IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 29, 2020, accepted December 21, 2020, date of publication December 25, 2020,
date of current version January 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3047537

Finding Hierarchical Structures of Disordered Systems: An Application for Market Basket Analysis

MAURICIO A. VALLE[1] **AND GONZALO A. RUZ**[2,3], **(Member, IEEE)**
[1]Facultad de Economía y Negocios, Universidad Finis Terrae, Santiago 7501015, Chile
[2]Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez, Santiago 7941169, Chile
[3]Center of Applied Ecology and Sustainability (CAPES), Santiago 8331150, Chile

Corresponding author: Mauricio A. Valle (mvalle@uft.cl)

This work was supported by the Agencia Nacional de Investigación y Desarrollo (ANID)—Chile under Grant Fondecyt 11160072 (M.A.V.) and Grant Fondecyt 1180706 (G.A.R).

ABSTRACT Complex systems can be characterized by their level of order or disorder. An ordered system is related to the presence of system properties that are correlated with each other. For example, it has been found in crisis periods that the financial systems tend to be synchronized, and symmetry appears in financial assets' behavior. In retail, the collective purchasing behavior tends to be highly disorderly, with a diversity of correlation patterns appearing between the available market supply. In those cases, it is essential to understand the hierarchical structures underlying these systems. For the latter, community detection techniques have been developed to find similar behavior clusters according to some similarity measure. However, these techniques do not consider the inherent interactions between the multitude of system elements. This paper proposes and tests an approach that incorporates a hierarchical grouping process capable of dealing with complete weighted networks. Experiments show that the proposal is superior in terms of the ability to find minimal energy clusters. These minimum energy clusters are equivalent to system states (market baskets) with a higher probability of occurrence; therefore, they are interesting for marketing and promotion activities in retail environments.

INDEX TERMS Boltzmann machine, clustering, disordered systems, greedy, hierarchical, market basket.

I. INTRODUCTION

The modeling of complex systems using networks has allowed large progress in understanding the interrelationship between a system's components and the complex interactions between them. From a network approach, the network's nodes or vertices represent the system's elements, and the edges are the connections between the nodes. These edges may also contain information regarding the intensity and direction of the interaction between two nodes. For example, the correlation between two financial assets or the coupling energy level needed to activate two neurons.

When the weights of these networks' edges are assumed to be i.i.d. random variables with a continuous distribution, we face disordered systems [1]. These systems are challenging to analyze due to the complexity and multiplicity of interactions resulting in multiple optimal energy states. Consequently, finding a global optimal or optimal configuration is an arduous task. These optimization problems are prominent in spin glasses [2], folding proteins [3], and the well-known traveling salesman problem [4].

The theory of disordered systems is interdisciplinary since problems in biology, finance, neurosciences, and recently in retail could be viewed as disordered systems solved with similar methods. The resolution of these problems is related to combinatorial optimization, where it is necessary to minimize a cost function (energy or distance) on a broad set of variables [5], [6]. In general, these cost functions have many local optima, which makes finding a global solution non-trivial. Despite this issue's difficulty, one tries to study these optimization problems on average, that is, by describing typical system configurations [7].

A classic example of a disordered statistical physics system is the spin glasses, which have random ferromagnetic and antiferromagnetic couplings that lead to frustration and a rugged energy landscape [8]. These characteristics seem to

The associate editor coordinating the review of this manuscript and approving it for publication was Thomas Canhao Xu.

be found in various phenomena and give rise to hierarchical structures characterized by ultrametrics. In this sense, the minimum spanning tree (MST) as a combinatorial optimization problem has found applications in this kind of random systems because the ground state can be mapped directly on the MST problem [9]. The MST can be seen as a property in the sense that its geometry is unaltered to the distribution of the system disorder, and its energy can be found for a given graph topology [10]. The disordered systems can be represented as complete networks (networks with N nodes and $N(N-1)/2$ edges), where nodes or spins can adopt only two states ($s_i = -1$ or $s_i = +1$). Each node is connected to all the other network nodes through couplings that determine the nature of each node's interaction. In those cases, the probability of a system state $P(\mathbf{s})$ can be considered proportional to the energy of the system, i.e., $P(\mathbf{s}) \sim \exp(-H(\mathbf{s})/Z)$, where Z is the partition function and $H(\mathbf{s}) = -\sum_{ik} s_i s_k J_{ik}$ is the energy function of the system depending of the interaction between nodes. This value describes the macroscopic energy of the system for a particular state $\mathbf{s}$. Note that with $J_{ik} > 0$, the energy decreases, and therefore, the probability increases. Thus, the J_{ik} couplings are directly related to the probability of a system state occurrence. Consequently, for a particular state, all possible pairs of couplings involved must be taken into account. The couplings of the network could be represented as the coupling matrix $\mathbf{J}$, which is equivalent to the weighted adjacency matrix, and it is the fingerprint of the system. An appropriate transformation to the values of $\mathbf{J}$ in an equivalent of the distance between nodes allows us to apply clustering methods to discover the hierarchical structure.

As an example of these complex systems, we have financial markets where the time series of stock returns are not predictable and can be described by random processes. In these cases, it is necessary to understand groups of stocks traded in the market that appear to have similar behavior and distinguish them from others with different behaviors [11]. The synchronous correlation coefficient of the daily difference of logarithm of closure prices of stocks is computed among pairs of stocks to obtain portfolio stocks' taxonomy [11], [12]. The correlation coefficient matrix is used to detect the hierarchical organization present inside the portfolio of stocks traded. A distance function is used on the correlation to obtain a metric that meets the axioms of distance metrics [13], [14]. The distance matrix is then used to determine the MST that connects the N stocks of the portfolio with $N-1$ edges. Thus, from a complete network connecting every pair of stocks through an edge with distance weights, one obtains a tree that describes a taxonomy of great interest from the economic point of view. This procedure has been repeated in numerous studies that establish the relationships between microscopic market interactions and macroscopic economic states [15], [16], and retail-related studies [17]. Note that the original network's weights are the pairwise correlations, which only measure the linear connection between pairs of system elements. The correlations assume that all the observations are independent of each other. The latter may not be correct due to the multitude of interactions between different system components. Instead, we try precisely to consider these relationships using couplings inferred from Boltzmann machines. Unlike a correlation, these couplings indicate the chance that a pair of elements are in a particular state.

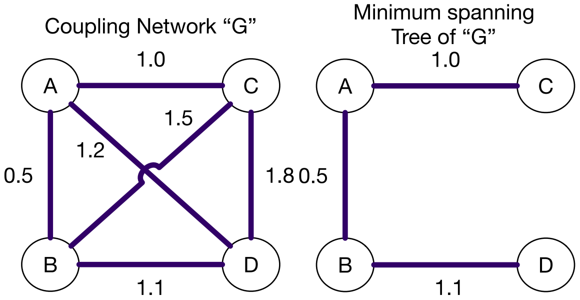

FIGURE 1. As an example, we have a simple coupling network G. It is a complete network with four nodes. The numbers on the edges indicate a distance between the node pair. The smaller the distance, the lower the bond energy between the two nodes. Thus, there is a higher probability of that state occurring. The minimum spanning tree of G is shown to the right with a minimal distance of 2.6. So, a hierarchical clustering using single linkage will be the following: the first cluster is $\{A, B\}$ at a distance of 0.5. Next, $\{A, B, C\}$ at a distance of 1, and finally, $\{A, B, C, D\}$ at a distance of 1.1. However, when we form the cluster $\{A, B, C\}$ the total distance-coupling involved is 3 (0.5 + 1.5 + 1.0). If we want to minimize the coupling energy, a better solution would be the cluster: $\{A, B, D\}$ at a distance of 2.8 (0.5 + 1.1 + 1.2), and finally $\{A, B, D, C\}$ at a distance of 7.4. Of course, the first and final clusters are the same in this example; however, the cluster $\{A, B, D\}$ has lower energy than the cluster $\{A, B, D\}$, so the first one represents a better representation of a cluster because it has a higher chance of occurrence.

The clustering structure has been studied using inferred couplings with Boltzmann Machines [18], [19]. Similar to the above, the related industry structures are based on MSTs, revealing industry sector clustering as a connected subset of the tree. In these cases, the MST has successfully revealed industrial groups of homogeneous behavior, but this does not necessarily reveal groups of system *states* that have a greater chance of behaving similarly. By definition, the resulting MST is derived from the Prim algorithm, which connects nodes with minimal energy (or minimal distance). In this process, the effect of other involved edges or couplings is discarded. Figure 1 shows the problem.

Market basket analysis aims to discover, among hundreds of thousands or millions of records, customer purchase patterns that can be useful for establishing promotional actions and bidding campaigns to maximize sales. In this area, the workhorse for discovering groups of products that occur together in a transaction set are the association rules (ARs) [20]. ARs are local-type models; that is, they look at parts of a data set (for example, a subset of variables or observations) to find rule-based patterns for a restricted set of binary variables. They are simply rules for joint occurrence between two item sets. In other words, the rules do not consider how a dependency between two items changes if the relationship between one of those items and another that is not in that relationship changes. However, other alternatives have been developed recently that look for global models. Thousand of customers' aggregate purchasing behavior has been studied, understood as a complex system of the market offer elements. For example, the hierarchical tree structure of the MST's subdominant ultrametric distances reveals the strongest dependencies between products of the same category, facilitating the retail manager, the proposal of different offers and promotions [17], [21], [22]. Again, there is a complete network of correlations computed from transactional data that indicate the presence and no-presence of various items or products in the market baskets of hundreds of thousands of customers. Then, the MST is found to reveal taxonomic structures. The problem with this approach is that it does not consider the possible effects of other interactions between *active* nodes, resulting in groups of items or products that do not reflect homogeneous groups of items with an activation behavior (simultaneous presence in the market basket). Our proposal is of a global nature; that is, we evaluate the joint distribution of the items, considering them as elements of a multi-state collective purchasing system, and offers parsiomonic groups of items that are more likely to occur.

The paper is organized as follows. A proposal of a coupling-based measure that considers all possible interactions for a potential cluster is presented. This measurement is the coupling energy for a particular system configuration. This is presented in the methods section. Based on the minimization of coupling energy, we study hierarchical structures of synthetic disordered systems and compare our algorithm's performance with two other well-known alternatives. This is presented in the results section and evaluated in the discussion section.

II. BACKGROUND

In this section, we define the problem by starting by describing a disordered Ising system. Although our proposal can be applied to any network with weights, with the only condition that the weights must be positive and greater than zero, we focus the development on networks that represent disordered systems, in which interactions must be inferred from observed states of the system. These allow us to establish a coherent method from the inference of network weights or couplings to discover the system's underlying hierarchy.

A. DESCRIBING THE PROBABILITY DISTRIBUTION OF STATES OF THE SYSTEM

The observation of a system reveals information about the states that manifest over some time. After gathering enough information, we can eventually find an empirical distribution of the realization of these states. Our interest is in finding a statistical model of this distribution. In these cases, the pairwise models are attractive because they can fit well with reasonable amounts of data [23].

Let be a vector $\mathbf{s} = (s_1, s_2, \ldots.., s_N)$, which denotes a state of the system, which in our case, represents a market basket with N items as part of the market offer. The component s_i can take $+1$ o -1 values depending on whether the i item is present in the basket. Let be $p(\mathbf{s})$ the probability of state $\mathbf{s}$ occurring. The probability $p(\mathbf{s})$ that the system is in such a micro-state, can be found by maximizing the Shannon information entropy $S = -\sum_{\mathbf{s}} p\log_2 p$ subject to the restriction that $p(\mathbf{s})$ is normalized, and that the first moments $\langle s_i \rangle$ and the second moments $\langle s_i s_k \rangle$ are equal to those observed from the sample [19]. The solution to this optimization problem results in the Boltzmann distribution:

$$p(\mathbf{s}) \sim \frac{e^{-H(\mathbf{s})}}{Z} \tag{1}$$

where $Z = \sum_i h_i s_i - \sum_{\mathbf{s}} e^{-H(\mathbf{s})}$ is the partition function, and $H(\mathbf{s}) = -\sum_{ik} s_i h_i - 0.5 \sum_{ik} J_{ik} s_i s_k$ is the energy function of the interaction between two particles i and k of the system. This value describes the energy of the system for a particular $\mathbf{s}$ state. The J_{ik} couplings represent the interaction between both particles (for example, the market offer elements, SKUs, or aggregated levels of categories). When $J_{ik} > 0$, there is a ferromagnetic interaction, which promotes that the spin i and j are aligned in the same direction. In other words, an item i and k tend to be present or absent simultaneously in a market basket. When $J_{ik} < 0$, the relationship is antiferromagnetic, where spin i and k tend to be aligned in opposite directions. That is, while one item is present or active in a market basket, the other is not. Thus, the magnitude and sign of each element J_{ik} in the matrix $\mathbf{J}$, defines a measure of the strength of the interaction between all pairs of items.

B. THE NETWORK OF COUPLINGS

The coupling matrix $\mathbf{J}$ is equivalent to the weighted adjacency matrix. This matrix defines the network of couplings. Since each system element eventually interacts with each of the other system elements, the network is complete. The h_i parameters, or fields, can be considered external agents' effect on the system, which influences only the spin i [19].

Without loss of generality, the coupling matrix could be correlations or some other measure of interaction. However, it makes more sense to talk about couplings because they have information about the energy needed to excite a pair of spins (nodes). Consequently, it is related to the probability of finding a particular state between a pair of nodes. In the following, we will continue talking about a matrix of couplings.

The network of couplings G is a collection of nodes $V = \{v_1, v_2, \ldots, v_N\}$ and a collection of edges E that connects the vertices, so $G = (V, E)$ with $E = \mid V \mid \times \mid V \mid$, is a complete network. An edge $e \in E$ connect a pair of vertices (v_i, v_k) with a coupling $J_{ik} \in \mathbf{J}$.

The graph G could have connections with positive and negative weights, denoting the variety of relationships between the nodes of the system giving rise to reinforcement and frustrations, similar to what happens in a spin-glass, where the couplings can be considered as a random variables that do not change in time, i.e., a quenched disorder. Couplings can adopt any positive or negative value, so to simplify the analysis, to the coupling matrix $\mathbf{J}$ is applied a nonlinear monotonic transformation, so that $d_{ik} = (\max(\mathbf{J}) - J_{ik})^{1/2}$. Thus, a positive interaction suggests a short distance between nodes. A negative one suggests a long distance. The result is an array **D** that plays the role of proxy for the energy couplings. This transformation is monotonic and does not alter the original order of the couplings.

C. MINIMUM BARRIER ENERGY

The minimum spanning tree (MST) can be considered a unique signature for each graph in random graphs. The MST energy for disordered systems is related to a scaling distribution, universal for all random graphs, and is not altered by the existing level of disorder [10]. Since the MST represents a network's skeleton in which all nodes can be traversed at minimum cost (minimum energy), it has been an essential part of hierarchical clustering models.

The MST $T(G, E^*)$ is a subgraph of G that has the same nodes as G, and it has a subset of edges $E^* \subseteq E(G)$ such that all the nodes are joined with the minimum distance, that is,$\sum_{i,k \in T(V,E^*)} d_{ik}$ is minimum, without loops which configures a tree with N nodes and $N-1$ edges. In disordered systems, this tree is equivalent to the path of minimum energy and, consequently, the one that offers the least energy barrier to move through the entire G network. These distances are the ones that have been used to find the hierarchical structures in previous studies.

D. COUPLING ENERGY

Each $v \in V$ has a state s that takes values -1 or $+1$, so that for any node i, it can adopt $s_i = -1$ (the item i not present on the market basket) or $s_i = 1$ (the item i is present on the market basket). The state of the G network system can be represented by a vector $S = \{s_1, s_2, \ldots, s_N\}$. We are interested in finding communities of nodes that have a high probability of simultaneous occurrence, which is equivalent to having a set of nodes or a community C, defined as $C = v' \subseteq V | s(\{v'\}) = 1$.

From the function $H(\mathbf{s})$, we are interested in coupling energy. Given the $G(V, E)$ network, with its corresponding coupling matrix **J**, the coupling energy of the system for a given state S is:

$$U(S) = -\sum_{\{i,k\} \in G(V,E)} J_{ik} s_i s_k, \quad J_{ik} \in \mathbf{J} \tag{2}$$

More negative values of this energy indicate a stable state of S or a greater probability of occurrence. Therefore, it is of interest to look for combinations of S that produce the minimum coupling energy. We can adapt the Equation 2 to indicate the coupling energy of a community C:

$$U(G, C) = -\sum_{\{i,k\} \in C} J_{ik}, \quad J_{ik} \in \mathbf{J} \tag{3}$$

The states s_i and s_k are not necessary to be included. They are already in the condition of $+1$. The same equation 3, can be expressed in terms of distance:

$$U^*(G, C) = \sum_{\{i,k\} \in C} d_{ik}, \quad d_{ik} \in \mathbf{D} \tag{4}$$

E. LEARNING THE COUPLINGS

The problem of inferring **J** (containing $N(N-1)/2$ couplings) is not trivial because, for a system with N spins, there are 2^N possible system phase states. This amount is usually much greater than the number of observed states, so an inference method is needed to solve the optimization problem with fewer observations than the number of possible system states. An alternative for this is the use of Boltzmann machines [21], [24] without hidden units. Unlike a restricted Boltzmann machine used as a feature extractor to learn features from data [25] or to predict links in dynamic networks [26], an unrestricted Boltzmann machine has no hidden neurons. All neurons are connected between them [24]. The learning process consists of finding the weights or couplings that define the Boltzmann distribution in Equation 1 using a set of system state vectors.

The algorithm starts with an initial guess of the couplings, which are used to carry out a Metropolis-Hasting simulation phase using a distribution with those parameters. Then, in the negative learning phase, the first and second moments are calculated from that simulation. The couplings are updated according to the process of contrastive divergence [24] following:

$$\begin{aligned} J^t_{ik} &= J^{t-1}_{ik} + \nu\big(\langle s_i s_k\rangle_{\text{data}} - \langle s_i s_k\rangle_{\text{model}}\big) \\ h^t_i &= h^{t-1}_i + \nu\big(\langle s_i\rangle_{\text{data}} - \langle s_i\rangle_{\text{model}}\big) \end{aligned} \tag{5}$$

where t indicates the epoch of iteration and ν is the learning rate.

The process is repeated as many times as necessary until the difference between the old and updated parameters is less than or equal to a predetermined value. The process gradually results in the distribution of the equation 1 in which the first moments of the spins (one-body averages) and the second moments (the pairwise connections) between those of the Boltzmann machine and the empiric ones tend to equality:

$$\begin{aligned} \langle s_i\rangle_{model} &= \langle s^*_i\rangle_{data} \\ \langle s_i s_k\rangle_{model} &= \langle s^*_i s^*_k\rangle_{data} \end{aligned} \tag{6}$$

As a result of the inference process, the coupling matrix **J** is obtained.

III. PROPOSED METHOD

As previously indicated, market basket analysis consists of looking for buying patterns that arise after many customer purchases. Association rules, for example, have played a vital role in this activity to find relationships between groups of items. The process of generating these association rules between products is local, which means that a rule does not consider other items' effects. In other words, it does not consider how a dependency between two items changes, if the relationship between one of those items and another that is not in that relationship changes. Instead, our proposal is global; that is, we evaluate the items' joint distribution, considering them as elements of a multi-state collective purchasing system. To better understand the process and analysis that we

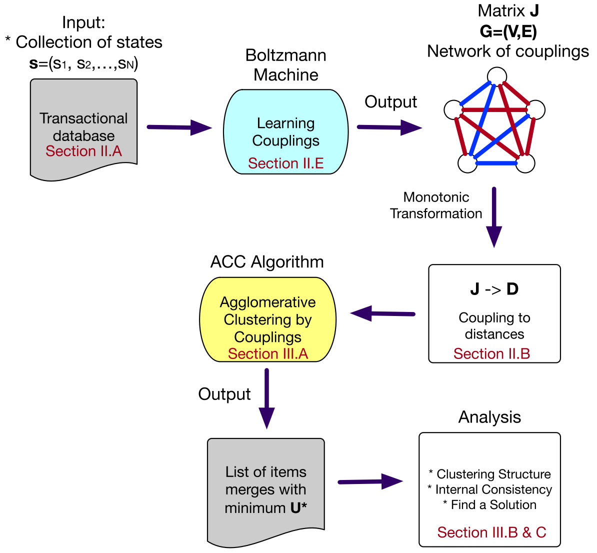

FIGURE 2. Global representation of the analysis process of hierarchical grouping, using the Agglomerative Clustering by the Couplings algorithm. The related section is indicated at each step.

Algorithm 1 Agglomerative Clustering by Couplings: ACC

```
Input
    $C = \{v \in V(G, E)\}$
    D    Distances of $\{E\}$
    $w =| V |$
procedure ComputeEnergy($C_i$, $C_j$)
    Compute $U^*(G, C_i \cup C_j)$
    return $U^*$
end procedure
procedure MakeJoin($C_i$, $C_j$)
    $w := w - 1$
    return $C - C_i - C_j + (C_i \cup C_j)$
end procedure
Main procedure:
while $w \neq 0$ do
    $\forall(C_i, C_j)$, ComputeEnergy($C_i$, $C_j$)
    Find the minimum $U^*(G, C_i^* \cup C_j^*)$
    $C_k$ := MakeJoin($C_i^*$, $C_j^*$)
end while
```

raised in this work, Figure 2 shows a flowchart that includes each step. The process considers two essential tasks: The first, relating to the problem of inference or learning of couplings J (Section II-E), and the second relating to agglomerative clustering. This section focuses precisely on this last step and on the analysis needed to validate the algorithm's solution and find a solution.

A. AGGLOMERATIVE CLUSTERING BY COUPLINGS (ACC) ALGORITHM

This section defines a greedy approach to find iteratively groups of nodes that minimize coupling energy U^*. The procedure for finding clusters and their hierarchies is based on a greedy approach. This approach provides a quick solution that can be easily implemented [27].

The hierarchical clustering is bottom-up. Communities of nodes are formed at each step $C = \{C_1, \ldots, C_n\}$. In the beginning each node $v \in V$ is a single community. Next, a pair of clusters are merged to form a cluster C_i, and the process is repeated until all nodes in V belongs to some cluster. Each cluster is made up of nodes in V of the graph G such that $C_j \cap C_i = \emptyset$, $(j \neq i)$, $\tilde{\bigcup}_{C_i \in C}^{n} C_i = V$. Each step of the algorithm is an optimization problem in that a decision must be taken: which pair of clusters to merge, such that the coupling energy $U^*(G, C_i, C_j)$ be minimum, i.e., form a new cluster $C_k = C_i \cup C_j$, such that:

$$C_k = \mathop{\arg\min}_{C_i, C_j \in C} U^*(G, C_i \cup C_j) \tag{7}$$

Thus, the algorithm selects the pair of clusters that provide the minimum coupling energy to form the C_k cluster.

Similar to the classic hierarchical clustering analysis, our algorithm reveals the hierarchical structure. The merge distances between clusters indicate the chance that nodes in those clusters are all together activated (or in the same market basket). The merging process can be visualized in a dendrogram representing the order in which the nodes are grouped into communities, capturing the entire hierarchical clustering process. An advantageous aspect of this methodology is that it does not violate the monotonic assumption, which indicates that the energy of successive merges is always increasing. This way, we are sure that we chose the best merge in each step of the algorithm.

An important issue to consider is that as the size of the clusters increases (more products in the state $s_i = 1$), the energy $U^*(G, C)$ needed for this to occur also increases. In this way, independently of the merge path of clusters that the algorithm finds, it will always end up with a cluster that merges all the system elements with single final coupling energy. Therefore, for a given problem, the algorithm may initially find very low energy clusters, but it will invariably end up with high energy clusters. Consequently, what we can modify is the trajectory of product aggregation. The ACC Algorithm always provides a single path that minimizes the coupling energy at each step (see Algorithm 1).

B. CLUSTERING STRUCTURE

To compare the structures that are formed with each algorithm, we use the Agglomerative clustering coefficient. This coefficient measures the amount of clustering structure found. Values closer to 1 indicate a balanced clustering structure, and values closer to 0 indicate no clustering structure. The coefficient achieved for network G, C_G was computed as:

$$C_G = \frac{1}{N-1} \sum_{i=1}^{N-1} \left(1 - \frac{U_i^*}{\max(U_i^*)}\right) \tag{8}$$

where the coupling distances U^* are taken for each iteration and N is the number of nodes of the coupling network.

C. INTERNAL CONSISTENCY

Each algorithm's result is a record of cluster merges that define in itself the structure achieved by the iterative clustering process. It is then necessary to look for a solution that indicates how many clusters there are or, in other words, to find the cutoff distance that defines the number of existing clusters. This requires some measurement that assesses the consistency of the clusters given this cutting distance.

In our case, we are not dealing with classical distance metrics calculated from a set of several variables (for example, the Euclidean or Manhattan distance), but with energy proxies. It is worth remembering that a high coupling distance means an anti-ferromagnetic relationship exists between the system elements. In contrast, a low coupling distance means that the system's pair of elements has a ferromagnetic relationship. Thus, we do not necessarily look for the element i to have a high distance with other items outside its cluster. This is equivalent to saying that we want the element i to have an anti-ferromagnetic relationship with all the other system components outside the cluster. It is not a meaningful task in frustrated systems where there are as many ferromagnetic as anti-ferromagnetic relationships. In this sense, it is not convenient to directly apply classical internal validity measures of clustering like the Davies-Bouldin index [28] and the Silhouette Width [29].

Thus, it is preferable to build an ad-hoc consistency measure for this application. Following Equation 4, and simplifying the notation, we will say that $U^*(C)$ is the coupling distance of the cluster C. It represents the level of internal coupling energy of the cluster.

On the other hand, we define the $b(i, C_k)$ as the coupling distance of the cluster C_k, $i \notin C_k$, that would be obtained if to this cluster is added the element i. That is to say:

$$b(i, C_k) = U^*(C_k) + \sum d_{ij} \quad \forall j \in C_k \tag{9}$$

This value represents the (energetic) convenience of keeping the i element in the cluster to which it belongs. We compute the coefficient $s_1(i, C_k) = \frac{b(i,C_k)}{U^*(C_k)}$. As $b(i, C_k) > U^*(C_k)$, then $s_1(i, C_k) > 1$. We are interested that by incorporating node i in the C_k cluster, the coupling distance of the cluster increases significantly, which means that it is not convenient that node i is in C_k. Consequently we are interested that s_1 be as large as possible.

If we calculate the average of all the $s_1(i, C_k)$ for all the existing clusters k (except the one where the i node lives) for all nodes in the system, S_1, we obtain a cohesiveness indicator of the clusters when the i element is incorporated as a solution in other clusters. The idea is that S_1 be as large as possible.

Now, we define $q(i)$ as the coupling energy of the cluster C when node i leaves the cluster, that is to say,

$$q(i, C) = U^*(C) - \sum d_{ij} \quad \forall i, j \in C \tag{10}$$

It represents how much coupling energy of the cluster C elements is affected when i is not present. We computed the coefficient $s_2(i, C) = \frac{q(i)}{U^*(C)}$. If the i node is very well coupled to the cluster, then its removal from the cluster will not affect much the coupling energy of the cluster, that is, $q(i, C) \approx U^*(C)$. Therefore, we are interested that $s_2 \approx 1$, which would indicate that the node i is correctly belonging to the cluster C. In other words, we are interested in S_1 being as large as possible. We compute the average of all s_2 values for each cluster and node in the system: S_2.

Note that for small cutoff distances, we will have many clusters with few nodes in each, so the values of s_1 will be higher compared to larger distances. Adding a node to a cluster with few members will have more relative importance than a cluster with many nodes. Thus, as the cutoff distance increases, the S_1 indicator decreases. On the other hand, with many clusters and few nodes, s_2 will be close to zero. Removing a node from a cluster with few nodes will have a greater impact than doing it in a cluster with many. Thus, as the cutting distance increases, the S_2 indicator increases.

We look for a solution that maximizes at the same time, S_1 and S_2. To get a global indicator of the consistency, it is possible to use a weighted function:

$$\begin{aligned} S(h) = \alpha\, (S_1(h) - \max(s_1))^2 \\ +(1-\alpha)\, (S_2(h) - \max(s_2))^2 \end{aligned} \tag{11}$$

where S_1 and S_2 depends on cutting distance h. We try to minimize $S(h)$. The α-value is the importance we give to the S_1 coefficient.

D. SPACE AND TIME COMPLEXITY

The ACC algorithm structure is not, in essence, very different from a bottom-up or hierarchical agglomerative algorithm. All distances between every pair of nodes must be computed in at least $\mathcal{O}(n^2)$ because each distance must be examined at least once. Similar as in single-link clustering, each node starts being its own cluster, and then, a couple of clusters form another higher hierarchy cluster. The process requires to find smallest distance for each data point and keep this information in the next array. The process continues in $N - 1$ steps of merging, and the distance matrix is updated in $\mathcal{O}(n)$ [30]. The difference lies basically in the concept used as a measure of distance between nodes, which, in our case, is of coupling energy between nodes, and not only the distance between two nodes. The space complexity is $\mathcal{O}(n)$ because we need to store the distance matrix.

The Figure 3 shows how the ACC algorithm scales in time and memory. In the case of time, the simulations show that the behavior can approach well to a quadratic relationship. The memory usage, as expected, is linearly proportional to the network size, however, for sizes over 500 nodes, the demand for increased storage of the distance matrix and the sequence of merges, requires an increase in the rate of memory usage for each additional node of the network. There are other faster variants for Hierarchical clustering [31], which could be implemented in the future for the ACC algorithm in order to improve time and memory usage.

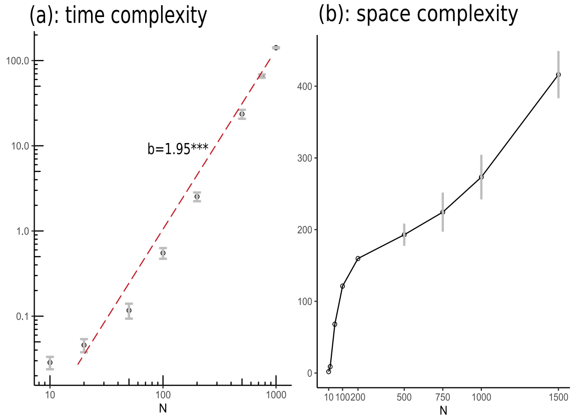

FIGURE 3. **(a): Log-Log plane of execution time of the ACC algorithm versis the size of the network. The line is an approximation of a linear fit to the data. The estimated coefficient was $\beta = 1.95$, significant at 0.1%. (b): Plot of memory usage of the ACC algorithm versus the size of the network. The simulations were performed on a PC-Desktop with MS Windows 10, processor Intel i7-6800K 6-core, 3.4Hz CPU, RAM: 16 GBytes.**

IV. EXPERIMENTS

A. SIMULATIONS

To compare the proposed algorithm's results, we started by simulating different coupling matrices **J** of different size and distribution. The results are measured in terms of the coupling distance $U^*(G, C)$ of the clusters that are being merged in each step. It should not be considered $U^*(G, C)$ as a performance measure between agglomerative algorithms, but a comparison measure that allows us to verify that our proposal produces clusters of elements with lower coupling energy about other algorithms. We consider two other agglomerative algorithms that allow finding structural hierarchies:

1) A hierarchical clustering using single linkage. It is the equivalent to the MST [32], which has been the standard for determining the hierarchical structure of financial systems.
2) A hierarchical clustering using modularity [33], [34] (MOD). The algorithm is also a greedy approach to find in successive iterations, the best combination of nodes that maximize modularity. The algorithm is agglomerative because, at each step, two clusters merge. The merge is decided by optimizing the weighted modularity. This method is possibly one of the most widespread and successful in many network applications [11]–[13], [15], [16].

We carried out several simulations with a different number of nodes. Figure 4 shows the energy paths achieved by every algorithm over 100 coupling matrix simulations each one. The couplings matrices **J** were simulated from a normal distribution with mean 0 and variance 1. Each matrix is then transformed into a distance matrix, which is the input for each of the algorithms under test. The error bars and the points show the variations and averages of each iteration's coupling energy, respectively.

First, we see that the greedy ACC Algorithm (red curve) is superior to single-linkage (MST) (green curve) and modularity maximization algorithm (blue curve). In general, the ACC approach that explicitly seeks to minimize the coupling distance manages to create clusters of nodes that have a smaller coupling distance than the MST and the modularity maximization approach in almost all the steps. This is expected since both algorithms do not work minimizing the coupling energy. Second, the approximation using MST has drastic increases of coupling distances indicating that some clusters merge at high distances. This occurs because the merging criterion is strictly local-based only on the edges of the MST. Instead, the ACC Algorithm seems to find well-defined groups of nodes that, for our purposes, represent market baskets of products that tend to be present simultaneously. Observing the energy paths for the MST algorithm, it is observed that the mergers are made at a high coupling distance, and the chaining effect is present, i.e., individual nodes are joined to larger clusters, one by one. Thirdly, the maximum coupling energy corresponds to the energy required to activate all the network nodes simultaneously. All the algorithms must finish on that coupling distance.

To get an idea of the solutions found for each algorithm, Figure 4(**b**) shows the resulting dendrograms of a simulation with $N = 25$. The ACC approach not only succeeds in merging nodes at a lower distance compared to their competitors. It also seems that ACC Algorithm achieves a more balanced node grouping structure. The agglomerative clustering coefficient for the ACC algorithm, MST, and modularity were 0.925, 0.803, and 0.902, respectively. Simulations were also carried out using uniform distributions of couplings between -3 and $+3$. The results are similar to those shown in Figure 4(**a**).

Figure 5 shows the results of the simulations using normal distributions for **J**. The greedy ACC Algorithm outperforms the other two algorithms in terms of coupling distances and clustering structure, being the clustering by MST the worst in terms of structure. Interestingly, the ACC Algorithm offers very little variability in results, which indicates that the clustering structures achieved are similar for a given size of the network.

It can be seen that in all the algorithms, there seems to be a very slight improvement (increase) of the coefficient as the size of the network increases. This is due to the effect of the maximum coupling energy $\max(U^*)$, which becomes larger and larger as the size of the system increases.

B. REAL EXAMPLE

To show our proposal's usefulness, we used a sample of real transactional data (also in [35]). The sample comprises 42077 purchase transactions between January and December 2007 of a national grocery chain in Chile. The data comes from a brach of the grocery located in the Capital of Chile, Santiago. Each record indicates the customer's ID, the identification and quantity of each item purchased, and the category and subcategory to which each item belongs. We have concentrated on carrying out the analysis in the 25 most

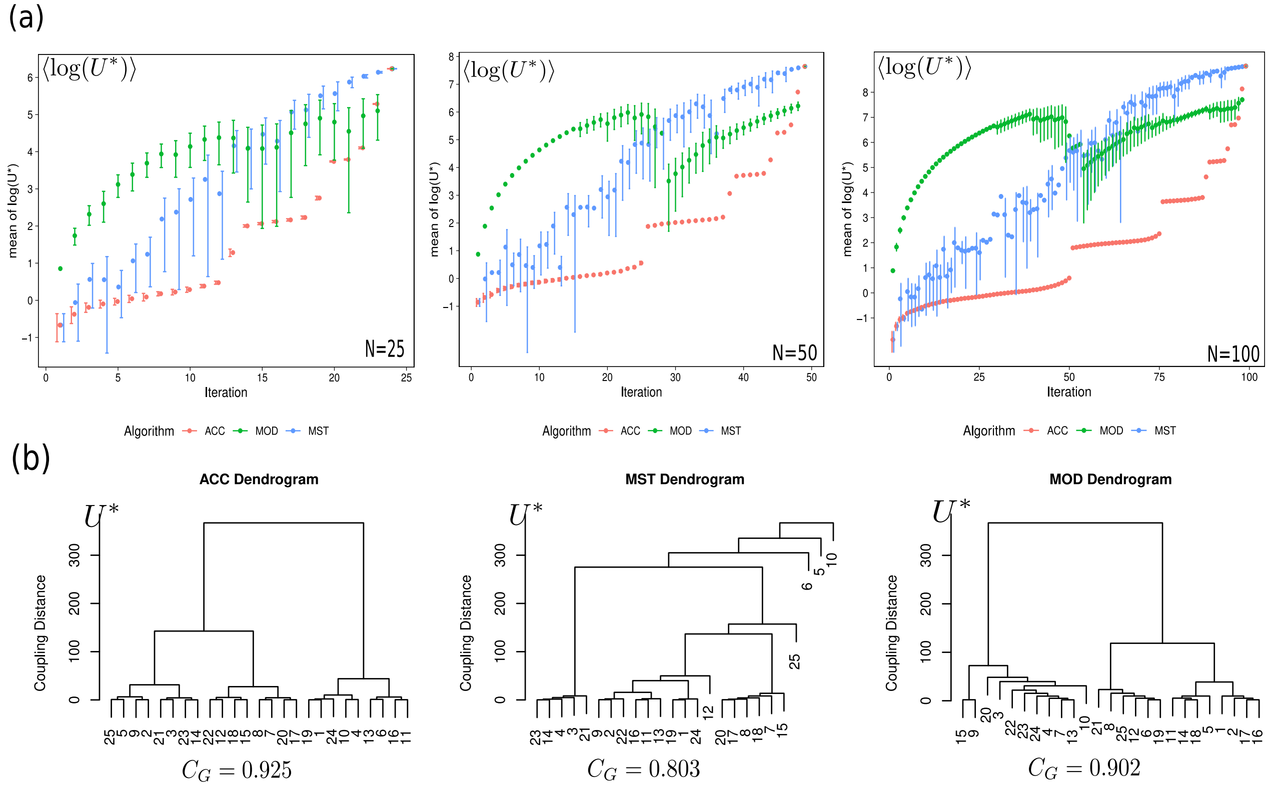

FIGURE 4. (a): Simulations of coupling matrices J with a size of 25, 50, and 100 nodes, as inputs to the MST, MOD, and ACC algorithms. There are 100 different random generations of J to calculate mean and standard deviations of every iteration for every size. The Figure shows the energy log(U*) at which the algorithm merges the clusters. The use of the logarithm is only to improve the graphical representation of the merge energies. (b): Dendrograms for a simulation with N = 25. The C_G values at the bottom indicate the clustering coefficients for the clustering results of each algorithm.

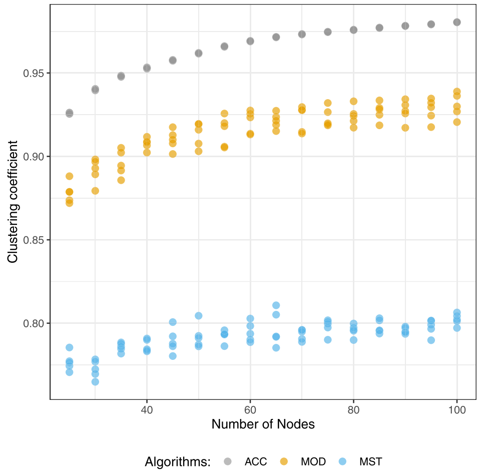

FIGURE 5. Agglomerative clustering coefficients (C_G) for different network sizes (N). For each size, 30 differents coupling matrices were simulated from a normal distribution with five differents means of couplings and standard deviation 1: $\langle J \rangle = -2$ (predominantly anti-ferromagnetic system with negative couplings), −1, 0, +1 and +2 (predominantly ferromagnetic system with positive couplings).

important subcategories of products in the branch. We represent each market basket as a vector S with length $N = 25$ with values of 1 and -1, indicating whether or not the subcategory to which the item belongs is present in the basket.

1) INFERENCE

The first step is to infer the $N(N-1)/2 = 300$ couplings between the $N = 25$ product subcategories. For this, we follow the same procedure used in [36] and described in the Previous Section. We use 400 steps of the Boltzmann learning with a decreasing learning rate of $\nu = 0.8$ and a decay of 0.0001. The means orientations and pair correlations were computed using the Metropolis-Hasting process with 40000 steps. In this way, we empirically assure the machine's convergence for the coupling parameters with an RMSE less than 10^{-3}.

At the end of the process, a test is made to verify that Equation 6 is met, that is, that the model reproduces the empirical data. This can be done by comparing the means and correlations of the actual data with those of the sample from the last Metropolis-Hasting process using the inferred couplings. If the empirical values of the spins orientations $\langle s_i \rangle$ and pairwise correlations $\langle s_i s_k \rangle$ and reproduced ones are the same, a scatterplot between them will show a perfect line with correlation $\rho = 1$ and a root mean square error $RMSE = 0$. The results indicate that for spin orientations and pairwise correlations, the correlation and RMS were 0.00058 ($\rho = 0.996$) and 0.00062 ($\rho = 0.998$) respectively.

The second step consists of transforming each coupling J_{ik} in equivalent distances d_{ik}, being the matrix $\mathbf{D}$ the input for

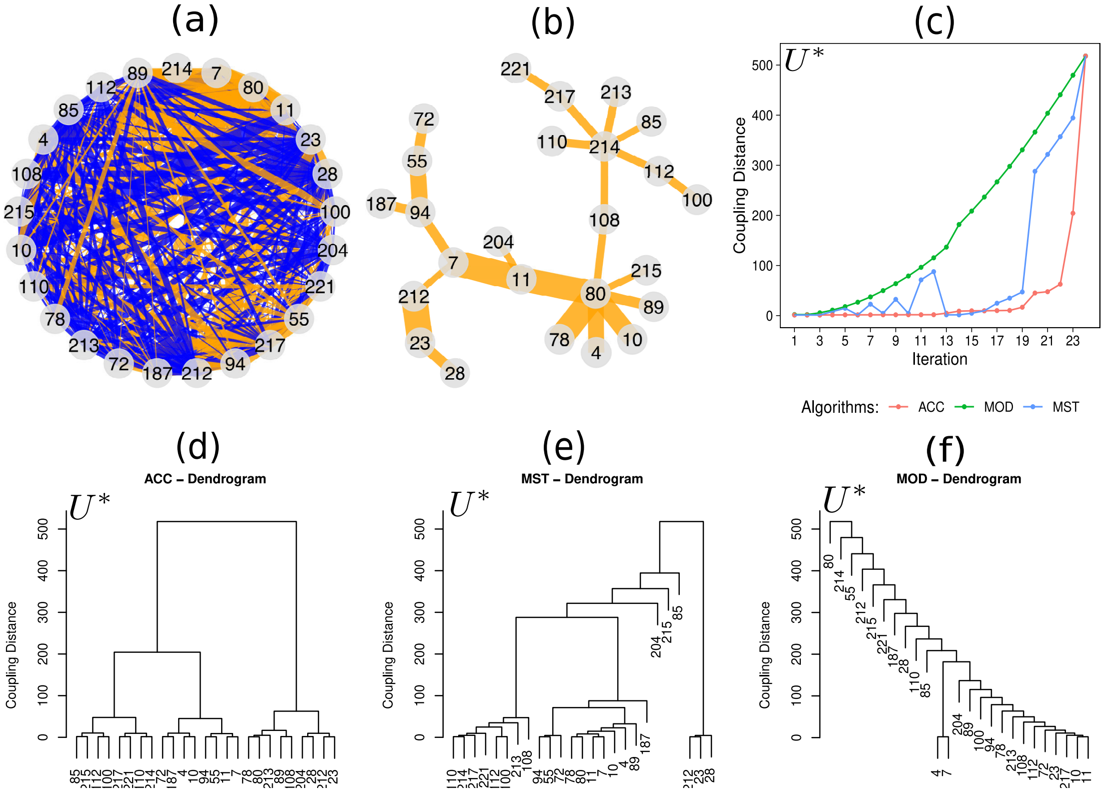

FIGURE 6. (a): The network of couplings. Orange edges are positive couplings, and green ones are negative couplings. (b): The corresponding MST of the network of coupling. The edges are the coupling-distance converted. The thicker the edge, the closer the pair of vertices are. The numbers in each vertex represent subcategories of items. For example, some of them are marigold seed oil (4), vegetable oil (7), rice grade 1 (10), rice grade 2 (11), family size coke (23), vitaminized noodle (80), flavored noodle (78), regular washing machine (94), packaged red wine (212), beaten yogurt (214). (c): Cluster merging paths for the real case dataset, using the ACC Algorithm (red line), the MST (light blue line), and weighted modularity-based clustering (green line). The y-axis represents the coupling distances U^* in each iteration of the process. (d), (e) and (f) are the dendrograms as a result of clustering results using ACC, MST and MOD algorithms.

the ACC Algorithm. These equivalent distances are also used as the input to the Prim algorithm to find the MST of the couplings network, which allows us to compare the achieved clustering with the MST traditional method.

Figure 6(**a**) shows the resulting network of couplings. The couplings' distribution follows a nearly-symmetrical pattern, with a coupling mean of 0.00613 (sd = 0.366). The maximum value is 1.409, and the minimum value is -1.257. 39.3% of the couplings (118) are positive, while the remaining 60.7% (182) is negative, which is typical of a frustrated interaction system [37] similar to what happens in the physical spin-glass [38].

Figure 6(**b**) it is shown the MST. The MST tree structure indicates the products' hierarchical structure, based on the minimum energy barriers, as we will see on the dendrograms. Figure 6(**c**) shows the cluster merge paths achieved for each algorithm. In the paths, each time there is a sudden drop in the clusters' coupling distance, it is because clusters with individual nodes are being formed. This happens a lot with the MST because the clusters' merge criterion is single-linkage using the MST adjacency matrix's coupling distances. The MST solution is a particular and restricted solution of the greedy algorithm because there are not many edges in the MST's coupling network's adjacency matrix. Thus, merges are limited only to the edges kept by the Prim algorithm to find the shortest distance (or coupling distance) among every node. When the path indicates a gradual and smooth increase in coupling distances, it indicates a chain effect, as in the case of modularity. In each iteration, a node is joined to a general cluster that increases in size. In the ACC algorithm, we see minimal increases in the distance, followed by sudden and significant increases in coupling distances. This indicates that the algorithm succeeds first in finding a variety of clusters with nodes at a very low merging distance and then merging groups of clusters, which inevitably must occur at more considerable distances. As seen before in the simulations, the ACC Algorithm is superior to MST and modularity in coupling distance. In most of the iterations, the greedy algorithm join clusters at a lower distance than other competitors. We see that the main difference between the clustering based on the MST and the ACC Algorithm is that the first one manages to find in the initial stages of the merge process, pairs of elements with very low coupling energy. At the same time, the MST tends to add elements to an already existing cluster from low coupling distance links. This is because the ACC Algorithm only merges elements

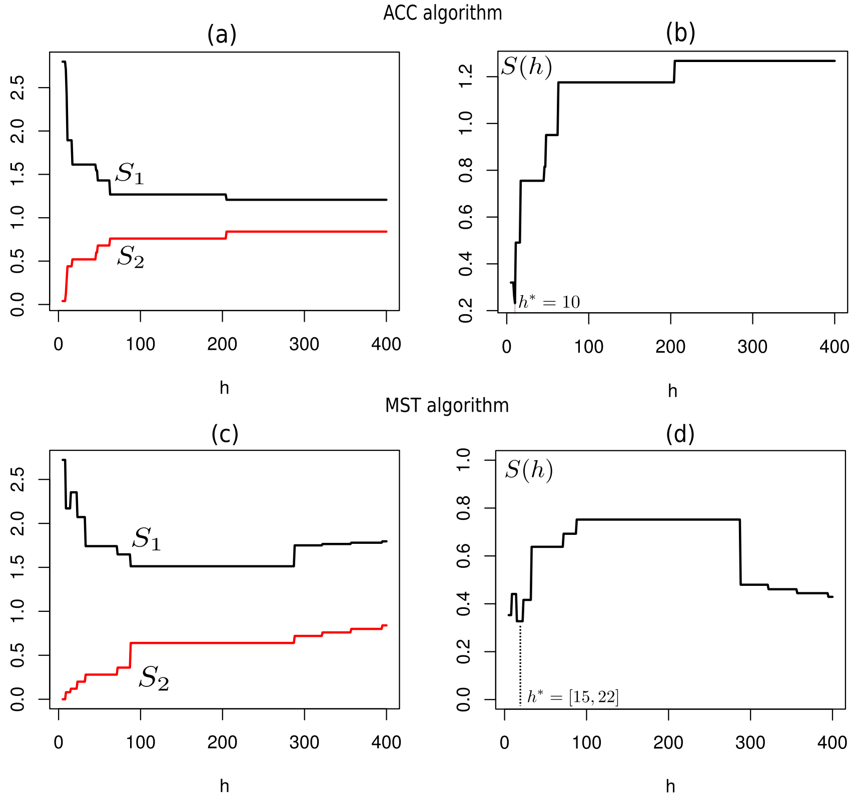

FIGURE 7. In (a) and (c) it is shown the different values of S_1 and S_2 as we increase the coupling distance cutoff h using the ACC algorithm and MST approach. In (b) and (d) it is shown the empirical $S(h)$.

into a new cluster when the coupling energy U^* of the new potential cluster is lower than other alternatives. In contrast, the MST only merges clusters searching at the lowest coupling distances d_{ij} available in the tree. For this example, in iteration 6, 13, 14, and 15, the MST manages to merge clusters at a very low distance, but the rest of the merges are made at a distance far above.

In terms of the clustering coefficient, the ACC Algorithm is superior to the rest. The coefficients are 0.922, 0.819, and 0.648 for the ACC Algorithm, MST, and modularity algorithms, respectively. In the case of modularity, the chain effect that occurs is clearly seen. In this case, each node is successively joined to the same cluster that grows in size, making it difficult to produce a clear clustering structure.

Figure 6(**d**), (**e**), and (**f**) shows the respective dendrograms. First, note that the tree achieved with the MST is the solution to the dendrogram of this algorithm. For example, items 80 (vitaminized noodle) and 78 (flavored noodle) form a cluster, then items 11 (rice grade 2), 7 (vegetable oil), 10 (rice grade 1), 4 (marigold seed oil), and 89 (canned mackerel) are added iteratively way. This group of subcategories forms the central cluster of the tree in Figure 6(**b**), being item 80, the root node of this bouquet of items.

Second, some clusters found the greedy algorithm with items that seem to match some MST clusters. For example, the terminal branch of the MST formed by the items 212 (packaged red wine), 23 (family size coke), and 28 (family size flavored sodas) is a cluster. These items are also present in the ACC Algorithm solution, including item 204 (full chicken nugget). This last item would never be part of this cluster in the MST because it does not allow a connection of item 204 to any of the other previous items. This edge is not part of the tree. However, in the ACC algorithm, item 204 is part of the cluster formed by these products because in terms of coupling distance U^*, they are very close to each other. This particular cluster of four items is merged at a coupling distance of $U^*(\{204, 28, 212.23\}) = 9.66$ in the ACC approach. In contrast, in the MST solution, these four items have managed to be part of the same cluster in the last iteration at the maximum coupling distance of 518.0.

The root node of the MST is subcategory 80 (vitaminized noodle). According to the MST, there is a cluster formed by at least 7 nodes (78, 80, 11, 7, 10, 4 and 89) (see dendrogram at Figure 6(**e**)). For example, let us suppose that the retail manager is interested in the pair of items 78 (flavored noodle), 80 (vitaminized noodle) and 11 (rice grade 2), 7 (vegetable oil), both of them at a very low coupling distance of 1.262 and 1.268 respectively. In the MST, these four items are merged at a distance of 8.14. However, this does not mean that this cluster of 4 subcategories is a frequent system state (or a set of products purchased simultaneously by customers). According to the ACC Algorithm, pairs 78,80 and 11,7 belong to

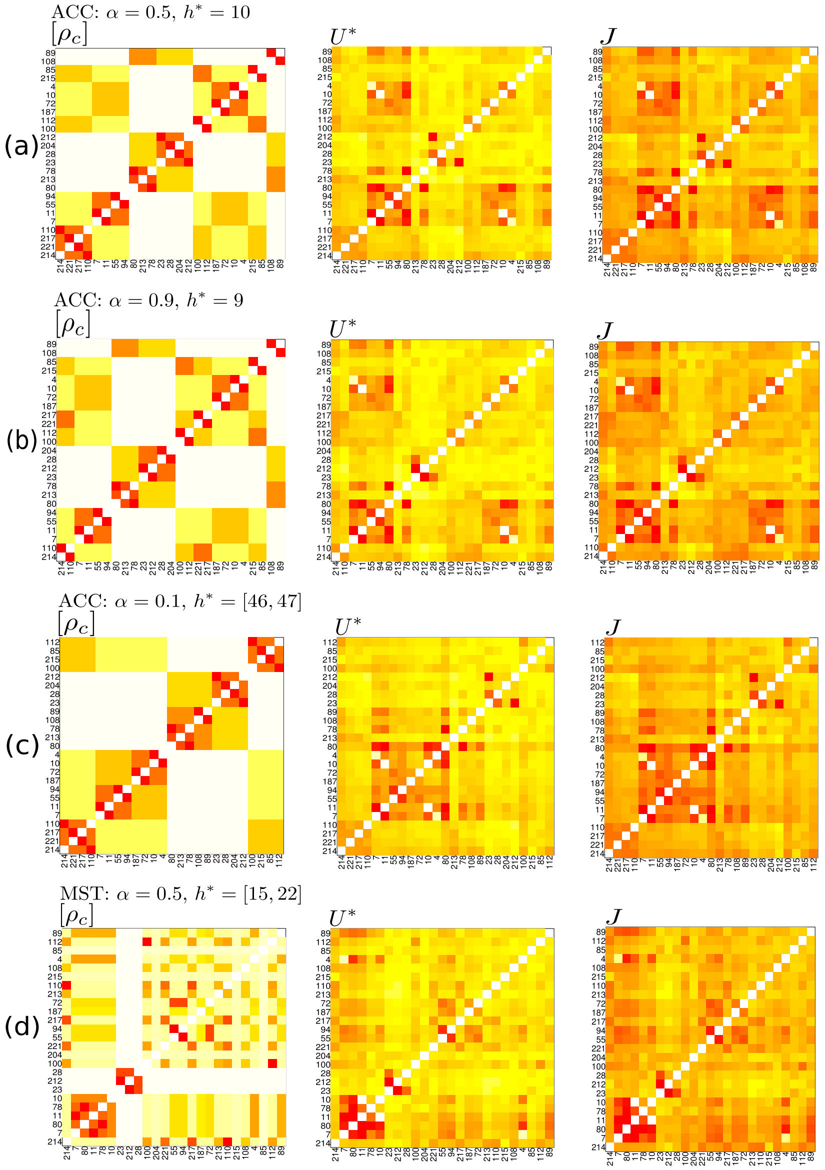

FIGURE 8. In (a), (b), and (c) it is shown the resulting cophenetic distances $[\rho_c]$, the coupling distances U^* and coupling matrix J for the solutions found at $\alpha = 0.5$, $\alpha = 0.9$, $\alpha = 0.1$, using ACC algorithm. The (d) row shows the same but the solution found using the MST approach with $\alpha = 0.5$. The order of items position for each case has been changed in order to group items belonging to the same cluster according to the specific solution. Red colors indicate low energies (low distances), and yellow-white colors indicate higher energies (higher distances).

very different clusters and are only merged at the maximum coupling distance of 518! According to the greedy algorithm, the recommendation to the retail manager is to consider the cluster formed by items 78 (flavored noodle), 80 (vitaminized noodle), and 213 (fruit-bearing yogurt), which form a cluster at a coupling distance of 4.82.

Given the structure found by the ACC-Algorithm, it is interesting to retrieve a spanning tree (ACC-Tree) from the coupling network, which connects each node in the network so that the connections between nodes are of minimum coupling distance between clusters. The length of the MST for this real case was 36.1, while that of the ACC-Tree was 37.2, as expected, longer than that of the MST. When looking at the degree of the nodes in each tree, we see substantial differences. For example, the two nodes with the highest degree in the MST are 80-Vitaminized noodle (d = 7) and 214-beaten yogurt (d = 6), while in the ACC-Tree, the nodes with the highest degree were 7-vegetable oil (d = 5), 214-beaten yogurt (d = 4), and 23-Coke Familiar Size (d = 4). In the first case, the 80 and 214 nodes manage to connect the most amount of just another at a minimum coupling distance, that is, minimum J_{ik}. On the other hand, nodes 7, 214, and 23 are characterized by connecting a greater number of other clusters at the shortest coupling distance U^*. In other words, for example, node 80 is well connected with 215, 89, 10, 4, and 78. In this sense, node 80 is a good candidate

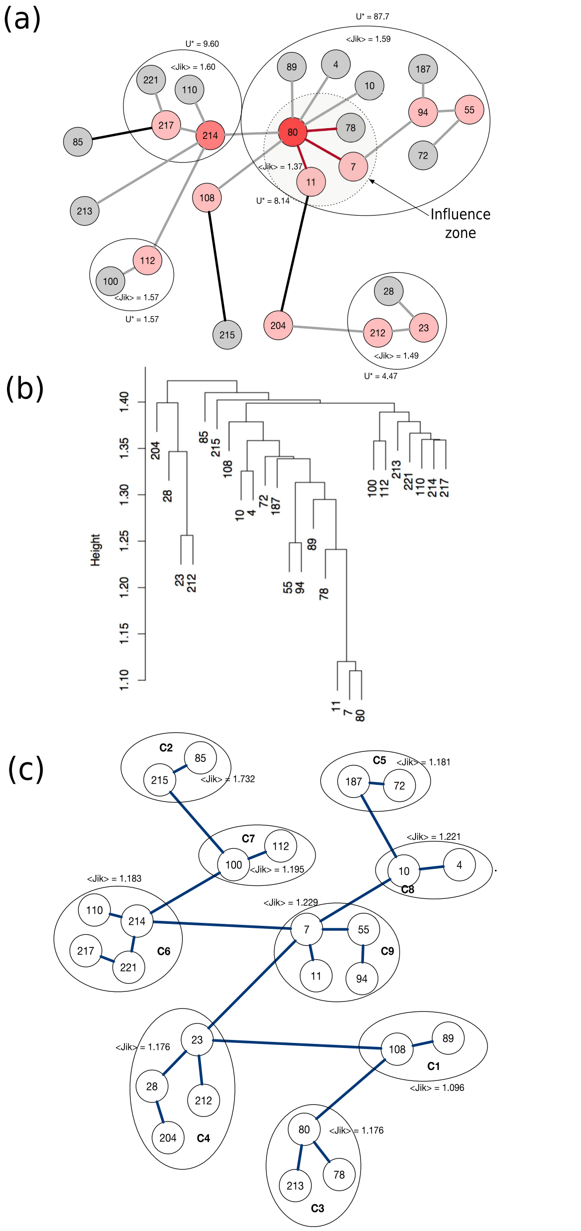

FIGURE 9. Comparison of the solution found using ACC algorithm with a solution using MSTcor. In (a), the MST is found using MSTcor methodology. The color of the edges represents the distance between nodes. A red colored edge indicates a short distance (distances are below the 10th percentile, which represents the influence zone), gray color are distances between the 10th and 90th percentile, and in black color larger distances (above the 90th percentile). Red nodes have importance equal to or above the 95th percentile (these are the most important nodes). Pink color nodes represents equal to or above the 90th percentile but below the 95th. Grey color nodes have importance are below the 90th percentile. (b): Dendrogram of the MSTcor methodology using average linkage, (c): The equivalent spanning ACC Tree for the dataset under analysis. The mean equivalent of the coupling J_{ik} for each cluster are shown, the same as shown in Table 1.

for marketing promotions. In contrast, node 7 has a greater number of connections to other clusters through nodes 214, 11, 55, and 10. In other words, an intervention in node seven could have a much more significant effect than intervening in node 80.

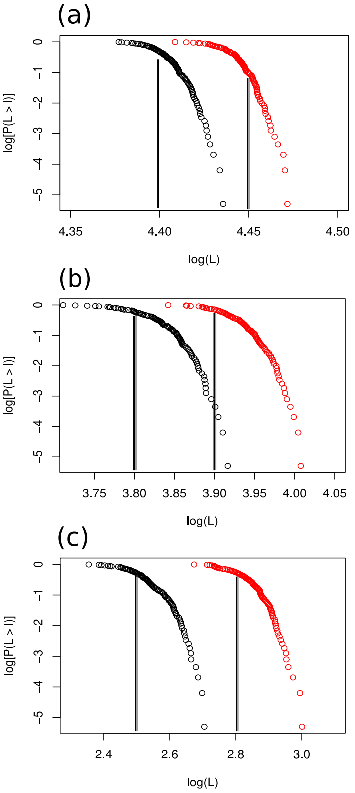

FIGURE 10. The probability distribution on a log-log plane of the tree lengths for simulation with $\langle J \rangle$ equal to −2, 0 and 2 in (a), (b) and (c) respectively. The black color is for the MST, and the red one is for the ACC-Tree. The tree-lengths of the MST is always lower than the ACC-Tree. The log-means of the tree lengths L were 4.40, 3.8 and 2.5 for the MST with J-means of −2, 0 and 2 respectively. The log-means of the tree lengths L were 4.45, 3.9 and 2.8 for the ACC-Tree with J-means of −2, 0 and 2 respectively.

2) FINDING A SOLUTION

Using the previous results, it is necessary to get a solution, i.e., to determine which is the best cutoff coupling distance h^*, which gives the best internal consistency $S(h^*)$. We used Equation 11 in with weight $\alpha = 0.5$. Figure 7 shows some interesting results.

For the ACC Algorithm, the best solution is found at $h^* = 10$ with $S(h^*) = 0.21$. On this location, $S_1 = 2.41$, $S_2 = 0.28$, nine clusters: 3 with four items, 1 with 3 items and 5 with two items. For comparison, the best solution in the MST approach is found at $h^* = [15, 22]$ with $S(h^*) = [0.33, 0.42]$. On this location, $S_1 = 2.35$, $S_2 = 2.07$, 19 clusters: 1 with 5 items, 2 with two items, and 12 clusters with just one item.

We see that the ACC algorithm has better internal consistency as the overall S-indicator is lower for the first case. Also, the ACC algorithm achieves a better balance on the number of elements in each cluster. Most of the clusters have size one on the MST approach and another cluster with size 5. This last one comprises items 7, 80, 11, 78, and 10, grouped in three different clusters on the ACC algorithm.

To appreciate the correspondence between the clustering solutions achieved by the algorithm and the original coupling energies, Figure 8 shows heatmaps representation of the coupling distances U^* and the original coupling energies $\mathbf{J}$, with a heatmap of the cophenetic distances, as a result of applying the ACC Algorithm with different α-values. In the first case, putting equal importance to S_1 and S_2 indexes, putting more importance to S_1 index, and putting more importance to S_2 index.

In our case, the cophenetic distance between two elements that have been merged is the coupling energy between groups, at which the two elements of the system have been combined in a cluster. By comparing the coupling distance and the cophenetic distance, it is possible to see the extent to which the clustering solution preserves the pairwise coupling distances.

As we have seen, with the ACC Algorithm, we obtain clusters with a good clustering coefficient, that is, we gain in obtaining structure, but apparently, it is not entirely a winner in preserving the original coupling distances. This is the cost to find a compelling structure in disordered systems. Comparing heatmaps of cophenetic distances and coupling distances, it can be carefully observed that the heatmap of the cophenetic distances, in general, coincide with the original distances, but not always. In fact, cophenetic correlation coefficient[1] was 0.22 for $\alpha = 0.5$, $\alpha = 0.9$ and $\alpha = 0.1$, which means that in there is a proportional relationship between the coupling distances and the cluster merge energies.

We see that the MST based clustering preserves the original distances (see heatmaps right column) better. The cophenetic correlation was 0.63, indicating a better correspondence between the pairwise coupling energy and the cluster merges energy than the ACC Algorithm. However, it should be remembered that the MST has the disadvantage of not taking into account energy interrelationships with other system elements. So, there is a trade-off between taking into account interrelationships between the system elements and preserving the original distances.

3) PROTOTYPICAL MARKET BASKETS

To get a practical idea of the results, we identified low coupling energy clusters called prototypical market baskets. They are groups of products that tend to be present simultaneously. To observe the results, we used a cut-off distance of $h^* = 15$ for the ACC Algorithm with $\alpha = 0.5$, which is the best solution found previously. The prototypical baskets are shown in Table 1. Some interesting insights can be obtained from these results that reveal the supermarket branch customers' purchase patterns. For example, soft drinks, wine, and chicken seem to be a typical market basket among supermarket branch customers, which may be the subject of special promotions or bundling. For example, the C9 cluster consists of four products: vegetable oil and rice grade 2, making much sense. The other two products are traditional chlorine bleach and regular dishwasher, which seems to make sense. However, these four products form a single cluster, indicating that these four products tend to be present simultaneously in the same basket. Again, this can be interesting as information for the marketing department might specialize in specific offers based on this kind of information.

4) RESULTS COMPARISON WITH OTHER GRAPH-ORIENTED METHOD

Our proposal for market basket analysis using graph methods is not the only one. Recently, minimum spanning trees (MSTs) have been used to reduce the huge amount of interactions that arise from aggregate purchasing behavior (see [21]). In this section, we carry out a brief comparison of the results with those that could be obtained following the proposal of [21] (hereafter, the MSTcor method). This graph-based methodology takes advantage of the MST's hierarchical tree structure to find groups of products that have strong interdependencies between them. In a similar way, the process starts the transactional data as indicated in our proposal, using market baskets represented by binary vectors that indicate the presence or absence of product categories. However, in [21] they directly compute the linear correlation between pairs of items and then transform those correlations into a distance metric. This information serves as input to compute the MST with the Prim Algorithm. They then perform a series of tests to detect edges of the MST that are independent and discard them, and also define a node importance metric (items). Finally, they establish the "zones of influence." These zones are important nodes and significant edges of interdependence that are susceptible and interesting for establishing promotional and marketing actions. The direct use of MSTs to find these influence zones makes sense considering that, as commented in section II-C, MSTs represent an inherent property of disordered systems and are equivalent to the optimal path that can be achieved at a minimum energy level.

It should be noted that this methodology does not provide a connection with the probability distribution of the purchase basket vectors, while the ACC algorithm in our work has a more solid foundation because it is based on the joint distribution of system states that reproduce the aggregate purchase behavior. This is the main difference between our approach and that of the MSTcor.

We take the 42077 purchase transaction database with 25 product subcategories to build the MST based on correlations, as described in [21]. There are no edges of the MST that were not significant. The result is shown in Figure 9(a). Node 80 (Vitaminized noodle) and 214 (beaten yogurt) are the most important items due to the proximity and connectivity with other products. The zone of influence is characterized by node 80 with other items with good connectivity and given by its proximity to other items (red edges).

The circles indicate clusters of items for a given solution with $k = 9$. This solution has been chosen only for comparison purposes with the 9 cluster solution with the

[1] The cophenetic correlation coefficient is the linear correlation between the coupling energies between every pair of spins and their corresponding cophenetic distances. In our case, the cophenetic distances are the coupling energies at which every element is merged in the same cluster.

TABLE 1. The cluster solution found at $\alpha = 0.5$ with the ACC Algorithm gives a total of 9 groups. The subcategories of products are indicated for each cluster, the total coupling distance of each cluster $U^*(C_m)$, and the mean equivalent of the coupling J_{ik} calculated as the total coupling distance divided by the number of edges. Low coupling distance indicates a more likely to find that market basket.

$\alpha = 0.5$, Total number of clusters = 9	$U^*(C_m)$	Mean $\langle J_{ik} \rangle$
C1: Sacked yogurt (215), Sliced gauda cheese (85)	1.80	1.80
C2: Full-cream milk powder (108), Canned mackerel (89)	1.73	1.73
C3: Vitaminized noodle (80), beaten yogurt with fruits (213), Flavoured noodle (78)	4.82	1.61
C4: Coca-Cola family size (23), Soft drinks family size (28), Whole chicken tutre (204), boxed red wine (212)	9.66	1.61
C5: Shampoo family size (187), washing machine powder (72)	1.60	1.60
C6: beaten yogurt (214), probiotic yogurt (221), yogurt with cereals (217), Flavoured milk 200 cc(110)	9.60	1.60
C7: Full-cream white milk 1L (100), Flavoured milk (112)	1.57	1.57
C8: Rice grade 1, (10), Sunflower oil (4)	1.51	1.51
C9: vegetable oil (7), rice grade 2 (11), traditional chlorine bleach (55), regular dishwasher (94)	8.92	1.49

ACC algorithm. For each cluster larger than two nodes, the total coupling distance of each cluster $U^*(C_m)$, and the mean equivalent of the coupling J_{ik} are shown. This information allows us to compare these values with those of the clusters of Table 1. It can be seen that in general, the means of coupling J_{ik} in the solution with ACC tends to be lower than in MSTcor, which indicates that the clusters with ACC have a higher probability of occurring simultaneously than those in MSTcor. Figure 9(b) shows the respective dendrogram using average linkage. The sequence of cluster agglomeration in MSTcor is completely different from that of the ACC algorithm. This is evident when comparing the ACC resultant dendrogram (Figure 6(d)) with that of MSTcor. In the latter methodology, pairs of nodes connected at a low distance are identified, that is, pairs of product items whose probability of being present in a shopping basket is high. In contrast, the solution indicated in Table 1 considers the influence of other products on a particular item.

It is included in Figure 9(c), an equivalent graphical representation of the solution with the ACC algorithm. This tree has a direct relationship with the dendrogram. It is another alternative to compare the clustering of items with other methods such as MSTcor. For details of the construction of the ACC spanning tree, see the Appendix. As it has been seen in simulations with other sizes of networks (see Figure 5), the ACC item agglomeration process achieves more balanced clusters compared to other methods, and in particular, more balanced than the MSTcor. This is a practical advantage since clusters of only one item are avoided. For promotional activities such as product bundling, it makes more sense to encourage the purchase of more than one product, such as clusters of 2 or 3 products.

V. CONCLUSION

The proposed algorithm's main advantage is that it manages to find a good structure level compared to clustering based on MST or modularity. However, this is achieved in exchange for sacrificing some consistency of the clusters. However, in practical terms, the structure achieved allows a decision-maker to have a clearer idea of potential groups of promotional or marketing activities. On the other hand, the MST-based approach produces long and unwieldy clusters due to the typical chain effect that manifests itself in complex systems with many interactions between the system's parts.

In practical terms, these results suggest that a product or item clustering should consider a solution based on the ACC algorithm as a necessary reference. The retail manager needs to know or identify all the small groups of products that usually are purchased together and represent clusters with a high probability of occurrence. From there, one wishes to discover groups of larger or more complex clusters. Although they have a lower chance of occurring as part of a market basket, follow a natural order as a cluster's size increases.

Unlike other classical methodologies in market basket analysis that seek to find relationships between product items, the proposal based on coupling energies considers the interdependence between multiple elements. This means that the approach is richer in providing information on how groups of items with similar behavior are formed. For example, association rules [20] offer partial and isolated relations between sets of items, making it more difficult for the analyst to have a global idea of the influence that a product may have on the set of other products. On the other hand, this methodology has the possibility of determining the force (couplings) to which the products are related, and not only the direction of the relationship. In this sense, we believe that the graph-theoretic approach of the purchase-market behavior, in which the system is more than the sum of its parts, offers a viable model to complement, but not to replace the Association Rules (ARs). Thus, using local models (as ARs) and global pattern-seeking models (as in the proposed methodology) can help the analyst achieve a more detailed understanding of customer behavior.

An indispensable input to the methodology is the coupling matrix **J**. Although there are methods for finding maximum entropy distributions consistent with the correlations and means of the empirical data [39]–[41], this task is usually not trivial, and it involves high computation time, especially when dealing with systems with a large number of nodes. An alternative to deal with the inverse Ising problem or parameter inference is to use the Mean-Field Theory [23] as an approximated method to find the couplings, which would allow a decrease in the computation time required to find the parameter values. Finally, it is worth that for both the inference of the couplings and the aggregation algorithm of products based on coupling energy, it was not necessary to make assumptions about the structure of the system or the distribution of the market baskets, which is an advantage over other market basket analysis models.

APPENDIX A
THE ACC SPANNING TREE

The network of couplings G has N^{N-2} spanning trees according to the Cayley's formula [42]. One such spanning tree is MST which, as noted above, represents a unique signature to the disordered system, and which establishes a hierarchical structure with the path of minimum energy. But this is not the only possible structure, and a solution that incorporates the interaction of all elements of the system simultaneously can be found using the ACC algorithm. Given this structure, it is possible to reconstruct a spanning tree of G, but not of minimum distance, that simplifies the complexity of the coupling network, and that keeps intact the ultrametric properties.

The ACC-Tree $T_{ACC}(G, E_{ACC})$ is a subgraph of G with the same nodes of G, with edges $E_{ACC} \subseteq E(G)$, but unlike the MST, they do not join all the nodes of G at a minimum distance, but join according to the minimum coupling distance between clusters formed by the ACC algorithm. It also has $N - 1$ edges, and from it, it is possible to identify nodes that have a high degree of connectivity with different other clusters in the system.

Let C_i be the cluster that is formed in the i^{th} iteration. The C_i cluster can be formed from other clusters and therefore, it is simply the union of clusters that precedes it, i.e., $C_i = C_q \cup C_w$ where $q < i$ and $w < i$. The coupling distance, of this merge in C_i es $U^*(C_i)$, which corresponds to the weights of the edges E_{ACC}. In the starting iterations, C_i is composed of individual nodes, for example, v_q and v_w. In these cases, the nodes form a connection at a given coupling distance between the two nodes d_{qw}. In cases where the merge that originates C_i occurs between a node v_j and a cluster C_k with $k \neq i$ y $v_j \nsubseteq C_k$, the connection occurs with some element of C_k such that it is minimum, that is to say, it looks for the minimum coupling distance between v_j and all the elements of C_k. This distance will be the weight for this edge. In cases where two clusters that both have more than two nodes are merged, the minimum coupling distance is searched for among all the combination pairs of nodes. The process of connecting nodes with the minimum coupling distance is repeated $N - 1$ times, from the first merge to the last one. The result will be $N - 1$ pairs of selected nodes that form the edges of the ACC spanning tree.

A. A SIMULATED EXAMPLE OF ACC-TRESS

As an example, to compare the behavior of MSTs and ACC-Trees with different disordered systems, we have made several simulations that create synthetic coupling matrices of size $N = 50$, and with different coupling means $\langle J \rangle$ equal to -2 (antiferromagnetic), 0 (neutral) and 2 (ferromagnetic). For each case, 200 realizations were made. In each of them, the solution of MST and ACC-Tree is obtained.

As expected, the lengths of the MST are always smaller than the lengths of the ACC-Tree. The interesting thing about the ACC-Trees is that they provide different information regarding the connectivity of the nodes. If we calculate the degree of each node of the MST and the ACC-Tree, we can see a small but positive correlation between both. This indicates that the structure derived from the coupling distance (ACC Algorithm), and not only from the minimum distance (MST), provides a different centrality characterization of nodes. Indeed, the correlation obtained between the degree of the nodes in MST and ACC-Tree were 0.10 (sd = 0.15), 0.12 (sd = 0.16) and 0.10 (sd = 0.15) with coupling means -1, 0 and 2 respectively. A hub node in the MST (high degree) indicates a high correlation between that node and those with whom it connects, while a hub node in the ACC-Tree indicates that it has a high degree of ferromagnetic relationship with other clusters.

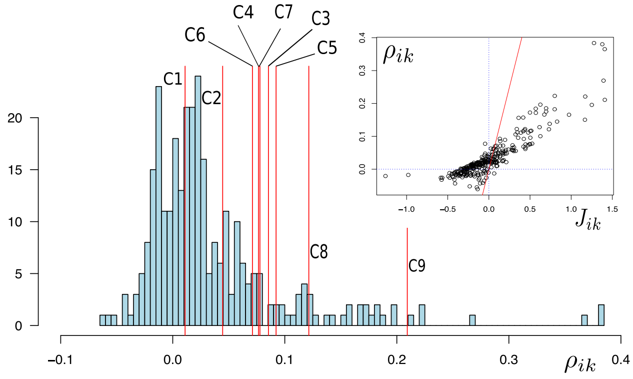

FIGURE 11. **The probability distribution of correlations among every pair of items of the real case dataset. The red lines indicates the mean of correlations between pair of items for each cluster solution found with the ACC algorithm using $\alpha = 0.5$. The inset shows a scatterplot between couplings and correlations. The red diagonal represent a perfect correspondence between them.**

Figure 11 shows the correlations between the items in the clusters found in relation to the distribution of correlations of the system. It would be expected that the correlations are positive, which is true for all correlations of the clusters. The correlations of cluster 9 and 8 are the highest of all of them. All the correlations are higher than the mean system correlations (equal to 0.0344) except the lowest of cluster 1 equal to 0.0110, which also coincides with the low coupling distance value equal to 1.096 (See Table 1). It is worth mentioning that the correlations measure only the linear association between pairs of items, while the couplings capture information of the non-linear interdependence between parts of the system. Thus, the relationship between couplings and correlations does not have to be perfect, as shown by the inset in Figure 11.

REFERENCES

[1] J. M. Ziman, *Models of Disorder: The Theoretical Physics of Homogeneously Disordered Systems*. Cambridge, U.K.: Cambridge Univ. Press, 1979.

[2] M. Mézard, G. Parisi, N. Sourlas, G. Toulouse, and M. Virasoro, "Nature of the spin-glass phase," *Phys. Rev. Lett.*, vol. 52, no. 13, p. 1156, 1984.

[3] A. Ansari, J. Berendzen, S. F. Bowne, H. Frauenfelder, I. Iben, T. B. Sauke, E. Shyamsunder, and R. D. Young, "Protein states and proteinquakes," *Proc. Nat. Acad. Sci. USA*, vol. 82, no. 15, pp. 5000–5004, 1985.

[4] S. Kirkpatrick and G. Toulouse, "Configuration space analysis of travelling salesman problems," *J. de Phys.*, vol. 46, no. 8, pp. 1277–1292, 1985.

[5] O. C. Martin, R. Monasson, and R. Zecchina, ''Statistical mechanics methods and phase transitions in optimization problems,'' *Theor. Comput. Sci.*, vol. 265, nos. 1–2, pp. 3–67, Aug. 2001.

[6] M. Mezard, M. Mezard, and A. Montanari, *Information, Physics, and Computation*. London, U.K.: Oxford Univ. Press, 2009.

[7] L. F. Cugliandolo. (Mar. 2020). *Disordered Systems (Lecture Notes)*. Cargèse, France. [Online]. Available: https://www.lpthe.jussieu.fr/~leticia/cargese-lectures.pdf

[8] K. Binder and A. P. Young, ''Spin glasses: Experimental facts, theoretical concepts, and open questions,'' *Rev. Mod. Phys.*, vol. 58, no. 4, p. 801, 1986.

[9] T. S. Jackson and N. Read, ''Theory of minimum spanning trees. I. mean-field theory and strongly disordered spin-glass model,'' *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 81, no. 2, Feb. 2010, Art. no. 021130.

[10] R. Dobrin and P. Duxbury, ''Minimum spanning trees on random networks,'' *Phys. Rev. Lett.*, vol. 86, no. 22, p. 5076, 2001.

[11] R. N. Mantegna and H. E. Stanley, ''Scaling behaviour in the dynamics of an economic index,'' *Nature*, vol. 376, no. 6535, p. 46, 1995.

[12] M. Keskin, B. Deviren, and Y. Kocakaplan, ''Topology of the correlation networks among major currencies using hierarchical structure methods,'' *Phys. A, Stat. Mech. Appl.*, vol. 390, no. 4, pp. 719–730, Feb. 2011.

[13] R. N. Mantegna, ''Hierarchical structure in financial markets,'' *Eur. Phys. J. B, Condens. Matter Complex Syst.*, vol. 11, no. 1, pp. 193–197, Sep. 1999.

[14] R. N. Mantegna, ''Information and hierarchical structure in financial markets,'' *Comput. Phys. Commun.*, vols. 121–122, pp. 153–156, Sep. 1999.

[15] G. Bonanno, G. Caldarelli, F. Lillo, and R. N. Mantegna, ''Topology of correlation-based minimal spanning trees in real and model markets,'' *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 68, no. 4, Oct. 2003, Art. no. 046130.

[16] L. Kullmann, J. Kertész, and R. N. Mantegna, ''Identification of clusters of companies in stock indices via potts super-paramagnetic transitions,'' *Phys. A, Stat. Mech. Appl.*, vol. 287, nos. 3–4, pp. 412–419, Dec. 2000.

[17] M. A. Valle, G. A. Ruz, and R. Morrás, ''Market basket analysis using minimum spanning trees,'' in *Network Intelligence Meets User Centered Social Media Networks. ENIC* (Lecture Notes in Social Networks), R. Alhajj, H. Hoppe, T. Hecking, P. Bródka, and P. Kazienko, Eds. Cham, Switzerland: Springer, 2018, doi: 10.1007/978-3-319-90312-5_11.

[18] S. S. Borysov, Y. Roudi, and A. V. Balatsky, ''U.S. Stock market interaction network as learned by the Boltzmann machine,'' *Eur. Phys. J. B*, vol. 88, no. 12, p. 321, 2015.

[19] T. Bury, ''Market structure explained by pairwise interactions,'' *Phys. A, Stat. Mech. Appl.*, vol. 392, no. 6, pp. 1375–1385, Mar. 2013.

[20] R. Agrawal and R. Srikant, ''Fast algorithms for mining association rules,'' in *Proc. 20th Int. Conf. Very Large Data Bases (VLDB)*, vol. 1215, 1994, pp. 487–499.

[21] M. A. Valle, G. A. Ruz, and R. Morrás, ''Market basket analysis: Complementing association rules with minimum spanning trees,'' *Expert Syst. Appl.*, vol. 97, pp. 146–162, May 2018.

[22] N. S. Magner, J. F. Lavin, M. A. Valle, and N. Hardy, ''The volatility forecasting power of financial network analysis,'' *Complexity*, vol. 2020, pp. 1–17, Sep. 2020.

[23] Y. Roudi, E. Aurell, and J. A. Hertz, ''Statistical physics of pairwise probability models,'' *Frontiers Comput. Neurosci.*, vol. 3, p. 22, 2009.

[24] G. Hinton, ''Boltzmann machines,'' in *Encyclopedia of Machine Learning and Data Mining*. New York, NY, USA: Springer, 2014, pp. 1–7.

[25] S. Guo, C. Zhou, B. Wang, and X. Zheng, ''Training restricted Boltzmann machines using modified objective function based on limiting the free energy value,'' *IEEE Access*, vol. 6, pp. 78542–78550, 2018.

[26] T. Li, B. Wang, Y. Jiang, Y. Zhang, and Y. Yan, ''Restricted Boltzmann machine-based approaches for link prediction in dynamic networks,'' *IEEE Access*, vol. 6, pp. 29940–29951, 2018.

[27] W. Gao, T. Friedrich, F. Neumann, and C. Hercher, ''Randomized greedy algorithms for covering problems,'' in *Proc. Genetic Evol. Comput. Conf.*, 2018, pp. 309–315.

[28] D. L. Davies and D. W. Bouldin, ''A cluster separation measure,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.

[29] P. J. Rousseeuw, ''Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,'' *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.

[30] C. D. Manning, H. Schütze, and P. Raghavan, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[31] W. H. E. Day and H. Edelsbrunner, ''Efficient algorithms for agglomerative hierarchical clustering methods,'' *J. Classification*, vol. 1, no. 1, pp. 7–24, Dec. 1984.

[32] J. C. Gower and G. J. Ross, ''Minimum spanning trees and single linkage cluster analysis,'' *J. Roy. Stat. Soc. C, Appl. Statist.*, vol. 18, no. 1, pp. 54–64, 1969.

[33] A. Clauset, M. E. J. Newman, and C. Moore, ''Finding community structure in very large networks,'' *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, no. 6, Dec. 2004, Art. no. 066111.

[34] M. E. J. Newman, ''Modularity and community structure in networks,'' *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 23, pp. 8577–8582, Jun. 2006.

[35] M. A. Valle and G. A. Ruz, ''Market basket analysis using Boltzmann machines,'' in *Artificial Neural Networks and Machine Learning—ICANN 2019: Text and Time Series* (Lecture Notes in Computer Science), vol. 11730, I. Tetko, V. Kurkova, P. Karpov, F. Theis, Eds. Cham, Switzerland: Springer, 2019.

[36] M. A. Valle, G. A. Ruz, and S. Rica, ''Market basket analysis by solving the inverse ising problem: Discovering pairwise interaction strengths among products,'' *Phys. A, Stat. Mech. Appl.*, vol. 524, pp. 36–44, Jun. 2019.

[37] V. Dotsenko, *An Introduction to the Theory of Spin Glasses and Neural Networks*, vol. 54. Singapore: World Scientific, 1995.

[38] D. L. Stein, ''Spin glasses: Old and new complexity,'' in *Proc. AIP Conf.*, 2011, vol. 1389, no. 1, pp. 965–968.

[39] E. Schneidman, M. J. Berry, R. Segev, and W. Bialek, ''Weak pairwise correlations imply strongly correlated network states in a neural population,'' *Nature*, vol. 440, no. 7087, pp. 1007–1012, Apr. 2006.

[40] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, ''A learning algorithm for Boltzmann machines,'' *Cognit. Sci.*, vol. 9, no. 1, pp. 147–169, 1985.

[41] T. R. Lezon, J. R. Banavar, M. Cieplak, A. Maritan, and N. V. Fedoroff, ''Using the principle of entropy maximization to infer genetic interaction networks from gene expression patterns,'' *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 50, pp. 19033–19038, Dec. 2006.

[42] L. Clarke, ''On Cayley's formula for counting trees,'' *J. London Math. Soc.*, vol. 1, no. 4, pp. 471–474, 1958.

MAURICIO A. VALLE received the B.Sc. degree in electronic engineering and the M.Sc. degree in industrial engineering from the Pontificia Universidad Católica de Valparaíso, Chile, in 2001 and 2003, respectively, and the Ph.D. degree in management from the Universidad Adolfo Ibáñez, Chile, in 2012. He is currently a Professor at the Faculty of Economics and Business, Universidad Finis Terrae, Santiago, Chile. He has participated in several interdisciplinary groups to analyze consumer behavior in the retail industry, employee turnover in call centers, and modeling for mining exploration activities using machine learning. His research interests include applying machine learning algorithms and physics models to understand human behavior and propose solutions to business problems and complex systems.

GONZALO A. RUZ (Member, IEEE) received the B.Sc., P.E., and M.Sc. degrees in electrical engineering from the Universidad de Chile, Santiago, Chile, in 2002 and 2003, respectively, and the Ph.D. degree from Cardiff University, U.K., in 2008. He is currently a Professor and the Research Director of the Faculty of Engineering and Sciences, Universidad Adolfo Ibáñez, Santiago, Chile. His research interests include machine learning, evolutionary computation, data mining, gene regulatory network modeling, and complex systems.