MultiAspectForensics: mining large heterogeneous networks using tensor

Koji Maruhashi*

Fujitsu Laboratories Ltd.,
Kawasaki, Kanagawa 211-8588, Japan
E-mail: maruhashi.koji@jp.fujitsu.com
*Corresponding author

Fan Guo and Christos Faloutsos

Carnegie Mellon University,
Pittsburgh, PA 15213, USA
E-mail: fanguo@cs.cmu.edu
E-mail: christos@cs.cmu.edu

Abstract: Modern applications such as web knowledge bases, network traffic monitoring and online social networks involve an unprecedented amount of 'heterogeneous' network data, with rich types of interactions among nodes. How can we find patterns and anomalies for heterogeneous networks with millions of edges that have high dimensional attributes, in a scalable way? We introduce *MultiAspectForensics*, a novel tool to automatically detect and visualise bursts of specific sub-graph patterns within a local community of nodes as anomalies in a heterogeneous network, leveraging scalable tensor analysis methods. One such pattern consists of a set of vertices that form a dense bipartite graph, whose edges share exactly the same set of attributes. We present empirical results of the proposed method on three datasets from distinct application domains, and discuss insights derived from these patterns discovered. Moreover, we empirically show that our algorithm can be feasibly applied to higher dimensional datasets.

Keywords: heterogeneous networks; tensor decomposition; pattern mining; large scale data; web knowledge bases.

Reference to this paper should be made as follows: Maruhashi, K., Guo, F. and Faloutsos, C. (2012) '*MultiAspectForensics*: mining large heterogeneous networks using tensor', *Int. J. Web Engineering and Technology*, Vol. 7, No. 4, pp.302–322.

Biographical notes: Koji Maruhashi received his BS and MS in Biological Science from Kyoto University, Japan, in 1997 and 1999. Since April 1999, he has worked at Fujitsu Ltd., and Fujitsu Laboratories Ltd. Since October 2009 to October 2010, he visited the Database group of the Carnegie Mellon University. His research interests include large scale graph mining, time series data mining, and bioinformatics.

Fan Guo is currently a Software Engineer with Facebook. He received his PhD at the Carnegie Mellon University.

Christos Faloutsos is a Professor at Carnegie Mellon University. He is an ACM fellow. He received the ICDM Research Contributions Award, the SIGKDD Innovations Award, 18 'best paper' awards and four teaching awards. His research interests include data mining for graphs and streams, fractals, database performance, and indexing for multimedia and bio-informatics data.

1 Introduction

Modern applications in the internet era, either data-informed or data-driven, have contributed to the boom of network data arising from a spectrum of domains, such as web knowledge bases (Bizer et al., 2009b), network traffic monitoring (Gómez et al., 2009) and online social networks (Boyd and Ellison, 2007). A glowing trend in the accumulation and analysis of such data is the emergence of heterogeneous interactions between nodes in the network, for which a vivid depiction is offered by the Facebook friendship page, with multiple page elements ranging from wall posts, comments, and photos, to mutual friends, shared interests and common networks between a pair of users. In web knowledge bases, the resource description framework (RDF) is a method for expressing knowledge as *triples* in the form of *subject-predicate-object* expressions that represents a heterogeneous interactions between *subject* nodes and *object* nodes with *predicate* edges (Klyne and Carroll, 2004), and OWL is a language based on RDF that describes the semantics of ontology (Bechhofer et al., 2004). The RDF-based knowledge is published as linked data (Bizer et al., 2009a). Browsing and navigation over such a space of information, despite its overwhelming scale and complexity, has been a challenging task commonly encountered in many fields. Yet the rather recent availability and popularity of these data, in addition to practical requirements over the efficiency, robustness and generalisability of the solution, has rendered the topic of pattern mining for heterogeneous network data a relatively under-explored one, where even the definition of interesting or abnormal *patterns* could become a non-trivial problem itself.

Many of pioneering studies on pattern discovery for graph and network data focused on frequent substructure mining, with heuristics motivated by information theory (Cook and Holder, 1994), mathematical graph theory (Yan and Han, 2002; Kuramochi and Karypis, 2004), inductive logic programming (Dehaspe and Toivonen, 1999), etc. An intimately related problem is the detection of rare event and anomalous behaviour, which has attracted wide interests thanks to its many well-recognised applications concerned with security, risk assessment, and fraud analysis. Noble and Cook (2003) were among the first to address this challenge on structured network data by providing solutions based on the minimal description length principle to search for abnormal sub-graphs. And many alternative approaches are now available to spot anomalous nodes (Akoglu et al., 2010), edges (Chakrabarti, 2004), or both (Eberle and Holder, 2007), with further elaboration adapted to bipartite graphs (Sun et al., 2005), and time-evolving graphs (Tong et al., 2008). This piece of work, by revealing two classes of patterns in the context of heterogeneous graphs, resembles a novel attempt to explore this relatively young realm of multi-aspect network data for state-of-the-art discoveries and developments.

A heterogeneous network can be represented as a graph with several statistics that differ according to the types of the relationships between nodes, and these statistics are carefully mixed up to process some data mining tasks such as clustering (Sun et al., 2009)

or classification (Ji et al., 2011). Instead we resort to a tensor-based representation and employ off-the-shelf decomposition algorithms (Kolda and Bader, 2009) as a starting point of the analysis. Previous research along this line has paid a great deal of attention on individual nodes, which play a central role in similarity ranking (Franz et al., 2009), personalised recommendation (Zheng et al., 2010), etc. The major finding in our study is that, for multiple heterogeneous network data across diverse application domains, we could always observe groups of elements with similar connections along one or more data modes, as implied by nearly-identical decomposition scores, which transform to quite visible spikes in histogram plots. While algorithms in aforementioned studies mostly look for elements with top eigenscores, our heuristic distinguishes itself by being able to capture patterns formed by less well-connected nodes in the network, which do not necessarily stand out in the eigenspace and are often ignored by other extant techniques.

In summary, we propose *MultiAspectForensics*, which starts with a data decomposition step for input heterogeneous networks, features a spike detection heuristic to reveal non-trivial substructure patterns. Our method also includes programs to automatically visualise the detected spikes and summarise the sub-graph patterns corresponding to the spikes. We demonstrate its effectiveness and efficiency by executing *MultiAspectForensics* on three datasets from distinct application scenarios, present empirical results and investigate the discovered patterns, which could be leveraged to suggest suspicious activities from network traffic logs such as port-scanning and denial-of-service attack, extract interesting facts from a web knowledge base such as punk musicians or low-cost airline destinations, and report gene function groups in a developmental biology consistent with established theories. Moreover, we empirically show that our algorithm can be feasibly applied to higher dimensional datasets.

This paper is a revised and expanded version of a paper entitled 'MultiAspectForensics: pattern mining on large-scale heterogeneous networks with tensor analysis' presented at ASONAM 2011, Kaohsiung, Taiwan, 25–27 July 2011 (Maruhashi et al., 2011).

The remainder of this paper is organised as follows: we first briefly sketch related literatures in Section 2, and then elaborate on *MultiAspectForensics* procedures step-by-step in Section 3. Experimental studies are covered in Section 4. Lastly, Section 5 concludes the discussion and highlights future directions.

2 Related work

2.1 Anomaly detection

Outlier detection, despite its wide interest across many application domains, is usually a challenging problem, as reflected in the fact that even a formal definition is not easy to make. A classical one was given by Hawkins (1980): "an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism".

Outlier detection methods can be categorised into two sets: parametric, statistical-based approaches, and non-parametric, model-free approaches. A common characteristic of methods in the former category is the existence of statistical assumptions about the underlying data distribution (Barnett and Lewis, 1994). The latter category

usually makes the call by resorting to distance computation (Knorr et al., 2000) or density estimation (Breunig et al., 2000; Jin et al., 2001). Besides, projection-based methods (Aggarwal and Yu, 2001) have been introduced for high-dimensional data. Moreover, clustering algorithms may output outlier labels as a by-product (e.g., Chaoji et al., 2008).

Compared to outlier detection, anomaly detection in structured data has only gained recent attention (Chandola et al., 2009), where we have reviewed relevant studies in the introductory section and claimed that there is no other attempt, to the extent of our knowledge, to discover similar patterns in heterogeneous network data as *MultiAspectForensics*.

2.2 Tensor analysis and graph mining

Tensor decomposition has been a basic technique well studied and applied to a wide range of disciplines and scenarios. An informative survey on tensor decompositions is presented by Kolda and Bader (2009) with many further references. Recent researches have further generalised the CP decomposition to handle incomplete data (Acar et al., 2010), or to produce non-negative components (Shashua and Hazan, 2005). Tucker decomposition, as the other well-known approach, is more flexible, although its application is usually limited by its limited scalability and vulnerability to noise. Notably, recent work on scalable alternatives such as Tsourakakis (2010) may open up the venue to enhance the *MultiAspectForensics* methodology with more powerful decomposition algorithms.

Quite a few popular implementations of tensor decomposition algorithms for academic researchers have been made publicly available. Examples are the *N*-way toolbox by Andersson and Bro (2000) and the more recent MATLAB tensor toolbox by Bader and Kolda (2010). The ALS method was proposed in the original papers by Carroll and Chang (1970) and Harshman (1970) to realise the CP decomposition, and it still remains the primary workhorse algorithm today due to its speed and ease of implementation (Tomasi and Bro, 2006).

Tensor analysis has also been applied to study the dynamics of graphs and networks (Sun et al., 2008). They commonly start by analysing graph/tensor snapshots within each timestamp, and take the output for subsequent time-series analysis. *MultiAspectForensics*, instead of focusing on the evolution between adjacent time-stamps, treats timestamp as another data mode to allow better discovery of global patterns in this trade-off.

3 Algorithm

MultiAspectForensics, in a nutshell, consists of the following steps:

- *data decomposition:* take the input heterogeneous network as a tensor and perform the CP decomposition to obtain an eigenscore vector along each data mode
- *spike detection in histograms:* iterate over all data modes to obtain histograms and apply the spike detection algorithm
- *visualisation:* create *attribute plots* and *histogram plots* with detected spikes highlighted

- *substructure discovery:* identify the induced sub-graph for each spike and summarise patterns discovered.

The above procedure just makes use of the strongest component after data decomposition. If the contribution of the top one eigen-component is not as large, the latter three steps should be carried out over multiple strongest components in a similar fashion.

The running example in this section comes from a snapshot of network traffic log which consists of packet traces in an enterprise network (Lawrence Berkeley National Laboratory and ICSI, n.d.). Each trace in the log is a triplet of (*IP-source*, *IP-destination*, *port-number*), which could be represented as a directed network of machine IP addresses with the only edge attribute 'port number' and number of packets as edge weights.

Table 1 lists definitions of symbols used in this paper.

Table 1 Symbol table

Symbol	*Definition*
$\mathcal{X}$	A tensor ('datacube')
$\mathcal{X}(i_1,\ldots,i_M)$	The *entry* of $\mathcal{X}$ with index $(i_1,\ldots,i_M)$
M	The *order* of the tensor [e.g., $M = 3$ for internet traffic (source-destination-port)]
$X_{(n)}$	The mode-n matricisation of a tensor $\mathcal{X}$
$E_{(n)}$	The elements of mode # n (e.g., the set of IP-sources, for $n = 1$)
I_n	The size of the n^{th} mode (i.e., $I_n = \mid E^{(n)} \mid$)
$e_i^{(n)}$	The i^{th} element of mode # n (e.g., '128.1.5.22')
R	The desired rank (# of components) in the decomposition
$A^{(n)} \in \mathbf{R}^{I_n \times R}$	The matrices for mode # n in the decomposition results of CP decomposition for $\mathcal{X}$
$\vec{a}_i^{(n)}$	The i^{th} row vector of $A(n)$ $(1 \leq i \leq R)$
$a_{ij}^{(n)}$	The j^{th} value, or eigenscore, in $\vec{a}_i^{(n)}$

3.1 Data decomposition

We first introduce a few definitions. A *tensor* can be represented as a multidimensional array of scalars. Its order is the dimensionality of the array, while each dimension is known as one *mode*, of which the value ranges over the set of *elements* for the specific mode. Thus, vectors are tensors of order one, and matrices are tensors with two modes. In Section 4 we will use *measure* to denote the unit of each *entry* in the multi-dimensional array.

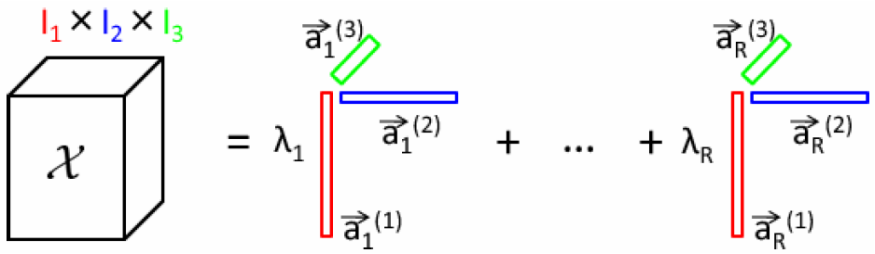
To transform a heterogeneous network into a tensor, every edge becomes a non-zero entry in the multi-dimensional array, where edge attributes, together with edge source and destination, make up different modes of the tensor. Edge weights naturally stay as entry values for weighted networks. Node attributes could also be incorporated by taking a Cartesian product over two end points of an edge, for instance, if a directed network contains nodes with seven different colours, we could have an edge attribute whose arity is $7^2 = 49$.

Tensor decomposition leverages multi-linear algebra to the analysis of high-order data. The canonical polyadic (CP) decomposition we applied in this paper generalises the singular value decomposition (SVD) for matrices. It factorises a tensor to the weighted sum of outer products of mode-specific vectors, as illustrated in Figure 1 for a three-order tensor. Formally, for an M-mode tensor $\mathcal{X}$ of size $I_1 \times I_2 \times \cdots \times I_M$, its CP decomposition of rank R yields

$$\begin{aligned} \mathcal{X}(i_1, \ldots) &\approx \sum_{r=1}^{R} \lambda_r \left(\overrightarrow{a_r^{(1)}} \times \ldots \times \overrightarrow{a_r^{(M)}} \right) \\ &= \sum_{r=1}^{R} \lambda_r \prod_{m=1}^{M} a_{ri_m}^{(m)} \end{aligned} \tag{1}$$

Similar to SVD, the approximation becomes closer as R enlarges, and would be exact if it equals the rank of the tensor [see Håstad (1990) for details]. We used the *cp_als* function in the MATLAB tensor toolbox (Bader and Kolda, 2010) which features the alternating least squares (ALS) method, a predominant implementation for CP decomposition.

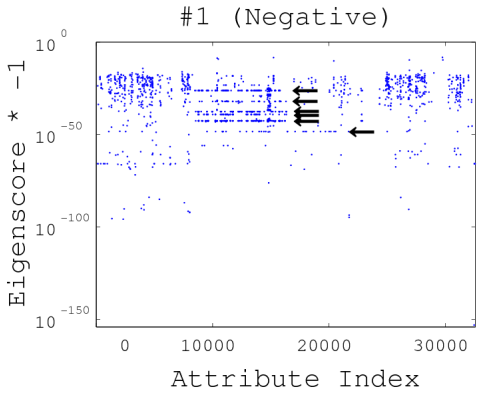
Figure 1 Illustration of the CP decomposition: the input three-mode tensor on the left is decomposed into R triplets of vectors on the right, reminiscing of the rank-R SVD of a matrix (see online version for colours)



3.2 Spike detection in histograms

Now that we have transformed complex structured data into a set of more manageable vectors, the next step is to spot *anomalous* patterns from these vectors. As a starting point, we visualise each vector by creating an *attribute plot*, which displays absolute values of eigenscores (y-axis) along its elements (indexed by the x-axis). An example of such plots is given in Figure 2. Note that the y-axis should be in *log* scale to emphasise on the relative difference. The black arrows indicate score values shared by many elements, which are not uncharacteristic in other dimensions and across different datasets. *This key observation* enables us to create effective heuristics to extract spikes from histograms as anomalies, and subsequently examine sub-graph patterns they imply in the next subsection. And the fact that many spikes do not appear at the very top of the figure with most significant eigenscore values makes it more difficult for many alternative methods to be effective.

Figure 2 An *attribute plot* which displays absolute values of eigenscores (y-axis in log-scale) along its elements (indexed by the x-axis) for the 'IP-source' mode with negative eigenscores for Lawrence Berkeley National Lab (LBNL) network traffic dataset (see online version for colours)



Notes: Elements are sorted such that IP-sources located in the same local network have similar attribute index. The black arrows point to common score values, illustrating an observation critical to the algorithmic design of *MultiAspectForensics*.

Algorithm 1 Spike detection algorithm (SDA)

Require: Eigenscore histogram vector H_o of size N

Ensure: The set indicating spikes detected S

sort the histogram in descending order s.t. $H_{o_1} \geq H_{o_2} \geq \cdots \geq H_{o_N}$

$S \leftarrow \phi$; $Q \leftarrow 0$; $Q_{SUM} \leftarrow \sum_{n=1}^{N} H_{o_n}^2$

for $k = 1,\ldots,K$ **do**

  $S \leftarrow S \cup \{o_k\}$

  $Q \leftarrow Q + H_{o_k}^2$

  if $Q / Q_{SUM} \geq s$ and $H_{o_k} / H_{o_1} < r$ **then**

    break

  end if

end for

if $Q / Q_{SUM} < s$ **then**

  $S \leftarrow \phi$

end if

return S

Prior to applying the spike detection heuristics, we obtain histogram data by equally dividing the range of eigenscores in log scale. The detection algorithm just needs to sort and traverse the histogram data until one of the following conditions is satisfied:

1 the energy as measured by sum of square values of frequencies covered is equal or more than a fraction of s, and the frequency is less than a fraction of r of the largest one

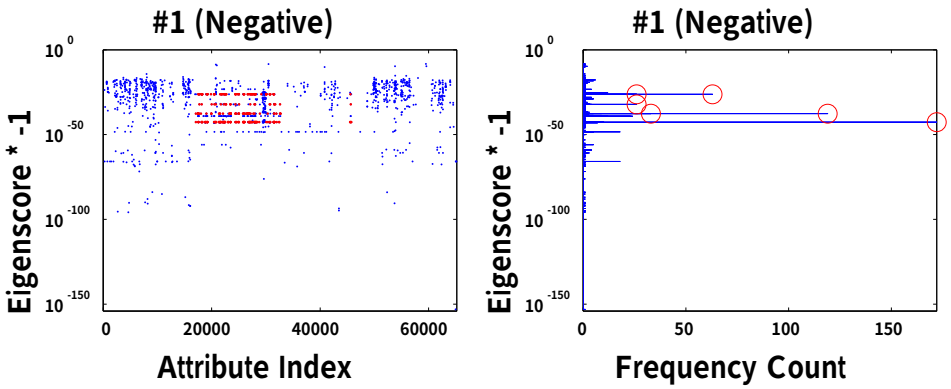
2 K bins are already inspected.

After the inspection, the sets of elements within each bin are extracted as spikes, as long as the energy covered is equal or more than a fraction of s. K is the desired number of spikes to be detected, and we used $K = 20$ as the number of spikes we can investigate practically. We used $r = 50\%$, because the bins with equal or more than half of the frequencies of the largest one should be extracted as spikes, as long as the condition of the energy is satisfied. We examined several number of s under $K = 20$ and $r = 50\%$, and we chose $s = 90\%$ so that many spikes are extracted for most of the datasets we used. The pseudo-code of the algorithm is listed in Algorithm 1 above.

3.3 Visualisation

Application of this algorithm to the data vector in Figure 2 yields Figure 3, where we put *attribute plot* on the left side-by-side with *histogram plot* on the right, high-lighting every spike in red. Using these plots, we can investigate the distributions of the attribute index of the elements within each spike. Figure 3 shows the IP-sources within the detected spikes have attribute index in a specific range. Because the elements are sorted such that IP-sources located in the same local network have similar attribute index, the *attribute plot* indicates that these IP-sources are located in the same local network.

Figure 3 An *attribute plot* (adopted from Figure 2) on the left side-by-side with the corresponding histogram plot showing the count of elements that have same eigenscores (indicated by x-axis) in the 'IP-source' mode for LBNL network traffic dataset (see online version for colours)

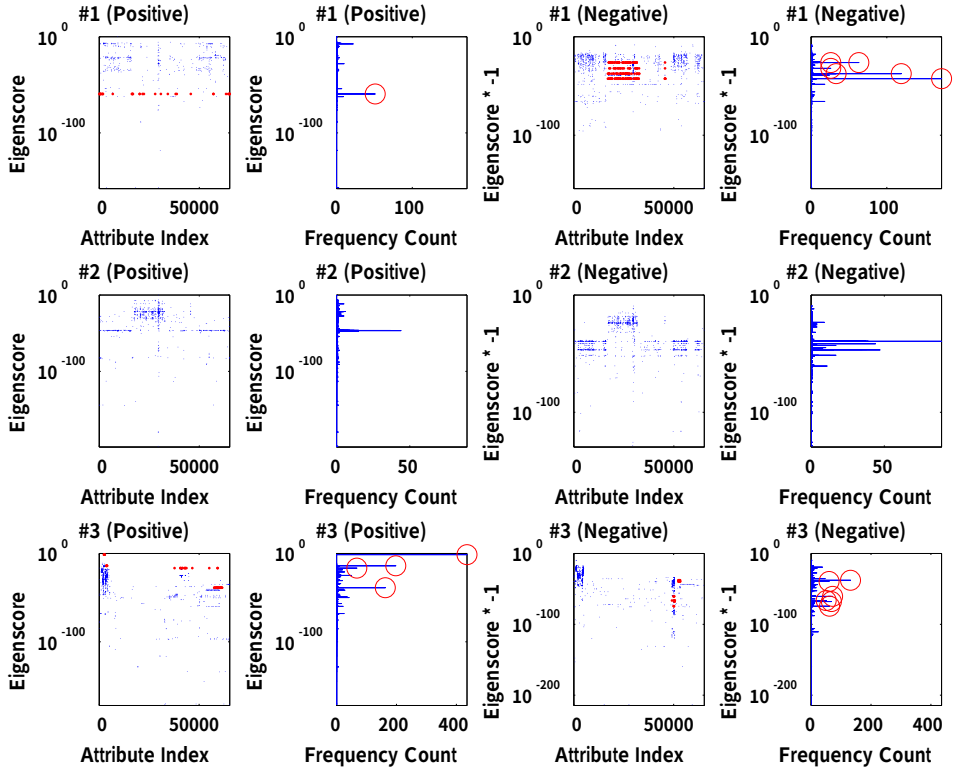


Notes: Detected spikes are indicated by red circles in *histogram plot* and red dots in *attribute plot*. We can find the elements within each spike have attribute index in a specific range, indicating that they are located in the same local network.

The collection of output plots of *MultiAspectForensics* is named *MultiAspectForensics* X-ray (*MAF-ray*), which puts together *attribute plots* and *histogram plots* of both positive and negative eigenscores. The rank 1 *MAF-ray* of LBNL network traffic dataset is shown in Figure 4. With *MAF-ray*, users can easily realise that there have occurred spikes in

histogram plot of some modes, whose corresponding elements have specific attribute index in *attribute plot*, like aforementioned spikes of IP-sources.

Figure 4 First MAF-ray for LBNL network traffic dataset: attribute plots and histogram plots, for 'IP-source' mode (the top row, #1) and 'IP-destination' mode (the middle row, #2) and 'port#' mode (the bottom row, #3), and for the positive (left) and negative (right) parts (see online version for colours)



Note: Using MAF-ray, we can quickly spot IP-sources (or IP-destinations, or port#), which have similar behaviour, forming one of the patterns in Section 3.4.

3.4 Substructure discovery

Having extracted *anomalous* sets of elements that form histogram spikes from each data mode, we head back to the input network data to examine corresponding local sub-networks to complete the final step of pattern discovery. Because the elements within the same spike are expected to behave similarly and specifically, the local patterns corresponding to detected spikes can be understood as bursts of common patterns shared by the elements. As the starting point for analysing the local patterns, we propose *spike table* that shows detected spikes along with frequency counts of elements within each spike, numbers of common patterns, and numbers of unique elements of other modes within the common patterns. A *spike table* in 'IP-source' mode (#1) in our running example is shown in Table 2. For example, most of the

119 IP-sources within spike #2 have three common patterns (131.243.143.66/534, 131.243.141.187/534, and 131.243.140.105/534 as IP-destination/port number), with three distinct IP-destinations (mode #2), and one distinct port number (mode #3). Patterns derived from *MultiAspectForensics* could be summarised into the following two categories:

3.4.1 Generalised star (g-Star)

This pattern can be detected as a spike in a mode, whose corresponding elements share a single common pattern of other modes. This is a sub-network which consists of conterminous edges that differ only in one data mode. It generalises the star pattern in two dimensional graphs, and makes up a continuous block along one dimension in the adjacency tensor, if elements along that dimension are ordered carefully. We can find three spikes of this category in the *spike table* shown in Table 2. They are groups of IP-sources sending packets to a single destination server using the same port [Figure 5(a) 'NCP']. Note that in a heterogeneous network, this category of patterns also includes multiple edges between one pair of nodes with differing attribute values, e.g., a good many port numbers in our running example, in which case the source machine may be either an administrator performing port screening or a suspect trying to exploit a vulnerable port [Figure 5(b) 'port scanning'].

3.4.2 Generalised bipartite-core (g-Bcore)

This pattern can be detected as a spike in a mode, whose corresponding elements share multiple common patterns of other modes. This is a sub-network that represents a dense bipartite structure similar to the bipartite-core pattern in regular graphs. More generally, it can be viewed as a *fibre* of continuous blocks along one dimension in higher-order tensors under specific element orders. Patterns of this category can be classified into some classes according to the number of unique elements of each mode within the common patterns. In the *spike table* shown in Table 2, spikes of this category are classified into two classes, spikes whose common patterns contain multiple unique elements only in mode #2 (spike #2 and #5), and spikes only in mode #3 (spike #3 and #7). The former class is a group of IP-sources sending packets to multiple destination servers with the same port [Figure 6(a) 'file sharing']. And the latter class is a group of IP-sources sending packets over different port numbers to the same server [Figure 6(b) 'multi purpose'], likely to happen during a distributed denial-of-service (DDoS) attack, a typical scenario of network intrusion, in which IP-sources play the role of malicious hosts sending huge volumes of packets to the target server as the victim. Note that this category of patterns can also include classes of spikes whose common patterns contain multiple unique elements in several modes, like spikes detected in the network traffic dataset with additional mode of time tick, as shown in Section 4.

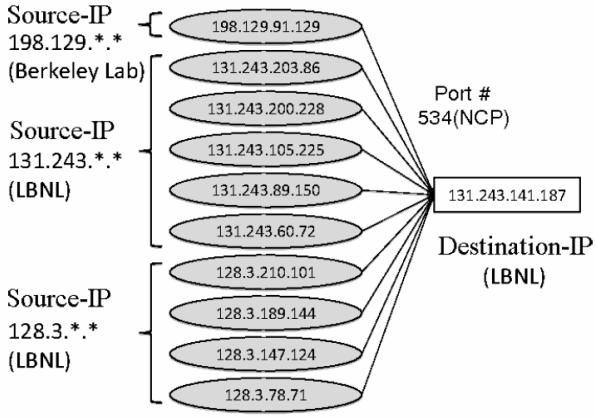
As a final remark, the statement that both patterns are belated to a block along one dimensions or a bundle of blocks in the high-order tensor only holds when elements of their respective data modes are ordered in specific ways. And the complexity to search for such an order is generally exponential, which reflects, in some sense, the power of the proposed approach.

Table 2 A spike table in 'IP-source' mode (#1) for the LBNL network traffic dataset

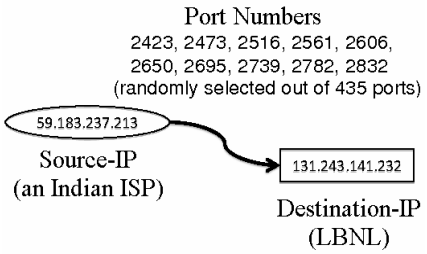
#	*Frequency count*	*# common patterns*	*# unique elements of mode #2;3*	*Description*
1	172	1	1; 1	*g-Star* ('NCP')
2	119	3	3; 1	*g-Bcore* ('file sharing')
3	63	5	1; 5	*g-Bcore* ('multi purpose')
4	51	1	1; 1	*g-Star*
5	33	4	4; 1	*g-Bcore*
6	26	1	1; 1	*g-Star*
7	26	2	1; 2	*g-Bcore*

Notes: Number of common patterns are the number of patterns of other modes shared by more than 90% of the elements within each spike. Numbers of unique elements of other modes within the common patterns are also shown.

Figure 5 Examples of generalised star patterns discovered in the LBNL network traffic dataset, (a) 'NCP': ten IP-sources (randomly selected out of 172 ones) are sending multiple packets to a server machine with Port# 524, which is a UDP port under the NCP protocol from a network OS for file sharing and printing services (b) 'Port scanning': the IP-source registered by an Indian ISP is sending packets to a host in LBNL via port numbers (ranging from 2,300 to 2,900) not usually intended for this type of communication, implying a suspicious activity



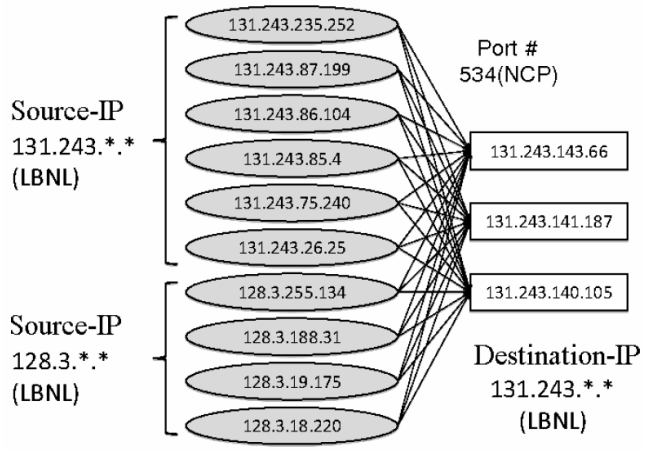
(a)



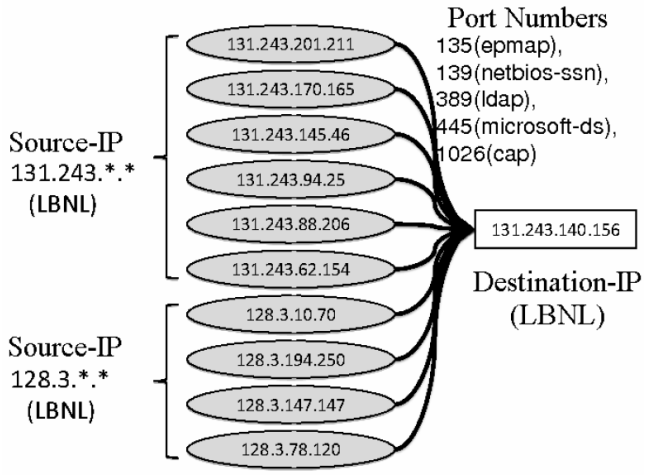
(b)

Note: Wavy arrows indicate multiple edges between the pair of nodes with a handful of distinct attribute values.

Figure 6 Examples of generalised bipartite-core patterns discovered in the LBNL network traffic dataset, (a) 'File sharing': ten IP-sources (randomly selected out of 119 ones) are sending multiple packets to an array of server machines over a port used for file sharing and printing services (b) 'Multi purpose': ten IP-sources (randomly selected out of 63 ones) are sending packets over different ports to a multi-purpose server machine



(a)



(b)

Note: Wavy arrows indicate multiple edges between the pair of nodes with a handful of distinct attribute values.

4 Empirical results

We commence this section with the description of datasets as well as experimental environment. It is followed by the discussion of respective patterns discovered by *MultiAspectForensics* in each of the three datasets.

4.1 Data and environment

Datasets are acquired from three dissimilar application domains: network traffic monitoring, knowledge networks, and bioinformatics. A summary is highlighted in Table 3.

Table 3 A summary of datasets

Dataset	*# modes*	*Measure*	*# non-zero*	*Dimensions*	*# spikes*
LBNL-sdp	3	# packets	27 K	2,345 IP-srcs	7
				2,355 IP-dsts	0
				6,055 port #'s	10
LBNL-sdpt	4	# packets	231K	3,610 time ticks	2
				2,345 IP-srcs	0
				2,355 IP-dsts	0
				6,055 port #'s	0
RTW	3	binary	10 K	3,641 subjects	15
				98 verbs	0
				3,929 objects	2
BDGP	3	binary	38 K	4,491 genes	5
				248 terms	2
				6 stages	0

Note: The numbers of spikes extracted by *MultiAspectForensics* are shown.

- *LBNL:* The network traffic log is made available through a research effort to study the characteristics of traffic for internet enterprises (Pang et al., 2005). The measurement was taken on servers within the LBNL from thousands of internal hosts over time, with millions of packet traces recorded. Each packet trace includes four data modes: IP-source, IP-destination, port number, and a time tick in second. With privacy in concern, lower 16 bits were randomly permuted to anonymise the host identity, whereas upper 16 bits were kept intact for proper identification of the location and service provider (Pang et al., 2006). We borrowed a subset of this dataset within one-hour time span in this section.
- *RTW:* This online knowledge base is the outcome of the Never-Ending Language Learning (NELL) system at Carnegie Mellon University (Carlson et al., 2010a). It employs natural language processing and machine learning techniques to constantly and automatically crawl web pages and extract facts (Carlson et al., 2010b). Each fact is a triplet of (subject, verb, object) such as (*Pittsburgh*, *city-located-in-state*, *Pennsylvania*), which could be represented as a directed graph made up of entities like *Pittsburgh* or *Pennsylvania*, edges with attributes like *city-located-in-state*. For better quality of results, we applied our algorithm on a pre-processed subset after noise removal (by courtesy of Dr. Byran Kisiel at Carnegie Mellon University).
- *Berkeley Drosophila Genome Project (BDGP):* The dataset is collected from the BDGP to study the spatial-temporal patterns of gene expression during the early

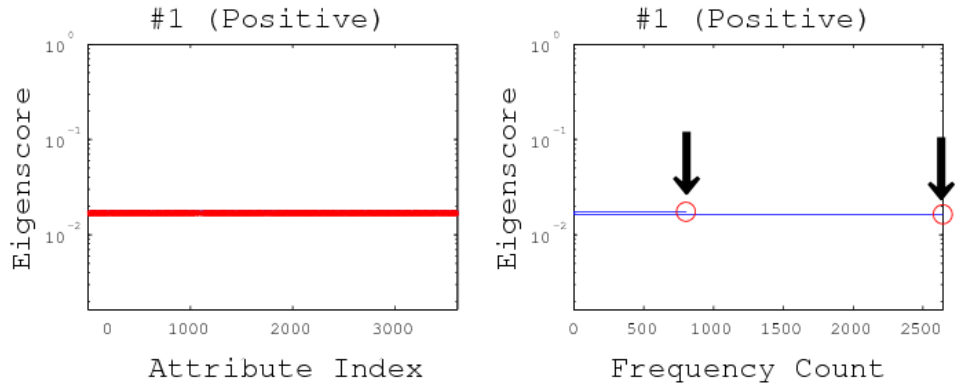
development of fruit fly (Tomancak et al., 2002, 2007). We selected three data modes from the database dump available at Berkeley Drosophilia Genome Project (n.d.), *Patterns of Gene Expression in Drosophila Embryogenesis*, which consists of 4,491 genes, 248 functional annotation terms from a specialised vocabulary, and 6 different developmental stages.

MultiAspectForensics was implemented in the MATLAB language, and all following experiments were performed on a Unix machine with four 2.8 GHz cores, and 16 GB memories. For every of these datasets, the wall-clock time was no more than 2 minutes to carry out the computation and generate *attribute plot* and *histogram plot* along all modes.

4.2 LBNL traffic log

We have already discussed patterns discovered from a snapshot of this dataset in Section 3.4, illustrated in Figures 5 and 6. With the additional mode of time tick (LBNL-sdpt), we found two dominating spikes for the 'time-ticks' mode (Table 3). The elements within these two big spikes in *histogram plot* (arrows on Figure 7) distributes on almost all the attributes in *attribute plot*, indicating the traffic corresponding to these spikes occurred at almost every time-tick. A *spike table* indicates both of the spikes are *g-Bcore* pattern, bipartite-cores between 'time-tick' elements and patterns of other modes (Table 4). Upon closer examination, we reported the following activities: the first spike is related to the HTTP traffic on port 80 between four servers in LBNL and three remote hosts in Chinese academic institutions, possibly executing scripts to crawl/download web pages. The second spike seems to be related to the same HTTP traffic as the first spike, with additional traffic between a server in LBNL and a remote host at India aforementioned. We traced further in time and found that the remote host never sent packets back to acknowledge the connection, suggestive of suspicious activities to be reported to domain experts.

Figure 7 An *attribute plot* on the left side-by-side with the corresponding *histogram* plot for the 'time' mode (#1) from the first *MAF-ray* for LBNL-sdpt (see online version for colours)



Note: The spikes indicated by black arrow are discussed in Section 4.2.

Table 4 A spike table in 'time ticks' mode (#1) for LBNL-sdpt

#	*Frequency count*	*# common patterns*	*# unique elements of mode #2;3;4*	*Description*
1	2,641	10	9; 9; 7	*g-Bcore*
2	803	11	9; 10; 7	*g-Bcore*

Notes: Number of common patterns are the number of patterns of other modes shared by more than 90% of the elements within each spike. Numbers of unique elements of other modes within the common patterns are also shown.

4.3 RTW knowledge base

Recall that each item in the knowledge database could be represented as a (subject, verb, object) triplet. *MultiAspectForensics* detected 15 spikes in 'subjects' mode and two spikes in 'objects' mode (Table 3). A *spike table* shows the spikes detected in 'subjects' mode in Table 5. Almost all of the spikes are g-Star pattern with exception of a spike of *g-Bcore* pattern having only two common patterns.

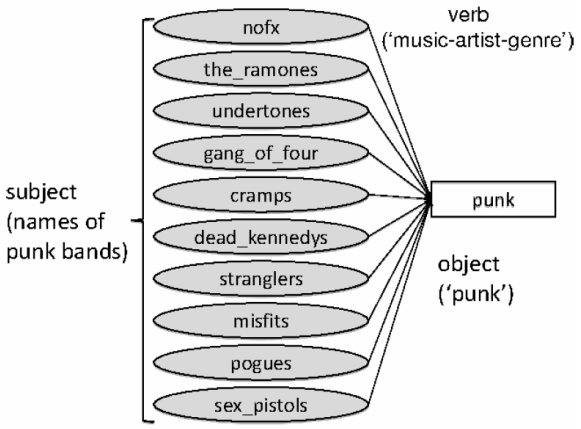
Table 5 A spike table in 'subjects' mode (#1) for RTW knowledge base

#	*Frequency count*	*# common patterns*	*# unique elements of mode #2;3*	*Description*
1	265	1	1; 1	*g-Star*
2	134	1	1; 1	*g-Star*
3	63	1	1; 1	*g-Star*
4	31	1	1; 1	*g-Star*
5	30	1	1; 1	*g-Star*
6	30	1	1; 1	*g-Star* ('punk')
7	29	1	1; 1	*g-Star*
8	27	1	1; 1	*g-Star* ('ryanair')
9	27	1	1; 1	*g-Star*
10	26	1	1; 1	*g-Star*
11	25	1	1; 1	*g-Star*
12	24	1	1; 1	*g-Star*
13	24	1	1; 1	*g-Star*
14	22	2	2; 2	*g-Bcore*
15	22	1	1; 1	*g-Star*

Notes: Number of common patterns are the number of patterns of other modes shared by more than 90% of the elements within each spike. Numbers of unique elements of other modes within the common patterns are also shown.

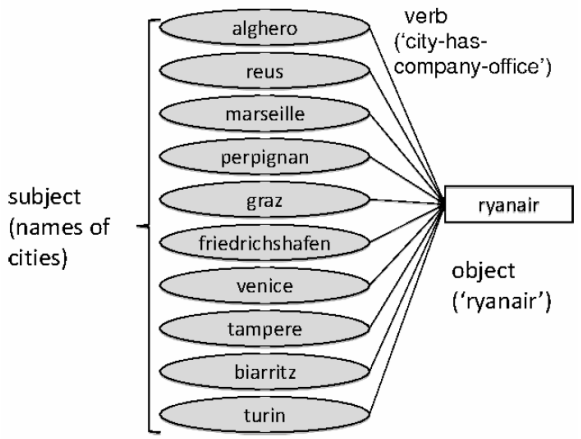
Figure 8 illustrates a sub-graph discovered revealing a *g-Star* pattern ('punk'). The music artists/bands listed here are specialised to punk music according to the knowledge base. And Figure 9 displays another *g-Star* pattern ('ryanair') between European cities and an Irish low-cost airline which flies to many regional or secondary airports to reduce cost, following a different business model and choice of destination from industrial giants.

Figure 8 'Punk': a *g-Star* pattern discovered from the RTW knowledge base about 49 punk music artists, of which a random selected set of ten are listed



Note: They are all specialised in punk or one of its sub-genres according to the knowledge base.

Figure 9 'Ryanair': a *g-Star* pattern discovered from the RTW knowledge base about 36 European destinations of the Ryanair, an Irish low-cost airline, of which a random selected set of ten are listed



Note: Many of these cities have only sparse connections with other verbs.

We should note that some elements within each spike have more versatile peers. For example, some of the musicians of 'subjects' mode in the 'punk' spike have patterns of verb 'music-artist-genre' with objects 'horror punk', 'proto punk', 'British punk' and 'punk rock' (not shown in the figure). In 'ryanair' spike, some of the cities of 'subjects' mode have patterns of verb 'city-located-in-country' with objects 'Finland', 'Norway', 'Austria', 'Scotland', 'Spain', 'Belgium' and 'Ireland' (not shown in the figure). These patterns are not be favourably selected by *MultiAspectForensics*, because they are less important in the first rank.

Moreover, as a sanity check, since node names are ordered alphabetically in this dataset, the pattern does not make a continuous block in the tensor without non-trivial permutation.

4.4 BDGP gene annotation

In this dataset *MultiAspectForensics* spots a spike of a set of genes known to be responsible for the *maternal effect* in the early development of fruit fly, which also provides hints to study other higher organisms including *Homo sapiens*. Products of such maternal effect genes, in the form of either protein or mRNA, play a critical role in the very early stage of embryo development, such as the first few cell divisions. For instance, four of such genes, including *bicoid*, *caudal*, *hunchback*, and *nanos*, is mostly responsible for the determination of anterior-posterior axis – which side of the embryo will be the future head and which other side will be the future tail (Lawrence, 1992).

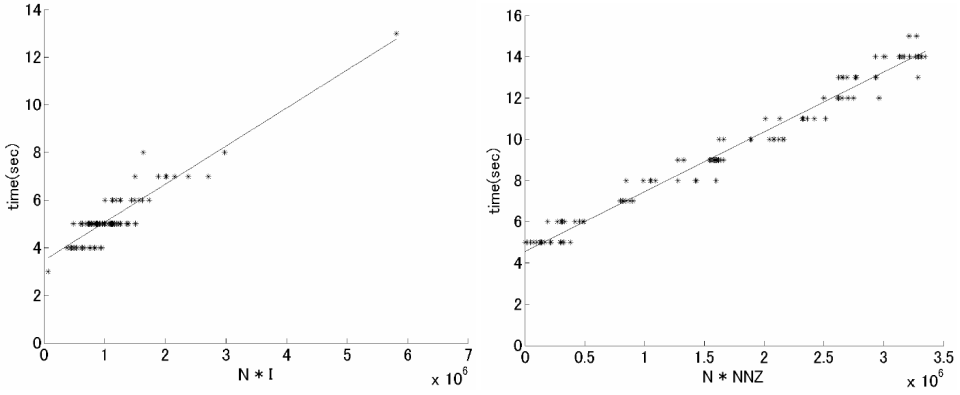
4.5 Scalability

Can *MultiAspectForensics* be feasibly applied to higher dimensional datasets? The most time-consuming part in *MultiAspectForensics* is CP decomposition for the input tensor, for which a predominant implementation is the ALS method, such as the *cp_als* function in the MATLAB tensor toolbox (Bader and Kolda, 2010). Given a mode-N sparse tensor of size $I_1 \times I_2 \times \cdots \times I_N$, the computational cost of each iteration depends on the larger of the following:

1 $I = \sum_{n=1}^{N} I_n$

2 NNZ = the number of non-zero elements.

Figure 10 Computational time of *MultiAspectForensics* on two different datasets, (a) $NNZ << I$ (b) $NNZ >> I$



(a) (b)

Notes: On the x-axis, N denotes the number of iterations, I equals the sum of sizes over all dimensions, and NNZ stands for the number of non-zero elements. The complexity of *MultiAspectForensics* in time is $O(N * \max(I, NNZ))$.

Figure 10 presents empirical results over two different datasets – (a) a traffic log from LBNL data for which the tensor is sparse ($NNZ << I$), and we added non-zero elements to (a) in order to create a more dense tensor in (b) for which $NNZ >> I$. These results support the complexity of *MultiAspectForensics* in time is $O(N * \max(I, NNZ))$. If the dimension is low and the tensor data stays in the category (b), the computational cost does not depend on the addition of the dimensions, and scales linearly as *NNZ* grows. When the dimension increases, the tensor data becomes sparser and will fall into the category (a). In such a case, even though the total volume of the high-dimensional space increases exponentially, the computational cost will increase linearly as the dimension grows, proportional with the sum of the number of attribute index. In each case *MultiAspectForensics* has a feasible time complexity for higher dimensional datasets.

5 Conclusions

We presented *MultiAspectForensics*, a novel and effective tool to automatically detect and visualise a category of anomalous patterns, including generalised star and generalised bipartite-core patterns. These patterns can be understood as bursts of specific sub-graph patterns within a local community of nodes in heterogeneous networks, even if they exist among less-well connected nodes which are more likely to be ignored by many extant methods. Empirical results exhibited valuable insights derived from pattern discovered, across multiple application domains such as network traffic monitoring, knowledge networks, and bioinformatics. These successes could be attributed to the fact that we resorted to a tensor-based representation to facilitate data decomposition, reached a key observation leading to spike patterns in histogram plots, and revealed typical substructures reflecting spectral properties of heterogeneous data. Moreover, *MultiAspectForensics* is scalable to higher dimensional datasets, as we have empirically shown.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants No. DBI-0640543 IIS-0705359 IIS0970179.

Research was sponsored by the Defense Threat Reduction Agency and was accomplished under contract No. HDTRA1-10-1-0120.

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the US Government. The US Government is authorised to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, or other funding parties.

References

Acar, E., Dunlavy, D.M., Kolda, T.G. and Mørup, M. (2010) 'Scalable tensor factorizations with missing data', *Proceedings of the 2010 SIAM International Conference on Data Mining (SDM'10)*, pp.701–712.

Aggarwal, C.C. and Yu, P.S. (2001) 'Outlier detection for high dimensional data', *SIG-MOD Record*, Vol. 30, No. 2, pp.37–46.

Akoglu, L., McGlohon, M. and Faloutsos, C. (2010) 'OddBall: spotting anomalies in weighted graphs', *Proceedings of the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'10)*, pp.410–421.

Andersson, C.A. and Bro, R. (2000) 'The N-way toolbox for MATLAB', *Chemometrics and Intelligent Laboratory Systems*, Vol. 52, No. 1, pp.1–4.

Bader, B.W. and Kolda, T.G. (2010) 'MATLAB Tensor Toolbox Version 2.4', available at http://csmr.ca.sandia.gov/tgkolda/TensorToolbox/ (accessed on 24 June 2010).

Barnett, V. and Lewis, T. (1994) *Outliers in Statistical Data*, 3rd ed., John Wiley and Sons, Chichester.

Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., PatelSchneider, P.F. and Stein, L.A. (2004) 'OWL web ontology language reference', Technical report, W3C, available at http://www.w3.org/TR/owl-ref/ (accessed on 20 March 2012).

Berkeley Drosophilia Genome Project (n.d.) *Patterns of Gene Expression in Drosophila Embryogenesis*, available at http://insitu.fruitfly.org/cgi-bin/ex/insitu.pl (accessed on 24 June 2010).

Bizer, C., Heath, T. and Berners-Lee, T. (2009a) 'Linked data – the story so far', *International Journal on Semantic Web and Information Systems (IJSWIS)*, Vol. 5, No. 3, pp.1–22.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R. and Hellmann, S. (2009b) 'Dbpedia – a crystallization point for the web of data', *Journal of Web Semantics*, Vol. 7, No. 3, pp.154–165.

Boyd, D.M. and Ellison, N. (2007) 'Social network sites: definition, history, and scholarship', *Journal of Computer-Mediated Communication*, Vol. 13, No. 1, pp.210–230.

Breunig, M.M., Kriegel, H-P., Ng, R.T. and Sander, J. (2000) 'LOF: identifying density-based local outliers', *SIGMOD Record*, Vol. 29, No. 2, pp.93–104.

Carlson, A., Betteridge, J., Kisiel, B., Settles, B.E.R.H., Jr. and Mitchell, T.M. (2010a) 'Toward an architecture for never-ending language learning', *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010)*, pp.1306–1313.

Carlson, A., Betteridge, J., Wang, R.C., Hruschka, E.R., Jr. and Mitchell, T.M. (2010b) 'Coupled semi-supervised learning for information extraction', *Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM'10)*, pp.101–110.

Carroll, J. and Chang, J-J. (1970) 'Analysis of individual differences in multidimensional scaling via an n-way generalization of 'eckart-young' decomposition', *Psychometrika*, Vol. 35, No. 3, pp.283–319.

Chakrabarti, D. (2004) 'AutoPart: parameter-free graph partitioning and outlier detection', *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'04)*, pp.112–124.

Chandola, V., Banerjee, A. and Kumar, V. (2009) 'Anomaly detection: a survey', *ACM Computing Surveys*, Vol. 41, pp.15:1–15:58.

Chaoji, V., Hasan, M. A., Salem, S. and Zaki, M.J. (2008) 'SPARCL: efficient and effective shape-based clustering', *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM'08)*, pp.93–102.

Cook, D.J. and Holder, L.B. (1994) 'Substructure discovery using minimum description length and background knowledge', *Journal of Artificial Intelligence Research*, Vol. 1, No. 1, pp.231–255.

Dehaspe, L. and Toivonen, H. (1999) 'Discovery of frequent datalog patterns', *Data Mining and Knowledge Discovery*, Vol. 3, No. 1, pp.7–36.

Eberle, W. and Holder, L. (2007) 'Discovering structural anomalies in graph-based data', *Proceedings of the Seventh International Conference on Data Mining Workshops (ICDMW'07)*, pp.393–398.

Franz, T., Schultz, A., Sizov, S. and Staab, S. (2009) 'TripleRank: ranking semantic web data by tensor decomposition', *Proceedings of the 8th International Semantic Web Conference (ISWC'09)*, pp.213–228.

Gómez, J., Gil, C., Padilla, N., Baños, R. and Jiménez, C. (2009) 'Design of a snort-based hybrid intrusion detection system', *Proceedings of the 10th International Work-Conference on Artificial Neural Networks: Part II: Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living (IWANN '09)*, pp.515–522.

Harshman, R. (1970) 'Foundations of the parafac procedure: Models and conditions for an 'explanatory' multi-modal factor analysis', UCLA Working Papers in Phonetics 16.

Håstad, J. (1990) 'Tensor rank is NP-complete', *Journal of Algorithms*, Vol. 11, pp.644–654.

Hawkins, D. (1980) *Identification of Outliers (Monographs on Statistics & Applied Probability)*, Chapman and Hall, London, New York.

Ji, M., Han, J. and Danilevsky, M. (2011) 'Ranking-based classification of heterogeneous information networks', *Proceeding of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11)*, pp.1298–1306.

Jin, W., Tung, A.K.H. and Han, J. (2001) 'Mining top-n local outliers in large databases', *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)*, pp.293–298.

Klyne, G. and Carroll, J.J. (2004) 'Resource description framework (RDF): concepts and abstract syntax', available at http://www.w3.org/TR/2004/REC-rdf-concepts-20040210 (accessed on 20 March 2012).

Knorr, E.M., Ng, R.T. and Tucakov, V. (2000) 'Distance-based outliers: algorithms and applications', *The VLDB Journal*, Vol. 8, Nos. 3–4, pp.237–253.

Kolda, T.G. and Bader, B.W. (2009) 'Tensor decompositions and applications', *SIAM Review*, Vol. 51, No. 3, pp.455–500.

Kuramochi, M. and Karypis, G. (2004) 'An efficient algorithm for discovering frequent subgraphs', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, No. 9, pp.1038–1051.

Lawrence Berkeley National Laboratory and ICSI (n.d.) 'LBNL/ICSI enterprise tracing project', available at http://www.icir.org/enterprise-tracing/ (accessed on 24 June 2010).

Lawrence, P.A. (1992) *The Making of a Fly: The Genetics of Animal Design*, Wiley-Blackwell, Chichester.

Maruhashi, K., Guo, F. and Faloutsos, C. (2011) 'Multiaspectforensics: pattern mining on large-scale heterogeneous networks with tensor analysis', *Proceedings of 2011 International Conference on Advances in Social Network Analysis and Mining (ASONAM 2011)*, pp.203–210.

Noble, C.C. and Cook, D.J. (2003) 'Graph-based anomaly detection', *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*, pp.631–636.

Pang, R., Allman, M., Bennett, M., Lee, J., Paxson, V. and Tierney, B. (2005) 'A first look at modern enterprise traffic', *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement (IMC'05)*, pp.2–2.

Pang, R., Allman, M., Paxson, V. and Lee, J. (2006) 'The devil and packet trace anonymization', *SIGCOMM Computer Communication Review*, Vol. 36, No. 1, pp.29–38.

Shashua, A. and Hazan, T. (2005) 'Non-negative tensor factorization with applications to statistics and computer vision', *Proceedings of the 22nd International Conference on Machine Learning (ICML'05)*, pp.792–799.

Sun, J., Qu, H., Chakrabarti, D. and Faloutsos, C. (2005) 'Neighborhood formation and anomaly detection in bipartite graphs', *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05)*, pp.418–425.

Sun, J., Tao, D., Papadimitriou, S., Yu, P.S. and Faloutsos, C. (2008) 'Incremental tensor analysis: theory and applications', *ACM Transactions on Knowledge Discovery from Data*, Vol. 2, pp.11:1–11:37.

Sun, Y., Yu, Y. and Han, J. (2009) 'Ranking-based clustering of heterogeneous information networks with star network schema', *Proceeding of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*, pp.797–806.

Tomancak, P., Beaton, A., Weiszmann, R., Kwan, E., Shu, S., Lewis, S., Richards, S., Ashburner, M., Hartenstein, V., Celniker, S. and Rubin, G. (2002) 'Systematic determination of patterns of gene expression during drosophila embryogenesis', *Genome Biology*, Vol. 3, No. 12, research0088.1{0088.14.

Tomancak, P., Berman, B., Beaton, A., Weiszmann, R., Kwan, E., Hartenstein, V., Celniker, S. and Rubin, G. (2007) 'Global analysis of patterns of gene expression during drosophila embryogenesis', *Genome Biology*, Vol. 8, No. 7, p.R145.

Tomasi, G. and Bro, R. (2006) 'A comparison of algorithms for fitting the parafac model', *Computational Statistics & Data Analysis*, Vol. 50, No. 7, pp.1700–1734.

Tong, H., Papadimitriou, S., Sun, J., Yu, P.S. and Faloutsos, C. (2008) 'Colibri: fast mining of large static and dynamic graphs', *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08)*, pp.686–694.

Tsourakakis, C.E. (2010) 'MACH: fast randomized tensor decompositions', *Proceedings of the 2010 SIAM International Conference on Data Mining (SDM'10)*, pp.689–700.

Yan, X. and Han, J. (2002) 'gSpan: graph-based substructure pattern mining', *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*, p.721.

Zheng, N., Li, Q., Liao, S. and Zhang, L. (2010) 'Flickr group recommendation based on tensor decomposition', *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'10)*, pp.737–738.