Modeling and Multiway Analysis of Chatroom Tensors

Evrim Acar, Seyit A. Çamtepe, Mukkai S. Krishnamoorthy, and Bülent Yener*

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

Abstract. This work identifies the limitations of n-way data analysis techniques in multidimensional stream data, such as Internet chatroom communications data, and establishes a link between data collection and performance of these techniques. Its contributions are twofold. First, it extends data analysis to multiple dimensions by constructing n-way data arrays known as *high order tensors*. Chatroom tensors are generated by a simulator which collects and models actual communication data. The accuracy of the model is determined by the Kolmogorov-Smirnov goodness-of-fit test which compares the simulation data with the observed (real) data. Second, a detailed computational comparison is performed to test several data analysis techniques including SVD [1], and multiway techniques including TUCKER1, TUCKER3 [2], and PARAFAC [3].

1 Introduction and Background

Internet Relay Chat (IRC) is a multi-user, multi-channel and multi-server communication medium that provides text-based, real-time conversation capability [4]. Chatroom communication data offer valuable information for understanding how social groups are established and evolve in cyberspace. Recently, there has been intense research focus on discovering hidden groups and communication patterns in social networks (see [5, 6, 7, 8, 9] and references therein).

Chatrooms are attractive sources of information for studying social networks for several reasons. First, chatroom data are public, and anyone can join into any chatroom to collect chat messages. Second, real identities of *chatters* are decoupled from the *virtual identities* (i.e., nick names) that they use in a chatroom. For example, a 50-year old male chatter can participate in a teenager chatroom with multiple virtual identities one of which could be associated with a female persona. Thus, there is no privacy in chatroom communications. Indeed, it is partially this total lack of privacy that makes chatrooms vulnerable to malicious intent and abuse, including terrorist activities. Third, chatroom data are obtained from streaming real-time communications, and contain multidimensional and noisy information. Extracting structure information without understanding

* This research is supported by NSF ACT Award # 0442154.

P. Kantor et al. (Eds.): ISI 2005, LNCS 3495, pp. 256–268, 2005.

the contents of chat messages (i.e., without semantic information) to determine how many topics are discussed, or which chatters belong to the same conversation topics, is quite challenging.

There are several efforts to extract information from chatroom communications [10, 11, 12, 8, 9]. However, current techniques have limited success since chatroom data may have high noise and multidimensionality. Thus, data analysis techniques, such as Singular Value Decomposition (SVD) [1] that rely on linear relationships in two-dimensional representation of data, may fail to capture the structure. In this work, we extend chatroom data analysis to multiple dimensions by constructing multiway data arrays known as *high order tensors*. In particular, we consider three-way arrays (i.e., cubes) with dimensions: (1) users (who), (2) keywords (what), (3) time (when). Accordingly, we consider generalization of SVD to higher dimensions to capture multiple facets of chatroom communications. Multidimensional SVD has been a focus of intensive research [13, 14, 15, 16, 17, 18, 19]. It is well understood that there is no "best" way to generalize SVD to higher dimensions. Computational methods favor greedy algorithms based on iterative computations such as *alternating least squares* (ALS) [16, 15, 13]. For example, most popular multiway data analysis techniques TUCKER3 [2] and PARAFAC [3] use ALS. While special cases (such as tensors with orthogonal decompositions [16]) are possible, in general enforcement of constraints in ALS remains as a challenge.

1.1 Our Contributions

The main goal of this work is to identify the limitations of n-way data analysis techniques, and establish a link between data collection (i.e., tensor construction) and performance of these techniques. More precisely, this paper has several contributions:

i. We present a model and its statistical verification using actual chatroom communications data. The model is used to implement a simulator for generating three dimensional **chatroom tensors** with *user* × *keyword* × *time*.
ii. We examine how two-way data analysis techniques such as SVD would perform on chatroom tensors to extract the structure information. We show that SVD may fail on chatroom tensors even with quite simple structure while three-way data analysis techniques such as TUCKER1 and TUCKER3 are successful.
iii. We investigate how the construction of chatroom tensors would impact the performance of both SVD and three-way data analysis techniques. In particular, we investigate the importance of noise filtering and dimensions of chatroom tensors, and show how sensitive the analysis techniques are.
iv. Finally we compare three-way analysis techniques with each other as a function of several metrics, such as number of components, explained variation, number of parameters and interpretability of the models. We show that high model complexity (w.r.t. the parameters), which is an indication of more modeling power and more explained variation, does not necessarily capture the right structure when data are noisy.

Organization of the paper: This paper is organized as follows. Section 2 describes the data collection procedure from chatroom channels and verifies the simulation model. In Section 3, we discuss the impact of data construction on two-way and multiway analysis techniques using both special cases and simulation data. This section also compares the performance of different data analysis techniques in extracting the internal structure of data.

2 Modeling and Simulating Chatroom Data

Data Collection: We have implemented an IRC bot similar to one used in [8]. The bot connects to an IRC server, and joins to given channel. It logs public messages and control sequences (nick, quit, kick, leave, etc.) flowing in the channel. We have collected 24 hours, 20 days (November 2004) of logs from *philosophy* channel in *dallas.tx.us.undernet.org undernet* server [1]. The log file is 25 MB in size, and includes 129,579 messages.

The logs are processed for 4-hour period between 16:00-20:00 for 20 days. There are average of 10 to 20 active users during this period. We keep track of *join* and *change nick* messages to determine a chain of nicks (e.g., change nick from A to B, B to C, etc.), and associate the chain with a single *user*.

Table 1 shows interarrival statistics obtained over 20-days of data. Rows of the table show the statistics when the interarrival time is bounded by the given value. Note that, 99.82% of the interarrival times are less then 300 seconds. Thus, we assume that no conversation could survive 300 seconds of silence.

Table 1. Interarrival time statistics over 20-day of log

Interarrival Time	Mean	Median	STD	Skewness	Kurtosis	Number of Messages	% of Messages
≤ 60	11.97	8	11.6	1.57	2.36	103,997	97.36
≤ 180	13.73	5	16.75	3.37	17.6	106,449	99.66
≤ 300	14.08	9	18.9	4.86	40.65	106,624	99.82

Table 2 presents the statistics for *message interarrival time*, *message size* in terms of word counts and *number of messages per user*. Message size and interarrival time fit to exponential distribution with parameters ($\mu = 0.0801$) and ($\lambda = 0.0677$), respectively. Number of messages per user obeys to a power law distribution with exponent ($\alpha = -1.0528$).

Identifying Keywords: Users in a chatroom talk about several topics which may overlap in time domain. We define a *conversation* as a sequence of posts made by at least two topic members.

[1] There was no specific reason for choosing the philosophy channel. It is one of the many channels with less junk information.

Table 2. Results of analysis on 4-hour x 20-day of log for message size, interarrival time and number of messages per user

	Mean	Std	Skewness	Kurtosis	# Samples	Distribution
Message Size	12.47	11.17	1.86	4.83	18,483	$f(x) = 0.0801\ e^{-0.0801x}$
Interarrival Time	14.76	19.29	4.68	37.58	18,430	$f(x) = 0.0677e^{-0.0677x}$
# Mess. per User	21.24	33.78	2.88	10.00	870	$f(x) = 0.2032x^{-1.0528}$

It is possible to find a set of specific keywords for each topic which are frequently used by the topic members. However, care must be taken to handle irregular verbs or verbs with $-ed$, $-ing$, $-s$ to treat them as the same word. We consult to the online *webster* (www.webster.com) dictionary to find the simple forms of these words. We consider common words among several topics as *noise* if they are not *specific keywords* of any topic. Noise also includes typos or other unresolved words by *webster*.

Model: We developed a model for chatroom communications based on the statistical observations on the real data. Model accepts five parameters: (i) distribution for interarrival time, (ii) distribution for message size in terms of word count, (iii) distribution for number of messages per user, (iv) noise ratio (NR), and (v) time period.

Given a topic-tuple $T_1, \cdots, T_n$ of n topics, the model computes the number of messages $m_{j,k}$ posted by user j on topic k. This number is assigned according to a power law distribution which is obtained from the statistics collected over real data. Once $m_{j,k}$ is determined, *message posting probability* for a user h is calculated as $m_{h,k} / \sum_{\forall j} m_{j,k}$. For the philosophy channel, interarrival time obeys exponential distribution which is generated by a Poisson arrival process with arrival rate of λ. Thus, conversation duration for a topic-tuple becomes: $\sum_{\forall j} \sum_{\forall k} m_{j,k} * 1/\lambda$. We model a chatroom log as a queue with multiple Poisson arrival processes. Suppose there are $T_1, ..., T_n$ of n topics each with $M_1, ..., M_n$ messages respectively. Then, the arrival rate for each topic will be $\lambda_1, ..., \lambda_n$ respectively where:

$$\lambda_i = \frac{M_i * \lambda}{\sum_{\forall j} M_j},\ 1 \leq i \leq n$$

Noise Modeling: In this work we use Gaussian noise to introduce a model parameter *noise ratio* (NR) as: *NR = (Topic Specific Words + Noise Words) / Noise Words*. Once message size is decided for a user, number of specific topic words and number of noise words are decided based on this ratio. Specific words are selected uniformly at random from the keyword set of the topic. Noise words are randomly selected according to Gaussian distribution. Gaussian distribution selects some of the words very frequently and some others very rarely. Frequently selected words represent the type of noise words which are used frequently by everybody in the chatroom. Rarely selected words represent typo like noise words which are used rarely in the chatroom. When all selected users in a topic post,

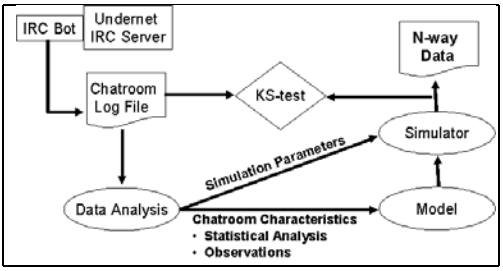

Fig. 1. System architecture for data collection, modeling and simulator

Table 3. Kolmogorov-Smirnov goodness-of-fit test (KS-test) results. *Interarrival time*, *message size* and *number of messages per user* data from model is compared to real chatroom data for the listed significance levels. KS-test does not reject null hypothesis which states that both *synthetic* and *chatroom data* come from the same distribution

	Synthetic Data # Samples	Chatroom Data # Samples	Asymptotic P-value	Significance Level
Interarrival Time	962	18,430	0.1800	10%
Message Size	1,034	18,483	0.2022	10%
# Mess. per User	57	870	0.0521	5%

number of posts for these selected users is decremented, and posting probabilities are recalculated.

Verification: Based on the model, we implement a simulator in Perl. Simulator receives its parameters from a configuration file, and generates chatroom-like communication logs according to the model to be used for verification. Figure 1 presents overall data flow. We perform goodness-of-fit test over synthetic and real data. Table 3 represents Kolmogorov-Smirnov goodness-of-fit test (KS-test) results for the listed significance levels. For all the cases, KS-test does not reject null hypothesis which states that both synthetic and chatroom data come from the same distribution.

3 Computational Comparison of 2-way and 3-way Data Analysis Techniques

We use specific datasets and simulation data to assess two-mode and three-mode analysis techniques. We demonstrate that two-way methods are not as powerful as three-way techniques in capturing the structure of data broken into a number of user groups. We define "user group" as the set of users who share a maximal keyword set in a given time period. Our analyses are conducted in Matlab using Tensor Class [20] for tensor operations and N-way ToolBox[21] for implementations of Tucker and PARAFAC models.

3.1 Impact of Tensor Data Construction

There are two types of data used in this section: (i) manually created data, and (ii) simulation data. For three-mode analysis, we rearrange the data into a tensor, $T \in R^{u \times k \times t}$, defined by *user x keyword x time* modes, where T_{ijk} shows the number of keyword j sent by user i during time slot k. For two-mode analysis based on SVD, we prepare two matrices: $UK \in R^{u \times k}$, where u and k are the number of users and keywords, respectively. Each entry UK_{ij} shows the number of keyword j sent by user i. Second matrix is matrix $UT \in R^{u \times t}$, where t is the number of time slots and UT_{ij} indicates the number of total keywords sent by user i in time slot j. Our objectives are twofold (i) to construct examples where SVD fails to discover the structure while three-way methods Tucker1 and Tucker3 succeed, and (ii) to generate tensors with the same properties as the actual chatroom-like communication data using the simulator and examine the impact of noise and time window size on the performance of analysis techniques.

Noise-Free Tensors with Disjoint Groups and Keywords: First dataset has a structure as shown in Table 4(a). Group 1 and 2 talk about the same topic using a common keyword set while Group 3 and 4 make use of a completely different keyword set. Group 1 and 3 always speak at odd time slots whereas Group 2 and 4 occupy even time slots. We note that data are *noise-free* thus there are no words that are not keywords and there are no users that do not belong to a group.

In such a setting, SVD on matrix UK tends to cluster the users using the same keywords. Similarly, SVD of matrix UT forms clusters containing the users that speak during the same time slots. Therefore, both methods fail to discover the internal structure of data, which actually contains 4 separate groups.

There are, in total, 10 users and 2 keyword sets each containing 2 keywords. Simulation time is 42 time slots. We can represent the sample data as a tensor A of size 10 x 4 x 42 or an unfolded matrix M with dimensions 10 x 168. Best ranks of matrices, UT, UK and M as well as best rank of each mode of tensor A are determined for rank reduction. The users are mapped on the spaces spanned by singular vectors chosen via rank reduction to identify the

Table 4. (a) First specific dataset where group membership and keywords are disjoint, (b) Second specific dataset where groups have common members but keywords are disjoint, and (c) Simulation dataset where groups have common members but keywords are disjoint

Groups	Members		
	(a)	(b)	(c)
1	User 1,2	User 1,2,3	User 1,2,3,4
2	User 3,4,5	User 2,3,4	User 3,4,5,6
3	User 6,7	User 5,6	User 7,8
4	User 8,9,10	User 7,8	User 9,10

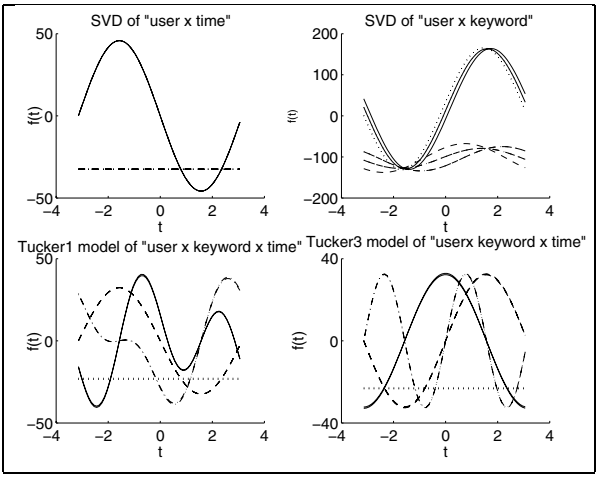

Fig. 2. SVD on *user x time*, UT and *user x keyword*, UK matrices are not powerful enough to find all four user groups while Tucker1 and Tucker3 are capable of extracting all groups from the data

clusters and the structure in the data. It is possible to have only two or three significant singular vectors but higher than three depending on the structure of data should also be anticipated. Let k be the number of most significant singular values identified by rank reduction. We multiply k significant singular values with their corresponding left singular vectors. If $U \in R^{nxm}$ and $S \in R^{mxm}$ represent left singular vectors and singular values of the original matrix, respectively, we compute matrix $F = U(:, 1:k) * S(1:k, 1:k)$ and regard $F \in R^{nxk}$ as a multidimensional dataset where each row represents a user and each column is one of the k properties of a user.

We represent the results of two-mode and three-mode methods using Andrew's curves [22], which transform multidimensional data into a curve, enable us to visualize graphically the structure of data stored in matrix F (i.e., how users are spread on the space spanned by more than 2 or 3 components) [1].

Noise Free Tensors with Overlapping Groups and Disjoint Keywords: The second dataset shown in Table 4(b) is similar to the first one except that overlapping user groups are allowed in order to inquire the performance of analysis methods in the presence of common users.

[1] To visualize the behavior of user i, i^{th} row of matrix F, F_i, is converted into a curve represented by the following function:

$$f_i(t) = \begin{cases} \frac{X_{i,1}}{\sqrt{2}} + X_{i,2}\sin(t) + X_{i,3}\cos(t) + ... + X_{i,p}\cos(\frac{p-1}{2}t) & \text{for } p \text{ odd} \\ \frac{X_{i,1}}{\sqrt{2}} + X_{i,2}\sin(t) + X_{i,3}\cos(t) + ... + X_{i,p}\sin(\frac{p}{2}t) & \text{for } p \text{ even} \end{cases}$$

where $t \in [-\pi, \pi]$

SVD of matrices UK and UT both perform poorly for this case: it can distinguish common users; but overlapping users are treated as a separate cluster. We consider this result as a failure because such analysis is capable of finding subsets of groups while missing the whole group structure. We note that in this case, Andrew's curves are not sufficient to differentiate common users. Therefore, we make use of an unsupervised clustering algorithm called fuzzy c-means [23], which returns membership degrees of each user for each group. Among many clustering algorithms, fuzzy c-means is an appropriate choice for chatroom data because it allows a data point to be in more than one cluster. C-means algorithm returns different membership values for each run since it is a nondeterministic algorithm. The results presented in Table 5, are the cases that represent the majority of 100 runs. SVD of UT gives the result in the table in 70% of the runs and SVD of UK returns the recorded result in 85% of the runs. The results for Tucker1 and Tucker3 are explained in detail below.

Rows named as "Groups", show which group each user is assigned to according to the results of membership values. SVD on UT, groups Users 1, 5 and 6

Table 5. Membership values for users given by fuzzy c-means clustering algorithm. Each user belongs to a group with certain probability represented by membership values. The highest probability determines the group each user belongs to

	User1	User2	User3	User4	User5	User6	User7	User8
SVD of UT								
Pr(Usr ∈ Grp1)	0.5000	0.0000	0.0000	0.0000	0.5000	0.5000	0.0000	0.0000
Pr(Usr ∈ Grp2)	0.0000	1.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Pr(Usr ∈ Grp3)	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	1.0000	1.0000
Pr(Usr ∈ Grp4)	0.5000	0.0000	0.0000	0.0000	0.5000	0.5000	0.0000	0.0000
Groups	1 or 4	2	2	3	1 or 4	1 or 4	3	3
SVD of UK								
Pr(Usr ∈ Grp1)	1.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
Pr(Usr ∈ Grp2)	0.0000	1.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Pr(Usr ∈ Grp3)	0.0000	0.0000	0.0000	0.0000	0.0016	0.0124	0.9984	0.9876
Pr(Usr ∈ Grp4)	0.0000	0.0000	0.0000	0.0000	0.9984	0.9876	0.0016	0.0124
Groups	1	2	2	1	4	4	3	3
Tucker1								
Pr(Usr ∈ Grp1)	1.0000	0.0042	0.0055	0.0530	0.0000	0.0000	0.0000	0.0000
Pr(Usr ∈ Grp2)	0.0000	0.0005	0.0007	0.0499	1.0000	1.0000	0.0000	0.0000
Pr(Usr ∈ Grp3)	0.0000	0.9948	0.9931	0.8429	0.0000	0.0000	0.0000	0.0000
Pr(Usr ∈ Grp4)	0.0000	0.0005	0.0007	0.0541	0.0000	0.0000	1.0000	1.0000
Groups	1	3	3	3	2	2	4	4
Tucker3								
Pr(Usr ∈ Grp1)	0.0596	0.0005	0.0007	0.0000	1.0000	1.0000	0.0000	0.0000
Pr(Usr ∈ Grp2)	0.8105	0.9927	0.9905	0.0000	0.0000	0.0000	0.0000	0.0000
Pr(Usr ∈ Grp3)	0.0653	0.0062	0.0080	1.0000	0.0000	0.0000	0.0000	0.0000
Pr(Usr ∈ Grp4)	0.0646	0.0006	0.0008	0.0000	0.0000	0.0000	1.0000	1.0000
Groups	2	2	2	3	1	1	4	4

together although they just share the time space and do not form a group. We observe the same behavior for Users 4, 7 and 8. This is an expected outcome because SVD on matrix UT maps the users with similar chatting pattern in time closer to each other in the space spanned by left singular vectors. Another important point is that common users are clustered as a completely separate group from other users whom they are talking to. SVD on matrix UK, also shows the same behavior. For overlapping groups model, SVD of matrix UK performs poorly compared to SVD of matrix UT because it cannot capture that Users 1 and 4 are in different groups. However, it outperforms the results of SVD on time mode by correctly extracting Groups 3 and 4 from data. All in all, we observe that two -way analysis results do not reflect the real structure of data. Similarly, at first sight, neither Tucker1 nor Tucker3 seems to capture the exact structure but the cases shown for Tucker1 and Tucker3 occur approximately 50% of the time. If User2 and User3 are clustered with User4 in half of the runs, then they are clustered with User1 in the other half. Therefore, probability of User2 and User3's being in the same cluster with User1 and User4 are both close to 0.5. Thus, we can make a hypothesis for group membership based on these probabilities.

Noisy Tensors and Impact of Noise Ratio: Using the simulator, we implement a model where we demonstrate the effect of noise in extracting the structure of data for two-way and three-way methods. Noise is introduced in keywords mode by the use of a number of words shared by all user groups.

Experimental model consists of 4 groups of users as shown in Table 4(c). As in the scenario of special cases, Group 1 and 2, and similarly Group 3 and 4 use the same keyword sets. Keyword sets are distinct and contain 10 keywords each. Groups making use of distinct keyword sets can talk during the same time period with equal probability.

We create separate chat logs for a simulation time of 4 hours for different noise levels indicated by NR. We set the minimum and maximum number of messages that can be sent by each user as 30 and 70, respectively. We assess the relative performance of SVD, Tucker1 and Tucker3 models on different noise levels and demonstrate our results in Table 6. After the selection of significant components for user mode, we run fuzzy c-means clustering algorithm 100 times to see how often the pattern discovered by clustering method coincides with the internal structure of data. SVD on matrix UK fails to find the right data pattern because it tends to cluster users talking about the same topic regardless of their conversation slots. SVD on matrix UT can capture the structure with moderate success ratios while Tucker1 and Tucker3 have the best results. As noise level increases, we observe that all algorithms start to suffer at some threshold values. Analysis results suggest that Tucker3 performs better than Tucker1 on noisy tensor data.

There are different concepts of noise but our approach in introducing noise tends to form a single keyword cluster. When we observe the behavior of singular values under the effect of noise, we clearly see that noisy data approach to rank-1 in keyword mode while becoming full rank in user mode.

Table 6. Impact of NR (noise ratio) on success ratios of two-way and three-way analysis methods when time window is 240 seconds and tensor is of form 10x100x60

NR	SVD userXkeyword	SVD userXtime	TUCKER1	TUCKER3
No Noise	0%	56%	100%	100%
5	0%	55%	100%	100%
3	0%	47%	73%	100%
2	0%	0%	3%	16%

Impact of Sampling Time Window: On the same experimental set up, we show how different time window sizes affect the performance of analysis techniques. We work on a noise-free dataset to be able to observe solely the effect of window size on success ratios. In this setting, minimum and maximum number of messages sent by a user are 1 and 200, respectively. We create a single chat log of 4 hours and generate data for different time window sizes. Analysis results in Table 7 display that there exists a threshold window size below which none of the analysis methods can discover the structure of data. When time window is roughly 200-300 seconds, three way models have the best performance while SVD on UT can capture the structure only to some extent. Under 180 seconds, none of the methods succeeds. We also observe a performance degradation in all methods as time window size increases considerably. This is an indication of an upper bound on time window size over which users talking at different time slots are considered in the same time period. This hides the communication pattern completely. It is important to observe the relative performance of algorithms in Table 7 rather than exact success ratios or exact time window threshold values.

Table 7. Impact of sampling time window on success ratios of two-way and three-way analysis methods for noise-free data. Total simulation time is 14400 seconds. Tensor is constructed as*user x keyword x time*. As sampling time window changes, dimension of the tensor in time mode is adjusted accordingly

Tensor	Time Window (seconds)	SVD userXkeyword	SVD userXtime	TUCKER1	TUCKER3
10x20x1	14400	0%	0%	0%	0%
10x20x2	7200	0%	0%	13%	17%
10x20x8	1800	0%	0%	17%	22%
10x20x12	1200	0%	25%	24%	27%
10x20x24	600	0%	28%	26%	26%
10x20x48	300	0%	20%	100%	100%
10x20x72	200	0%	14%	100%	100%
10x20x80	180	0%	0%	0%	0%
10x20x160	90	0%	0%	0%	0%

Similar to the effect of noise case, when we inspect the behavior of singular values in user mode, we observe that as time window size gets smaller, we observe a structure close to full rank.

3.2 Performance Comparison of Multiway Techniques

We assess the performance of Tucker1, Tucker3 and PARAFAC with respect to several metrics such as number of components, explained variation, number of parameters and interpretability of models. Performance analysis results suggest that Tucker3 model provides the best interpretation. In case of noise-free data, there is no difference in using Tucker1 or Tucker3 decomposition in terms of data interpretation. However, for Tucker1, number of parameters, which is the total number of entries in matrices/ core tensors produced in tensor decomposition, is much larger than the number of parameters in Tucker3 and more parameters introduce complication in data interpretation. PARAFAC is not appropriate for modeling our data because of its strict modeling approach. It does not allow extraction of different number of components in different modes. Besides, while Tucker3 model enables us to decompose a tensor into orthogonal component matrices X, Y, and Z and estimate orthonormal bases, in PARAFAC, we can only do that if tensor is diagonalizable. In Table 8 we present the results of

Table 8. Performance comparison of N-way analysis techniques for time window 240 seconds and tensor 10x100x60. Tucker1, Tucker3 and PARAFAC are compared based on explained variation, number of parameters used in each model and success ratio of capturing the structure. Comparison of the models is presented for two different noise levels, NR=0 and NR=3

	NR	Number of Components	Explained Variation	Number of Parameters	Structure
Tucker1	0	5	84.6493	30050	100%
Tucker3	0	5 5 5	76.9814	975	100%
Tucker3	0	5 2 5	76.5944	600	100%
Parafac	0	5	48.753	850	0%
Tucker1	0	8	95.1406	48080	75%
Tucker3	0	8 8 8	84.8294	1872	100%
Tucker3	0	8 2 8	83.7563	888	100%
Parafac	0	8	49.4294	1360	0%
Tucker1	3	5	77.4148	30050	5%
Tucker3	3	5 5 5	62.7061	975	7%
Tucker3	3	5 2 5	62.2053	600	11%
Parafac	3	5	40.7648	850	0%
Tucker1	3	4	69.5659	24040	69%
Tucker3	3	4 4 4	58.3426	744	64%
Tucker3	3	4 2 4	58.2482	512	100%
Parafac	3	4	40.3238	680	0%

performance comparison for multiway techniques. Note that even if we extract the same number of components in each mode, Tucker3 model is more robust.

When data are noisy, we observe performance degradation in terms of interpretability in Tucker1 while Tucker3 can still capture the structure successfully if right number of components is determined for each mode. Table 8 demonstrates the importance of right component numbers in success ratio of data interpretation. Similarly, it gives an example of a case where selection of component numbers just taking into fit of the model into account does not necessarily imply better interpretation of data.

4 Conclusions

In this work we show how to generate three-way chatroom tensors and examine the performance of data analysis algorithms. We show that three-dimensional chatroom tensors contain multilinear structure that cannot be detected by SVD. The performance gap between SVD and multiway analysis techniques Tucker1 and Tucker3 grows as a function of increasing noise in the data. We also show that construction of the chatroom tensor with respect to sampling window size has significant impact on the performance of analysis techniques. We examine the performance of Tucker1, Tucker3 and PARAFAC with respect to several metrics such as number of components, explained variation, number of parameters and interpretability of the models. Our results suggest that there is no difference in using Tucker1 or Tucker3 decomposition if the data are noise-free. In general, Tucker3 model provides the best interpretation and has the advantage of less number of parameters compared to Tucker1. We note that one of the challenges left for further research is to determine the optimal number of components to obtain the most accurate structure information. It is evident from our study that how data are collected and represented have significant impact over discovering the structure hidden in them.

References

1. Golub, G., Loan, C.: Matrix Computations. 3 edn. The Johns Hopkins University Press, Baltimore, MD (1996)
2. Tucker, L.: Some mathematical notes on three mode factor analysis. Psychometrika **31** (1966) 279–311
3. Harshman, R.: Foundations of the parafac procedure: Model and conditions for an explanatory multi-mode factor analysis. UCLA WPP **16** (1970) 1–84
4. Kalt, C.: Internet Relay Chat. RFC 2810, 2811, 2812, 2813 (2000)
5. Krebs, V.: An introduction to social network analysis. http:// www.orgnet.com/ sna.html (accessed February 2004) (2004)
6. Magdon-Ismail, M., Goldberg, M., Siebecker, D., Wallace, W.: Locating hidden groups in communication networks using hidden markov models. In: Intelligence and Security Informatics (ISI'03). (2003)

7. Goldberg, M., Horn, P., Magdon-Ismail, M., Riposo, J., Siebecker, D., Wallace, W., Yener, B.: Statistical modeling of social groups on communication networks. In: First conference of the North American Association for Computational Social and Organizational Science (NAACSOS'03). (2003)
8. Camtepe, S., Krishnamoorthy, M., Yener, B.: A tool for internet chatroom surveillance. In: Intelligence and Security Informatics (ISI'04). (2004)
9. Camtepe, S., Goldberg, M., Magdon-Ismail, M., Krishnamoorty, M.: Detecting conversing groups of chatters: A model, algorithms, and tests. In: IADIS International Conference on Applied Computing. (2005)
10. Mutton, P., Golbeck, J.: Visualization of semantic metadata and ontologies. In: Seventh International Conference on Information Visualization (IV03), IEEE (2003) 300–305
11. Mutton, P.: Piespy social network bot. http://www.jibble.org/piespy (accessed January 2005) (2001)
12. Viegas, F., Donath, J.: Chat circles. In: ACM SIGCHI (1999), ACM (1999) 9–16
13. Kroonenberg, P.: Three-mode Principal Component Analysis: Theory and Applications. DSWO press, Leiden (1983)
14. Leibovici, D., Sabatier, R.: A singular value decomposition of a k-ways array for a principal component analysis of multi-way data, the pta-k. Linear Algebra and its Applications **269** (1998) 307–329
15. Lathauwer, L., Moor, B., Vandewalle, J.: On the best rank-1 and rank-(r1,r2,...,rn) approximation of higher-order tensors. SIAM J. Matrix Analysis and Applications **21** (2000) 1324–1342
16. Zhang, T., Golub, G.: Rank-one approximation to higher order tensors. SIAM J. Matrix Analysis and Applications **23** (2001) 534–550
17. Kolda, T.: Orthogonal tensor decompositions. SIAM J. Matrix Analysis and Applications **23** (2001) 243–255
18. Kofidis, E., Regalia, P.: On the best rank-1 approximation of higher-order supersymmetric tensors. SIAM J. Matrix Analysis and Applications **22** (2002) 863–884
19. Kolda, T.: A counter example to the possibility of an extension of the eckart-young low-rank approximation theorem for the orthogonal rank tensor decomposition. SIAM J. Matrix Analysis and Applications **24** (2003) 762–767
20. Kolda, T., Bader, B.: Matlab tensor classes for fast algorithm prototyping. Technical Report SAND2004-5187, Sandia National Laboratories (2004)
21. Andersson, C., Bro, R.: The N-way Toolbox for MATLAB. Chemometrics and Intelligent Laboratory Systems. (2000)
22. Andrews, D.: Plots of high-dimensional data. Biometrics **28** (1972) 125–136
23. Bezdek, J.: Pattern Recognition with Fuzzy Objective Function Algoritms. Plenum Press, New York (1981)