A Model to Summarize User Action Log Files

Eleonora Gentili
Supervisor: Alfredo Milani

Dipartimento di Matematica e Informatica, Università degli Studi di Perugia
via Vanvitelli 1 – 06123 Perugia, Italy
eleonora.gentili@dmi.unipg.it

Abstract. Social networks, web portals or e-learning platforms produce in general a large amount of data everyday, normally stored in its raw format in log file systems and databases. Such data can be condensed and summarized to improve reporting performance and reduce the system load. This data summarization reduces the amount of space that is required to store software data but produces, as a side effect, a decrease of their informative capability due to an information loss. In this work we study the problem of summarizing data obtained by the log systems of chronology-dependent applications with a lot of users. In particular, we present a method to reduce the data size, collapsing the descriptions of more events in a unique descriptor or in a smaller set of descriptors and pose the optimal summarization problem.

1 Introduction

During last years we have seen an impressive growth and diffusion of applications shared and used by a huge amount of users around the world. Network traffic data log systems, alarms in telecommunication networks and web portals which records the user activities are examples of chronology-dependent applications (CDA) producing in general large amount of data in the form of log sequences.

But log files are usually big and noisy, and the difficulty of finding patterns is very high as well as the number of patterns discovered can be very large [6]. For this reason, data in log files can be condensed and summarized to improve reporting performance and analyze the system behavior.

In this work, a new method to produce a concise summary of sequences of events related to time is presented, which is based on the data size reduction obtained merging time intervals and collapsing the descriptions of more events in a smaller set of descriptors. Moreover, in order to obtain a data representation as compact as possible, an abstraction operation allowing an event generalization process (as in [5]) is defined. The summarized time sequence can substitute the original time sequence in the data analysis.

The reduction of the amount of data produces also as a side effect, a decrease of their informative capability due to an information loss. For this reason, we formally define the summarization problem as an optimization problem that balances between shortness of the summary and accuracy of the data description.

1.1 Related Works

In [5], Pham et al. propose an algorithm that achieves time sequence summarization based on a generalization, grouping and concept formation process, maintaining the overall chronology of events. Our summarization method overcomes the time limitation of this procedure, using time intervals, instead only time instants.

In [6], [4], [3], authors propose methods to produce a comprehensive summary for the input sequence, focused on the frequency changes of event types across adjacent segments. In particular, in [4], Kiernan and Terzi rely to the Maximum Description Length (MDL) principle to produce the summarized time sequence balancing the shortness of the summary and accuracy of the data description. Moreover, in [3], the presented framework summarizes an event sequence using inter-arrival histograms to capture the temporal relationships among events using the MDL. Unlike these works, where are presented methods which partition the time sequence into segments and study global relationships on each segment, and among segments, our method takes into account an overview of the time sequence to solve the summarization problem.

In [1], Chandola et al. formulates the problem of summarization as an optimal summarization problem involving two measures, Compaction Gain and Information Loss, which assess the quality of the summary.

2 The model

In this work we assume that each event is described by a set of users which made actions over objects either at a particular instant or during an interval. Moreover, the assumed time model is discrete, where events are instantaneous and are representable by a tuple (u, a, o, t), with which we describe *users*, *actions*, *objects* and *time* of the actions.

In order to represent more general situations and potentially aggregate information of similar events, we define events as tuples involving sets of users, actions and objects possibly occurred during an interval.

Definition 1. *An event descriptor is a t-uple $X = (U, A, O, I, \delta)$ representing a set of actions A made by a set of users U over a set of objects O, during a given time interval I according to the covering index δ that is defined as the ratio by the number of points in which the actions in A are actually executed and all the points of I.*

Example 1. $X = (\{admin\}, \{login, logout\}, \{IT\}, [10, 50], 0.30)$ represents that *admin* made *login* and *logout* in ITcourse in the 30% of the points of $I = [10, 50]$.

We assume that the labels in the sets U, A and O can be organized in taxonomies, which are organized in hierarchies with multiple inheritance: each taxonomy is associated to an *abstraction operator* ($\uparrow$), allowing to climb the hierarchy.

The abstraction operator applied to a node of the taxonomy returns all the fathers of the node, while, when it is applied to a set of nodes $S = \{s_1, \ldots, s_n\}$, the result is a set $\uparrow (S) = S'$ where at least a s_i is substituted with $\uparrow (s_i)$. Let's two different sets S_1 and S_2, the *minimal abstracting set* S is defined as the first not null set of common ancestor of S_1 and S_2 computed by climbing the taxonomy graph associated to S_1 and S_2, i.e. such that $S = \uparrow (S_1) = \uparrow (S_2)$.

Definition 2. *A time sequence $\boldsymbol{X} = (X_1, \ldots, X_m)$ is a sequence of m event descriptors X_i; m is called size of the time sequence, or data size.*

Given a *time sequence*, we aim to provide methods to reduce its data size, collapsing the descriptions of more events in a smaller set of event descriptors.

Definition 3. *Let's Ω the set of all event descriptors, $X_1 = (U, A, O, I_1, \delta_1)$ and $X_2 = (U, A, O, I_2, \delta_2)$ two event descriptors, $I_1 = [t'_1, t''_1]$ and $I_2 = [t'_2, t''_2]$, we define the merging operator as*

$$\begin{gathered} \oplus : \Omega \times \Omega \rightarrow \Omega \\ ((U, A, O, I_1, \delta_1), (U, A, O, I_2, \delta_2)) \mapsto (U, A, O, I, \delta) \end{gathered} \tag{1}$$

such that $I = [min(t'_1, t'_2), max(t''_1, t''_2)]$ and

$$\delta = \frac{\delta_1 |I_1| + \delta_2 |I_2| - \min(\delta_1 |I_1|, \delta_2 |I_2|, |I_1 \cap I_2|)}{|I|} \tag{2}$$

The *merging operator* collapses intervals of *event descriptors* with identical label sets. δ is computed considering that events happening in both I_1 and I_2 coincide as much as possible; it is simple to prove that $\delta \leq \max(\delta_1, \delta_2)$.

Example 2. Given the time sequence $\mathbf{X} = \{X_1, X_2, X_3, X_4\}$, where

$$X_1 = (\{user_1, user_2\}, \{login\}, \{objA\}, [1,1], 1.0), X_2 = (\{user_2\}, \{send\}, \{objA\}, [2,2], 1.0),$$
$$X_3 = (\{user_1\}, \{read\}, \{objA\}, [4,4], 1.0), X_4 = (\{user_2\}, \{send\}, \{objA\}, [5,5], 1.0).$$

We can apply the *merging operator* to X_2 and X_4 obtaining

$$X_{24} = X_2 \oplus X_4 = (\{user_2\}, \{send\}, \{objA\}, [2,4], 0.67).$$

The new *time sequence* $\mathbf{X} = \{X_1, X_{24}, X_3\}$ has a smaller size but less information about events. And no more *merging operation* can be applied.

Definition 4. *Let's Ω the set of all event descriptors and given an event descriptor $X = (U, A, O, I, \delta)$, the abstraction operator $\uparrow_S$ is defined as*

$$\begin{gathered} \uparrow_S : \Omega \rightarrow \Omega \\ (U, A, O, I, \delta) \mapsto (U', A', O', I, \delta) \end{gathered} \tag{3}$$

where $S \in \{U, A, O\}$ and $S' = \uparrow (S)$ is obtained applying the abstraction operator.

The *abstraction operator* will make mergeable *event descriptors* having different label sets, generalizing labels in the sets U, A, O.

Definition 5. *Let's $\{X_i : X_i = (U_i, A_i, O_i, I_i, \delta_i), i = 1, \ldots, n\}$ a set of event descriptors, $X_i^* = (U, A, O, I_i, \delta_i)$ is the minimal abstracting event for X_i if each label set U, A, O is the minimal abstracting set respectively for $\{U_i\}$, $\{A_i\}$, $\{O_i\}$.*

For instance, let consider the taxonomy graphs depicted in Fig.1, and given

$$X_1 = (\{user_1\}, \{Create.folder, Save\}, \{log_1\}, I_1, \delta_1),$$
$$X_2 = (\{user_1, user_2\}, \{Disk.op, Write.mail\}, \{log_1\}, I_2, \delta_2),$$

the two *minimal abstracting events* for X_1 and X_2 are $X_1^* = (\{user\}, \{User.op\}, \{log_1\}, I_1, \delta_1)$ and $X_2^* = (\{user\}, \{User.op\}, \{log_1\}, I_2, \delta_2)$ respectively.

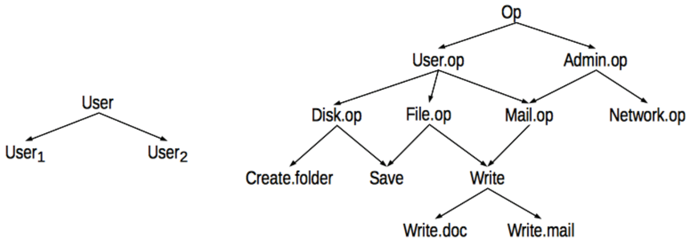

Fig. 1. An example of taxonomy graphs respectively over the sets U and A.

Considering that creating a summary produces information loss, the optimal summarization problem aims to maximize the reduction of data size minimizing the information loss. It is clear that the optimal summarization is a question of tradeoff between application of the *merging operator* and the *abstraction operator* to the *event descriptors*.

3 The optimal summarization problem

Let's $\mathbf{X}_0$ the time sequence of the initial volume of data and $\mathbf{X}$ a summarized time sequence, we define some metrics to assess the quality of $\mathbf{X}$ with respect to $\mathbf{X}_0$.

Definition 6. *The compaction gain of* $\boldsymbol{X}$ *is define as* $\mathcal{C}(\boldsymbol{X}, \boldsymbol{X}_0) = \frac{|\boldsymbol{X}_0|}{|\boldsymbol{X}|}$.

Definition 7. *Given* $I_i = [t'_i, t''_i]$ *and* $G_i = [t''_i, t'_{i+1}]$*, the covering accuracy of* $\boldsymbol{X}$ *is*

$$\mu(\boldsymbol{X}) = \frac{\sum_{i=1}^{n} \delta_i |I_i| + \sum_{i=1}^{m} |G_i|}{\sum_{i=1}^{n} |I_i| + \sum_{i=1}^{m} |G_i|}. \tag{4}$$

The gap intervals G_i are considered as intervals with $\delta_{G_i} = 1.0$, because there are no events happening in each G_i. It easy to prove that $0 \le \mu(\mathbf{X}) \le 1$. In particular, $\mu(\mathbf{X}) = 1$ is verified if and only if $\delta_i = 1$, $\forall i = 1, \ldots, n$.

Definition 8. *Given* $\boldsymbol{X}$*, the description accuracy of* $\boldsymbol{X}$ *is defined as*

$$\eta(\boldsymbol{X}) = \min_{X \in \boldsymbol{X}} \left(\min(\omega_U \eta(U), \omega_A \eta(A), \omega_O \eta(O))\right), \tag{5}$$

where $\omega_U, \omega_A, \omega_O \ge 0$ *are the weights of the label sets, and*

$$\eta(S) = \min_{n \in S} \frac{d(r, n)}{h(n)},$$

where $S \in \{U, A, O\}$*,* r *is the root of the taxonomy* T_S *and* $h(n)$ *is the longest distance from* n *to a leaf.*

Note that $0 \leq \eta(S) \leq 1$. In particular, $\eta(S) = 1$ is verified when n is a leaf, and $\eta(S) = 0$ when n coincides with the root.

Definition 9. *Given $\boldsymbol{X}$ and $\boldsymbol{X}_0$, the information loss of the summarization process is defined as*

$$\mathcal{I}(\boldsymbol{X}, \boldsymbol{X}_0) = \alpha(\mu(\boldsymbol{X}_0) - \mu(\boldsymbol{X})) + \beta(\eta(\boldsymbol{X}_0) - \eta(\boldsymbol{X})). \quad (6)$$

Definition 10. *Given $\boldsymbol{X}_0$ and a real number $\gamma > 0$, we define the Optimal Summarized Time Sequence $\overline{\boldsymbol{X}}$ such that the parameterized ratio between $\mathcal{I}(\boldsymbol{X}, \boldsymbol{X}_0)$ and $\mathcal{C}(\boldsymbol{X}, \boldsymbol{X}_0)$ is minimal, i.e.*

$$\overline{\boldsymbol{X}} = \underset{\boldsymbol{X}}{\operatorname{argmin}} \frac{\mathcal{I}(\boldsymbol{X}, \boldsymbol{X}_0)}{[\mathcal{C}(\boldsymbol{X}, \boldsymbol{X}_0)]^{\gamma}}. \quad (7)$$

4 Conclusions and future works

In this work the problem of summarizing data obtained by the log systems of applications with a lot of users is studied. We have presented a new method to produce a concise summary of sequences of events related to time, based on the data size reduction obtained merging time intervals and collapsing the descriptions of more events in a unique descriptor or in a smaller set of descriptors. Moreover, in order to obtain a data representation as compact as possible, an abstraction operation allowing an event generalization process is defined.

Moreover, we are studying about the formalization and the implementation of an optimal algorithm for the *Optimal Summarization Problem*. The idea is to use suboptimal algorithms to obtain the best summarization algorithm for our method.

References

1. V. Chandola and V. Kumar. *Summarization–compressing data into an informative representation.* Knowledge and Information Systems, 12(3):355–378, 2007.
2. E. Gentili, A. Milani and V.Poggioni. *Data summarization model for user action log files.* In Proceedings of ICCSA 2012, Part III, LNCS 7335, pp. 539–549. Springer, Heidelberg, 2012.
3. Y. Jiang, C.S. Perng, and T. Li. *Natural event summarization.* In Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pages 765–774. ACM, 2011.
4. J. Kiernan and E. Terzi. *Constructing comprehensive summaries of large event sequences.* In Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 417–425. ACM, 2008.
5. Q.K. Pham, G. Raschia, N. Mouaddib, R. Saint-Paul, and B. Benatallah. *Time sequence summarization to scale up chronology-dependent applications.* In Proceeding of the 18^{th} ACM conference on Information and knowledge management, pages 1137–1146, 2009.
6. P. Wang, H. Wang, M. Liu, and W. Wang. *An algorithmic approach to event summarization.* In Proceedings of the 2010 international conference on Management of data, pages 183–194. ACM, 2010.