper**				**Time Period**
DBLP	112,157	117,934	55	33,285	62,710				35 years

Table 3: Four Twitter datasets (New York 2014, Los Angeles 2014, Super Bowl 2013 and Grammy Awards 2013) and the DBLP data.

information-theoretically principled metric that defines the meaningfulness of including or excluding any element.

- **Effectiveness:** we evaluate the scalable CATCHTARTAN on synthetic data, four Twitter datasets and the DBLP data. We show both quantitative and qualitative results: CATCHTARTAN provides comprehensive behavioral summaries.

2. RELATED WORK

Traditional approaches model behaviors in three ways: graphs, tensors/cubes and multidimensional itemsets. However, none of the above can represent the dynamic multicontextual patterns in the human behavioral data. Table 2 gives a visual feature-based comparison of CATCHTARTAN with the existing methods.

Graph data summarization. Graph is common to represent the binary relations inside human behaviors. GRAPHSCOPE [28] uses graph search for hard-partitioning of temporal graphs to find dense temporal cliques and bipartite cores. VOG [18] and TIMECRUNCH [27] use MDL to label subgraphs in terms of stars, (near) cliques, (near) bipartite cores and chains: the former approach works on static graphs, while the latter focuses on dynamic graphs. SLASHBURN [16] is a recursive node-reordering approach to leverage run-length encoding for graph compression. Toivonen et al. [31] uses structural equivalence to collapse nodes/edges to simplify graph representation. These approaches work on flat representations, while the behavioral dataset itself is naturally multidimensional.

Tensor decomposition and cube analysis. Tensor decompositions [29, 17, 11] conduct multidimensional analysis; COM2 [2] uses CP/PARAFAC tensor decomposition with MDL. However, the tensor has a big flaw: it has to drop the behaviors in which some dimension is missing. On the cube side, TOPICCUBE [15] proposes a topic-concept cube that supports online multidimensional mining of query log. GRAPHCUBE [35] defines analysis cubes and OLAP operations on cubes over graphs. EVENTCUBE [30] performs multidimensional search and analysis of large collections of free text. Our CATCHTARTAN proposes a totally different representation for the behaviors and it has a principled scoring function to select the dimensions for the summaries.

Frequent pattern mining and multidimensional clustering. We can adopt the concept of itemsets in both the frequent pattern mining [20, 8] and multidimensional data clustering [23, 14, 1, 19] to represent the behavioral contexts. F. Cordeiro et al. [7] proposes BOW method for clustering very large and multidimensional datasets with MAPREDUCE. However, the mixture of the dimensional values (itemsets in the above methods) kills the selectivity of meaningful dimensions and thus fails to describe the multicontextual patterns. The timestamp clustering cannot describe the dynamic patterns either.

MDL theory and applications. Rissanen [25] proposes optimal encoding for integers greater than or equal to 1, which minimizes the description length one obtains estimates of the integer-valued structure parameters. Cilibrasi et al. [3] proposes a hierarchical clustering method on compression using the non-computable notion of Kolmogorov complexity. Faloutsos et al. [6] demonstrates that compression and Kolmogorov complexity can measure structure and order. The MDL principle aims to be a practical version of Kolmogorov Complexity [21]. Vreeken et al. [32] uses the MDL principle to catch large groups of patterns essentially describing the same set of transactions in the data.

To summarize, our CATCHTARTAN is unique for its (1) novel representations for behaviors and summaries to capture dynamic multicontextual patterns; (2) principled scoring function with no user-defined parameters; and (3) scalable propagation algorithm.

3. BEHAVIORAL REPRESENTATION AND SUMMARIZATION

In this section, we first introduce several datasets of behaviors and preliminarily analyze the multicontextual characteristic. Then we propose our representations for the behaviors and summaries, following by the problem definition of behavioral summarization.

3.1 The Multicontextual Behaviors

Datasets. We use four large Twitter datasets as well as the DBLP data (see Table 3). The tweets were collected from different sources: (1) NYC14 and LA14 were crawled using Twitter Streaming API[1] from August 1^{st} to November 30^{th} 2014. The NYC14 dataset consists of 10 million tweets in New York and the LA14 consists of 0.4 million in the Greater Los Angeles Area. (2) SPB13 (Super Bowl 2013) and GRM13 (the Grammy Awards 2013) were collected by Techtunk[2], each of which has over 2 million tweets. We extract many hard-encoded dimensions such as location, hashtag, URL,

[1]https://dev.twitter.com/streaming/overview

[2]http://www.techtunk.com/

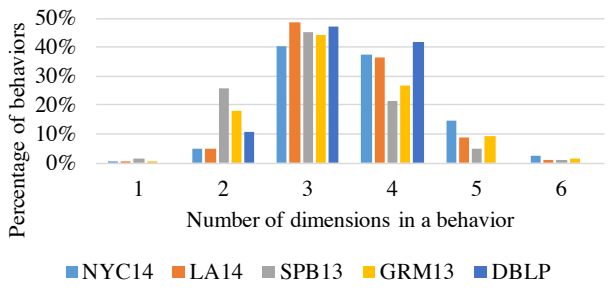


Figure 3: The distribution of #dimensions in human behaviors.

@User, as well as the rich phrase dimension. For the DBLP data, we have the author, venue, keyword and cited-paper dimensions.

Dimension distributions. Given the number of dimensions in a behavior, Figure 3 shows the percentage of the behaviors of that many dimensions in the datasets. The most frequent number in all the datasets is 3: (User, Phrase, Location/Hashtag/URL) in the Twitter data, and (Author, Venue, Keyword) in the DBLP. The behaviors are allowed to have various dimensions and for each dimension, they are allowed to have multiple values (a few phrases or a few keywords). A specific behavioral intention shares a set of specific contextual factors and creates a pattern of specific dimensions and values. For example, advertisers often generate tweets of similar phrases and the same URL; local events often share a group of hashtags and phrases. This is so called "multicontextual".

3.2 "Two-level Matrix" and the Tartan

Now we know that the behavioral data include temporal and contextual information, because every behavior has its timestamp and a set of contexts, or called dimensional values. On the contextual side, suppose the data have D dimensions, and for each dimension $d \in [1, D]$, there are N_d values. On the temporal side, suppose the data can be divided into T time slices, and for each time slice $t \in [1, T]$, there are $E^{(t)}$ behaviors. The symbols are their definitions are given in Table 4.

Figure 1b has illustrated our proposed "two-level matrix" to represent human behaviors. The formal definition is as follows.

DEFINITION 1 (TWO-LEVEL MATRIX (BEHAVIORAL DATA)). *A two-level matrix $\mathcal{X}$ consists of $\sum_{d=1}^{D} N_d$ columns (dimensional values) and $\sum_{t=1}^{T} E^{(t)}$ rows (behaviors), in which $\mathcal{X}_d^{(t)}(b, i)$ denotes how many times the i-th value in the d-th dimension appears in the b-th behavior at the t-th time slice. The top level consists of D dimensions and T time slices.*

Note that our definition can represent any dimensional setting and any type of values including non-value and set-values. A behavioral summary is a subset of the data that creates a representative pattern. Specifically, the definition is as follows.

DEFINITION 2 (TARTAN (BEHAVIORAL SUMMARY)). *A behavioral summary $\mathcal{A}$ has five components:*

- *a set of dimensions $\mathcal{D} \subseteq \{1, \dots, D\}$;*
- *a set of values $\mathcal{V}_d \subseteq \{1, \dots, N_d\}$ in the dimension $d \in \mathcal{D}$;*
- *a list of consecutive time slices $\mathcal{T} = [t_{start}, t_{end}] \subseteq [1, T]$;*
- *a set of behavior entries $\mathcal{B}^{(t)} \subseteq \{1, \dots, E^{(t)}\}$ in the time slice $t \in \mathcal{T}$;*
- *the behavior-value entries $\{\mathcal{X}_d^{(t)}(b, i) | d \in \mathcal{D}, t \in \mathcal{T}, b \in \mathcal{B}^{(t)}, i \in \mathcal{B}^{(t)}\}$*

The size of the first four components are denoted by $1 \le D^{\mathcal{A}} \le D$, $1 \le n_d \le N_d$, $1 \le T^{\mathcal{A}} \le T$ and $1 \le e^{(t)} \le E^{(t)}$.

As shown in Figure 1b and 4, the Tartan is named after its particular shape in the two-level matrix.

Symbol	Definition
$\mathcal{X}$	The "two-level" matrix: the behavioral data
$\mathcal{A}$	The Tartan: the behavioral summary
$\mathcal{X}^{\mathcal{A}}$	The first-level submatrix of $\mathcal{X}$ that includes $\mathcal{A}$
D	Number of dimensions in the data
T	Number of time slices in the data
d	The dimension index
t	The time slice index
N_d	The size of the d-th dimension
$E^{(t)}$	The size of the t-th time slice
V	The volume of $\mathcal{X}^{\mathcal{A}}$
C	The sum of non-zero entries in $\mathcal{X}^{\mathcal{A}}$
$\mathcal{D}$	The set of dimensions in $\mathcal{A}$
$\mathcal{T}$	The consecutive time slices in $\mathcal{A}$: $[t_{start}, t_{end}]$
$\mathcal{V}_d$	The set of values on the d-th dimension in $\mathcal{A}$
$\mathcal{B}^{(t)}$	The set of behaviors at the t-th time slice in $\mathcal{A}$
$D^{\mathcal{A}}$	The number of dimensions in $\mathcal{A}$
$T^{\mathcal{A}}$	The number of time slices in $\mathcal{A}$
n_d	The number of values on the d-th dimension in $\mathcal{A}$
$e^{(t)}$	The number of behaviors at the t-th time slice in $\mathcal{A}$
v	The volume of the Tartan $\mathcal{A}$
c	The sum of non-zero entries in $\mathcal{A}$

Table 4: Symbols and their definitions.

3.3 The Behavioral Summarization Problem

In this paper, the ultimate goal is to summarize the behaviors, in other words, to find the behavioral summaries in the temporal multidimensional data. With the above representations for the behavior and summary, we define the problem of behavioral summarization as follows, equally as catching Tartans in the two-level matrix.

PROBLEM 1 (BEHAVIORAL SUMMARIZATION). ***Given*** *the behavioral data (a two-level matrix) $\mathcal{X} = \{D, N_d|_{d=1}^{D}, T, E^{(t)}|_{t=1}^{T}\}$,* ***find*** *a list of behavioral summaries (Tartans) $\tilde{\mathcal{A}} = \{\dots, \mathcal{A}, \dots\}$ ordered by a principled metric function $f(\mathcal{A}, \mathcal{X})$ which defines how well the sets of meaningful dimensions, values, time slices and behaviors are partitioned and how well the meaningful subset of data is summarized, where $\mathcal{A} = \{\mathcal{D}, \mathcal{V}_d|_{d\in\mathcal{D}}, \mathcal{T}, \mathcal{B}^{(t)}|_{t\in\mathcal{T}}\}$.*

Good summarization including good partitions will be determined in an information-theoretic manner. We would like to emphasize that we solve the problem in a *parameter-free* and *scalable* way.

4. PROPOSED METHOD: CATCHTARTAN

Our CATCHTARTAN method is based on the Minimum Description Length (MDL) principle and employs a lossless encoding scheme for the temporal multidimensional data. Our objective function estimates the number of bits that encoding the Tartan can save from merging this meaningful knowledge into the data. In this section, we will address the proposed problem in Section 3 by answering three questions: (1) how to derive the cost of encoding the Tartan; (2) how to define the principled scoring function for optimization; (3) how to develop a scalable algorithm to catch the Tartans.

Encoding the Tartan.

Figure 4 illustrates the MDL-based scheme for encoding the five components of the Tartan. The components are (1) first-level columns (dimensions), (2) second-level columns (dimensional values), (3) first-level rows (time slices), (4) second-level rows (behaviors), and (5) the behavior-value entries.

Encoding the dimensions. Suppose $D = 5$ and the set of dimensions in the Tartan $\mathcal{A}$ is $\mathcal{D} = \{1, 2, 4\}$ ($D^{\mathcal{A}} = 3$), the binary string

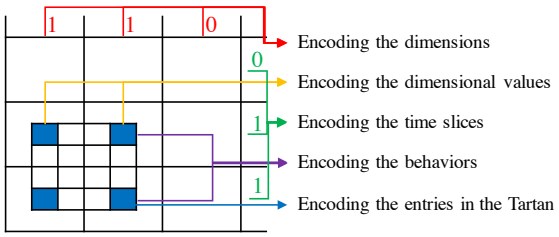


Figure 4: Encoding the 5 components when encoding a Tartan.

to encode the set is 11010. The length of this string is D and the number of 1s is $D^{\mathcal{A}}$. To further save space, we can adopt Huffman coding or arithmetic coding to encode the binary string, which formally can be viewed as a sequence of realizations of a binomial random variable X. We denote by $H_{\mathcal{D}}(X)$ the entropy:

$$\begin{aligned} H_{\mathcal{D}}(X) &= -\sum_{x\in\{0,1\}} P(X=x)\log P(X=x) \\ &= -\left(\tfrac{D^{\mathcal{A}}}{D}\log\tfrac{D^{\mathcal{A}}}{D} + \tfrac{D-D^{\mathcal{A}}}{D}\log\tfrac{D-D^{\mathcal{A}}}{D}\right). \end{aligned}$$

Additionally, two integers need to be stored: D and $D^{\mathcal{A}}$. The cost for storing these integers is $(\log^* D + \log^* D^{\mathcal{A}})$ bits, where $\log^* x$ is the universal code length for an integer x [26]. Therefore, the description length is

$$\begin{aligned} L_{\mathcal{D}}(\mathcal{A}) &= \log^* D + \log^* D^{\mathcal{A}} + D\cdot H_D(X) \\ &= \log^* D + \log^* D^{\mathcal{A}} + g(D, D^{\mathcal{A}}), \end{aligned}$$

where $g(x,y) = x\log x - (x-y)\log(x-y) - y\log y, y \le x$; x is the total number of values and y is the number of selected values.

Encoding the dimensional values. For each dimension $d \in \mathcal{D}$, the binary string is of the length N_d and has n_d 1s. Therefore, the entropy is

$$H_{\mathcal{V}_d}(X) = -\left(\tfrac{n_d}{N_d}\log\tfrac{n_d}{N_d} + \tfrac{N_d-n_d}{N_d}\log\tfrac{N_d-n_d}{N_d}\right).$$

The total description length is

$$L_{\mathcal{V}}(\mathcal{A}) = \sum_{d\in\mathcal{D}}\left(\log^* N_d + \log^* n_d + g(N_d, n_d)\right).$$

Encoding the time slices. The set of consecutive time slices in the Tartan $\mathcal{A}$ is $\mathcal{T} = [t_{start}, t_{end}] \subseteq [1, T]$, where $t_{end} = t_{start} + T^{\mathcal{A}} - 1$. Thus, the description length is

$$L_{\mathcal{T}}(\mathcal{A}) = \log^* T + \log^* T^{\mathcal{A}} + \log^* t_{start}$$

Encoding the behaviors. For each time slice $t \in \mathcal{T}$, the binary string is of the length $E^{(t)}$ and has $e^{(t)}$ 1s. The entropy is

$$H_{\mathcal{B}^{(t)}}(X) = -\left(\tfrac{e^{(t)}}{E^{(t)}}\log\tfrac{e^{(t)}}{E^{(t)}} + \tfrac{E^{(t)}-e^{(t)}}{E^{(t)}}\log\tfrac{E^{(t)}-e^{(t)}}{E^{(t)}}\right).$$

The total description length is

$$L_{\mathcal{B}}(\mathcal{A}) = \sum_{t\in\mathcal{T}}\left(\log^* E^{(t)} + \log^* e^{(t)} + g(E^{(t)}, e^{(t)})\right).$$

Encoding the entries in the Tartan. The entries in the Tartan $\mathcal{A}$ are non-negative counts instead of binary values. The volume, i.e., the length of the non-negative integer string, is

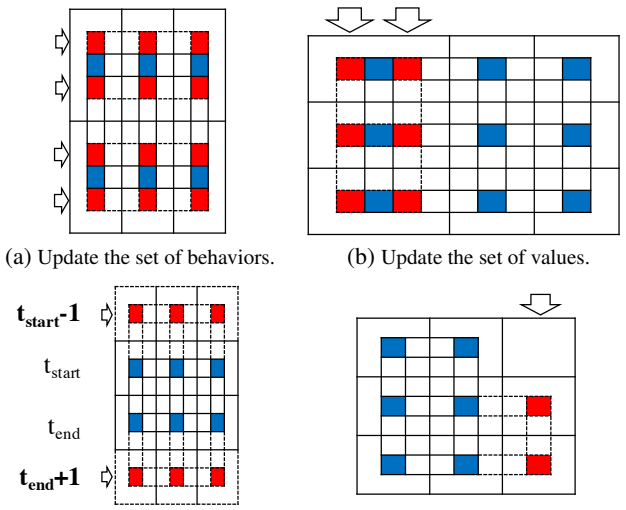
$$v = \left(\sum_{d\in\mathcal{D}} n_d\right)\left(\sum_{t\in\mathcal{T}} e^{(t)}\right).$$

The sum of the non-negative counts is

$$c = \sum_{d\in\mathcal{D},t\in\mathcal{T}}\sum_{b\in\mathcal{B}^{(t)},i\in\mathcal{V}_d}\mathcal{X}_d^{(t)}(b,i).$$

It is straightforward to add bits in order to store the integer string as a binary string. For example, if the string is 2 1 0 4 0, it can be stored as 110 10 0 11110 0, where $1\cdots10$ encodes a non-negative integer x with x 1s. Therefore, the binary string is of the length $v+c$ and has c 1s. The entropy is

$$H_{\mathcal{A}}(X) = -\left(\tfrac{c}{v+c}\log\tfrac{c}{v+c} + \tfrac{v}{v+c}\log\tfrac{v}{v+c}\right).$$



(a) Update the set of behaviors. (b) Update the set of values.

(c) Update the consecutive time slices. (d) Update the set of dimensions.

Figure 5: Updating the four elements of the Tartan till convergence: time slices, behaviors, dimensions and dimensional values.

The description length is

$$L_{\mathcal{A}}(\mathcal{A}) = (v+c)H_{\mathcal{A}}(X) = g(v+c, c).$$

The *entire encoding cost* of the Tartan $\mathcal{A}$ is

$$L(\mathcal{A}) = L_{\mathcal{D}}(\mathcal{A}) + L_{\mathcal{V}}(\mathcal{A}) + L_{\mathcal{T}}(\mathcal{A}) + L_{\mathcal{B}}(\mathcal{A}) + L_{\mathcal{A}}(\mathcal{A}).$$

The principled scoring function.

The goal is to find the Tartan with high "meaningfulness score". The scoring function is defined as the number of bits (description length) that encoding the Tartan $\mathcal{A}$ saves from encoding every individual entry in the first-level matrix $\mathcal{X}^{\mathcal{A}}$:

$$\boxed{f(\mathcal{A},\mathcal{X}) = L(\mathcal{X}^{\mathcal{A}}) - L(\mathcal{A}) - L(\mathcal{X}^{\mathcal{A}}\backslash\mathcal{A}).} \quad (1)$$

where $\mathcal{X}^{\mathcal{A}}\backslash\mathcal{A}$ is the individual entries in $\mathcal{X}^{\mathcal{A}}$ except the Tartan $\mathcal{A}$.

Encoding the individual entries in the first-level matrix. $\mathcal{X}^{\mathcal{A}}$ includes every value from the dimension in the set $\mathcal{D}$ and every behavior from the time slice in the set $\mathcal{T}$:

$$\mathcal{X}^{\mathcal{A}} = \{\mathcal{X}_d^{(t)}(b,i) | d\in\mathcal{D}, t\in\mathcal{T}, i\in\{1,\ldots,N_d\}, b\in\{1,\ldots,E^{(t)}\}\}.$$

The volume of this first-level matrix is

$$V = \left(\sum_{d\in\mathcal{D}} N_d\right)\left(\sum_{t\in\mathcal{T}} E^{(t)}\right).$$

Its sum of the non-negative counts is

$$C = \sum_{d\in\mathcal{D},t\in\mathcal{T}}\sum_{b\in\{1,\ldots,E^{(t)}\},i\in\{1,\ldots,N_d\}}\mathcal{X}_d^{(t)}(b,i).$$

Therefore, the description length of $\mathcal{X}^{\mathcal{A}}$ is

$$\begin{aligned} L(\mathcal{X}^{\mathcal{A}}) &= g(V+C, C) + L_{\mathcal{D}}(\mathcal{A}) + L_{\mathcal{T}}(\mathcal{A}) \\ &\quad + \sum_{d\in\mathcal{D}}\log^* N_d + \sum_{t\in\mathcal{T}}\log^* E^{(t)}. \end{aligned}$$

Given the $\mathcal{A}$ and $\mathcal{X}^{\mathcal{A}}$, the #bits to encode the individual entries is

$$L(\mathcal{X}^{\mathcal{A}}\backslash\mathcal{A}) = g(V+C-v-c, C-c);$$

Our proposed scoring function encodes different partitions including the dimensions, dimensional values, as well as the time slices and behaviors in the time slice, in order to achieve a concise description of the data. The fundamental trade-off that decides the

"best" summaries is between (1) the number of bits needed to describe the Tartan, and (2) the number of bits needed to describe the individual entries in the data.

Properties. We list several good properties that agree with intuition of the function $f(\mathcal{A}, \mathcal{X})$, which directs us to a propagation algorithm that updates the Tartan for a high score. These properties are proved in the Appendix.

Property 1. A Tartan of a higher sum c saves more bits, when other variables are fixed (which is assumed for all properties).

Property 2. A Tartan of a smaller volume v saves more bits.

Property 3. The first-level data of a smaller sum C saves more bits.

Property 4. The data of a bigger volume V saves more bits.

The scalable algorithm to catch the Tartans.

We propose a greedy search algorithm for optimal partitions in the Tartans. However, finding the optimal solution is NP-hard[3]. So we present an iterative alternating optimization where we find the optimal set of dimensions, values, time slices and behaviors while holding other variables in the Tartan. We run this sequence of updates until convergence. The algorithm is scalable to run on multiple threads sharing the memory of the dataset.

Algorithm 1 CATCHTARTAN : Catching the dynamic multicontextual Tartans for behavioral summaries

Require: the behavioral data $\mathcal{X} = \{D, N_d|_{d=1}^{D}, T, E^{(t)}|_{t=1}^{T}\}$
1: $\tilde{\mathcal{A}} = \{\}$
2: **while** the threads run **do**
3: generate a seed Tartan $\mathcal{A} = \{\mathcal{D}, \mathcal{V}_d|_{d\in\mathcal{D}}, \mathcal{T}, \mathcal{B}^{(t)}|_{t\in\mathcal{T}}\}$
4: **while** not converged **do**
5: **for** each time slice $t \in \mathcal{T} = [t_{start}, t_{end}]$ **do**
6: Update the set of behaviors $\mathcal{B}^{(t)}$ (see Figure 5a) by maximizing the scoring function $f(\mathcal{A}, \mathcal{X})$
7: **end for**
8: **for** each dimension $d \in \mathcal{D}$ **do**
9: Update the set of values $\mathcal{V}_d$ (see Figure 5b)
10: **end for**
11: Update the consecutive time slices: check if includes the (t_{start}-1)-th and (t_{end}+1)-th slices (see Figure 5c)
12: **for** each dimension $d \notin \mathcal{D}$ **do**
13: Check if includes the dimension (see Figure 5d)
14: **end for**
15: **end while**
16: $\tilde{\mathcal{A}} \leftarrow \tilde{\mathcal{A}} \cup \mathcal{A}$ sorted in descending order by $f(\mathcal{A}, \mathcal{X})$
17: **end while**
18: **return** $\tilde{\mathcal{A}}$: the list of Tartans in $\mathcal{X}$

Seed selection. We recommend three ways of generating seed Tartans: (1) one or several random behaviors in a single time slice, (2) several popular dimensional values, and (3) high-order SVD on the partial data. Experimental results in Section 5 show that the first, simple setting performs well and runs fast.

Complexity. The properties with guarantees ensure that the top behaviors in $\mathcal{B}^{(t)}$ and top dimensional values in $\mathcal{V}_d$. Intuitively, a higher score looks for a better compression, i.e., few behaviors/values in the time slice/dimension that give large sums of counts. Therefore, the optimization can be solved via a quick sorting of the values. The time complexity is $\mathcal{O}(\sum_d N_d \log N_d + \sum_t E^{(t)} \log E^{(t)})$.

5. EXPERIMENTS

In this section, we evaluate CATCHTARTAN and seek to answer the following questions. Can it accurately catch the Tartans from the temporal multidimensional data? Is it scalable? For real-world behaviors, are their patterns dynamic and multicontextual? If they are, what Tartan structures do we see and what do they mean?

[3]It is NP-hard since, even allowing only column re-ordering, a reduction to the TSP problem can be found [13].

5.1 Quantitative Analysis

It is impracticable to evaluate the behavioral summaries in real datasets. Thus, we generate the synthetic datasets and report the quantitative results.

Synthetic datasets and experimental setup. We generate random "behavioral" data and inject a Tartan into it. We set up extensive experiments with many parameters on (1) the Tartan distribution:

1. $T^{\mathcal{A}} \in [2,\ 9]$, the number of consecutive time slices in the Tartan $\mathcal{A}$, 4 as default;
2. $e^{(t)} \in [100,\ 2,000]$, the number of behaviors in the time slice, $1,000$ as default;
3. $D^{\mathcal{A}} \in [2,\ 9]$, the number of dimensions in $\mathcal{A}$, 3 as default;
4. $n_d \in [50,\ 200]$, the number of values per dimension in $\mathcal{A}$, 100 as default;
5. $\rho \in [1,\ 10]$, the average number of values per dimension in the behaviors, 3 as default;

and (2) the data distribution:

6. $T \in [5, 30]$, the total number of time slices in the dataset, 10 as default;
7. $E^{(t)} \in [1,000,\ 10,000]$, the number of behaviors per time slice in the dataset, $5,000$ as default;
8. $N_d \in [1,000,\ 2,000]$, the number of values per dimension in the data, $1,000$ as default.

Our task is to catch the Tartan, which has two binary classification subtasks: (1) detecting the set of behaviors in the Tartan, (2) detecting the set of dimensional values (contexts) in the Tartan.

We adopt the following methods as the baselines:

- FSG (Frequent Subgraph Discovery) [20]: this is a frequent itemset discovery algorithm that discovers subgraphs that occur frequently over the entire set of graphs.
- EIGENSPOKE [24]: this is a SVD-based method that can detect communities from large graphs by reading the singular vectors of the adjacency matrix.
- NMF (Nonnegative Matrix Factorization) [33]: it factorizes the matrices of complex networks to find the close relationship between clustering methods.

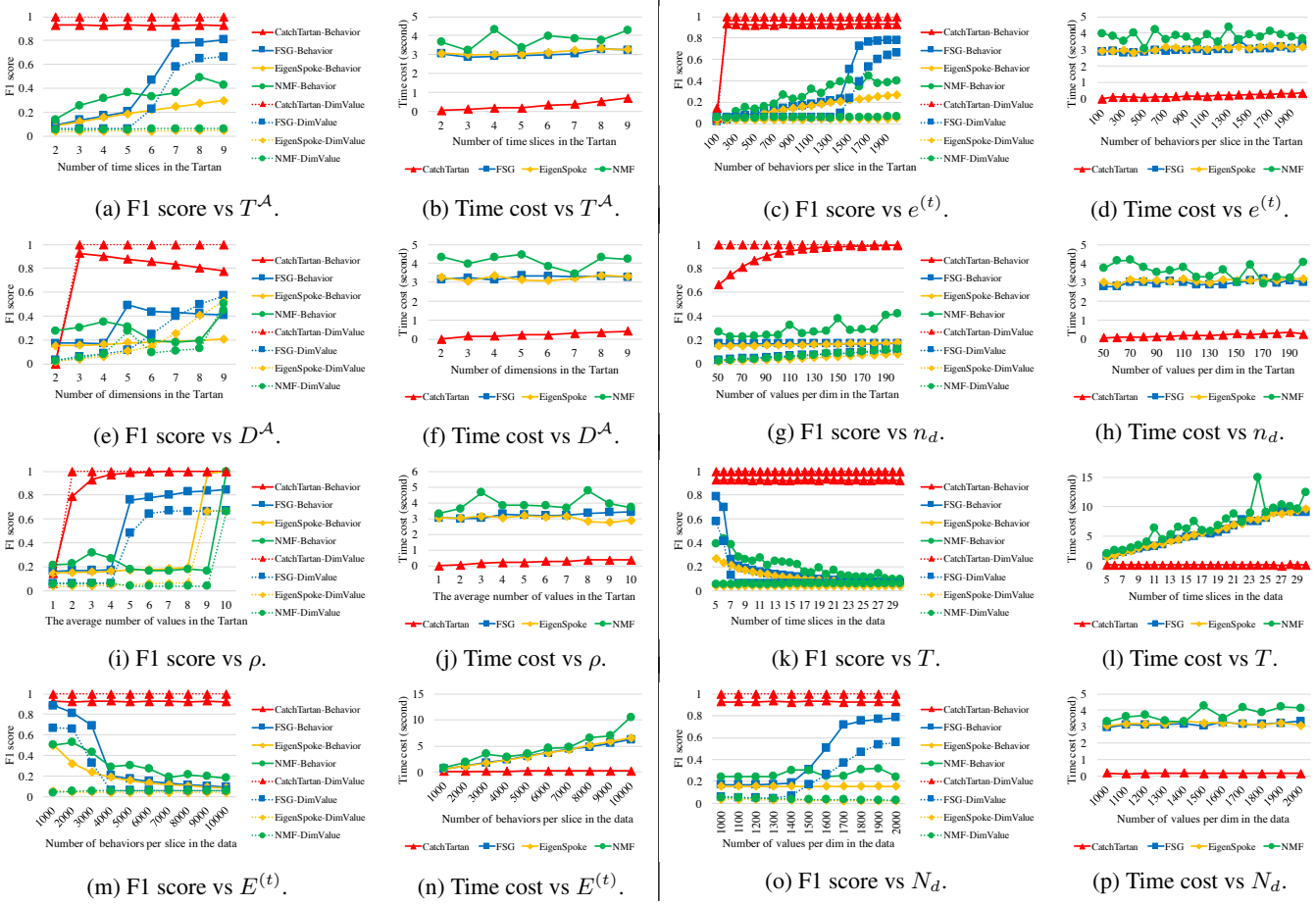
The experiments were conducted on a machine with 20 cores of Intel(R) Xeon(R) CPU E5- 2680 v2 @ 2.80GHz. We set up 10 threads and each thread searches the Tartans with 2 seeds.

Note that none of the above methods selects the dimensions nor time slices. The tensor-based methods including decompositions [29, 11] and the CROSSSPOT [9] fail to represent the multicontextual data. They lose lots of information and gave poor performances in the Tartan detection, so we do not show their results in this paper.

We evaluate the performance of our CATCHTARTAN and the baselines from two perspectives, (1) accuracy: F1 score that is the harmonic mean of precision and recall, (2) efficiency: the cost of running time. A high F1 score indicates accurate performance and a small time cost indicates high efficiency.

Accuracy and efficiency. Figure 6 presents the extensive experimental results. As every parameter varies in a big range, it shows the accuracy (on the 1st and 3rd columns in the figure) and the time cost (on the 2nd and 4th columns) of our CATCHTARTAN and the baseline methods. The solid lines are for the behavior detection, and the dashed lines are for the dimensional value detection. The symbols of our CATCHTARTAN is the red triangles. We have the following observations.

- CATCHTARTAN *consistently outperforms the baselines and the F1 score is close to the perfect 1.* FSG performs well



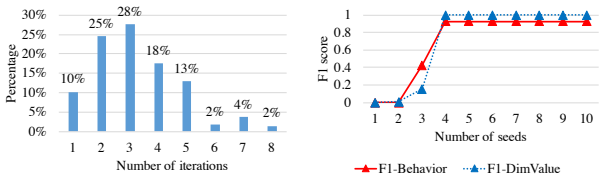
(a) F1 score vs $T^{\mathcal{A}}$. (b) Time cost vs $T^{\mathcal{A}}$. (c) F1 score vs $e^{(t)}$. (d) Time cost vs $e^{(t)}$.

(e) F1 score vs $D^{\mathcal{A}}$. (f) Time cost vs $D^{\mathcal{A}}$. (g) F1 score vs n_d. (h) Time cost vs n_d.

(i) F1 score vs ρ. (j) Time cost vs ρ. (k) F1 score vs T. (l) Time cost vs T.

(m) F1 score vs $E^{(t)}$. (n) Time cost vs $E^{(t)}$. (o) F1 score vs N_d. (p) Time cost vs N_d.

Figure 6: Synthetic experimental results demonstrate the effectiveness and efficiency of our CATCHTARTAN : the red triangle gives high and stable F1 scores on the tasks of both catching the behaviors and catching the values; it also costs much less running time than the baselines.



(a) Taking few iterations. (b) Taking few seeds.

Figure 7: Our CATCHTARTAN takes fewer than 5 iterations in over 90% runs; it requires 4 seeds to reach a higher-than-0.92 F1 score.

when the number of time slices $T^{\mathcal{A}}$ or the behaviors $e^{(t)}$ in the Tartan is big, however, CATCHTARTAN can catch the Tartan when the time period is short, which is a common case in real data. When the number of dimensions $D^{\mathcal{A}}$ is bigger or the number of values in the dimension becomes smaller, the F1 score of CATCHTARTAN gradually decreases since the Tartan becomes more like a high-dimensional block but it is still higher than the baselines. Moreover, from Figure 6k, 6m and 6o, we demonstrate that CATCHTARTAN is robust to the data distributions, especially when the baselines are inaccurate for the too big data scale of both the temporal and contextual dimension.

- CATCHTARTAN *consistently costs much less time than the baselines.* Our method is cheaper for its counting and sorting operations instead of the high-order decomposition. It has a quasi-linear complexity while the complexity of the traditional approaches are quadratic. From the 2nd and 4th columns of the figure, we spot that CATCHTARTAN spends less time to reach a much better performance. For the default setting, the CATCHTARTAN uses 0.155 second, while FSG, EIGENSPOKE and NMF use 3.25, 3.08 and 3.98 seconds, respectively: our method has a 20× speed.

The robustness to the number of iterations and seeds. Figure 7a shows that over 92% of the processes of catching the Tartans take fewer than 5 iterations. In Figure 7b, we spot that in the default setting, CATCHTARTAN requires only 4 seeds to reach an as-high-as-0.92 F1 score for both the detection tasks of behaviors and dimensional values. CATCHTARTAN requires no user-defined parameters, and it is robust to the number of iterations and seeds.

5.2 Qualitative Analysis

In this section, we discuss qualitative results from applying CATCHTARTAN to the tweet datasets and DBLP data mentioned in Table 3.

"Super Bowl 2013" event summaries. We have introduced Figure 2 in which the seven of the Tartans summarize the behavioral patterns in the data. They present five phases of the event such as the score prediction, first half, half-time show, second half and sentiments after a win/loss. The Tartans consist of different numbers of dimensions and different consecutive time slices, which indicates the advertising campaigns, local trends and topical discussions.

"Grammys Award 2013" event summaries. Figure 9 presents ten Tartans caught by our method from the GRM13 data. The Tartans have meaningful dimensional settings: (User, Phrase, @User) in-

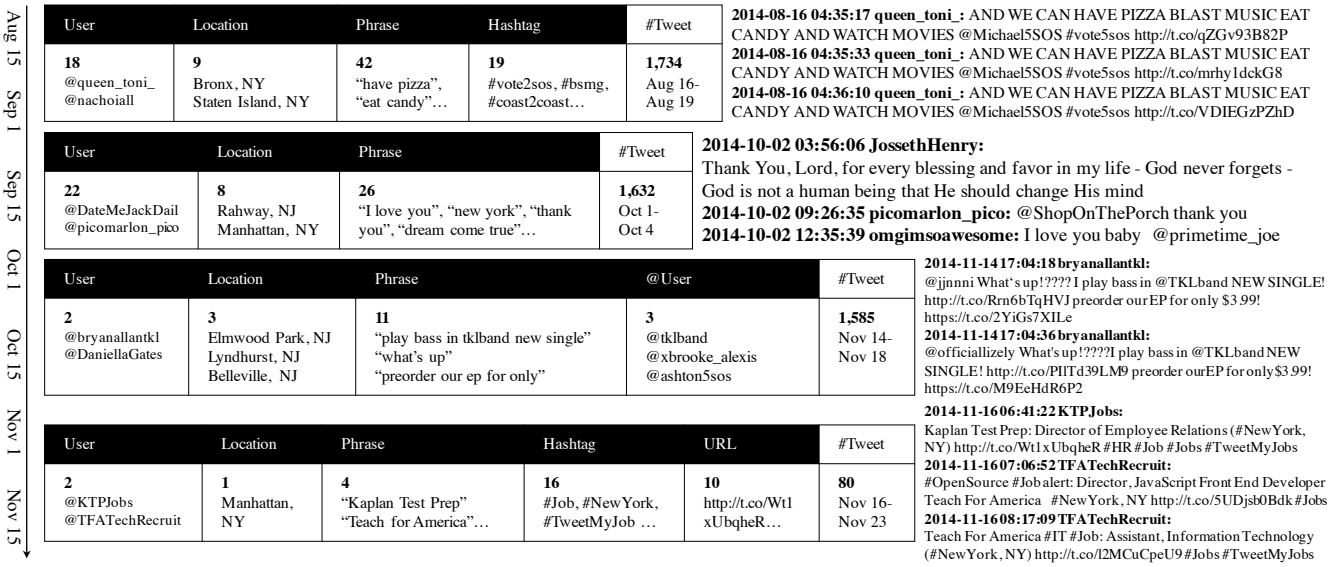


Figure 8: Tweet summaries of the NYC14 data: four Tartans of different sets of dimensions indicate behavioral patterns (e.g., advertising).

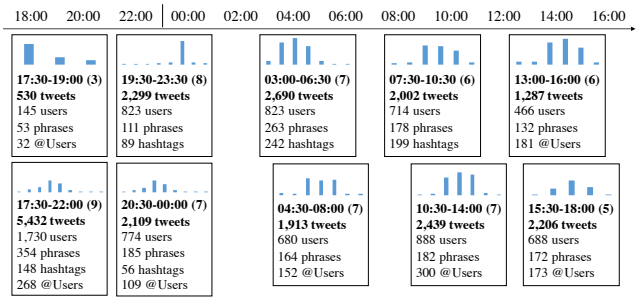


Figure 9: Tweet summaries of the GRM13 (Grammys): Tartans of different consecutive half-hours and different dimensions.

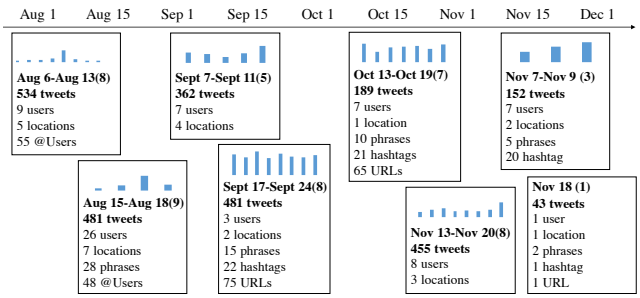
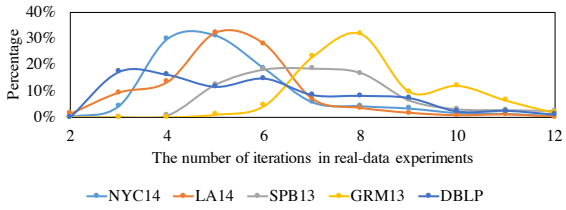


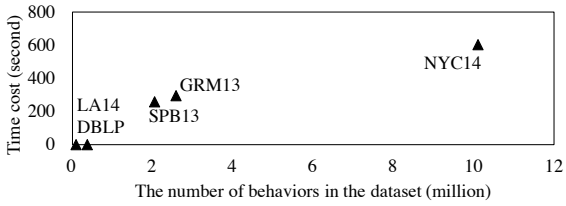
Figure 10: Tweet summaries of the LA14 (Los Angeles): Tartans of different consecutive days and different dimensions.

dicating that a group of users communicated with each other about the same words, (User, Phrase, Hashtag) indicating that the users discussed similar topics with a group of phrases; besides these three-dimensional Tartans, there are two four-dimensional ones. All these Tartans include a list of consecutive time slices for their dynamic behavioral patterns.

"Los Angeles 2014" tweet summaries. Figure 10 presents eight Tartans from the LA14 data. The Tartans have various dimensional settings: three are five-dimensional, (User, Location, Phrase, Hashtag, URL) for well-designed advertising campaigns; two are four-dimensional, (User, Location, Phrase, Hashtag) for local topics and (User, Location, Phrase, URL) for advertisements; one is three-dimensional, (User, Location, @User); and two are two-dimensional, (User, Location). These Tartans show not only the multicontextual view but also the dynamic patterns for 3 days, 5 days or even one



(a) CATCHTARTAN takes fewer than 15 iterations.



(b) The time cost is linear in the number of behaviors.

Figure 11: The scalability in real-data experiments.

week. CATCHTARTAN provides comprehensive behavioral summaries about the information at the Greater Los Angeles Area.

"New York City 2014" tweet summaries. Figure 8 takes more space to introduce four Tartans in the NYC14 data. The first Tartan has four dimensions. It has 18 users from 9 locations, talking about 42 phrases and 19 hashtags during 4 days in August, 2014. It has as many as 1,734 tweets. The tweets show that this Tartan encodes a campaign that promotes the band *5SOS* (5 Seconds of Summer).

The second Tartan has 1,632 tweets with 22 users, 8 locations and 26 phrases. This 3-dimensional Tartan also takes 4 days but in October, 2014. The tweets encode the positive sentiments of the New York citizens.

The third Tartan has four dimensions, user, location, phrase and @user. The volume is quite small: only 2 users, 3 locations, 11 phrases and 3 users who were mentioned, but the sum of tweet counts is big. The 1,585 tweets were generated from November 14 to November 18, which promoted the EP by the band *TKLband* (The Killing Lights). The messages are too similar to be generated by the two "legitimate" users: they are high probably created by some scripts. We even doubt whether the message "preorder our EP for only $3.99" is true or false or fraudulent.

The last example has five dimensions: 2 users, 1 location, 4

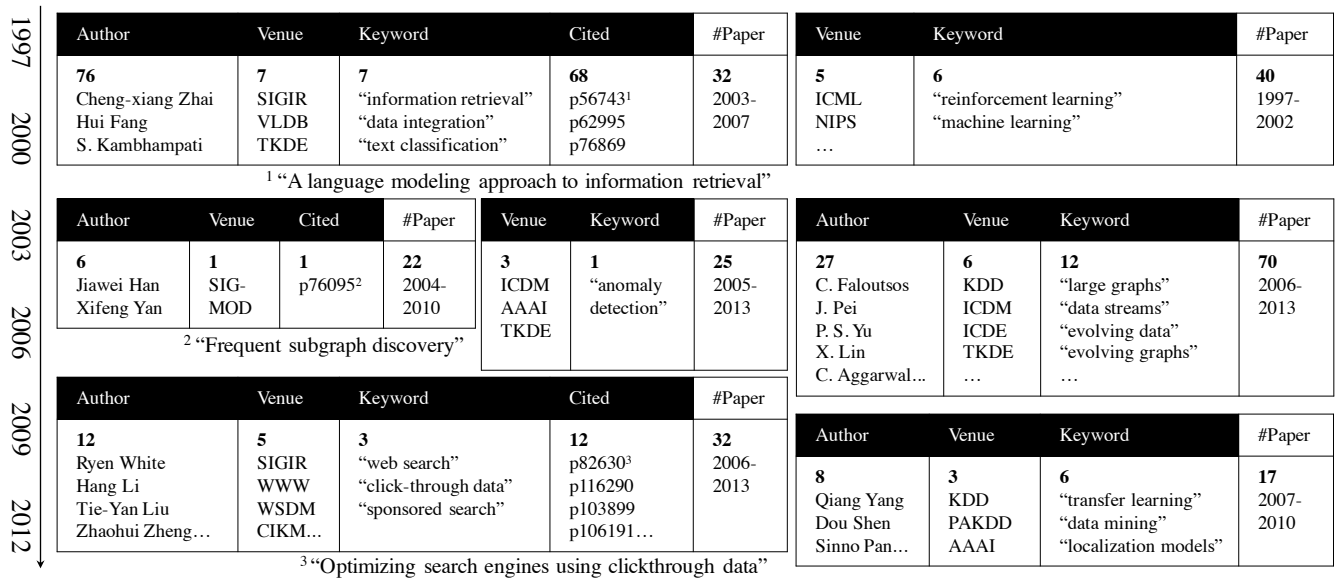


Figure 12: Publishing-paper behavioral summaries: seven Tartans about DM and ML show the dynamics and contexts of our community.

phrases, 16 hashtags and 10 URLs. The messages are a large group of interesting news about job hunting in the Manhattan, NY. The number of the messages is 80. It looks small but the messages are from a even smaller number of users and locations. Compressing these messages as a whole can save lots of bits from compressing the messages individually.

"DBLP" summaries. Figure 12 presents seven (but not the least) Tartans in the DBLP data. They are shown for their relatedness to the area of database, data mining, machine learning and so on.

The first Tartan has four dimensions, author, venue, keyword and cited paper. From 2003 to 2007, 76 authors published papers about "information retrieval", "data integration" and "text classification" on the SIGIR, VLDB conferences and TKDE journal. The paper *p56743* ("A language modeling approach to information retrieval") was frequently cited by these papers. The authors have all read this paper and explored new techniques to address the problems.

The second Tartan has only two dimensions, the venue and keyword. There were 40 papers about "reinforcement learning" on the ICML and NIPS conferences from 1997 to 2002. The author and cited-paper dimensions are missing as many scholars cited many related papers in their publications. The Tartan automatically excluded both the dimensions because no leading authors nor cited papers could be summarized.

The third Tartan has three dimensions, author, venue and cited paper. A group of six famous researchers, e.g., Dr. Jiawei Han and Dr. Xifeng Yan, published 22 papers on the SIGMOD conference which cited the same paper *p76095* ("Frequent subgraph discovery") from 2004 to 2010. The 22 papers used different phrases because the area of "subgraph mining" or "frequent subgraph pattern mining" was promising but not mature: many different methods, models and algorithms were proposed.

The forth Tartan has two dimensions, venue and keyword. 25 papers about "anomaly detection" were published in the ICDM, AAAI conferences and TKDE journal from 2005 to 2013. The papers were written by a large number of authors to address the detection problem in many applications with different methods.

The fifth Tartan is relatively large. It has 3 dimensions of 27 authors, 6 venues and 12 keywords. These 70 papers were published from 2006 to 2013. The group of researchers studied the "large graphs", "data streams", "evolving data" and "evolving graphs" on the KDD, ICDM, ICDE conferences and TKDE journal.

The sixth Tartan presents the behaviors by the "web search" community. It has four dimensions. A group of 12 authors studied 3 keywords "web search", "click-through data" and "sponsored search" and published 32 papers on the SIGIR, WWW, WSDM, CIKM conferences from 2006 to 2013. These conferences are the major information retrieval conferences. There are 12 highly-cited papers. The most representative paper is *p82630* ("Optimizing search engines using clickthrough data").

Finally, the small but meaningful Tartan represents the "transfer learning" community. 8 authors, e.g., Dr. Qiang Yang and Dr. Dou Shen, published 17 papers about "transfer learning" and " data mining" on the KDD and AAAI conferences. This is a 3-dimensional Tartan during the year 2007-2010.

Efficiency. Figure 11a shows the distributions of the number of iterations until convergence when we apply CATCHTARTAN to different datasets: CATCHTARTAN takes fewer than 15 iterations, and the most frequent number is consistently smaller than 10. Figure 11b shows that the time cost is linear in the number of behaviors in the real-data experiments, which demonstrates the scalability.

6. CONCLUSIONS

In this paper, we uncovered the dynamic and multicontextual patterns of human behaviors and focused on the problem of behavioral summarization. We proposed a novel representation called the Tartan that includes a set of dimensions, sets of dimensional values, consecutive time slices, and sets of behaviors in the slices. We proposed a parameter-free and scalable method CATCHTARTAN to capture the Tartan summaries with a principled scoring function. We applied our CATCHTARTAN to the synthetic data, DBLP data and Twitter datasets. The experimental results including the comprehensive event summaries have demonstrated the effectiveness and efficiency of our proposed CATCHTARTAN .

7. ACKNOWLEDGEMENTS

We thank the reviewers for their insightful comments. This work was sponsored in part by the U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), National Science Foundation CNS-1314632, IIS-1408924, IIS-1017362, IIS-1320617, IIS-1354329, HDTRA1-10-1-0120, and grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov), and

MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory, the U.S. Government, or other funding parties. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

8. REFERENCES

[1] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data. *DMKD'05*.

[2] M. Araujo, S. Papadimitriou, S. Günnemann, C. Faloutsos, P. Basu, A. Swami, E. E. Papalexakis, and D. Koutra. Com2: fast automatic discovery of temporal ("comet") communities. In *PAKDD'14*.

[3] R. Cilibrasi and P. Vitanyi. Clustering by compression. *TIT'05*.

[4] P. Cui, H. Liu, C. Aggarwal, and F. Wang. Computational modeling of complex user behaviors: Challenges and opportunities. *IEEE Intelligent Systems*, 31(2):78–81, 2016.

[5] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 2000.

[6] C. Faloutsos and V. Megalooikonomou. On data mining, compression, and kolmogorov complexity. *DMKD'07*.

[7] R. L. Ferreira Cordeiro, C. Traina Junior, A. J. Machado Traina, J. López, U. Kang, and C. Faloutsos. Clustering very large multi-dimensional datasets with mapreduce. In *KDD'11*.

[8] J. Han, H. Cheng, D. Xin, and X. Yan. Frequent pattern mining: current status and future directions. *DMKD'07*.

[9] M. Jiang, A. Beutel, P. Cui, B. Hooi, S. Yang, and C. Faloutsos. A general suspiciousness metric for dense blocks in multimodal data. In *ICDM'15*.

[10] M. Jiang, P. Cui, and C. Faloutsos. Suspicious behavior detection: Current trends and future directions. *Intelligent Systems, IEEE*, 31(1):31–39, 2016.

[11] M. Jiang, P. Cui, F. Wang, X. Xu, W. Zhu, and S. Yang. Fema: flexible evolutionary multi-faceted analysis for dynamic behavioral pattern discovery. In *KDD'14*.

[12] M. Jiang, P. Cui, F. Wang, W. Zhu, and S. Yang. Scalable recommendation with social contextual information. *TKDE'14*.

[13] D. Johnson, S. Krishnan, J. Chhugani, S. Kumar, and S. Venkatasubramanian. Compressing large boolean matrices using reordering techniques. In *VLDB'04*.

[14] K. Kailing, H.-P. Kriegel, and P. Kröger. Density-connected subspace clustering for high-dimensional data. In *SDM'04*.

[15] D. Kang, D. Jiang, J. Pei, Z. Liao, X. Sun, and H.-J. Choi. Multidimensional mining of large-scale search logs: a topic-concept cube approach. In *WSDM'11*.

[16] U. Kang and C. Faloutsos. Beyond "caveman communities": Hubs and spokes for graph compression and mining. In *ICDM'11*.

[17] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 2009.

[18] D. Koutra, U. Kang, J. Vreeken, and C. Faloutsos. Vog: Summarizing and understanding large graphs. *SDM'14*.

[19] H.-P. Kriegel, P. Kröger, M. Renz, and S. Wurst. A generic framework for efficient subspace clustering of high-dimensional data. In *ICDM'05*.

[20] M. Kuramochi and G. Karypis. An efficient algorithm for discovering frequent subgraphs. *TKDE'04*.

[21] M. Li and P. Vitányi. *An introduction to Kolmogorov complexity and its applications*. 2013.

[22] S. Padmanabhan, B. Bhattacharjee, T. Malkemus, L. Cranston, and M. Huras. Multi-dimensional clustering: a new data layout scheme in db2. In *SIGMOD'03*.

[23] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *KDD'04*.

[24] B. A. Prakash, A. Sridharan, M. Seshadri, S. Machiraju, and C. Faloutsos. Eigenspokes: Surprising patterns and scalable community chipping in large graphs. In *PAKDD'10*.

[25] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5), 1978.

[26] J. Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of statistics*, 1983.

[27] N. Shah, D. Koutra, T. Zou, B. Gallagher, and C. Faloutsos. Timecrunch: Interpretable dynamic graph summarization. In *KDD'15*.

[28] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *KDD'07*.

[29] J. Sun, D. Tao, and C. Faloutsos. Beyond streams and graphs: dynamic tensor analysis. In *KDD'06*.

[30] F. Tao, K. H. Lei, J. Han, C. Zhai, X. Cheng, M. Danilevsky, N. Desai, B. Ding, J. G. Ge, H. Ji, et al. Eventcube: multi-dimensional search and mining of structured and text data. In *KDD'13*.

[31] H. Toivonen, F. Zhou, A. Hartikainen, and A. Hinkka. Compression of weighted graphs. In *KDD'11*.

[32] J. Vreeken, M. Van Leeuwen, and A. Siebes. Krimp: mining itemsets that compress. *DMKD'11*.

[33] F. Wang, T. Li, X. Wang, S. Zhu, and C. Ding. Community discovery using nonnegative matrix factorization. *DMKD'11*.

[34] T. Zhang, P. Cui, C. Song, W. Zhu, and S. Yang. A multiscale survival process for modeling human activity patterns. *PloS one*, 11(3):e0151473, 2016.

[35] P. Zhao, X. Li, D. Xin, and J. Han. Graph cube: on warehousing and olap multidimensional networks. In *SIGMOD'11*.

APPENDIX

Proofs of the Properties in Section 4. Suppose the partitions have much smaller encoding cost than the data entries, the scoring function can be written as

$$f(\mathcal{A}, \mathcal{X}) = g(V + C, C) - g(v + c, c) - g(V + C - v - c, C - c).$$

Since $g'(x) = \log x - \log(x - y)$ and $g'(y) = \log(x - y) - \log y$, we have the derivatives of the function as follows.

$$\frac{\partial f}{\partial c} = \log \frac{\frac{(C-c-v)c}{Vv} + \frac{c}{v}}{\frac{(C-c-v)c}{Vv} + \frac{C}{V}} \quad , \quad \frac{\partial f}{\partial v} = \log \frac{\frac{V-v-c}{V} + \frac{C}{V}}{\frac{V-v-c}{V} + \frac{c}{v}}$$

$$\frac{\partial f}{\partial C} = \log \frac{\frac{(V+C-c)C}{Vv} - \frac{c}{v}}{\frac{(V+C-c)C}{Vv} - \frac{C}{V}} \quad , \quad \frac{\partial f}{\partial V} = \log \frac{\frac{C+V-v}{v} - \frac{C}{V}}{\frac{C+V-v}{v} - \frac{c}{v}}.$$

In a summary, the density of the Tartan is higher than the data: $\frac{c}{v} > \frac{C}{V}$. Thus, we obtain that $\frac{\partial f}{\partial c} > 0$ (Property 1), $\frac{\partial f}{\partial v} < 0$ (Property 2), $\frac{\partial f}{\partial C} > 0$ (Property 3), and $\frac{\partial f}{\partial V} < 0$ (Property 4).