Data clustering using topological features

Brazilian Conference on Intelligent Systems, 3th, 2014, São Carlos.
http://www.producao.usp.br/handle/BDPI/48592

Downloaded from: Biblioteca Digital da Produção Intelectual - BDPI, Universidade de São Paulo
Abstract—Clustering is one of the most used data mining techniques, while computational topology is a very recent field bridging abstract mathematics with concrete computational techniques. In this paper, we explore the hypothesis that topologically-similar clusters may indicate meaningful relationships. Our approach has an efficient implementation based on computing Minimum Spanning Trees to obtain topological information of each cluster. We then compute a discreteness and a disconnectedness index, used to characterize each cluster, thus allowing the retrieval of equivalence classes. We show that for a real-world high-dimensional network intrusion data set, the topologically-similar clusters retrieved by our approach do indeed correspond to meaningful equivalence classes present in the data set.

I. INTRODUCTION

Clustering [1] is one of the most used data mining techniques [2], which aims to place similar objects together and separate them from dissimilar ones. It is a specially interesting technique for dealing with large data sets in which supervision may not be available, but structure wants to be found nevertheless.

Within the many clustering paradigms, one of the most successful is the density-based paradigm. It allows one to find clusters that correspond to regions of high density that are separated from one another by regions of low density [3]. The advantages of density-based clustering are that 1) it is a very intuitive notion of clustering and 2) clusters of virtually any shape can be found, as long as they are separated by low-density regions and their intra-cluster densities do not vary widely from one another. For this reason, in our proposed approach, we use a classical density-based clustering algorithm, namely DBSCAN [4].

In this paper, we investigate the hypothesis that clusters with similar topological features can indicate meaningful relationships. We present results on synthetic data that confirms the ability of our approach to find topologically related clusters. We also analyze a real-world challenge data set on network intrusion from KDD’99 [5], in which we show that topologically-similar clusters belong to the same type of network attack.

Computational Topology [6] is a recent field that has been created to fill a gap in obtaining qualitative information about data sets. Traditionally, data analysis has focused on computing similarities of objects simply in terms of their quantitative spatial closeness, ignoring morphological and more qualitative information. Topology is precisely the branch of mathematics that studies qualitative geometric information, thus it is a natural choice for filling this gap.

Topology studies the connectivity information of a space, including the identification of loops, voids and higher dimensional surfaces. Quantitative distance values are replaced by the notion of infinite closeness of a point to a subset of the space, which is useful when there is limited understanding of the metric space the data is embedded in.

We give a brief introduction to the notation and concepts surrounding topological spaces [6], [7]. Formally, a topology on a set $X$ is a subset $\tau \subseteq 2^X$ such that: 1) if $S_1, S_2 \in \tau$, then $S_1 \cap S_2 \in \tau$; 2) if $\{S_j\}_{j \in J} \subseteq \tau$, then $\bigcup_{j \in J} S_j \in \tau$; and 3) $\emptyset, X \in \tau$. This implies that topology is simply a system of subsets that describe the connectivity of a set.

The pair $(X, \tau)$ of a set $X$ and a topology $\tau$ is a topological space. $\tau$ is used as notation for a topological space $X$, with $\tau$ implicit. In data mining, we are used to metric spaces, specially Euclidean space. A metric space is a topological space, which is given by a set $X$ with a metric function $d$. Euclidean space can thus be defined as the Cartesian product of $n$ copies of $\mathbb{R}$ along with the Euclidean metric: $d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$, where $u_i$ is the $i$-th Cartesian coordinate function, thus forming the $n$-dimensional Euclidean space $\mathbb{R}^n$.

Topology is inherently a classification system, thus given a set of topological spaces, we are interested in partitioning the set into sets of spaces that are connected in the same way. A partition of a set is a decomposition of the set into subsets (cells) such that every element of the set is in one and only one of the subsets, i.e., creating disjoint sets.

Let $S$ be a nonempty set and let $\sim$ be an equivalence relation on $S$. Then, $\sim$ yields a natural partition of $S$, where $\pi = \{x \in S | x \sim a\}$. $\pi$ represents the subset to which $a$ belongs to. Each cell (subset) $\pi$ is an equivalence class.

A homeomorphism $f : X \rightarrow Y$ is a bijective function such that both $f$ and $f^{-1}$ are continuous. We say that $X$ is homeomorphic to $Y$, $X \approx Y$ and that $X$ and $Y$ have the same topological type. Homeomorphisms partition the class of topological spaces into equivalence classes of homeomorphic spaces. We will later develop, in our methodology, a way to find topological features that can be used to partition clusters into equivalence classes.

A useful topological space on which to develop an intuitive notion of homeomorphisms is a manifold, which is, informally, a space that locally behaves as $\mathbb{R}^n$. That is, each point admits a coordinate system, consisting of coordinate functions on the points of the neighborhood, determining its topology. We use a homeomorphism to define a chart. A chart at $p \in X$ is a function $\varphi : U \rightarrow \mathbb{R}^d$, where $U \subseteq X$. 

Data clustering using topological features

Cássio M. M. Pereira
Institute of Mathematical and Computer Sciences
São Carlos, SP 13566-590, Brazil
Email: cpereira@icmc.usp.br

Rodrigo F. de Mello
Institute of Mathematical and Computer Sciences
São Carlos, SP 13566-590, Brazil
Email: mello@icmc.usp.br

978-1-4799-5618-0/14 $31.00 © 2014 IEEE
DOI 10.1109/BRACIS.2014.71

2014 Brazilian Conference on Intelligent Systems
is an open set containing \( p \) and \( \varphi \) is a homeomorphism onto an open subset of \( \mathbb{R}^d \). The dimension of the chart is \( d \). The coordinate functions of the chart are \( x^i = u^i \circ \varphi : U \to \mathbb{R} \), where \( u^i : \mathbb{R}^n \to \mathbb{R} \) are the standard coordinates on \( \mathbb{R}^d \). Because a chart is a homeomorphism, it has an inverse \( \varphi^{-1} \) and the inverse is also continuous.

In computational topology, we do not have continuous objects in space to study, but only a finite set of points with no topology associated. In order to study its topological properties, first a discretization of the space has to be done.

One of the most common discretizations in computational topology, especially common in the field of persistence homology, is done by using the structure of a simplicial complex \([6]\) \( \sigma \), i.e., the convex hull of \( p + 1 \) affinely independent points \( x_0, x_1, \ldots, x_p \in \mathbb{R}^d \). More intuitively, simplicial complexes are solid polyhedrons. A 0-simplex is a vertex, a 1-simplex is an edge, a 2-simplex a triangle, a 3-simplex a tetrahedron, and so forth. The difficulty with this approach is its computational cost. For points in 2-d and 3-d, there are known algorithms for computing the convex hull in \( O(n \cdot \log h) \), where \( n \) is the number of input points and \( h \) the number of points in the convex hull. However, for \( d > 3 \), the time for computing the convex hull is \( O(n^{d/2}) \).

In our proposed approach, the discretization step happens after the initial clusters are obtained. To discretize the space within each cluster, we instead construct the Minimum Spanning Tree (MST) of its data points, which allows for an efficient multi-resolution analysis, as the MST needs to be computed only once for all distance values desired to be studied \([8]\).

After obtaining topological spaces of each cluster, we compute properties based on their connectivities in order to attain a characterization of the spaces. Currently, we compute two exponents, \( \gamma \), a disconnectedness index and \( \delta \), a discreteness index \([8]\), detailed later. Next, clusters are compared in the space of topological features and partitioned into equivalence classes, which allow us to extract knowledge from the structure found.

This paper is organized as follows. In the next section, we present related work that has been developed bridging computational topology and clustering. Section III presents our proposed approach, explaining each step of the methodology. Next, in Section IV, our initial experiments are reported, using synthetic and real data sets. Finally, in Section V, we present our conclusions and prospects for future work.

II. RELATED WORK

As far as we are aware, no other work has investigated the analysis of clusters according to their topological features, however we present here recent work that has employed topological methods along with clustering techniques to analyze data sets.

Chazal et al. \([9]\) proposed a persistence-based clustering scheme that combines a mode-seeking phase with a cluster merging phase in the density map. The first step, mode-seeking, consists in finding local peaks of the density function to use as cluster centers. Next, the authors propose to merge clusters. This is done through a persistence diagram, in which points that lie far from the diagonal serve as basis for merging. The difference between their approach and ours is that theirs uses topological features of the data set and not of each cluster, aiming to find only spatially close points.

Johansson et al. \([10]\) introduced a framework for clustering periodic, quasi-periodic and recurrent dynamical systems based on topological features detected with persistent cohomology. The authors use an embedding theorem to reconstruct time-dependent signals in the phase space. By using Vietoris-Rips simplicial complexes, a filtration is produced, which reflects the topology at different scales. Next, a persistence diagram is constructed. Circular coordinates of high-relevance 1-cocycles are then computed, which allow behavior modeling: periodic systems correspond to circles, quasiperiodic to \( k \)-dimensional torus and recurrent to a bouquet of circles, such as in the Lorenz system. Their work is similar to ours in the premise that topological features correspond to meaningful properties of the mathematical objects being studied, which in our cases are clusters.

Lum et al. \([11]\) proposed a methodology to apply topological methods to study high dimensional data sets. The method takes as input one or more filter functions, a resolution parameter and a percentage overlap parameter. The resolution determines a set of intervals with a uniform overlap of their lengths. The filter is used to determine in which of the intervals a data point falls in. The next step is to use a clustering algorithm on each interval. A network is then built, such that the nodes are the partial clusters found in each bin and two nodes are connected with an edge if they have at least one data point in common. The constructed network permits the identification of subgroups of interest. This is the work most similar to ours, as it focuses on comparing meaningful clusters, found according to topological methods. We will see in the next section how our methodology differs from theirs, mainly in that it has a more principled approach to extracting topological features.

III. PROPOSED APPROACH FOR TOPOLOGICAL CLUSTERING

Our proposed approach to finding topologically-similar clusters is composed of three main steps, as outlined in Figure 1. Initially, we are given an unsupervised point cloud data set \( S \). In step 1, we execute a clustering algorithm to obtain an initial partition of the data set into clusters. In the example, three clusters are found, the red, yellow and blue ones. Next, in step 2, a discretization of each cluster is performed by obtaining its Minimum Spanning Tree. We regard the results of that step as a collection of \( k \) topological spaces \( X = \{ X_1, X_2, \ldots, X_k \} \), such that \( \bigcup_{i=1}^k X_i = S \) and \( X_i \cap X_j = \emptyset, \forall i \neq j \).

Finally, in step 3, each topological space found in the previous step is analyzed in terms of its topological properties, generating a 2-d space composed of the two exponents we compute: \( \gamma \) and \( \delta \). The analysis is concluded by clustering the projected points in the topological feature space, which then allow us to interpret the results. In the example of Figure 1, despite the blue and yellow clusters being spatially closer and representing the same geometrical figure, our proposed approach identifies that the blue cluster shares a topological
property with the red one, namely that the points seem to surround a hole in the middle, while the yellow cluster does not. In traditional topology, we would say that the red and blue clusters are homeomorphic to each other, in the same sense that a solid cube and a sphere are, or a tea cup and a torus [6]. Thus, in the final clustering phase, that identifies equivalence classes, the red and blue clusters are joined together before the yellow one. In the following, we explain in detail how each step in our approach is performed.

Our analysis begins by submitting an unsupervised data set to a traditional clustering algorithm. As mentioned before, we chose DBSCAN [4] because it is able to find arbitrarily shaped and sized clusters while being resistant to noise. One of the difficulties in using DBSCAN, specially when visual inspection is not possible on high-dimensional data sets, is determining the values for its parameters; $\theta$, which defines the neighborhood for a core point, and $MinPts$, which defines the minimum number of points in the neighborhood of a core point. This difficulty can be met by evaluating the behavior of the distance from a point to its $i$-th nearest neighbor, or $i$-dist [3]. Points that belong to the same cluster have small $i$-dist if $i$ is not larger than the cluster size. For points not in a cluster, such as noise, the $i$-dist will likely be larger. Thus, by computing all $i$-dist, sorting them by increasing value for some $i$, and plotting the results, we can obtain a reasonable estimation for $\delta$ as the value in which a sharp change occurs, i.e., a knee in the graph. A suitable value for $MinPts$ is then the selected $i$ used to produce the plot.

The second step of our analysis involves a discretization of each cluster, in order to obtain connectivity information, and thus a topology. For that step, we construct the Minimum Spanning Tree of each cluster using Prim’s algorithm with a priority queue to speed up neighbor selection. We note the correspondence between each cluster endowed with this connectivity information and a topological space $\mathbb{X}$. Our intuition is that, by extracting features of each topological space, they can be clustered together into equivalence classes, which leads to the third step in our methodology.

The third step can be subdivided into two more. The first substep revolves around extracting topological features of each topological space identified in step 2. For this process, we compute two exponents, previously proposed by Robins [8], namely $\gamma$, a disconnectedness index and $\delta$, a discreteness index. We explain them in detail next.

To compute the exponents we first need to define quantities that they take as input. First, let $\epsilon$ be a distance value used to identify connected vertices on the MST. Two vertices are considered to be connected if their distance is less than $\epsilon$ or there is an $\epsilon$-chain connecting them, i.e., a path in which no consecutive points are farther apart than $\epsilon: x_1, \ldots, x_n$, such that $d(x_i, x_{i+1}) < \epsilon$ for $i = 1, \ldots, n$. A connected component of a point $x \in X$ is the largest connected subset of $X$ containing $x$. If the connected component of every point is only the point itself, then the set is totally disconnected.

A subset $A \subset X$ is an $\epsilon$-component if $A$ is $\epsilon$-connected and $d(A, X \setminus A) \geq \epsilon$. Given a value of $\epsilon$, we can decompose $X$ into the disjoint union of its $\epsilon$-components. We call the number of $\epsilon$-components as $C(\epsilon)$.

Another quantity we can extract, besides $C(\epsilon)$, is the diameter of the set. Given a resolution $\epsilon$, $D(\epsilon)$ indicates the set of diameters of the $\epsilon$-components. The diameter $D(\epsilon)$ is then defined as $D(\epsilon) = \max(D(\epsilon))$, i.e., the largest diameter considering all $\epsilon$-components. Note that $D(\epsilon)$ is a monotonic, non-increasing, non-negative function, so the limit at $\epsilon \to 0$ exists.

With these two quantities, we can proceed to define $\gamma$ and $\delta$. The first exponent, $\gamma$, is called the disconnectedness index, or component growth rate, and is computed according to Equation 1. A positive value of $\gamma$, i.e., a positive derivative or upwards trend, in the log-log graph of $1/\epsilon$ vs $C(\epsilon)$ implies that the set has infinitely many components, or, still in other terms, as the resolution gets smaller, the number of connected components increases.

$$\gamma = \lim_{\epsilon \to 0} \frac{\log C(\epsilon)}{\log(1/\epsilon)}$$  \hspace{1cm} (1)

The second exponent, $\delta$, is called the discreteness index and is computed according to Equation 2. If $\delta$ is positive, the set must be totally disconnected. $D(\epsilon)$ relates the size of $\epsilon$-components to the distance between them, such that $\delta$ measures the relative rate of decrease of component and gap sizes. When both $\gamma$ and $\delta$ are zero, the set is connected or has a finite number of components.
The practical ideas behind these exponents are that when ϵ is large, all points are ϵ-connected and thus $C(\epsilon) = 1$, whose diameter $D(\epsilon)$ is the maximum diameter of the data set. This persists until $\epsilon$ shrinks to the largest interpoint spacing, at which point $C(\epsilon) = 2$ and $D(\epsilon)$ shrinks to the largest diameter of the two subsets. When $\epsilon$ reaches the smallest interpoint spacing, every point is an $\epsilon$-connected component, thus $C(\epsilon) = I(\epsilon)$, which is the number of points in the data set and $D(\epsilon) = 0$.

$$\delta = \lim_{\epsilon \to 0} \frac{\log D(\epsilon)}{\log \epsilon}$$ (2)

To compute numerically the quantities and exponents discussed, we first build the MST. Afterwards, $C(\epsilon)$ is computed as just the number of edges with length greater than $\epsilon$ in the MST plus one. $D(\epsilon)$ is computed as the maximum diameter of all $\epsilon$-components found when edges greater than $\epsilon$ were removed from the MST. Finally, to compute $\gamma$ and $\epsilon$, we take the linear regression of their respective graphs and use the angular coefficients as estimates.

After the exponents are obtained, we can project their values for all clusters in the topological feature space of $\gamma$ vs $\delta$ and analyze how they cluster together. Our hypothesis is that, in various scenarios, topologically similar clusters can reveal meaningful properties they share in common. To conduct this analysis we can plot a similarity graph of the pairwise Euclidean distance and cluster the points using a hierarchical clustering algorithm, such as single linkage. By studying how points cluster in the topological feature space, we can derive equivalence classes. In the next section, we present initial experiments that corroborate our ability to find topologically-similar clusters and meaningful equivalence classes.

IV. Experiments

We present in this section two experiments on synthetic data sets which show that our proposed approach can identify topologically similar clusters, and one experiment, on a real-world data set, that shows the equivalence classes we derive are meaningful. The code was developed partly in Java and Python. For preprocessing steps we used the Weka 3.7.10 API [12]. The discretization step, computing the MST, and the exponents calculations, were coded by us in Java. The final single-linkage clusterings were coded in Python using scipy 0.13.1 [13].

In the first experiment, we generated a synthetic data set consisting of ellipsoid and rectangular objects, some filled with points and others surrounding a hole. By applying the strategy for estimating DBSCAN parameters, discussed in the previous section, we estimated $\theta = 0.0025$ and $MinPts = 3$. Figure 2 presents the results of the DBSCAN run, which found 6 clusters, as expected.

After the initial clusters were obtained by the DBSCAN run, we discretized each one of them by constructing their Minimum Spanning Trees. Next, the values of $C(\epsilon)$ and $D(\epsilon)$ were calculated for a range of 20 $\epsilon$ values from 0.1 to 2.0. The limits of the range can be reasonably chosen by looking at the min/max intra-cluster dissimilarities. Table I presents the results of the DBSCAN run, which found 6 clusters, as expected.

The second artificial data set we generated simulated clusters homeomorphic to a 2-d torus, along with clusters surrounding no holes. It is a harder problem than the first one, as clusters have different sizes and very different densities. Also, note that the two clusters surrounding a hole are very far apart from each other, and the upper left one is closer to a cluster not surrounding a hole. Proceeding with the same to one equivalence class, while 2, 4, 5 and 6 to another. This can also be seen in the dendrogram of Figure 3, produced by running single-linkage clustering on the projected points in the topological feature space.

Table I: Projections in the topological feature space of the clusters found for the first synthetic data set.

<table>
<thead>
<tr>
<th>Cluster</th>
<th>$\gamma$</th>
<th>$\delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.82</td>
<td>0.02</td>
</tr>
<tr>
<td>2</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>0.85</td>
<td>0.04</td>
</tr>
<tr>
<td>4</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>5</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>6</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

The second artificial data set we generated simulated clusters homeomorphic to a 2-d torus, along with clusters surrounding no holes. It is a harder problem than the first one, as clusters have different sizes and very different densities. Also, note that the two clusters surrounding a hole are very far apart from each other, and the upper left one is closer to a cluster not surrounding a hole. Proceeding with the same
strategy to estimate DBSCAN parameters, we selected $\theta = 0.1$ and $MinPts = 5$. Results of the DBSCAN run are illustrated in Figure 4, which highlight 4 clusters, as expected. After the discretization step, we computed $C(\epsilon)$ and $D(\epsilon)$ for 50 $\epsilon$ values ranging from 0.01 to 0.4. The estimated values of $\gamma$ and $\delta$ for each cluster are displayed on Table II. We can see that the two equivalence classes were correctly retrieved, as can also be confirmed by the dendrogram of Figure 5.

Table II: Projections in the topological feature space of the clusters found for the second synthetic data set.

<table>
<thead>
<tr>
<th>Cluster</th>
<th>$\gamma$</th>
<th>$\delta$</th>
<th>Major Class</th>
<th>Purity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.48</td>
<td>0.19</td>
<td>Neptune</td>
<td>1.0</td>
</tr>
<tr>
<td>2</td>
<td>0.13</td>
<td>0.004</td>
<td>Satan</td>
<td>1.0</td>
</tr>
<tr>
<td>3</td>
<td>0.11</td>
<td>0.002</td>
<td>Ipsweep</td>
<td>0.8</td>
</tr>
<tr>
<td>4</td>
<td>0.45</td>
<td>0.18</td>
<td>Smurf</td>
<td>0.76</td>
</tr>
</tbody>
</table>

The first two experiments confirmed the ability of our proposed approach to identify equivalence classes, i.e., topologically similar clusters. In the final experiment, we used a real-world data set from the KDD’99 network intrusion challenge [5]. The data set consists of highly-imbalanced supervised data with approximately five million records of TCP dump data. Connections are either labeled as normal or as an attack, which can be one of a set of twenty two. Because the data set is very large, we used only a sample for our experiment. This was done using the Resample filter from Weka with parameters $B = 0$, $S = 1$, $Z = 0.1$ and no replacement. We also removed nominal attributes from the data set by using Weka’s Remove filter with parameters $-R 2, 3, 4, 7, 12, 21, 22$. As a final preprocessing step, we normalized each attribute to be in the range $[0, 1]$. Note we did not do z-score normalization, but only changed the range of each attribute. The final compressed data set we used is available at https://sites.google.com/site/cassiomartini/publications/kdd99sampledfilterednormalized.arff.gz. In our subsampled data set, only points of the classes neptune (1091), satan (14), ipsweep (12), smurf (2762), pod (1), nmap (4), portsweep (9) and normal (1004) were present.

Using the DBSCAN parameter estimation strategy previously discussed, we estimated $\theta = 0.5$ and $MinPts = 6$. That step results in the discovery of nine clusters plus noise. It is important to note that for the topological analysis step we removed all points that were classified as noise by DBSCAN. Analyzing intra-cluster dissimilarities, we chose to run the topological step over fifty values of $\epsilon$ ranging from 0.01 to 0.4. The estimated values of $\gamma$ and $\delta$ for each cluster are displayed on Table III, which also indicates the majority class in each cluster and its purity, i.e., the fraction of points of the majority class over the total number of points in the cluster, e.g., 1.0 being 100% purity. We observe that most clusters are totally pure.

Table III: Projections in the topological feature space of the clusters found for the real-world network intrusion data set.

<table>
<thead>
<tr>
<th>Cluster</th>
<th>$\gamma$</th>
<th>$\delta$</th>
<th>Majority Class</th>
<th>Purity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.34</td>
<td>0.48</td>
<td>Neptune</td>
<td>1.0</td>
</tr>
<tr>
<td>2</td>
<td>0.96</td>
<td>1.22</td>
<td>Satan</td>
<td>1.0</td>
</tr>
<tr>
<td>3</td>
<td>0.07</td>
<td>0.26</td>
<td>Ipsweep</td>
<td>0.8</td>
</tr>
<tr>
<td>4</td>
<td>1.49</td>
<td>0.72</td>
<td>Smurf</td>
<td>0.76</td>
</tr>
<tr>
<td>5</td>
<td>0.91</td>
<td>1.42</td>
<td>Normal</td>
<td>1.0</td>
</tr>
<tr>
<td>6</td>
<td>1.45</td>
<td>1.24</td>
<td>Normal</td>
<td>1.0</td>
</tr>
<tr>
<td>7</td>
<td>1.00</td>
<td>1.54</td>
<td>Normal</td>
<td>1.0</td>
</tr>
<tr>
<td>8</td>
<td>1.24</td>
<td>1.33</td>
<td>Normal</td>
<td>1.0</td>
</tr>
<tr>
<td>9</td>
<td>1.08</td>
<td>0.45</td>
<td>Neptune</td>
<td>1.0</td>
</tr>
</tbody>
</table>

To facilitate identifying similarities among clusters, Figure 6 presents a graphical representation of the Euclidean dissimilarity matrix for the clusters in the topological feature space. What is most interesting to observe is that cluster formations in the topological feature space do indeed correspond to meaningful behaviors in the real-world data. Normal behavior was clustered together, while Neptune and Smurf attacks were also close together, which is interesting, since both are a type of Denial of Service (DoS) attack. We believe that the Ipsweep cluster was not close to any other because it is not really an active attack, but a passive scan. Also, the Satan type of attack was close to the normal behavior, but Satan is a tool that only evaluates vulnerabilities, not exploits them, thus it is conceivable it could have characteristics close to normal behavior. Figure 7 presents the single linkage clustering in
Figure 6: Dissimilarity matrix corresponding to the Euclidean distance between projected cluster points in the topological feature space for the real-world network intrusion data set. Darker tones indicate more similarity (smaller distance). The blue square indicates a block of similar normal (non-attack) behavior (clusters 5–8). The yellow row indicates that cluster 3 (IPSweep attack) did not cluster well with any other. The red squares indicate that cluster 1 (Neptune attack) was very topologically similar to another Neptune attack (cluster 9) and a Smurf attack (cluster 4), which is very interesting, as both belong to a Denial of Service Attack (DoS) category.

Figure 7: Single-linkage dendrogram for the projected points of the real-world network intrusion data set in the topological feature space.

V. CONCLUSIONS

We have presented in this paper an approach to detect topologically-similar clusters. Our hypothesis is that topologically-similar clusters may indicate meaningful relationships, not easily observed in complex data sets. Initial experiments with synthetic data sets have confirmed our ability to detect clusters with topologically-similar features. We have not considered data sets with overlapping clusters because the topology might be lost or changed in that situation, however we see as future work evaluating how much overlapping would be needed for losing topological distinction. An experiment using real-world data of a network intrusion challenge showed that topologically-similar clusters, found through our approach, had a correspondence with equivalence classes of network attacks present in the data. Those results support our initial hypothesis. As future work, we will explore the computation of more topological characteristics, besides the current exponents $\gamma$ and $\delta$. We will also make further experiments with real-world data to further corroborate our hypothesis that topologically-similar clusters indicate meaningful relationships.

ACKNOWLEDGMENT

This work has been funded by FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) under grant 2013/04453-0 and CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) under grant 303280/2011-5. We would also like to thank the anonymous reviewers who suggested improvements to the presentation of the paper.

REFERENCES