

Research Article

Temporal Semantic Analysis Based Human Interaction Pattern Mining using Partial Ancestral Graph

¹S. Uma and ²J. Suguna

¹School of Computer Science, CMS College of Science and Commerce, Coimbatore, India

²Department of Computer Science, Vellalar College for Women, Erode, Tamil Nadu, India

Abstract: In modern life, interactions between human beings occur frequently in meeting discussions. Semantic knowledge of meetings can be revealed by discovering interaction patterns from those meetings. Human interaction flow in a discussion session is used to extract the frequent pattern interaction. In this study Partial Ancestral Graph (PAG) meet method is proposed to mine frequent interaction among patterns. The experimental results shows that the proposed method can extract several interesting patterns that are useful for the interpretation of human behavior in meeting discussions, such as determining frequent interactions, typical interaction flows and relationships between different types of interactions.

Keywords: Directed Acyclic Graph (DAG), human interactions, Partial Ancestral Graph (PAG), tree based mining

INTRODUCTION

Data mining, which is an important technique for discovering original information, is extensively adopted in various fields such as bioinformatics, marketing and security. KDD is the process of discovering original patterns from large data sets concerning the methods at the grouping of artificial intelligence, machine learning, statistics and database systems (Karthika and RangaRaj, 2013; Yang and Wu, 2006). The process of data mining is to extract knowledge from a dataset in a human understandable structure. Frequent patterns are item sets, subsequences, or substructures that appear in a data set with frequency not less than a user-specified threshold. Finding frequent patterns plays an essential role in mining associations, correlations and many other interesting relationships among data. Moreover, it helps in data indexing, classification, clustering and other data mining tasks as well. Thus, frequent pattern mining has become an important data mining task and a focused theme in data mining research. Human interaction determines whether the meeting was well organized or not. It is one of the main issues in the meetings (Nandha Kumar and Baskar, 2013).

LITERATURE REVIEW

To acquire the semantic information from a meeting, researchers extracted the meeting contents and represented them in a machine readable format. For instance, Waibel *et al.* (1998) presented a meeting browser that describes the dynamics of human

interactions. McCowan *et al.* (2005) recognized group actions in meetings by modeling the joint behavior of participants and expressed group actions as a two-layer process by a hidden Markov model framework. Otsuka *et al.* (2007) used gaze, head gestures and utterances to determine who responds to whom in multiparty face-to-face conversations. Otsuka *et al.* (2007) proposed a multimodal approach for interaction recognition. Yu *et al.* (2012) used a tree-based mining method to discover frequent patterns from human interactions occurred in meetings. Such a method focuses mostly on capturing direct parent-child relationship.

There have been several works done in discovering human behavior patterns by using stochastic techniques. Bakeman and Gottman (1997) applied sequential analysis to observe and analyze human interactions. Magnusson (2000) proposed a pattern detection method, called T-pattern to discover hidden time patterns in human behavior. T-pattern has been adopted in several applications such as interaction analysis (Anolli *et al.*, 2005) and sports research (Yu *et al.*, 2009).

To discover the frequent patterns in a tree Yu *et al.* (2010) introduces a novel algorithm to discover all frequent pattern subtrees in a tree-plant with a novel data structure called scope-list. Barnard *et al.* (2005) systematically expanded the two algorithm pattern growth methods for drawing out frequent tree patterns.

Casas-Garriga (2003) proposed algorithms to mine unbounded episodes from a sequence of events on a time line. The work is generally used to extract frequent episodes. Morita *et al.* (2005) proposed a pattern mining method for the interpretation of human interactions in a

Corresponding Author: S. Uma, School of Computer Science, CMS College of Science and Commerce, Coimbatore, Tamil Nadu, India

This work is licensed under a Creative Commons Attribution 4.0 International License (URL: <http://creativecommons.org/licenses/by/4.0/>).

poster exhibition. It extracts simultaneously occurring patterns of primitive actions such as gaze and speech.

Discovering semantic knowledge is significant for understanding and interpreting how people interact in a meeting discussion. Becker and Yu (2012) introduced a mining method to extract frequent patterns of human interaction based on the captured content of face-to-face meetings. Human interaction flow in a discussion session is represented as a tree. The Weighted Interesting Pattern mining algorithm is used for discovering patterns by calculating weight confidence and similar patterns are grouped by using similar weights (Uma and Suguna, 2013).

METHODOLOGY

Temporal semantic analysis for partial ancestral graph based pattern mining: In human interaction, patterns can be triggered or influenced by multiple interactions. The extent of influence can be significantly dependent on the weight/rank of the person triggering that interaction. In this study, first, Temporal Semantic Analysis (TSA) is used to leverage temporal information and compute a refined metric of semantic relatedness. Second, PAG-based mining method is used for extracting weighted temporal frequent interaction patterns from meetings.

Temporal Semantic Analysis (TSA): TSA is based on associating each word with a weighted vector of concepts. Such concepts can be derived from human interaction flow from meetings. Thus, instead of representing a word with a vector of unit concepts, vectors of time series are manipulated, where each time series describes concept dynamics over time. Concepts that behave similarly over time are semantically related. Such a rich representation of words could facilitate the discovery of implicit semantic relationships between the original words.

Let c be a concept represented by a sequence of words that might be positive or negative opinion related for each and every human interaction flow wc_1, \dots, wc_k . Let d be a document or human interaction flow. Say that c appears in d if its words appear in the document with a distance of at most ϵ words between each pair wc_i, wc_j , where ϵ is a proximity relaxation parameter. That is, a concept appears in a document if there is a window of size ϵ .

Let t_1, \dots, t_n be a sequence of consecutive discrete time points. Let $H = \{D_1, \dots, D_N\}$ be a history represented by a set of document collections, where D_i is a collection of documents associated with time t_i . Then the dynamics of a concept c related to the human interaction flow in meetings is defined as:

$$\text{Dynamics}(c) = \left\langle \frac{|\{d \in D_1 | \text{appears}(c, d)\}|}{|D_1|}, \dots, \frac{|\{d \in D_n | \text{appears}(c, d)\}|}{|D_n|} \right\rangle$$

Algorithm 1: Procedure Semantic Relatedness (t_1, t_2):

- 1) $C(t_1) = \{ts_1^1, \dots, ts_n^1\}$
- 2) $C(t_2) = \{ts_1^2, \dots, ts_m^2\}$
- 3) $R(t_1, t_2) \leftarrow 0$
- 4) Repeat Min (m, n) times
- 5) $\langle t\hat{s}_1, t\hat{s}_2 \rangle = \arg \max_{(ts_1, ts_2) \in C(t_1) \times C(t_2)} Q(ts_1, ts_2)$
- 6) $R(t_1, t_2) \leftarrow R(t_1, t_2) + Q(t\hat{s}_1, t\hat{s}_2)$
- 7) $C(t_1) \leftarrow C(t_1) \setminus \{t\hat{s}_1\}$
- 8) $C(t_2) \leftarrow C(t_2) \setminus \{t\hat{s}_2\}$
- 9) Return $R(t_1, t_2)$

The relatedness Q between two concepts is determined by comparing their dynamics. For comparing the concepts associated with time series, two methods namely cross correlation and Dynamic Time Warping (DTW) is used.

Cross correlation: In statistics, cross correlation is a method for measuring statistical relations, e.g., measuring similarity of two random variables. Here words are represented as time series. Words, whose frequencies correlate in volume, but with a time lag, will be identified as similar. To evaluate the correlation of the two words time-series, compare the time series starting from the first time point they both started appearing, until the time point when one of the words stopped appearing.

Algorithm 2: Procedure cross correlation (ts_1, ts_2):

- Similarity (ts_1, ts_2) = 0
 Cov (ts_1, ts_2) = 0
 For delay = $\{-delay_{min}, delay_{max}\}$
 For index = $\{0, \dots, \min(|ts_1|, |ts_2|)\}$
 Delays index = index + delay
 $Con(ts_1, ts_2) \leftarrow cov(ts_1, ts_2) + (ts_1[index] - E(ts_1)) * (ts_2[delayedIndex] - E(ts_2))$

$$corr@delay(ts_1, ts_2) \leftarrow \frac{cov(ts_1, ts_2)}{N\sigma_{ts_1}\sigma_{ts_2}}$$

$$\text{Similarity}(ts_1, ts_2) \leftarrow \text{Max}(\text{similarity}(ts_1, ts_2), corr@delay((ts_1, ts_2)))$$

Return similarity (ts_1, ts_2)

Dynamic time warping: The DTW algorithm measures the similarity between two time series that may differ in time scale, but similar in concepts. The algorithm defines a local cost matrix $C \in \mathbb{R}^{|ts_1| \times |ts_2|}$ of two time series ts_1 and ts_2 as:

$$C_{i,j} = ||ts_1[i] - ts_2[j]||, i \in \{1, \dots, |ts_1|\}, j \in \{1, \dots, |ts_2|\}$$

where, $||ts_1[i] - ts_2[j]||$ is a distance metric between two points of the time series. Given this cost matrix,

DTW constructs an alignment path that minimizes the cost over this cost matrix. This alignment p is called the "warping path" and defined as a sequence of points pairs $p = (pair_1, \dots, pair_k)$ where $pair_1 = (i, j) \in \langle 1, \dots, |ts_1| \rangle \times \langle 1, \dots, |ts_2| \rangle$ is a pair of indexes in ts_1 and ts_2 respectively. Each consequent pair preserves the ordering of the points in ts_1 and ts_2 and enforces the first and last points of the warping path to be the first and last points of ts_1 and ts_2 .

Algorithm 3: Procedure $DTW(ts_1, ts_2, C)$:

```

n ← Min ([ts1], [ts2])
dtw(ts1, ts2) ← new[[ts1] × [ts2]]
for i = {1, .. n}
dtw(i, 1) ← dtw(i - 1, 1) + c(i, 1)
dtw(1, i) ← dtw(1, i - 1) + c(1, i)
for i = {1, .. n}
for j = {1, .. n}
dtw(i, j) = ||ts1(i) - ts2(j)|| + Min(dtw(i - 1, j), dtw(i, j - 1), dtw(i - 1, j - 1))
Return dtw(n, n)
    
```

Human interaction flow: Human interactions occurred in the meetings can be mainly categorized into the following seven classes:

- **PRO:** A participant proposes an idea.
- **ASK:** A participant asks for opinion regarding a proposal.
- **POS:** A participant expresses positive attitude towards a proposal.
- **NEG:** A participant expresses negative attitude towards a proposal.
- **ACK:** A participant agrees on some other's comment, decision, or attitude.
- **COM:** A participant comments on another action (PRO, ACK, POS, etc.).
- **REQ:** A participant requests information regarding an issue.

Partial ancestral graph: A PAG (Partial Ancestral Graph) is used to represent any subset of Equiv ($G(O, L)$) (Equivalence graph). A PAG is an extended graph that consists of set of vertices O and a set of edges between vertices. There may be the following kind of edges: $A \leftrightarrow B, A o - oB, A o \rightarrow B, A \leftarrow oB, A \rightarrow B$ or $A \leftarrow B$. In PAG the A end point of $A \rightarrow B$ is " $-$ "; A end point of an $A \leftrightarrow B$ is " $<$ "; and then A end point of $A o \rightarrow B$ is " o ". The conventions of B end point are analogous. In addition pair of edge points may be connected by underlining (Werth *et al.*, 2008). A partial Ancestral Graph for the set of directed graph G each carrying the same set of Observed variables O , contain partial information about the ancestral relation in G , namely only those ancestral relations is common to all members of G . There can be more than one PAG representing a given set G . Thus the PAG can be used

to represent both the ancestor relation among the members of O common to the members of G and the set conditional independence relation among the members of O in G .

Support calculation: Given (i) a sub-PAG and (ii) a database DB , the support is defined by the following equation:

$$\text{Sup} = \frac{(\text{No of temporal frequent pattern in graph})}{(\text{total no of temporal patterns in DB})}$$

Mining temporal frequent interaction patterns from meeting database: Once a graph and its traversals are specified, valuable information can be retrieved through graph mining. Normally they are in the form of patterns. Frequent patterns which are sub traversals occurred in a large ratio are considered for analysis.

Algorithm 4: PAG meet:

Input: (1) Meeting Database of PAG, (2) minsup threshold with DTW

Output: A set temporal frequent patterns TF of frequent PAG interaction patterns

```

Begin
TF ← all the frequent nodes in all the PAGs in DB
tmpTF ← F
while tmpTF ≠ ∅ do
tmp ← ∅
for Tf ∈ tmpTF do
tmp ← tmp ∪
expandWithRoots (DB, tf, minsup, dtw (n, n))
tmp ← tmp ∪
expandWithNewLevel (DB, tf, minsup, dtw (n, n))
tmp ← tmp ∪ expandWithNewNode
(DB; f; minsup, dtw (n, n))
tmp ← tmp ∪
expandWithNewEdge (DB, tf, minsup, dtw (n, n))
end for
tmp ← pruneNonCanonical (tmp; minsup, )
tmp ← filterNotWantedFragments (tmp; minsup)
tmp ← filterNotFrequentFragments (tmp; minsup)
TF ← TF ∪ findConnectedPAG (tmp)
tmpTF ← tmp
end while
end
    
```

To discover the PAG's i.e., sub graphs, PAG meet algorithm are used which derive closed frequent sets. It replaces closed frequent PAG mining problem with the problem of closed frequent item-set mining on edges with the restriction that all the labels of the vertices in a PAG must be distinct.

EXPERIMENTAL RESULTS

The datasets are collected from the open source. Each dataset has respective interaction values. The goal

of this research is to discover frequent interaction trees and analyze the behavior of the algorithms on the data set, focusing on the effect of threshold.

Results: The tree-based method misses some important frequent patterns because it does not capture all triggering relations. As illustrated in Fig. 1 only one triggering relation is captured in the tree database for each triggered interaction. For instance, the tree captures the interaction ASK triggered by PRO but misses the one triggered by POS. Similarly, the tree captures the interaction NEG triggered by PRO but misses the one triggered by ASK. As such, the tree-based method does not generate the pattern POS-ASK-NEG as these three nodes are not directly connected in the tree. In fact, fragments containing siblings or ancestor's siblings of a node might not be connected without the absence of

their common ancestor in a tree. Hence, if the common ancestor is not frequent, the tree mining method fails to mine such fragments as a temporal frequent pattern. This kind of temporal frequent patterns reveals highly correlated interactions.

During the mining process, any frequent pattern must be connected because neither PAG meet nor the tree-based method can search DB for a pattern or fragment that is not connected. The PAG meet algorithm did not miss any frequent pattern. Moreover, PAG meet algorithm used weighted nodes for representing the rank of persons triggering each interaction. This criterion decreased the number of frequent patterns discovered by PAG meet. In contrast, the tree-based method did not distinguish multiple interactions. The PAG meet captured all triggering and temporal relations, it generated fewer frequent patterns.

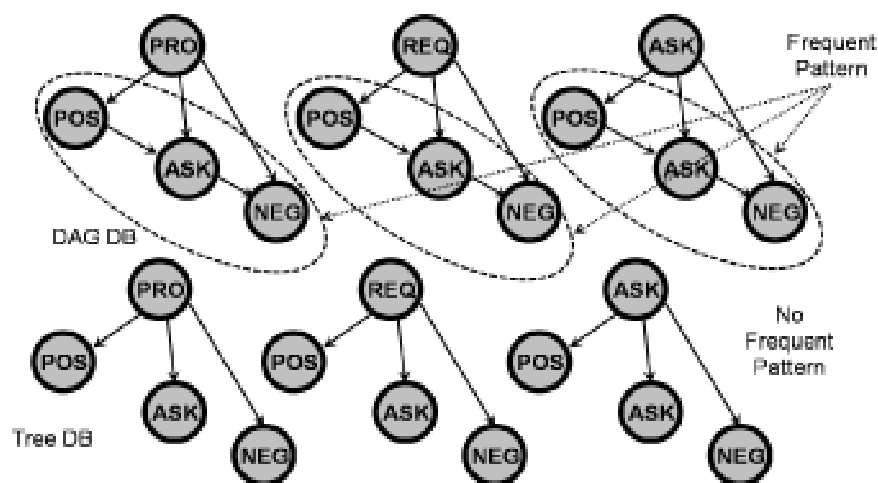


Fig. 1: PAG representation

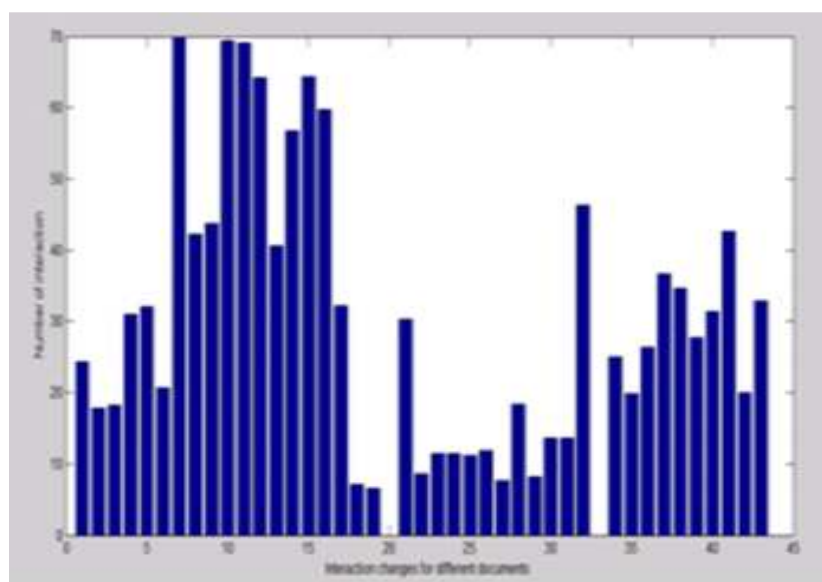


Fig. 2: Interaction change

Table 1: Performance analysis

Minsupport (%)	Frequent pattern in tree based mining	Temporal frequent patterns in PAG meet
43	79	17
40	70	13
32	53	9

Table 1 shows the performance comparison between the existing tree based method and the proposed PAG meet method based on the threshold value.

Figure 2 depicts the number of interaction change for different documents. From this it is possible to find the best document form the given set of documents based on the maximum interaction change.

CONCLUSION

In this study, we propose a Partial Ancestral Graph (PAG) Based Pattern Mining method for discovering temporal frequent interaction. The weight indicates the rank or importance of the person who initiates one of the seven classes of interactions. Such a PAG based representation of interaction flow captures both:

- Temporal relations
- Triggering relations in meetings

The proposed PAG meet algorithm mines weighted PAG-based meeting for frequent interaction patterns. The key idea is to model each session of a meeting using PAGs. Performance results shows that it is capable to mine large frequent sub-graphs in a bigger graph set with lower support. The results are significantly better than the Tree Based Pattern Mining algorithm.

REFERENCES

Anolli, L., S. Duncan Jr., M.S. Magnusson and G. Riva, 2005. The Hidden Structure of Interaction: From Neurons to Culture Patterns. Emerging Communication: Studies in New Technologies and Practices in Communication, IOS Press, Apr. 2005.

Bakeman, R. and J.M. Gottman, 1997. Observing Interaction: An Introduction to Sequential Analysis. 2nd Edn., Cambridge Univ. Press, New York.

Barnard, M., S. Bengio, D. Gatica-Perez, G. Lathoud, I. McCowan and D. Zhang, 2005. Automatic analysis of multimodal group actions in meetings. IEEE T. Pattern Anal., 27(3): 305-317.

Becker, C. and Z. Yu, 2012. Tree-based mining for discovering patterns of human interaction in meetings. Proceeding of 8th IEEE International Conferences on Knowledge Discovery and Data Mining (PAKDD'10), pp: 107-115.

Casas-Garriga, G., 2003. Discovering bounded episodes in sequential data. Proceeding of European Conferences on Principles and Practice of Knowledge Discovery in Databases (PKDD '03), pp: 83-94.

Karthika, D. and R. RangaRaj, 2013. A graph-based interaction pattern discovery for human meetings. Int. J. Adv. Comput. Sci. Technol., 2(8).

Magnusson, M.S., 2000. Discovering hidden time patterns in behavior: T-patterns and their detection. Behav. Res. Meth. Ins. C., 32(1): 93-110.

McCowan, I., D. Gatica-Perez, S. Bengio, G. Lathoud, M. Barnard and D. Zhang, 2005. Automatic analysis of multimodal group actions in meetings. IEEE T. Pattern. Anal., 27(3): 305-317.

Morita, T., Y. Hirano, Y. Sumi, S. Kajita and K. Mase, 2005. A pattern mining method for Interpretation of interaction. Proceeding of International Conferences on Multimodal Interfaces (ICMI '05), pp: 267-273.

Nandha Kumar, A. and N. Baskar, 2013. An efficient interaction pattern discovery for human meetings. Int. J. Comput. Trends Technol. (IJCTT), 4(5).

Otsuka, K., H. Sawada and J. Yamato, 2007. Automatic inference of cross-modal nonverbal interactions in multiparty conversations: "who responds to whom, when, and how?" from gaze, head gestures, and utterances. Proceedings of the 9th International Conference on Multimodal Interfaces (ICMI, 2007). ACM Press, New York, pp: 255-262.

Uma, S. and J. Suguna, 2013. Tree-based weighted interesting pattern mining approach for human interaction pattern discovery. Int. Rev. Comput. Software (IRECOS), 8(11): 2570-2575.

Waibel, A., M. Bett, M. Finke and R. Stiefelwagen, 1998. Meeting browser: Tracking and summarizing meetings. Proceeding of DARPA Broadcast News Transcription and Understanding Workshop, pp: 281-286.

Werth, T., A. Dreweke, M. Wörlein, I. Fischer and M. Philippsen, 2008. DAGMA: Mining directed acyclic graphs. Proceeding of IADIS European Conference on Data Mining 2008, pp: 11-18.

Yang, Q. and X. Wu, 2006. 10 challenging problems in data mining research. Int. J. Inf. Tech. Decis., 5(4): 597-604.

Yu, Z., Z. Yu, Y. Ko, X. Zhou and Y. Nakamura, 2009. Inferring human interactions in meetings: A multimodal approach. In: Zhang, D. *et al.* (Eds.), UIC 2009. LNCS 5582, Springer-Verlag, Berlin, Heidelberg, pp: 14-24.

Yu, Z., Z. Yu, X. Zhou, C. Becker and Y. Nakamura, 2012. Tree-based mining for discovering patterns of human interaction in meetings. IEEE T. Knowl. Data En., 24(4): 759-768.

Yu, Z.W., Z.Y. Yu, H. Aoyama, M. Ozeki and Y. Nakamura, 2010. Capture, recognition, and visualization of human semantic interactions in meetings. Proceeding of 8th IEEE International Conferences on Pervasive Computing and Communications (PerCom '10), pp: 107-115.