Collective Sampling and Analysis of High Order Tensors for Chatroom Communications

Evrim Acar, Seyit A. Çamtepe, and Bülent Yener *

Department of Computer Science, Rensselaer Polytechnic Institute
110 8^{th} Street, Troy, NY 12180.
{acare, camtes, yener}@cs.rpi.edu

Abstract. This work investigates the accuracy and efficiency tradeoffs between centralized and collective (distributed) algorithms for (i) sampling, and (ii) n-way data analysis techniques in multidimensional stream data, such as Internet chatroom communications. Its contributions are threefold. First, we use the Kolmogorov-Smirnov goodness-of-fit test to show that statistical differences between real data obtained by collective sampling in time dimension from multiple servers and that of obtained from a single server are insignificant. Second, we show using the real data that collective data analysis of 3-way data arrays *(users x keywords x time)* known as *high order tensors* is more efficient than centralized algorithms with respect to both space and computational cost. Furthermore, we show that this gain is obtained without loss of accuracy. Third, we examine the sensitivity of collective constructions and analysis of high order data tensors to the choice of server selection and sampling window size. We construct 4-way tensors *(users x keywords x time x servers)* and analyze them to show the impact of server and window size selections on the results.

1 Introduction and Background

Chatroom communications are attractive sources of information since they are in public domain and *real identities* are decoupled from the *virtual identities* (i.e., nicknames). However, chatroom communications generate real-time stream data that may have nonlinear structure [1] which is difficult to extract without semantic interpretation of the messages. Thus, data analysis techniques, such as Singular Value Decomposition (SVD) [2] that rely on linear relationships in a matrix representation of data, may fail to capture important structure information [1]. In particular, we showed that constructing multiway data arrays known as *high order tensors* such as data cubes with (users x keywords x time) modes can discover the subgroups that cannot be detected by SVD [1].

In this work, we extend centralized data collection and analysis of multiway data arrays to collective sampling and analysis. In particular, we consider how to

* This research is supported in part by NSF ACT Award # 0442154 and NSF ITR Award # 0324947.

build and analyze three-way and four-way data arrays from chatroom communications collected by sampling the data from multiple servers in time dimension. We discuss and compare collective sampling and analysis approach to a centralized one. Our motivations are twofold. First, a distributed (collective) approach would be more suitable to real-time data streams. Furthermore, it can eliminate the drawbacks of a centralized approach including being a single point of failure and becoming a performance bottleneck. Second, collective n-way data analysis may reduce the time and space complexity of a centralized one. In this work, we verify by using the real data that collective n-way data analysis provides significant saving with respect to space and computational cost.

1.1 Our Contributions

In this paper we report following contributions:

i. We present a simple distributed sampling approach, and analyze the statistical properties of data obtained by this approach. Sampling is done in time domain. Statistical comparison, based on Kolmogorov-Smirnov goodness-of-fit test, between data collected from multiple servers and data collected from a single server is provided. Thus we report that collective sampling method can produce chatroom logs which are statistically good fit to centralized ones.
ii. We construct 3-way tensors with modes of (users x keywords x time) at each server, and employ Tucker3 model to analyze them. We collect summaries of data generated by Tucker3 analysis at different servers and analyze them collectively using SVD at a central location.
iii. We compare the performance of collective tensor analysis approach with central tensor analysis in terms of space complexity, ease of determining model parameters and computation cost. We emphasize that same user clusters are identified by using both collective and central analysis of chatroom tensors.
iv. We rearrange 3-way tensors from multiple servers into a 4-way tensor with modes of (users x keywords x time x servers) to inspect the sensitivity of Tucker3 analysis with respect to server selection.

Organization of the paper This paper is organized as follows. Section 2 discusses data collection and the statistical comparisons of data obtained from a single server with data constructed from multiple servers. Section 3 explains the methodology for collective 3-way data analysis. In this section we also provide a cost comparison between centralized analysis and collective one with respect to space and computations cost. In Section 4 we present a sensitivity study for the collective data analysis. Finally, we conclude in Section 5.

2 Collective Sampling of Chatroom Data

In this work, we collected *philosophy* chatroom data from eight Undernet IRC servers located in USA, Canada, Netherland, Austria, and Croatia. We used

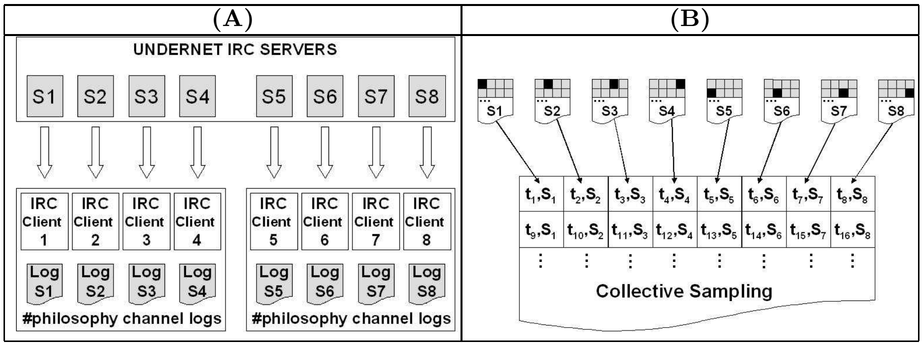

Fig. 1. (A) Centralized collection of *philosophy* chatroom data from eight *Undernet* IRC servers (S_1, S_2, ..., S_8) which are located in USA, Canada, Netherland, Austria, and Croatia. Eight copies of IRC clients running on two computers connected to servers and generated individual log files for 17 days (Jan 4th - 20th, 2006). (B) Collective sampling of *philosophy* chatroom data from eight *Undernet* IRC servers. Time is divided into time windows (150, 180, 210, 240, 270 and 300 seconds), and at each time window chatroom data coming from a specific server are accepted; at time window T_i, chatroom data coming from server S_j where $j = ((i-1)\ mod\ 8) + 1$ are accepted.

eight copies of IRC clients running on two computers as described in Figure 1-A. There are several challenges for collecting data from multiple servers. First of all, chatroom operators don't like silent listeners. Therefore, they frequently *disconnect* such users. They can *ban* IP addresses of such users, even sometimes whole IP domain. Use of public proxy servers or any other anonymity networks may sometimes be useless because of two reasons: (i) most of the servers permit only three or four connections from each IP address, (ii) a proxy or anonymity server may get banned because of the offensive acts of another IRC client sharing the same proxy. It is also possible that one or more servers are disconnected from the remaining IRC servers in which case views of the same chatroom will be completely different in terms of users and their messages. Due to these reasons, data collected from each server are different then the others. Table 1 lists number of messages collected from each server for 17 days (Jan 4th - 20th, 2006). We collected 23530 messages from the server S_7 while we collected 59320 messages from the server S_4. In a centralized approach, where chatroom data are collected only from server S_7, it would not be possible to obtain a good view of chatroom data. Instead of centralized data collection where an IRC client connects one IRC server, we use collective sampling technique where chatroom data are collected from multiple IRC servers.

In Figure 1-B, collective sampling technique is illustrated. In this work, we use several different time window values to analyze effectiveness of collective sampling. We first use centralized data collection technique to collect chatroom logs from eight servers for 17 days as illustrated in Figure 1-A. We simulate collective sampling on these chatroom logs to obtain collective chatroom data for different time windows (150, 180, 210, 240, 270 and 300 seconds). Number of messages for both centralized and collective chatroom data are given in Table 1. In the next section, we statistically compare centralized chatroom data with collective ones.

2.1 Statistical Comparison of Collective v.s. Centralized Data Collection

We first try to find suitable upper and lower bounds on time windows for collective sampling of chatroom data. Interarrival time distribution, as given in Table 1, states that 99% of the interarrival times are less than 150 seconds. When chatroom data are divided into time windows larger than 150 seconds, we expect more than one message in each time window with high probability. For the upper bound of 300 seconds, we consider the gaps in the centralized chatroom data due to connection problems. When we use collective sampling, too big time windows may cause such gaps to be transferred to collective chatroom data. Therefore we use time windows of 150, 180, 210, 240, 270 and 300 seconds. As shown in Table 1, collective data provide similar percentages as centralized data for these time windows.

Table 2 provides interarrival time and message size (in word count) statistics for centralized and collective data. Mean, median, standard deviation, skewness and kurtosis statistics for centralized and collective data are provided. Our first observation is that in both data sets, mean and standard deviation values are very close; this highlights distributions such as exponential where mean and standard deviation are the same. Positive skewness values for both data sets indicate a distribution with an asymmetric tail extending towards more positive values. Positive kurtosis values indicate a distribution with a peak. As the kurtosis statistic gets larger with a positive value, it indicates the possibility of a tall distribution. These statistics support findings in [1] that interarrival and message size fit to exponential distributions. Table 3 provides results of statistical comparison between collective and centralized data based on Kolmogorov-Smirnov goodness-of-fit test (kstest). Centralized data are compared to collective data in terms of interarrival time and message size distributions.

2.2 Collective Construction of Chatroom Tensors

3-way (users × keywords × time) and 4-way (users × keywords × time × servers) tensors of Figure 3 are generated from centralized and collective data for 2-hour interval from $17:00$ to $19:00$ on Jan 07, 2006. For tensor generation, first step is to generate list of keywords and users active in 2-hour time interval. We used a dictionary of 5000 most frequent words to eliminate frequently used words (i.e. the, they, etc.). Next, simple forms of the irregular verbs and verbs with *-ed*, *-ing*, *-s* are found by using online *webster* dictionary. *Webster* is also used to fix simple typos. Once the list of keywords is obtained, a user list is generated. Finally, 2-hour data are divided into time window intervals. Each time window corresponds to a time slot in *time* dimension of (user × keywords × time) tensor.

Centralized Chatroom Data						Collective Chatroom Data					
Server	# Mess.	< 150	< 180	< 240	< 300	Time Win.	# Mess.	< 150	< 180	< 240	< 300
S1	49179	0.995	0.996	0.997	0.998	150 sec.	45497	0.965	0.976	0.983	0.985
S2	48626	0.994	0.995	0.997	0.997	180 sec.	45340	0.970	0.971	0.985	0.987
S3	41862	0.994	0.995	0.997	0.997	210 sec.	45057	0.973	0.974	0.982	0.987
S4	59320	0.994	0.996	0.997	0.998	240 sec.	44982	0.976	0.977	0.978	0.987
S5	46679	0.994	0.995	0.996	0.997	270 sec.	45233	0.979	0.980	0.981	0.987
S6	42728	0.994	0.995	0.996	0.997	300 sec.	45076	0.981	0.981	0.983	0.983
S7	23530	0.995	0.996	0.997	0.998						
S8	49630	0.995	0.996	0.997	0.998						

Table 1. Number of messages and percentage of interarrival times smaller than 150, 180, 240, and 300 seconds for centralized and collective data. Analysis made over 17 days of data (Jan 4th - 20th, 2006).

Entry *(i,j,k)* of the tensor indicates the number of times keyword j is used in time slot k by user i. Once 3-way tensors for centralized data (chatroom logs from eight IRC servers) and collective data are generated, 4-way tensor is obtained by the join of these nine tensors.

Collective chatroom data are obtained by receiving data from server S_j in time window T_i if $j = ((i-1) \ mod \ 8) + 1$. When 3-way tensor is generated from the collective data, values for time slot T_i in the tensor corresponds to messages coming from the server S_j. We used the same time window value for collective sampling chatroom data and tensor generation. Matrix slice for collective data tensor for time slot T_i will be equivalent to matrix slice for centralized data tensor of server S_j for time slot T_i. Thus, the server S_j may send its matrix slice for time slot T_i instead of messages, and collective chatroom data tensor can be collectively constructed.

3 Collective 3-way Analysis of Chatroom Data

3.1 Methodology

Tucker Model/ HOSVD: We employ one of the most common multiway analysis models, i.e. Tucker [9], in chatroom data analysis. For a 3-way tensor T of size I x J x K, Tucker3 model decomposes the tensor in the following form:

$$T_{ijk} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} G_{r_1 r_2 r_3} \mathbf{A}_{ir_1} \mathbf{B}_{jr_2} \mathbf{C}_{kr_3} + E_{ijk}$$

where $\mathbf{A} \in R^{IxR_1}$, $\mathbf{B} \in R^{JxR_2}$, $\mathbf{C} \in R^{KxR_3}$ are component matrices for first, second and third mode, respectively, $G \in R^{R_1 x R_2 x R_3}$ is the core tensor and $E \in R^{IxJxK}$ represents the error term. Tucker model is not limited to 3-way arrays and can be generalized to high-order datasets. Different constraints such as non-negativity, unimodality or orthogonality can also be enforced on the component matrices. We constrain component matrices to be orthogonal.

	Centralized Chatroom Data						Collective Chatroom Data					
	Server	Mean	Med	Std	Skew	Kurt	Time Win.	Mean	Med	Std	Skew	Kurt
Inter-arrival Time	S1	15	10	17	3	13	150 sec.	17	11	21	3	17
	S2	16	11	19	2	11	180 sec.	16	11	19	3	14
	S3	18	12	20	2	11	210 sec.	17	11	19	2	11
	S4	16	11	18	2	11	240 sec.	16	11	18	2	12
	S5	17	12	20	2	9	270 sec.	17	11	19	2	11
	S6	20	13	22	2	8	300 sec.	17	11	19	3	13
	S7	18	12	20	3	13						
	S8	17	11	19	2	12						
Message Size in word counts	S1	11	9	10	2	6	150 sec.	11	8	10	2	5
	S2	11	9	10	2	5	180 sec.	11	8	10	2	6
	S3	11	8	10	1	5	210 sec.	11	8	9	1	5
	S4	11	8	10	1	5	240 sec.	11	8	10	1	5
	S5	10	8	9	1	5	270 sec.	11	8	10	1	5
	S6	11	8	10	2	5	300 sec.	11	8	10	1	5
	S7	10	8	10	2	7						
	S8	11	8	10	2	6						

Table 2. Interarrival time and message size (in word count) statistics (mean, median, standard deviation, skewness and kurtosis) for centralized and collective data. Analysis made over 17 days of data (Jan 4th - 20th, 2006). Only the messages in time interval $17:00$ and $19:00$ of each day is considered for the analysis because these are the times when eight Undernet IRC servers have minimum connectivity problem throughout 17 days of data.

Tucker3 model with orthogonality constraints is rather referred as High-Order Singular Value Decomposition (HOSVD) or multilinear SVD [6].

Collective Tensor Analysis: Collective Tensor Analysis approach analyzes multiple tensors simultaneously and then transfers summaries of data from each tensor to a central location. Those summaries are combined together to capture the structure in the mode, which we want to explore. In the context of chatroom communications, collective method assumes that data are sampled by different servers and arranged as a tensor at each sampling site. These tensors are locally analyzed at each sampling site by fitting a multiway model, i.e. Tucker3. Summaries of data representing the user space are collected at a central location. Matrix formed by gathering user space summaries from each server is then analyzed using SVD to capture the structure in the whole user space.

Let T_i be an n-way tensor constructed at the i^{th} server by rearranging sampled data as a tensor. Each T_i, for $i = 1, 2, ...s$, is decomposed by Tucker3, whose structural model can also be represented as $\mathbf{T}_i = \mathbf{A}_i\mathbf{G}_i(\mathbf{C}_i \otimes \mathbf{B}_i)^T$. $\mathbf{T}_i$ and $\mathbf{G}_i$ are matrices, which are matricized forms of tensors T_i and G_i in the first mode and $\mathbf{A}_i$, $\mathbf{B}_i$ and $\mathbf{C}_i$ are the component matrices corresponding to first, second and third mode, respectively. We are interested in extracting the structure of first (user) mode so we collect $\mathbf{A_i}$'s, the singular vectors for the first mode and

Interarrival Time							Message Size (in word counts)						
	150 s.	180 s.	210 s.	240 s.	270 s.	300 s.		150 s.	180 s.	210 s.	240 s.	270 s.	300 s.
S1	0.12	0.4	0.09	0.23	0.09	0.05	S1	0.9	0.94	0.88	1	0.96	0.99
S2	0.99	0.96	1	0.99	0.99	0.7	S2	0.79	0.67	0.56	0.96	0.73	0.65
S3	0.17	0.03	0.13	0.04	0.22	0.18	S3	0.99	1	0.91	1	0.99	0.99
S4	0.8	1	0.9	0.99	0.92	0.49	S4	1	1	1	0.99	1	1
S5	0.29	0.03	0.25	0.09	0.34	0.33	S5	0.97	0.96	0.99	0.71	0.96	0.97
S6	0.01	2e-4	0.01	6e-4	0.02	0.04	S6	0.99	0.99	0.99	0.98	0.99	0.99
S7	0.1	0.01	0.09	0.03	0.1	0.12	S7	0.63	0.65	0.71	0.32	0.46	0.75
S8	0.90	0.23	0.89	0.44	0.82	0.97	S8	0.92	0.97	0.99	0.71	0.91	0.95

Table 3. Centralized data are compared to collective data with time windows 150, 180, 210, 240, 270 and 300 seconds. Table lists resulting P values of Kolmogorov-Smirnov Goodness-of-fit tests (kstest) for interarrival time and message size (in word count). P values: (i) > 0.05 mean difference between two data sets is statistically insignificant, (ii) 0.01 to 0.05 mean difference is significant, (iii) 0.001 to 0.01 mean difference is very significant, and (iv) < 0.001 mean difference is extremely significant. P values should be > 0.05 to be able to conclude that there is no sufficient evidence to reject the hypothesis that two data sets are coming from the same distribution. For interarrival time, P values > 0.05 for the servers S_1, S_2, S_4 and S_8 which have highest message counts as given in Table 1. There is no sufficient evidence to reject that interarrival time distributions of these pairs of data sets are the same. For message size, all P values are > 0.05, meaning that, difference between two data sets is statistically insignificant and there is no sufficient evidence to reject that message size distributions of these pairs of data sets are the same.

${\Sigma_i}^n$, the singular values corresponding to first mode ($n = 1$) to construct matrix M [6]. This matrix is then analyzed using SVD and significant left singular vectors, U, and corresponding singular values, S, are used to extract the structure in user space.

$$\begin{aligned}
\mathbf{T}_i &= \mathbf{A}_i \mathbf{G}_i (\mathbf{C}_i \otimes \mathbf{B}_i)^T \text{ where i=1,2,...s} \\
{\mathbf{\Sigma}_i}^n &= diag({\sigma_1}^n, {\sigma_2}^n, ..., {\sigma_R}^n) \text{ where R is the rank of } n^{th} \text{ mode} \\
\mathbf{M} &= [\mathbf{A_1} * \mathbf{\Sigma_1} | \mathbf{A_2} * \mathbf{\Sigma_2} | ... | \mathbf{A_s} * \mathbf{\Sigma_s}] = \mathbf{U} * \mathbf{S} * \mathbf{V^T}
\end{aligned} \tag{1}$$

Steps of collective analysis of multiple tensors are listed in Equation 1. This approach is a generalized version of Collective Principal Component Analysis (CPCA)[3] to high-order datasets. Collective tensor analysis employs the same approach used in multiway multiblock component models [8] except for the minimization of a common objective function. The objective function is defined to be the sum of the residuals in multiple tensor decompositions and residual of the final step: 2-way component analysis in [8]. We, on the other hand, handle modeling of each tensor independently at each sampling site.

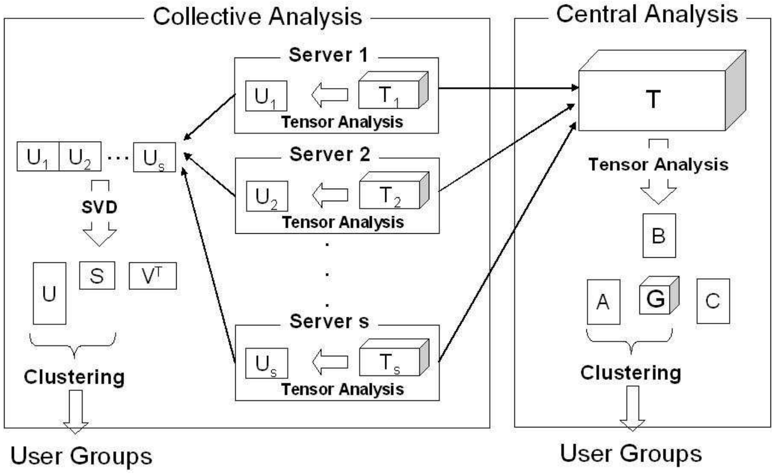

Fig. 2. Collective and Central Analysis of Tensors. In collective analysis of a tensor, partitions of the tensor are decomposed by tensor decomposition locally at different servers. User clusters are found by analyzing the collection of data summaries. On the other hand, central analysis decomposes one large tensor and user groups are determined using data summaries obtained from tensor analysis.

3.2 Collective Analysis of Chatroom Data Cubes

Tensors constructed at different servers contain only a specific portion of the data depending on the sampling scheme. Therefore, these tensors may not contain information regarding to every user. Table 4 shows how many users are logged as active users by different servers. We decompose these small tensors, sizes of which are given in Table 4 using Tucker3 model. Around $85 - 90\%$ percent of the data fits the model at each sampling site.

Tensor decomposition at each sampling site provides component matrices and singular values corresponding to the singular vectors in the user mode. Since different sets of users are logged at different servers, we pad the rows of $\mathbf{A_i^*\Sigma_i}$ with zeros for the users which are not in the set of users logged by server i. Matrix $\mathbf{M}$, is then formed as in Equation 1 and decomposed by SVD.

An alternative analysis approach is to collect the tensors constructed by different servers at a central location and then decompose one large tensor containing data for the logs of all users during whole period of conversation. We apply Tucker3 analysis on the large tensor and determine the component numbers such that Tucker model fits around 80% of the data compatible with the percent of the data modeled using reduced SVD of matrix $\mathbf{M}$. Let $\mathbf{A}$ be the component matrix for the user mode obtained by the decomposition of large tensor. We also find the singular values corresponding to the singular vectors in this component matrix ($\mathbf{\Sigma}$) as we have done in small tensor analysis and compute $\mathbf{A}^*\mathbf{\Sigma}$.

Last step of the analysis is to cluster user groups. We use $\mathbf{U}^*\mathbf{S}$ from SVD of matrix $\mathbf{M}$ and $\mathbf{A}^*\mathbf{\Sigma}$ from Tucker3 decomposition of the large tensor to find

Server Id	# Users (m1)	# Keywords (m2)	# Time Samples (m3)	# Tucker3 Models	# Entries
1	12	77	3	350	2772
2	13	91	3	406	3549
3	16	99	3	601	4752
4	11	100	3	297	3300
5	14	80	3	467	3360
6	15	68	3	532	3060
7	13	61	3	406	2379
8	14	79	3	467	3318
			Total	*3526*	*26490*
Complete Tensor	28	501	24	117847	336672

Table 4. Size of the tensors collected at each server for time window = 300 seconds. Maximum number of possible Tucker3 models that can be fit in collective tensor analysis is much smaller than maximum number of Tucker3 models for the complete tensor. Maximum number of Tucker3 models for tensor X is the number of all possible component number combinations $[R_1 R_2 R_3]$ such that $R_1 \leq R_2 * R_3$, $R_2 \leq R_1 * R_3$ and $R_3 \leq R_1 * R_2$. Total number of entries in collective tensors are around 8% of the entries in complete tensor.

and compare the user groups identified by collective and central analysis of chatroom tensors. We apply K-means algorithm [7] with different number of clusters, $k = 1, 2, ..6$ and observe that both central and collective analysis of 2-hour chatroom data for time windows 150 and 300 seconds identify the same user clusters. Complete procedure for central and collective tensor analysis is summarized in Figure 2.

3.3 Performance Comparison

Collective analysis of chatroom tensors has several advantages over central analysis of one large tensor. First of all, small tensors do not store the user entries if users do not speak in sampled time windows. Keywords, which are not used during those time windows, are not stored, either. Since most of the zero-entries in complete tensor are omitted, total number of entries shrinks in collective tensor analysis. Table 4 demonstrates that number of entries we keep track of in collective tensor analysis is approximately 8 % of the entries we use in centralized analysis.

Second, determining number of components for Tucker3 model is much easier compared to the central case. Techniques such as residual analysis, DIFFIT [4, 5] etc. determine the right number of components in Tucker3 model based on model fit values. For a 3-way tensor of size I x J x K, model fit should be computed only for the component number combinations, where $IJ \geq K$ and $IK \geq J$ and $JK \geq I$. For the cases, when $IJ < K$, we obtain the same model fit as $IJ = K$ [10]. Number of components are easily determined in collective tensor

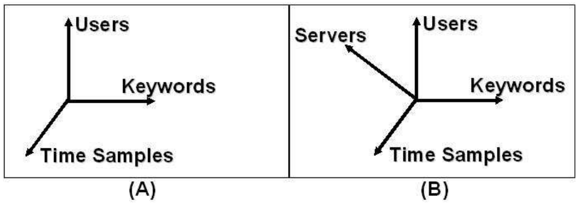

Fig. 3. Chatroom Tensors. 3-way tensors with modes of users, keywords and time samples are constructed at each server. 4-way tensors with modes of users, keywords, time samples and servers are used in sensitivity analysis of chatroom data with respect to different servers.

analysis because total number of Tucker3 models that can be fit to small tensors drops dramatically compared to the number of possible Tucker models for the central tensor. In Table 4, we show that total number of possible models for collective analysis is around 3% of the models that can be fit to the large tensor. Third, tensor analysis of multiple tensors is computationally more efficient than multiway analysis of one large tensor. Computational cost of Tucker3 model using ALS (Alternating Least Square) algorithm is $\prod_{j=1}^{n} m_j * 3R^2$ per iteration, where m_j is the number of dimensions in the j^{th} mode, n is the number of modes and R is the maximum of the component numbers used in Tucker3 analysis. We clearly observe:

$$\sum_{i=1}^{\#\text{ of servers}} ((\prod_{j=1}^{3} m_{ij} * 3r_i^2) * (\#\text{ of iterations})) < \prod_{j=1}^{3} M_j * 3R^2 * (\#\text{ of iterations})$$

where m_{ij} is the dimension of the j^{th} mode of the tensor constructed in server i, M_j is the dimensionality of the j^{th} mode in the complete tensor and r_i is the maximum of component numbers in Tucker3 analysis in server i. Iteration numbers are observed to be approximately the same in both central and collective analysis of tensors.

4 4-way Analysis of Chatroom Data for Sensitivity

4.1 Impact of Server Selection

We construct 3-way tensors with users, keywords and time samples using the chatroom data logged during a specific time period at different servers. Our goal is to explore how tensors formed by different servers compare to each other. At this step, unlike collective tensor analysis, each server logs total chatroom conversation for a specific time period.

Data collected by different servers are arranged into a 4-way tensor, where first, second, third and fourth modes are users, keywords, time samples and servers, respectively (Figure 3 B). We are particularly interested in extracting the structure in the server mode. Therefore, after fitting Tucker model to 4-way tensor, we examine the component matrix corresponding to server mode.

2-hour chatlog is arranged into a 3-way tensor for each server and these are used to construct a 4-way dataset. Server mode also contains the tensor formed by combining partial data from several servers. It is essential to compare this tensor formed from data samples from different servers to all other tensors collected at a single server in order to show the validity of collective partial chatroom analysis.

We fit Tucker model to the 4-way data such that model explains around 95% of the data. This fit value can be achieved by extracting only one component from the server mode. Singular value corresponding to that single singular vector captures 96.97%, 99.88%, 97.40% and 96.36% of the variation for time window sizes of 150 sec., 180 sec., 240 sec. and 300 sec., respectively. Explained variations demonstrate that rank-one reduction in the server mode is enough and coefficients in the extracted singular vector reveal that each server contributes almost equally to the first component. The results also indicate that collection of chatroom data at different servers does not make any difference in terms of the analysis of chatroom tensors.

4.2 Impact of Sampling Window Size

In order to inspect the effect of time window size in the comparison of tensors constructed at different servers, 2-hour chatroom log is arranged as a 4-way tensor using different time window sizes. Our analyses with different time window sizes give the same strong rank-one reduction in the server mode as we have indicated in the previous section. Therefore, we demonstrate that collection of chatroom data at different servers with the given time window sizes does not make any difference in terms of constructed chatroom tensors.

5 Conclusions

In this paper we consider how to collect and analyze multilinear stream data from multiple servers in a distributed way. As an example of such data we consider Internet chatroom communications as our case study to demonstrate the results. We show that sampling in time domain by multiple servers can be used to obtain data with no statistical difference from the data obtained by a centralized approach. Consequently, we discuss how to construct 3-way data arrays and how to analyze the structure of multilinear data represented as high order tensors. Our collective analysis algorithm is based on constructing smaller tensors at each server (sampling site) and applying a tensor decomposition technique to obtain component matrices. The component matrix of interest (e.g., corresponding user groups) from each site is combined into a one larger component matrix which is then analyzed using SVD. We show that this approach compared to constructing a single tensor with full information and analyzing it with the same tensor decomposition technique gives the same structural information for the data. Since we establish the accuracy of the collective approach, we also compare the space and computation cost of collective analysis to the centralized one. We show that on our chatroom communication data, collective analysis provides significant savings. We define an equation that shows the computational cost relationship between centralized and collective analysis approach.

References

1. Acar E., Camtepe S. A., Krishnamoorthy M. and Yener, B. Modeling and Multiway Analysis of Chatroom Tensors. IEEE ISI 2005.
2. Golub G.H. and Loan C.F.V. Matrix Computations. 3rd edn. The Johns Hopkins University Press, Baltimore, MD (1996).
3. Kargupta H., Huang W., Sivakumar K. and Johnson, E. Distributed Clustering Using Collective Principal Component Analysis Knowledge and Information Systems Journal, 3 (2001), 4, pp. 422-448.
4. Timmerman M. and Kiers H.A.L. Three-mode principal component analysis: Choosing the numbers of components and sensitivity to local optima British Journal of Mathematical and Statistical Psychology, 53 (2000), pp.1-16.
5. Kiers H.A.L. and der Kinderen A. A fast method for choosing the numbers of components in Tucker3 analysis British Journal of Mathematical and Statistical Psychology, 56 (2003), pp.119-125.
6. Lathauwer L.D., Moor B. D. and Vanderwalle J. A Multilinear Singular Value Decomposition. SIAM Journal on Matrix Analysis and Applications, Vol 21, No.4, (2000), pp. 1253-1278.
7. MacQueen J.B. Some Methods for classification and Analysis of Multivariate Observations Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1 (1967), pp.281-297.
8. Smilde Age K., Westerhuis J.A. and Boque R. Multiway Multiblock Component and Covariates Regression Models J. Chemometrics, 14 (2000), pp. 301-331.
9. Tucker L. Some mathematical notes on three mode factor analysis. Psychometrika, 31 (1966), pp. 279-311.
10. Wansbeek T. and Verhees J. Models for multidimensional matrices in econometrics and psychometrics R.Coppi and S. Bolasco (Eds.), Multiway Data Analysis, Amsterdam: North Holland.