

Multi-Type relational clustering for enterprise cyber-security networks

Elizabeth Riddle-Workman^{a,**}, Marina Evangelou^a, Niall M. Adams^{a,b}

^aDepartment of Mathematics, Imperial College London, London, SW7 2AZ, UK

^bData Science Institute, Imperial College London, London, SW7 2AZ, UK

ABSTRACT

Several cyber-security data sources are collected in enterprise networks providing relational information between different types of nodes in the network, namely computers, users and ports. This relational data can be expressed as adjacency matrices detailing inter-type relationships corresponding to relations between nodes of different types and intra-type relationships showing relationships between nodes of the same type. In this paper, we propose an extension of Non-Negative Matrix Tri-Factorisation (NMTF) to simultaneously cluster nodes based on their intra and inter-type relationships. Existing NMTF based clustering methods suffer from long computational times due to large matrix multiplications. In our approach, we enforce stricter cluster indicator constraints on the factor matrices to circumvent these issues. Additionally, to make our proposed approach less susceptible to variation in results due to random initialisation, we propose a novel initialisation procedure based on Non-Negative Double Singular Value Decomposition for multi-type relational clustering. Finally, a new performance measure suitable for assessing clustering performance on multi-type relational data sets is presented for evaluating performance on unlabelled data. Our algorithm is assessed on both a simulated and real computer network against standard approaches showing its strong performance.

1. Introduction

Clustering, a type of unsupervised learning, aims to partition data from a single data type into groups. Many data sets such as document-word matrices benefit from clustering both the rows and columns, known as bi-clustering. However, in many cases multiple data sets may exist referring to the same data object where the aim is to form a clustering that utilises information from multiple data sets, incorporating the interrelationships between the rows and the columns of different matrices. Kumar et al. (2011) discussed that jointly clustering over multiple networks can improve clustering accuracy over single network clustering methods. We are concerned with relational type data where inter-type relationships correspond to the relations between data objects of different types and intra-type relationships refer to relations between data objects of the same type. This clustering problem is referred to as *multi-type relational clustering*.

Cyber-security data lends itself naturally to be represented using relational matrices to describe the interactions between users, computers and network ports within an enterprise computer network. Clustering these different node types into groups of nodes performing similar activity is useful for improved situational awareness for network administrators and could be used for detection of intruders within the network. A challenge often associated with cyber-security data is the size of the networks and the high volume of events, requiring the methods to be scalable. Additionally, as the network is continuously changing, the method must be computationally fast to allow for cluster updates.

Recently there has been an influx of research performed on Multi-View Clustering where clustering is performed over multiple “views” of the data (all with the same dimensions referring to the same entities). Liu et al. (Liu et al., 2013) applied Non-Negative Matrix Factorisation to address this problem where the clustering for each view is regularised towards a consensus clustering. Alternatively Yan et al. (Yan et al., 2020) apply an adaptive k-means approach for multi-view clustering. In contrast, we work on the problem of clustering over different matrices that either have common rows or columns, called multi-type clustering. Other work in multi-type clustering includes joint

^{**}Corresponding author:

e-mail: elizabeth.riddle-workman13@imperial.ac.uk (Elizabeth Riddle-Workman)

clustering over multiple entity classes using a relational graph model (Banerjee et al., 2007). A probabilistic approach has also been implemented that allows for both hard and soft clustering of the data (Long et al., 2007). Non-Negative Matrix Factorisation (NMF) has been widely applied for multi-type clustering e.g. (Ni et al., 2015) and (Pei et al., 2015). Despite the success of these methods, due to large matrix computations, performing such decompositions is slow.

In this paper we propose a novel procedure called Simple Non-Negative Matrix Tri-Factorisation (Simple NMTF) that can cluster large sparse relational data matrices, quickly with comparable performance to established methods in this domain. Simple NMTF additionally benefits from requiring no hyper-parameters to be set empirically unlike many existing procedures. For consistent results, we build upon the popular NNDSVD initialisation procedure (Boutsidis and Gallopoulos, 2008) for application to multi-type relational data.

In many domains, including cyber-security, labels for the entities are unknown hence internal measures of performance must be utilised. Existing measures only evaluate the performance for a single entity type as opposed to measuring over multiple data sets and entity types. In this work we propose a novel multi-type relational clustering performance measure called Node Cluster Similarity (NCS) which quantifies the similarity between the nodes within their cluster based on their edge similarity and cluster connection similarity.

2. Notation and Problem Formulation

Throughout this paper, matrices are represented by boldface uppercase letters where we use $\mathbf{A}^{ij}$ to denote the i -th row and j -th column entry of matrix $\mathbf{A}$. Additionally $\mathbf{A}^i$ and $\mathbf{A}^j$ refer to the i -th row and j -th column of $\mathbf{A}$ respectively. The data matrices presented are adjacency matrices which detail the connections between two nodes where $\mathbf{A}^{ij} = 1$ if there is a connection between nodes i and j and $\mathbf{A}^{ij} = 0$ otherwise. The rows and columns may either correspond to the same entity type representing a uni-partite graph or if the rows and columns refer to different entity types, this describes a bi-partite graph. Let Ψ represent the set of *cluster indicator matrices*. An indicator matrix $\mathbf{G} \in \Psi^{m \times c}$ is a non-negative matrix where for each integer $1 \leq i \leq n$, $\mathbf{G}^i$ has all entries equal to 0 except for a single entry equal to 1 corresponding to the cluster membership for data object i .

In the context of cyber-security, we are interested in monitoring the actions of different node types where we are specifically concerned with computers, users and ports and their interactions. Ports are communication endpoints which can help to identify the type of network connection (e.g. port 80 is used for HTTP traffic). We focus only on computer-computer, computer-user and computer-port relations. Let $\mathbf{A}_{CC}$ be the adjacency matrix detailing the connections between the source and destination computers (i.e. the computers the connection is initiated and terminated by respectively) which is constructed as an undirected graph ($\mathbf{A}_{CC} = \mathbf{A}_{CC}^T$) where the source and destination computers are contained in a single set. $\mathbf{A}_{CU}$ details the relationship between source computers and users. Finally $\mathbf{A}_{CP}$

details relationship between source computers and destination ports. Let n_c , n_u and n_p represent the number of computers, users and ports within the network. Let c_c , c_u and c_p represent the number of computer, user and port clusters where these are specified by the user. Finally, $\|\cdot\|_F^2$ refers to the squared Frobenius norm and $\|\cdot\|^2$ refers to the squared L^2 norm. The aim of this paper is to produce a single computer, user and port clustering that incorporates information from these three graphs, represented by their adjacency matrices, simultaneously.

3. Background and Related Work

Non-Negative Matrix Factorisation (NMF) is used extensively for decomposing non-negative matrices into two lower rank non-negative matrices where under orthogonality constraints, this is equivalent to k-means clustering (Ding et al., 2005) with the following interpretation; the first matrix represents the cluster centroids and the second acts as a cluster indicator matrix. NMF has been applied to a variety of areas such as text mining (Pauca et al., 2004) and recommender systems (Xin Luo et al., 2014) and was popularised by (Lee and Seung, 2001) who developed a multiplicative update rule.

NMF is extended by Long et al. (Long et al., 2005) to Non-Negative Matrix Tri-Factorisation (NMTF) where instead they utilise the duality of the rows and columns, clustering both simultaneously. NMTF factorises input non-negative matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$ into three non-negative matrices by minimising the objective,

$$J = \|\mathbf{X} - \mathbf{FSG}^T\|_F^2 \text{ s.t. } \mathbf{F} \geq 0, \mathbf{S} \geq 0 \text{ \& } \mathbf{G} \geq 0 \quad (1)$$

where $\mathbf{F} \in \mathbb{R}^{n \times k}$, $\mathbf{S} \in \mathbb{R}^{k \times l}$ and $\mathbf{G} \in \mathbb{R}^{m \times l}$. Generally $k \ll n$ and $l \ll m$. The matrices $\mathbf{F}$ and $\mathbf{G}$ contain the row and column association respectively with their clusters and $\mathbf{S}$ is a compact representation of $\mathbf{X}$ (Long et al., 2005). This $\mathbf{S}$ matrix is needed to absorb the different scales of $\mathbf{X}$, $\mathbf{F}$ and $\mathbf{G}$ (Ding et al., 2006). These matrices are found through performing either: multiplicative, alternating least squares, projected gradients or coordinate descent updates, the details of which are given in (Čopar et al., 2019). These updating mechanism are often slow due to large matrix multiplications within the updates, rendering them infeasible for application to large data matrices. This procedure also requires an additional post-processing step to obtain the cluster assignment of the rows and columns that may lead to non-unique solutions Wang et al. (2011b).

As suggested by Wang et al. (Wang et al., 2011b), to avoid the issues associated with standard NMTF, the matrices $\mathbf{F}$ and $\mathbf{G}$ are instead indicator matrices. Through enforcing this constraint on these matrices, they are now orthogonal allowing for clustering interpretations where $\mathbf{G}$ and $\mathbf{F}$ contain the cluster assignments of the rows and columns of $\mathbf{X}$ respectively. Wang et al. (Wang et al., 2011b) method is called *Fast NMTF* which minimise the following objective function:

$$J = \|\mathbf{X} - \mathbf{FSG}^T\|_F^2 \text{ s.t. } \mathbf{F} \in \Psi^{n \times k}, \mathbf{S} \in \mathbb{R}^{k \times l} \text{ \& } \mathbf{G} \in \Psi^{m \times l}, \quad (2)$$

where Ψ is the set of all cluster indicator matrices. The solution is found by alternatively solving for each matrix. First keeping

both $\mathbf{F}$ and $\mathbf{G}$ fixed and minimise J w.r.t. $\mathbf{S}$ (by taking derivative and setting to 0) gives,

$$\mathbf{S} = (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T \mathbf{X} \mathbf{G} (\mathbf{G}^T \mathbf{G})^{-1}. \quad (3)$$

Keeping $\mathbf{F}$ and $\mathbf{S}$ fixed, J is minimised by $\mathbf{G}$, an indicator matrix where $\mathbf{G}$ is such that,

$$g_{ij} = \begin{cases} 1 & \text{if } j = \operatorname{argmin}_k \|\mathbf{X}^i - (\mathbf{F}\mathbf{S})^k\|^2 \\ 0 & \text{otherwise} \end{cases}. \quad (4)$$

A similar expression can be calculated for $\mathbf{F}$. As the updates do not require large matrix multiplications, the clustering is fast to compute. Additionally the algorithm converges to a local minimum due to the nature of alternating optimisation. An additional benefit of this method is its interpretability. $\mathbf{S}\mathbf{G}^T$ and $\mathbf{F}\mathbf{S}$ contain the row and column clusters centres respectively (represented by their mean).

We now introduce two competitive multi-type clustering approaches. Wang et al. (Wang et al., 2011a) propose Fast Non-Negative Matrix Tri-Factorisation (F-NMTF), where both inter-type and intra-type relationship matrices are used to simultaneously cluster multiple data objects. Their work uses NMTF to cluster the data where they also perform hard clustering to improve computational speed. Manifold regularisation is used for including the intra-type relationships within the objective function. Pei et al. (Pei et al., 2015) also simultaneously cluster over multiple networks using a method called R-NMTF for clustering in social media networks imposing orthogonality constraints on the cluster matrices. Both algorithms presented in (Wang et al., 2011a) and (Pei et al., 2015) require hyper-parameters to be set empirically using a grid search minimising external performance measures that utilise a known clustering for the data however in practice, this is not feasible as labelled data often does not exist.

4. Simple Non-Negative Matrix Tri-Factorisation (Simple NMTF) for Multi-Type Bi-clustering

In this section we present our novel multi-type clustering procedure in a general framework that simultaneously clusters over multiple entities whose behaviour is described using a number of adjacency matrices.

4.1. Objective Function

In many settings, particularly enterprise networks, high dimensional data sets are generated hence clustering procedures must be computationally fast and scalable. Interpretability is highly important in many fields including cyber-security. Similarly to Fast NMTF, we restrict the factor matrices within NMTF to be cluster indicator matrices to produce hard clustering results for faster computational speeds and better interpretability.

The aim is to cluster simultaneously over multiple relation matrices consisting of both intra and inter-type relations for multiple entity types. Let $\mathbf{A}_{k(i)k(i)}^{(i)}$ be the i^{th} intra-type relational adjacency matrix involving entity type $k(i)$. Let $\mathbf{A}_{l(i)m(i)}^{(i)}$ be the

i^{th} inter-type relational adjacency matrix between entity types $l(i)$ and $m(i)$. Let Z be the set of entity types. From these matrices we seek a single clustering for each of the entity types that incorporates information from the different relations. By incorporating multiple matrices, the latent representation of the entities shared between the data sets can be better realised and can help reduce the noise present in each. To find this shared clustering structure, multiple objective functions are minimised simultaneously in the following manner:

$$J = \sum_{i=1}^q \|\mathbf{A}_{k(i)k(i)}^{(i)} - \mathbf{G}_{k(i)} \mathbf{S}_{k(i)}^{(i)}\|_F^2 + \sum_{i=1}^r \|\mathbf{A}_{l(i)m(i)}^{(i)} - \mathbf{G}_{l(i)} \mathbf{S}_{l(i)m(i)}^{(i)} \mathbf{G}_{m(i)}^T\|_F^2 \quad \text{s.t. } \mathbf{G}_z \in \Psi^{n_z \times c_z}, \mathbf{S}_{zw} \in \mathbb{R}^{c_z \times c_w} \text{ \& } z, w \in Z \quad (5)$$

where n_z and c_z are the numbers of nodes and clusters respectively for entity type z . Here the intra relationship matrices $\mathbf{A}_{k(i)k(i)}^{(i)}$ are not bi-clustered as it is a uni-partite matrix. Additionally when the objective for $\mathbf{A}_{k(i)k(i)}^{(i)}$ is minimised in isolation, this is equivalent to k-means clustering. The inter relationship matrices are however bi-clustered so that information about both node types are incorporated during the clustering. This function is minimised with respect to $\mathbf{G}_z$ and $\mathbf{S}_{zw}$ for $z, w \in Z$. This formulation also allows for multiple views of the same relation.

4.2. Optimisation Strategy

As the optimisation function is not convex over all matrices, the solution is instead found by iteratively solving for each matrix separately until convergence. First, by fixing all matrices but $\mathbf{S}_{k(i)k(i)}^{(i)}$ and setting the derivative with respect to $\mathbf{S}_{k(i)k(i)}^{(i)}$ of the full objective to zero gives,

$$\mathbf{S}_{k(i)k(i)}^{(i)} = (\mathbf{G}_{k(i)}^T \mathbf{G}_{k(i)})^{-1} \mathbf{G}_{k(i)}^T \mathbf{A}_{k(i)k(i)}^{(i)}. \quad (6)$$

Similarly consider when all matrices but the inter $\mathbf{S}_{l(i)m(i)}^{(i)}$ matrix are held fixed. By setting the derivative of the objective function with respect to $\mathbf{S}_{l(i)m(i)}^{(i)}$ to zero gives,

$$\mathbf{S}_{l(i)m(i)}^{(i)} = (\mathbf{G}_{l(i)}^T \mathbf{G}_{l(i)})^{-1} \mathbf{G}_{l(i)}^T \mathbf{A}_{l(i)m(i)}^{(i)} \mathbf{G}_{m(i)} (\mathbf{G}_{m(i)}^T \mathbf{G}_{m(i)})^{-1}. \quad (7)$$

Suppose $\mathbf{G}_z$ for $z \in Z$ is associated with multiple norms then when optimising for this clustering all related terms are considered. Keeping all other matrices but $\mathbf{G}_z$ fixed, the problem is decoupled to the following problem for each node i ($1 \leq i \leq n_z$),

$$\min_{\mathbf{G}_z \in \Psi} \sum_{j \in \{t \in \{1, \dots, r\} | k(t)=z\}} \|\mathbf{A}_{k(j)k(j)}^{(j)} - \mathbf{G}_z^i \mathbf{S}_{k(j)k(j)}^{(j)}\|_F^2 + \sum_{j \in \{t \in \{1, \dots, r\} | l(t)=z\}} \|\mathbf{A}_{l(j)m(j)}^{(j)} - \mathbf{G}_z^i \mathbf{S}_{l(j)m(j)}^{(j)} \mathbf{G}_{m(j)}^T\|_F^2 + \sum_{j \in \{t \in \{1, \dots, r\} | m(t)=z\}} \|\mathbf{A}_{l(j)m(j)}^{(j)} - \mathbf{G}_{l(j)} \mathbf{S}_{l(j)m(j)}^{(j)} (\mathbf{G}_z^T)^i\|_F^2. \quad (8)$$

Algorithm 1 Simple NMTF

Input: Intra-type relational matrices $\mathbf{A}_{k(i)k(i)}^{(i)}$, inter-type relational matrices $\mathbf{A}_{l(i)m(i)}^{(i)}$ and cluster sizes c_z for each entity type $z \in Z$.

Initialisation: Initialise $\mathbf{G}_z$ for $z \in Z$ using the NNDSVD initialisation procedure detailed in Section. 5.1

repeat

- 1: Update intra $\mathbf{S}$ matrices using Eq. (6).
- 2: Update inter $\mathbf{S}$ matrices using Eq. (7).
- 3: Update $\mathbf{G}_z$ for $z \in Z$ using Eq. (9).

until Converges

Output: $\mathbf{G}_z$ for $z \in Z$ indicating cluster membership for each entity type.

As $\mathbf{G}_z$ is a cluster indicator matrix, the optimal matrix is given by,

$$\mathbf{G}_z^{i,j} = \begin{cases} 1 & j = \underset{s}{\operatorname{argmin}} \sum_{p \in \{t \in \{1, \dots, r\} | k(t)=z\}} \|\mathbf{A}_{k(p)k(p)}^{(p)j} - \mathbf{S}_{k(p)k(p)}^{(p)s}\|^2 \\ & + \sum_{p \in \{t \in \{1, \dots, r\} | l(t)=z\}} \|\mathbf{A}_{l(p)m(p)}^{(p)j} - (\mathbf{S}_{l(p)m(p)}^{(p)} \mathbf{G}_{m(p)}^T)^s\|^2 \\ & + \sum_{p \in \{t \in \{1, \dots, r\} | m(t)=z\}} \|\mathbf{A}_{l(p)m(p)}^{(p)j} - (\mathbf{G}_{l(p)} \mathbf{S}_{l(p)m(p)}^{(p)})^s\|^2 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

The Simple NMTF procedure is summarised by Algorithm

1. In our examples we perform Simple NMTF over a single intra matrix $\mathbf{A}_{CC}$ and two inter-type relation matrices $\mathbf{A}_{CU}$ and $\mathbf{A}_{CP}$. Unlike the two multi-type clustering procedures detailed in Section 3, our clustering algorithm does not require any hyper-parameters to be set.

4.3. Convergence and Complexity

Alternating optimisation is used in Simple NMTF where convergence to a local minimum is guaranteed (Wang et al., 2011b). Assuming the time complexity of multiplying matrices $\mathbf{A}$ ($m \times n$) and $\mathbf{B}$ ($n \times p$) is $O(mnp)$, the time complexity of Algorithm 1 is $O(r(c_r^2 n_c + c_n n_c^2 + c_c^3 + c_c n_c n_u + c_n n_u c_u + n_u c_u^2 + c_u^3 + c_c n_c n_p + c_n n_p c_p + n_p c_p^2 + c_p^3))$ where r is the total number of iterations for convergence for our enterprise network case.

5. Proposed Multi-Network Extensions

5.1. Cluster Initialisation Strategy

In the majority of papers that utilise NMTF, a random initialisation procedure is used however the solutions for these algorithms are often not stable for each run and produce wildly different clusterings. Alternatively, Non-Negative Double Singular Value Decomposition (NNDSVD) proposed by Boutsidis and Gallopoulos (Boutsidis and Gallopoulos, 2008) for initialising the two components within standard NMF. This widely applied method utilises the positive section of the k leading singular triplets of the matrix. Alternate SVD based initialisation procedures have also been proposed (Atif et al., 2019) and (Qiao, 2015). We too propose utilising SVD for initialisation within our algorithm and is an extension of NNDSVD.

For matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ of rank $r \leq \min(m, n)$, the optimal rank- k approximation, $\mathbf{A}^{(k)}$ ($k \leq r$), with respect to the Frobenius norm is given by the SVD rank- k approximation (Eckart and Young, 1936),

$$\mathbf{A}^{(k)} = \sum_{j=1}^k \sigma_j \mathbf{u}_j \mathbf{v}_j^T = \sum_{j=1}^k \sigma_j \mathbf{C}^{(j)}, \quad (10)$$

where σ_j are the non-negative singular values in decreasing order and $\mathbf{u}_j$ and $\mathbf{v}_j$ are the corresponding left and right singular vectors respectively where $\mathbf{C}^{(j)} = \mathbf{u}_j \mathbf{v}_j^T$. The best non-negative approximation of $\mathbf{C}^{(j)}$ with regards to the Frobenius norm is given by its non-negative part $\mathbf{C}_+^{(j)}$ (Boutsidis and Gallopoulos, 2008). For standard NMF, NNDSVD initialises the two factors using the left and right singular vectors of $\mathbf{C}_+^{(j)}$ respectively multiplied by the square root of the singular value to give $\mathbf{W}^*$ and $\mathbf{H}^*$, the initialised values for standard NMF. We extend this methodology to initialise bi-clustering for hard clustering. The NNDSVD initialisation procedure is first applied to the bipartite adjacency matrices for producing initial cluster indicator matrices:

$$\mathbf{F}_{\text{initial}}^{i,j} = \begin{cases} 1 & j = \underset{s}{\operatorname{argmax}} \mathbf{W}^{*sj} \\ 0 & \text{otherwise} \end{cases} \quad \mathbf{G}_{\text{initial}}^{i,j} = \begin{cases} 1 & j = \underset{s}{\operatorname{argmax}} \mathbf{H}^{*sj} \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

This is equivalent to the initial clustering being chosen as the singular vector with the largest value for each node and initial $\mathbf{S}$ is set as the identity matrix.

If the rows and columns are to be clustered into c_r and c_c clusters respectively, first NNDSVD is performed using $\max(c_r, c_c)$ as the rank for the decomposition. When calculating $\mathbf{W}^*$ and $\mathbf{H}^*$, only the leading c_r and c_c singular value/vector pairs are kept respectively before using Eq. (11) to get the initial cluster indicator matrices. For initialisation of multi-type clustering, averaging the relevant $\mathbf{W}^*$ and $\mathbf{H}^*$ referring to each entity type before applying Eq. (11) is performed.

Empirical results on simulated graphs showed the results are more stable with significantly less variation between the runs compared to random initialisation with similar model performance. On real data the initialisation procedure improved the model performance. For brevity, the comparison results for the random and proposed initialisation procedure are not presented.

5.2. Performance Evaluation

Clustering performance techniques are divided into two categories: external and internal performance measures (Jain and Dubes, 1988). External performance measures evaluate how similar the clustering is to a known true clustering of the data. These measures are useful for simulated examples where the true clustering is known. Internal measures instead utilise the data to assess the clustering performance where a desired property of the clustering is measured such as having compact clusters.

In this study, the *Adjusted Rand Index* (ARI) (Hubert and Arabic, 1985), an external performance measure, is used to evaluate the similarity between the clustering generated by an

algorithm and the true clustering. Let n_{ij} represent the number of objects in common between cluster i of the algorithm's clustering and true cluster j , then the ARI is given by,

$$\text{ARI} = \frac{\sum_{i,j} \binom{n_{ij}}{2} - \sum_i \binom{n_{i.}}{2} \sum_j \binom{n_{.j}}{2} / \binom{n}{2}}{\frac{1}{2} [\sum_i \binom{n_{i.}}{2} + \sum_j \binom{n_{.j}}{2}] - \sum_i \binom{n_{i.}}{2} \sum_j \binom{n_{.j}}{2} / \binom{n}{2}}, \quad (12)$$

where $n = \sum_{i,j} n_{ij}$. The ARI has values between -1 and 1 where 0 corresponds to random labelling independent of the numbers of clusters and 1 signifies the clusterings are identical (up to permutation). As this is an external measure, the true clustering must be known but in application labels are often not available.

We now propose a novel similarity based clustering measure for assessing the performance of the generated clustering for multi-type relational data. The majority of internal based measures such as Dunn's Index (Dunn, 1974) focus on identifying compact well separated clusters but are not suited for evaluating bi-clustering. Our performance measure seeks to measure how similar nodes within each cluster are in terms of similarity in connections and similarity in the clusters it has connections to, where higher similarity is desirable. It does not however take dissimilarity between clusters into consideration where having multiple similar clusters is not penalised.

More formally let $\mathbf{a}_i$ be a binary vector detailing the other nodes for which node i connects to and $\mathbf{c}_i$ details the number of connections node i has to each cluster. We propose the *Node Cluster Similarity* (NCS) to measure the similarity between nodes i and j given by:

$$s_{ij} = \frac{\text{sim}(\mathbf{a}_i, \mathbf{a}_j) + \text{sim}(\mathbf{c}_i, \mathbf{c}_j)}{2} \quad (13)$$

where cosine similarity is used here. This similarity score has values between 0 and 1 where 1 signifies both nodes have identical connections. For a single node set of size N divided into K clusters C_k , the performance over all nodes is calculated as:

$$\text{NSC}_{\text{single entity}} = \frac{2}{N} \sum_{k=1}^K \frac{1}{2} \mathbb{1}_{|C_k|=1} + \sum_{i,j \in C_k, i < j} \frac{s_{ij}}{|C_k|-1} \mathbb{1}_{|C_k| \neq 1}. \quad (14)$$

Eq. (14) can be extended for bipartite graphs. If $\{C_k, k = 1, \dots, K_1\}$ and $\{C_k, k = K_1 + 1, \dots, K_1 + K_2\}$ are the sets of clusters for node set 1 and 2 respectively and N_1 and N_2 are the number of nodes in these sets respectively, then the NCS for bi-clustering is:

$$\text{NCS}_{\text{bi-clustering}} = \frac{2}{N_1 + N_2} \sum_{k=1}^{K_1+K_2} \frac{1}{2} \mathbb{1}_{|C_k|=1} + \quad (15)$$

$$\sum_{i,j \in C_k, i < j} \frac{s_{ij}}{|C_k|-1} \mathbb{1}_{|C_k| \neq 1} \quad (16)$$

For multi-type data, the clustering performance can be evaluated using either Eq. (14) or (16) on a matrix with the relevant matrices concatenated along the shared axis. If the bipartite formulation is used on this matrix, a single score detailing the performance of the clustering algorithm over both the rows and columns is produced.

Number Computers / Clusters	5000 / 200
Number Users / Clusters	5000 / 200
Number Ports / Clusters	1000 / 50
CC Number Cluster Connections	{1,2,3}
CU Number Cluster Connections	{1,2}
CP Number Cluster Connections	{1,2,...,7}
Probability of Connection CC	{0.1,0.15,0.2}
Probability of Connection CU	{0.01,0.05,0.1}
Probability of Connection CP	{0.3,0.4,0.5,0.6}

Table 1. Description of simulated data.

6. Experiments

We have compared our proposed clustering algorithm to a number of competitive methods on real and simulated data sets for our specified problem. The competitive clustering methods applied are divided into the following three classes:

Single Entity Clustering Methods: The uni-partite clustering is applied to each entity type separately. For the computer clustering, all three data matrices were concatenated to produce a single computer clustering. The methods applied are the standard NMF algorithm (Lee and Seung, 2001) and SVD+K-means (Zha et al., 2001).

Bi-Clustering Methods: Orthogonal Non-Negative Matrix Tri-Factorisation (ONMTF) (Ding et al., 2006) is applied to each adjacency matrix separately. Similar to the single entity methods, for the computer entities, clustering is performed over the concatenated matrix to produce a single clustering for this entity type. Utilising ONMTF instead of standard NMTF allows for clustering interpretations as discussed earlier.

Multi-Type Clustering Methods: The F-NMTF (Wang et al., 2011b) approach, detailed in Section 3, is used which utilises a manifold regularisation for intra-type relations. Additionally R-NMTF proposed by Pei et al. (2015) is implemented where we have translated their method to fit our problem. For these methods, hyper-parameters suggested by the authors are used.

For all competitive approaches except F-NMTF and R-NMTF an initialisation procedure has either been proposed by the cited author or if it is a common approach such as NMF, a standard initialisation procedure is employed. Following the authors of F-NMTF and R-NMTF, a random initialisation is applied for these methods.

6.1. Simulated Data

To generate the adjacency matrices, a stochastic block model (Holland et al., 1983) is employed where connections between clusters occur with a certain probability and each cluster connects to a specified number of clusters. Stochastic block models lend themselves well to the type of clustering we are aiming to carry out where nodes in each cluster connect to similar nodes or clusters. More specifically, for the simple uni-partite graph case let N nodes be divided into K clusters, $\{C_1, C_2, \dots, C_K\}$. Let $\mathbf{P} \in \mathbb{R}^{N \times N}$ contain the probability of an edge between the clusters where $\mathbf{P}^{ij}$ is the probability of an edge between nodes in clusters i and j . The edges within the graph are sampled at random where for any two nodes $x \in C_i$ and $y \in C_j$ an edge

Table 2. Average ARI, NCS and time (in minutes) comparison for the different methods for the 20 simulated networks. Best performer in bold, second best in bold grey.

	Entity	True	NMF	KMeans	ONMTF	F-NMTF	R-NMTF	Simple NMTF
ARI	Computer		0.280(0.030)	0.367(0.061)	0.160(0.004)	0.496 (0.030)	0.465(0.106)	0.705 (0.053)
	User		0.284(0.053)	0.008(0.002)	0.100(0.016)	0.120(0.016)	0.340 (0.081)	0.393 (0.042)
	Port		0.928 (0.031)	0.402(0.071)	0.559(0.016)	0.806(0.022)	0.692(0.098)	0.870 (0.059)
	All		0.299(0.031)	0.052(0.005)	0.098(0.007)	0.235(0.020)	0.383 (0.079)	0.515 (0.037)
NCS	CC-C	0.534(0.005)	0.165(0.014)	0.291 (0.017)	0.172(0.005)	0.266(0.015)	0.267(0.042)	0.405 (0.015)
	CU-C	0.547(0.012)	0.109(0.010)	0.308 (0.014)	0.250(0.015)	0.234(0.018)	0.292(0.037)	0.413 (0.018)
	CU-U	0.547(0.014)	0.303(0.028)	0.336 (0.031)	0.163(0.008)	0.336 (0.022)	0.318(0.060)	0.460 (0.014)
	CP-C	0.722(0.004)	0.461(0.021)	0.519(0.022)	0.365(0.004)	0.593 (0.010)	0.547(0.040)	0.664 (0.011)
	CP-P	0.727(0.004)	0.706 (0.006)	0.492(0.030)	0.566(0.001)	0.593(0.021)	0.625(0.034)	0.641 (0.030)
	Computer	0.653(0.005)	0.383(0.017)	0.460(0.015)	0.330(0.001)	0.497 (0.015)	0.469(0.038)	0.586 (0.011)
	Overall	0.611(0.006)	0.391(0.019)	0.391(0.015)	0.275(0.001)	0.433 (0.015)	0.414(0.033)	0.533 (0.008)
Time			9.883(0.701)	3.314 (0.135)	61.513(42.0193)	19.748(0.795)	35.926(312.038)	0.755 (0.061)

occurs with probability $\mathbf{P}^{i,j}$ where $i, j \in \{1, \dots, K\}$. Bi-partite graphs are simulated in a similar manner where instead let there be N_1 nodes of type 1 and N_2 nodes of type 2 where these nodes are split into K_1 and K_2 clusters respectively with the following cluster sets: $\{C_1^1, C_2^1, \dots, C_{K_1}^1\}$ and $\{C_1^2, C_2^2, \dots, C_{K_2}^2\}$. $\mathbf{P} \in \mathbb{R}^{N_1 \times N_2}$ contains the probability of an edge between the clusters of the two node types.

For each simulation a single computer, user and port clustering is generated by random assignment for the desired cluster numbers. This clustering is used for the generation of all three relational graphs. The following restrictions are implemented to aid in generating graphs that exhibit similar properties to observed network data. For each cluster, the number of connecting clusters is restricted where this number is chosen at random from a finite set of possible values. Probability of connections between clusters is also chosen from a restricted set of probabilities at random. Table 1 summarises the parameters used to generate the data. As initialisation affects the performance for a number of the competitors methods, each method was run 20 times on 20 simulated networks with results averaged over the runs and networks. All experiments are performed on a four 16-core AMD Opteron CPU at 3.0GHz and 512G bytes memory.

6.2. Simulated Data Results

Table 2 presents the mean and standard deviation for ARI, NCS and run time of our proposed Simple NMTF to the other competitive methods across the 20 simulated networks. Simple NMTF achieves the highest ARI for computer, user and overall whereas NMF performs well for identifying the port clustering only. The poor user clustering performance for all algorithms is due to the weak clustering structure present for this node type by construction. Additionally although the clustering structure within the CC and CU graphs is much less prominent than that of CP, the clustering algorithm is able to utilise information from all matrices to uncover the true computer clustering with high similarity to the truth. Similar results are observed for NCS over all matrices and node types, highlighting its consistent good clustering performance. Simple NMTF is

over 4 times faster than the second fastest approach, KMeans with significantly better performance than all other algorithms.

6.3. Real Data Set Description

To assess the performance of the algorithm on a real cybersecurity data set, the publicly available ‘‘Unified Host and Network Data Set’’¹ (Turcotte et al., 2017) from Los Alamos National Laboratory (LANL) is used which contains both Netflow and Windows Logging Service (WLS) data over a 90 day period. The Netflow data details the interactions between the computers within the network whereas the WLS data logs local activities such as authentication events occurring on each host. Due to space limitations, only a 24 hour window of data from day 22, a Monday, is used to create the adjacency matrices detailed in Section 2 whose size is described in Table 3. We reduce the ports used to the top 2000 ports within the investigated period as these provide information about the connections performed. As the true labels are unknown, it is assessed only via NCS.

6.4. Real Data Results

All clustering algorithms ran a total of 50 times on the real data set where the average NCS values and running times are shown in Table 4. Simple NMTF outperforms the other algorithms when clustering on the CC matrix for computers. Additionally, it consistently performs well and is generally either the second or third best method across the entities. In contrast, none of the other algorithms performance have a consistent behaviour between the matrices and nodes. For example, the computer clustering from F-NMTF performs very well for the computers on the CP matrix with poor performance for the computers on the CC and CU matrices.

The clear advantage of Simple NMTF is the computational speed which is significantly faster than any other algorithm, irrespective of the large magnitude of data. For example, Simple NMTF is over 72 times faster than ONMTF, the overall second best algorithm. The computational speed and overall consistent

¹<https://csr.lanl.gov/data/2017.html>

Table 4. NCS and time (in minutes) comparison for the different methods for LANL data. Best performer in bold, second best in bold grey.

	Entity	NMF	KMeans	ONMTF	F-NMTF	R-NMTF	Simple NMTF
NCS	CC	0.452(0.009)	0.513(0.008)	0.613 (0.000)	0.542(0.014)	0.587(0.028)	0.746 (0.002)
	CU-C	0.043(0.003)	0.190 (0.046)	0.131(0.000)	0.042(0.003)	0.054(0.009)	0.176 (0.006)
	CU-U	0.579(0.011)	0.482(0.032)	0.642 (0.000)	0.634 (0.011)	0.529(0.016)	0.631(0.004)
	CP-C	0.655(0.016)	0.660(0.011)	0.696(0.000)	0.762 (0.012)	0.716(0.017)	0.761 (0.001)
	CP-P	0.507(0.020)	0.462(0.011)	0.550 (0.000)	0.287(0.015)	0.548 (0.005)	0.399(0.002)
	Comp	0.592(0.014)	0.603(0.010)	0.654(0.000)	0.696 (0.014)	0.668(0.015)	0.730 (0.001)
	Overall	0.578(0.004)	0.534(0.015)	0.639 (0.000)	0.628(0.008)	0.593(0.009)	0.653 (0.002)
Time		46.349(4.741)	5.627(1.176)	186.042(44.595)	47.907(8.193)	101.778(0.916)	2.562 (0.665)

Table 3. Summarising features of LANL 24 hour data of day 22.

Number Computers	9321
Number Users	9413
Number Ports	2000
Number Computer Clusters	300
Number User Clusters	300
Number Port Clusters	100

performance of Simple NMTF on all matrices makes it more suitable for real time monitoring compared to the other methods.

7. Conclusion

In this paper, we proposed Simple NMTF, a multi-type relational clustering algorithm suitable for clustering over both intra and inter-type relational matrices for multiple entity types. Due to constraining the factor matrices to indicator matrices, the clustering approach is computationally fast and scalable for large computer network adjacency matrices and can be used for improved situational awareness. We also presented an extension of the NNDSVD initialisation procedure for multi-type relational data that allows for consistent results between runs. A new internal performance measure, NCS, was proposed which is suitable for assessing cluster performance based on similarity of nodes clustered together. Experiments on both simulated and real data sets showed the consistency in the performance of Simple NMTF and highlighted the extreme improvement in computational speed Simple NMTF offers over existing methods, which is an essential property for real time monitoring in enterprise cyber-security networks.

Acknowledgements

This work is funded by The Engineering and Physical Sciences Research Council (EPSRC) Centres for Doctoral Training (CDT) in Financial Computing & Analytics studentship.

References

Atif, S.M., Qazi, S., Gillis, N., 2019. Improved SVD-based Initialization for Nonnegative Matrix Factorization using Low-Rank Correction. *Pattern Recognition Letters* 122, 53–59. doi:10.1016/j.patrec.2019.02.018.

Banerjee, A., Basu, S., Merugu, S., 2007. Multi-way Clustering on Relation Graphs, in: *Proceedings of the 2007 SIAM International Conference on Data Mining*, pp. 145–156.

Boutsidis, C., Gallopoulos, E., 2008. SVD Based Initialization: A Head Start for Nonnegative Matrix Factorization. *Pattern Recognition* 41, 1350–1362.

Čopar, A., Zupan, B., Zitnik, M., 2019. Fast Optimization of Non-Negative Matrix Tri-Factorization. *PLOS ONE* 14.

Ding, C., He, X., Simon, H.D., 2005. On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering, in: *Proceedings of the 2005 SIAM International Conference on Data Mining*, pp. 606–610.

Ding, C., Li, T., Peng, W., Park, H., 2006. Orthogonal Nonnegative Matrix Tri-Factorizations for Clustering, in: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '06*, p. 126.

Dunn, J.C., 1974. Well-Separated Clusters and Optimal Fuzzy Partitions. *Journal of Cybernetics* 4, 95–104.

Eckart, C., Young, G., 1936. The Approximation of One Matrix by Another of Lower Rank. *Psychometrika* 1, 211–218.

Holland, P.W., Laskey, K.B., Leinhardt, S., 1983. Stochastic Blockmodels: First Steps. *Social Networks* 5, 109–137.

Hubert, L., Arabic, P., 1985. Comparing Partitions. *Journal of Classification* 2, 193–218.

Jain, A.K., Dubes, R.C., 1988. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs.

Kumar, A., Rai, P., Daumé III, H., 2011. Co-regularized Multi-view Spectral Clustering, in: *NIPS'11 Proceedings of the 24th International Conference on Neural Information Processing Systems*, pp. 1413–1421.

Lee, D.D., Seung, H.S., 2001. Algorithms for Non-Negative Matrix Factorization, in: *Advances in Neural Information Processing Systems*, pp. 556–562.

Liu, J., Wang, C., Gao, J., Han, J., 2013. Multi-View Clustering via Joint Nonnegative Matrix Factorization, in: *Proceedings of the 2013 SIAM International Conference on Data Mining*, pp. 252–260.

Long, B., Zhang, Z.M., Yu, P.S., 2005. Co-Clustering by Block Value Decomposition, in: *Proceeding of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining - KDD '05*, ACM Press, pp. 635–640.

Long, B., Zhang, Z.M., Yu, P.S., 2007. A Probabilistic Framework for Relational Clustering, in: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '07*, ACM Press, pp. 470–479.

Ni, J., Tong, H., Fan, W., Zhang, X., 2015. Flexible and Robust Multi-Network Clustering, in: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15*, pp. 835–844.

Pauca, V.P., Shahnaz, F., Berry, M.W., Plemmons, R.J., 2004. Text Mining using Non-Negative Matrix Factorizations, in: *Proceedings of the 2004 SIAM International Conference on Data Mining*, pp. 452–456.

Pei, Y., Chakraborty, N., Sycara, K., 2015. Nonnegative Matrix Tri-Factorization with Graph Regularization for Community Detection in Social Networks. *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2083–2089.

Qiao, H., 2015. New SVD Based Initialization Strategy for Non-Negative Matrix Factorization. *Pattern Recognition Letters* 63, 71–77. doi:10.1016/j.patrec.2015.05.019.

Turcotte, M.J.M., Kent, A.D., Hash, C., 2017. Unified Host and Network Data Set. *Technical Report arXiv:1708.07518*.

Wang, H., Nie, F., Huang, H., Ding, C., 2011a. Nonnegative Matrix Tri-Factorization based High-Order Co-Clustering and its Fast Implementation,

- in: Proceedings - IEEE International Conference on Data Mining, ICDM, IEEE. pp. 774–783.
- Wang, H., Nie, F., Huang, H., Makedon, F., 2011b. Fast Nonnegative Matrix Tri-Factorization for Large-Scale Data Co-Clustering, in: IJCAI International Joint Conference on Artificial Intelligence, pp. 1553–1558.
- Xin Luo, Mengchu Zhou, Yunni Xia, Qingsheng Zhu, 2014. An Efficient Non-Negative Matrix-Factorization-Based Approach to Collaborative Filtering for Recommender Systems. IEEE Transactions on Industrial Informatics 10, 1273–1284.
- Yan, F., Wang, X.d., Zeng, Z.q., Hong, C.q., 2020. Adaptive Multi-View Subspace Clustering for High-Dimensional Data. Pattern Recognition Letters 130, 299–305. doi:10.1016/j.patrec.2019.01.016.
- Zha, H., He, X., Ding, C., Simon, H., Gu, M., 2001. Spectral Relaxation for K-means Clustering, in: Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic, MIT Press. pp. 1057–1064.