On methods to assess the significance of community structure in networks of financial time series

Argimiro Arratia and Martí Renedo

Universitat Politècnica de Catalunya, Barcelona, SPAIN
argimiro@cs.upc.edu, marti.renedo@estudiant.upc.edu

Abstract. We consider the problem of determining whether the community structure found by a clustering algorithm applied to financial time series is statistically significant, or is due to pure chance, when no other information than the observed values and a similarity measure among time series are available. As a subsidiary problem we also analyse the influence of the choice of similarity measure in the accuracy of the clustering method.
We propose two raw-data based methods for assessing robustness of clustering algorithms on time-dependent data linked by a relation of similarity: One based on community scoring functions that quantify some topological property that characterises ground-truth communities, and another based on random perturbations and quantification of the variation in the community structure. These methodologies are well-established in the realm of unweighted networks; our contribution are versions of these methodologies properly adapted to complete weighted networks.

Keywords: clustering time series, ground-truth communities, similarity measures, Forex network

1 Introduction

We treat in this work the problem of determining the intrinsic structure of clustered data, where the clusters are based on some measure of similarity affecting all pairs of data points. From a network analysis perspective we are concerned with assessing the significance of communities formed by some unsupervised classification algorithm (i.e. clustering procedure) applied to fully-connected weighted networks.

We are motivated by research in community structure and their dynamics in financial market networks, which are characterise by a fixed number of nodes, each representing a financial time series, and links among all pairs of nodes weighted by the values of a measure of similarity, commonly based on pairwise correlation, between pairs of time series (see, e.g., [3],[5],[9],[12],[11]). In our previous work [14] we presented empirical evidence of the impact of the chosen similarity measure on the clustering results: In a foreign exchange (FX) network, and clustering based on the Girvan-Newman modularity maximisation algorithm [10, 2], we analysed the qualitative differences in the clusterings obtained under three different correlation measures: Pearson, Kendall and the most recent distance correlation [4]. As an application of the statistical and topological criteria that we developed and present here to assess robustness of clustering on weighted networks, we shall give quantitative measures of the nature of the clustering obtained by considering similarity either based on Pearson or on distance correlation.

To assess the significance of communities structure in complete weighted networks, we developed a collection of cluster scoring functions that measure some topological characteristic of the ground-truth communities as defined by Yang and Leskovec in [15] for unweighted networks. Our scoring functions are proper extensions of theirs to weighted networks. We then combined these topological measures of robustness of clusters with an analysis of the variation of successive random perturbations of the original network. The perturbations consist on changing the weights distribution and degenerate the original network, the cluster variation is measured in terms of the change of information (in the sense of Shannon's Theory of Information [7]). The idea is that a robust community should differ in its structural properties from the random perturbations inasmuch as these affect greater proportions of the network. Additional clustering robustness tests and fine details of the ones presented here can be read in the extended report [13].

2 Basic definitions

2.1 The Forex Network

The networks of exchange rates studied in [14][3] are built by considering the exchange rates as vertices and drawing edges between these vertices, weighted by the similarity between the returns of the pair of chosen exchange rates. We will focus on two possible similarity measures: one based on the Pearson correlation and the other based on the distance correlation[4][1].

For the Pearson similarity network, the adjacency matrix is defined as

$$A^{\rho}_{ij} = \frac{1}{2}(\rho(r^i, r^j) + 1) - \delta_{ij}. \tag{1}$$

This scales the Pearson correlation $\rho(\cdot, \cdot)$ from $[-1, 1]$ to $[0, 1]$, while the Kronecker delta δ_{ij} removes self-edges. In the graph with adjacency matrix A^{ρ} exchange rates with positively linearly correlated returns will be connected by edges of weight close to 1, and weight near 0 if the correlation is negative. Edges connecting non correlated exchanges will have weights closer to the center of the interval $[0, 1]$.

[1] In previous work [14] we considered in addition Kendall correlation measure. We omit it in this study since our main purpose here is to put to test the clustering performance measures, as opposed to compare clustering methods.

In the case of the distance correlation $\mathcal{R}(\cdot,\cdot)$, the network is simply built from the matrix of distance correlations, $A^{\mathcal{R}}$ by removing self edges. For each pair of exchange rate returns r^i, r^j,

$$A^{\mathcal{R}}_{ij} = \mathcal{R}(r^i, r^j) - \delta_{ij} \tag{2}$$

2.2 Community Detection

The partition of the networks into communities is done using the Potts method. It consists on minimising an objective function, the Potts Hamiltonian, which evaluates the strength[2] of a partition of the graph. This can be seen as a generalisation of the modularity function[10].

Definition 1. *The modularity of the partition $\mathcal{P}$ of a weighted undirected graph with adjacency matrix A is given by*

$$Q(\mathcal{P}) = \frac{1}{2m}\sum_{ij}[A_{ij} - P_{ij}]\delta(c_i, c_j) \tag{3}$$

where c_i is the community of the node i in the partition $\mathcal{P}$ (so $\delta(c_i, c_j)$ is 1 when i and j are in the same community and 0 otherwise), P_{ij} is the expected weight of the edge ij in a null model and m is the sum of the weights of all edges in the graph.

Definition 2. *The Hamiltonian of the Potts system of the partition $\mathcal{P}$ of a weighted undirected graph with adjacency matrix A is given by*

$$H(\mathcal{P}) = -\sum_{ij}[A_{ij} - \gamma P_{ij}]\delta(c_i, c_j)$$

where γ is a parameter which determines how likely vertices are to form communities.

The algorithm used to minimise the Potts Hamiltonian has been adapted from the modularity maximisation algorithm in [2] to suit weighted networks and this objective function.

3 Cluster scoring functions

Here we will provide functions which will evaluate the division of networks into clusters, specifically when the edges have weights. Using the scoring functions for communities in unweighted networks given in [15] as a reference, we propose generalisations of some of them to the weighted case.

[2] Considering a strong partition one that has strong links inside the communities and weak links between them.

Basic definitions. Let $G(V, E)$ be an undirected graph of order $n = |V|$ and size $m = |E|$. In the case of a weighted graph $\tilde{G}(V, \tilde{E})$[3], we will denote $\tilde{m} = \sum_{e \in \tilde{E}} w(e)$ the sum of all edge weights. Given $S \subset G$ a subset of vertices of the graph, we have $n_S = |S|$, $m_S = |\{(u,v) \in E : u \in S, v \in S\}|$, and in the weighted case $\tilde{m}_S = \sum_{(u,v) \in \tilde{E}: u,v \in S} w((u,v))$. Note that if we treat an unweighted graph as a weighted graph with weights 0 and 1 (1 if two vertices are connected by an edge, 0 otherwise), then $m = \tilde{m}$ and $m_S = \tilde{m}_S$ for all $S \subset V$.

The following definitions will also be needed later on:

– $c_S = |\{(u,v) \in E : u \in S, v \notin S\}|$ is the number of edges connecting S to the rest of the graph.
– $\tilde{c}_S = \sum_{(u,v) \in E: u \in S, v \notin S} w_{uv}$ is the natural extension of c_S to weighted graphs; the sum of weights of all edges connecting S to $G \setminus S$.
– $\tilde{d}(u) = \sum_{v \neq u} w_{uv}$ is the natural extension of the vertex degree $d(u)$ to weighted graphs; the sum of weights of edges incident to u.
– $d_S(u) = |\{v \in S : (u,v) \in E\}|$ and $\tilde{d}_S(u) = \sum_{v \in S} w_{uv}$ are the (unweighted and weighted, respectively) degrees[4]restricted to the subgraph S.
– d_m and $\tilde{d}_m$ are the median values of $d(u)$, $u \in V$.[5]

Scoring functions. The left column in Table 1 shows the community scoring functions for unweighted networks defined in [15]. These functions characterise some of the properties that are expected in networks with a strong community structure, with more ties between nodes in the same community than connecting them to the exterior. There are scoring functions based on internal connectivity (internal density, edges inside, average degree), external connectivity (expansion, cut ratio) or a combination of both (conductance, normalised cut, and maximum and average out degree fractions)

On the right column we propose generalisations to the scoring functions which are suitable for weighted graphs while most closely resembling their unweighted counterparts. Note that for graphs which only have weights 0 and 1 (1 indicates that an edge exists, 0 that it doesn't) each pair of functions is equivalent (any definition that didn't satisfy this wouldn't be a generalisation at all).

– **Internal density, edges inside, average degree:** These definitions are easily and naturally extended by replacing the number of edges by the sum of their weights.
– **Expansion:** Average number of edges connected to the outside of the community, per node. For weighted graphs, average sum of edges connected to the outside, per node.

[3] For every variable or function defined over the unweighted graph, will use a "∼" to denote its weighted counterpart
[4] We assume the weight function w_{uv} is defined for every pair of vertices u,v of the weighted graph, with $w_{uv} = 0$ if there is no edge between them.
[5] To prevent confusion between the function $d_S(\cdot)$ and the median value (which only depends on G) d_m we will always refer to subgraphs of G with uppercase letters.

Table 1. Community scoring functions for weighted and unweighted networks.

	unweighted	weighted
Internal density	$f(S) = \frac{m_S}{n_S(n_S-1)/2}$	$f(S) = \frac{\tilde{m}_s}{n_S(n_S-1)/2}$
Edges Inside	$f(S) = m_S$	$f(S) = \tilde{m}_S$
Average Degree	$f(S) = \frac{2m_S}{n_S}$	$f(S) = \frac{2\tilde{m}_S}{n_S}$
Expansion	$f(S) = \frac{c_s}{n_s}$	$f(S) = \frac{\tilde{c}_s}{n_s}$
Cut Ratio	$f(S) = \frac{c_s}{n_s(n-n_s)}$	$f(S) = \frac{\tilde{c}_s}{n_s(n-n_s)}$
Conductance	$f(S) = \frac{c_s}{2m_s+c_s}$	$f(S) = \frac{\tilde{c}_s}{2\tilde{m}_s+\tilde{c}_s}$
Normalised Cut	$f(S) = \frac{c_s}{2m_s+c_s}$	$f(S) = \frac{\tilde{c}_s}{2\tilde{m}_s+\tilde{c}_s}$
Maximum ODF	$f(S) = \max_{u\in S} \frac{\lvert\{(u,v)\in E: v\notin S\}\rvert}{d(u)}$	$f(S) = \max_{u\in S} \frac{\sum_{v\notin S} w_{uv}}{\tilde{d}(u)}$
Average ODF	$f(S) = \frac{1}{n_s}\sum_{u\in S} \frac{\lvert\{(u,v)\in E: v\notin S\}\rvert}{d(u)}$	$f(S) = \frac{1}{n_s}\sum_{u\in S} \frac{\sum_{v\notin S} w_{uv}}{\tilde{d}(u)}$

- **Cut Ratio:** Fraction of edges leaving the cluster, over all possible edges. The proposed generalisation is reasonable because edge weights are upper bounded by 1 and therefore relate easily to the unweighted case. In more general weigthed networks, however, this could take values well over 1 while lacking many "potential" edges (as edges with higher weights would distort the measure). In general bounded networks (with bound other than 1) it would be reasonable to divide the result by the bound, which would result in the function taking values between 0 and 1 (0 with all possible edges being 0 and 1 when all possible edges reached the bound).
- **Conductance and normalised cut:** Again, these definitions are easily extended using the methods described above.
- **Maximum and average Out Degree Fraction:** Maximum and average fractions of edges leaving the cluster over the degree of the node. Again, in the weighted case the number of edges is replaced by the sum of edge weights.

Clustering coefficient. Another possible scoring function for communities is the clustering coefficient or transitivity: the fraction of closed triplets over the number of connected triplets of vertices. A high internal clustering coefficient (computed on the graph induced by the vertices of a community) matches the intuition of a well connected and cohesive community inside a network, but its generalisation to weighted networks is not trivial.

There have been several attempts to come up with a definition of the clustering coefficient for weighted networks. One is proposed in [1] and is given by $c_i = \frac{1}{\tilde{d}(i)(d(i)-1)} \sum_{j,h} \frac{w_{ij}+w_{ih}}{2} a_{ij}a_{jh}a_{ih}$. Note that this gives a local (*i.e.* defined for each vertex) clustering coefficient.

While this may work well on some weighted networks, in the case of complete networks, such as those built from correlation of time series,

$$c_i = \frac{1}{\tilde{d}(i)(d(i)-1)} \sum_{i,j} \frac{w_{ij} + w_{ih}}{2} = \frac{\sum_j w_{ij} + \sum_h w_{ih}}{\tilde{d}(i)(n-2)\cdot 2} = \frac{2\tilde{d}(i)}{\tilde{d}(i)(n-2)\cdot 2} = \frac{1}{n-2} \quad (4)$$

is constant on all edges and doesn't give any information about the network.

An alternative was proposed in [6] with complete weighted networks (with weights in the interval $[0,1]$) in mind, which makes it more adequate for our case:

- For $t \in [0,1]$ let A_t be the adjacency matrix with elements $a^t_{ij} = 1$ if $w_{ij} \geq t$ and 0 otherwise.
- Let C_t the clustering coefficient of the graph defined by A_t.
- The resulting weighted clustering coefficient is defined as

$$\tilde{C} = \int_0^1 C_t \; dt \quad (5)$$

Since C_t can only take as many different values as the number of different edge weights in the network, the integral is actually a finite sum. However, computing C_t (which is not computationally trivial) potentially as many as $n(n-1)$ times would be very costly for large values of n , so this function has been implemented by approximating the integral dividing the interval [0,1] into `n_step` parts (where `n_step`[6]is much smaller than $n(n-1)$).

It is worth noting that some of the introduced functions (internal density, edges inside, average degree, clustering coefficient) take higher values the stronger the clusterings are, while the others (expansion, cut ratio, conductance and normalised cut) do the opposite.

3.1 Variation of information

To compare and measure how similar two clusterings of the same network are, we will use the variation of information; a criterion introduced in [7] and which is based on information theory.

Definition 3. *The entropy of a partition $\mathcal{P} = \{\mathcal{P}_1, ..., \mathcal{P}_K\}$ of a set is given by:*

$$\mathcal{H}(\mathcal{P}) = -\sum_{k=1}^{K} \frac{|\mathcal{P}_k|}{n} \log\left(\frac{|\mathcal{P}_k|}{n}\right), \quad (6)$$

where n is the size of the set and $\mathcal{P}_k$ is the k-th cluster of the partition.

[6] In this case we set `n_step`=100. This gives a reasonable resolution while keeping computations fast.

Definition 4. *Given* $P(k,k') = \frac{|\mathcal{P}_k \cap \mathcal{P}'_{k'}|}{n}$ *the joint probability distribution of elements belonging to clusters* $\mathcal{P}_k$ *and* $\mathcal{P}_{k'}$*, the mutual information is defined as:*

$$I(\mathcal{P},\mathcal{P}') = \sum_{k=1}^{K} \sum_{k'=1}^{K'} P(k,k') \log \frac{P(k,k')}{P(k)P'(k')} \tag{7}$$

Definition 5. *The variation of information between partitions* $\mathcal{P}$ *and* $\mathcal{P}'$ *is given by:*

$$VI(\mathcal{P},\mathcal{P}') = \mathcal{H}(\mathcal{P}) + \mathcal{H}(\mathcal{P}') - 2I(\mathcal{P},\mathcal{P}') \tag{8}$$

Intuitively, the mutual information measures how much knowing the membership of an element of the set in partition $\mathcal{P}$ reduces the uncertainty of its membership in $\mathcal{P}'$. This is consistent with the fact that the mutual information is bounded between zero and the individual partition entropies

$$0 \leq I(\mathcal{P},\mathcal{P}') \leq \min\{\mathcal{P},\mathcal{P}'\}, \tag{9}$$

and the right side equality holds if and only if one of the partitions is a refinement of the other. Consequently, the variation of information will be 0 if and only if the partitions are equal (up to permutations of indices of the parts), and will get bigger the more the partitions differ. It also satisfies the triangle inequality, so it is a metric in the space of clusterings of any given set.

4 Generating a random graph

The algorithm proposed here to generate a random graph which will serve as a null model is a modification of the switching algorithm described in [8]. Each step of this algorithm involves randomly selecting two edges AC and BD and replacing them with the new edges AD and BC (provided they didn't exist already). This leaves the degrees of each vertex A, B, C and D unchanged while shuffling the edges of the graph.

One way to adapt this algorithm to our weighted graphs (more specifically, complete weighted graphs, with weights in $[0,1]$) is, given vertices A,B,C and D, transfer a certain weight $\bar{w}$ from w_{AC} to w_{AD}, and from w_{BD} to w_{BC}[7]. We will select only sets of vertices A, B, C, D such that $w_{AC} > w_{AD}$ and $w_{BD} > w_{BC}$, that is, we will be transferring weight from "heavy" edges to "weak" edges. For any value of $\bar{w}$, the weighted degree of the vertices remains constant, but if it is not chosen carefully there could be undesirable consequences.

4.1 Selection of $\bar{w}$

Choosing large values of $\bar{w}$ could result in edge weights falling outside of the $[0,1]$ interval in which all of our original values are contained, but small values will hardly have similarly small effects on the network. Restricting $\bar{w}$ to be as

[7] Here, w_{ij} refers to the weight of the edge between vertices i and j

large as possible without edge weights falling out of $[0,1]$, however, will favour a degenerate network in which most of the edge weights are either 0 or 1, which is also undesirable and unlike any network that could be obtained from correlations of time series.

If we bound the transferred weight to the difference between the strong and weak edges, the new weights will be upper and lower bounded by the initial strong and weak weights, respectively, which would avoid this issue entirely. In this case, the maximum transferred weight would have to be $\bar{w} = \min(w_{AC} - w_{AD}, w_{BD} - w_{BC})$. This results in one of the pairs of edges being exchanged, while in the other a certain weight equal or smaller than their difference is transferred. In this second case, it is important to note that the difference between the new edge weights will be smaller than the difference of the original weights (strictly smaller if $w_{AC} - w_{AD} \neq w_{BD} - w_{BC}$).

The effect this has on the variance of the weights of the network can be seen in Figure 1. Unfortunately, as soon as the network starts to be significantly shuffled, the variance starts to fall. If we iterate the algorithm until the variation of information stops increasing, the variance has more than halved in our sample network.

As an alternative, we can impose the sample variance (i.e. $\frac{1}{n-1}\sum_{i,j=1}^{n}(w_i j - m)^2$, where m is the mean) to remain invariant after applying the transformation, and find the appropriate value of $\bar{w}$. The variance remains constant if and only if the following equality holds:

$$
\begin{aligned}
& (w_{AC} - m)^2 + (w_{BD} - m)^2 + (w_{AD} - m)^2 + (w_{BC} - m)^2 \\
= \;& (w_{AC} - \bar{w} - m)^2 + (w_{BD} - \bar{w} - m)^2 + (w_{AD} + \bar{w} - m)^2 + (w_{BC} + \bar{w} - m)^2 \\
\iff \;& 4\bar{w}^2 + 2\bar{w}(-(w_{AC} - m) - (w_{BD} - m) + (w_{AD} - m) + (w_{BC} - m) = 0 \\
\iff \;& 2\bar{w}^2 + \bar{w}(-w_{AC} - w_{BD} + w_{AD} + w_{BC}) = 0. \qquad (10)
\end{aligned}
$$

The solutions to this equation are $\bar{w} = 0$ (which is trivial and corresponds to not applying any transformation to the edge weights) and $\bar{w} = \frac{w_{AC}+w_{BD}-w_{AD}-w_{BC}}{2}$.

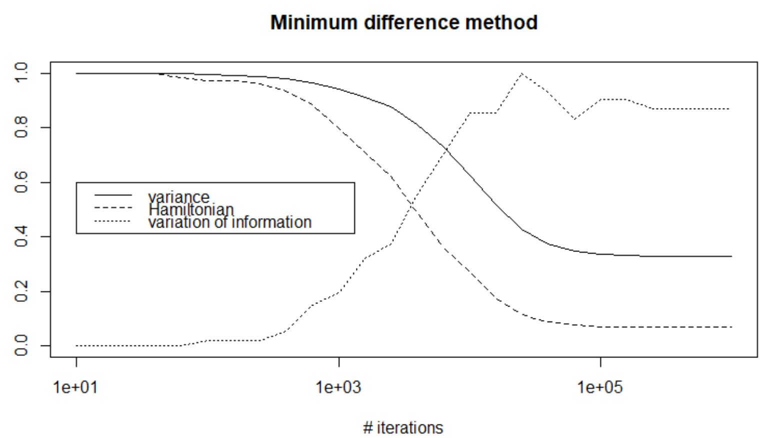
While this alternative can result in some weights falling outside of the interval $[0,1]$, in the networks we studied it is very rare, so it is enough to discard these few steps to obtain the desired results.

Note that if all edge weights are either 0 or 1, in both cases this algorithm is equivalent to the original switching algorithm for discrete graphs, as in every step the transferred weight will be one if the switch can be made without creating double edges, or zero otherwise (which corresponds to the case in which the switch cannot be made).

4.2 Number of iterations

To determine how many iterations of the algorithm are enough to sufficiently "shuffle" the network, we study the variation of information of the resulting clustering respect to the initial one (Figure 1). As the algorithm transfers weight between the edges, the variation of information increases, until it stabilises roughly after 10^4 iterations. Then, running 10^5 iterations to generate each random graph will be more than enough (there will be no improvement by iterating further) while still being very fast to compute. This is also consistent with the number of iterations found to be enough for the discrete case in [8].

Fig. 1. Normalised variance, Potts Hamiltonian and variation of information after applying the proposed algorithm with the minimum difference method. Horizontal axis is on logarithmic scale.



5 Clustering validation

To check that the results given by the clustering algorithm when applied to our FX networks are significant, we generate a random network using the method described in Section 4 for every month in the 2009-2016 period. Ideally, we would expect to see that the clusters found in the real networks are much stronger than those in the randomised networks, which shouldn't have any meaningful community structure. And indeed, we found that the clusterings of the randomised networks have many isolated vertices, and those that are grouped together are in smaller clusters than those we find in the original networks. (Nice pictures of these clusterings can be seen in the full report [13].)

In Figure 2 we verify that, across the entire observed period, the studied FX networks form larger communities than their randomised counterparts, and the number of nodes which are isolated or on very small communities is much smaller. Moreover, the value of the FX network Hamiltonian is consistently at least four times that of its corresponding randomised network using the distance correlation (Figure 3). With the Pearson correlation, the Hamiltonian varies

Fig. 2. Distribution of vertices grouped by the size of their communities, across the entire 2009-2016 period, for both the original and randomized networks.

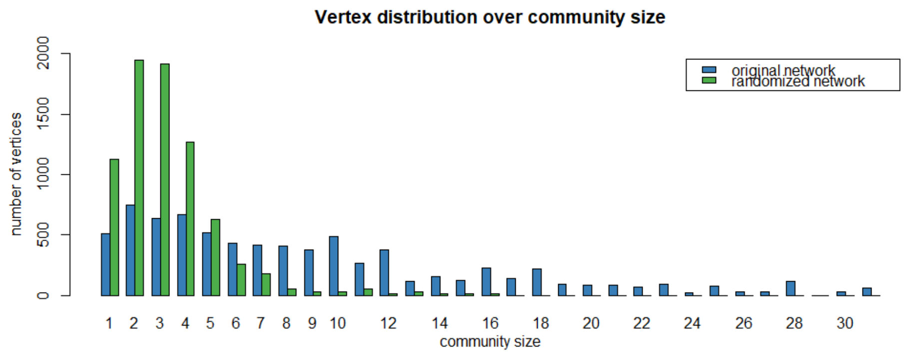
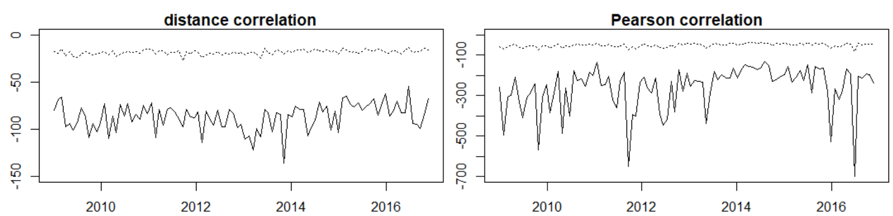


Fig. 3. Hamiltonians for the original and randomised networks (solid and dashed lines, respectively), for both the distance and Pearson correlation methods.



more but is also much lower than in the randomised network. Note though that the Hamiltonians cannot be compared across the different clustering methods, because with the Pearson correlation we need to take the inverses of each time series, resulting in a graph twice the order and four times the size.

Regarding the evolution of the scoring functions introduced in Section 3, in all cases the values of the original networks are better than those of their corresponding randomised ones[8]. We note that not only are the average scores better, but the results are consistent across all functions and periods of time. This, together with the much lower values achieved for the Hamiltonian, the objective function of the clustering algorithms, suggests that the observed community structure on our networks is significant and consistent.

While most of the values given by the scoring functions cannot be compared across the two clustering methods due to differences in the networks (their size, for example), Table 2 gives the percentage of increase of the real networks respect to the randomised models. We have observed the most dramatic increases on

[8] We omit the tables showing these explicit values due to space restrictions. Details can be seen in the full report [13]

the internal connectivity based functions on the Pearson correlation networks (probably related to the inclusion of inverse exchange rates in the network), but the decreases in external connectivity (expansion, cut ratio) are better in the distance correlation networks. The distance correlation method also performs better according to the clustering coefficient measure, with an increase that almost doubles that of the Pearson correlation.

As for the improvements in the hamiltonian, the rates of increase for both methods are very similar, but the consistency observed by the distance correlation as opposed to the highs and lows observed over time with the Pearson correlation Hamiltonian (Figure 3) could make it preferable.

Table 2. Means of the scoring functions over the 2009-2016 period for the randomised and observed networks, as well as the percentage of increase of the latter respect to the former.

	distance correlation			Pearson correlation		
	original	randomised	variation	original	randomised	variation
internal.density	0.83	0.81	1.90 %	0.85	0.91	-6.33%
edges.inside	24.02	3.00	701.49%	89.21	3.70	2313.02%
av.degree	4.44	1.62	174.37%	8.87	2.07	329.50%
expansion	16.97	18.70	-9.24%	35.46	37.96	-6.57 %
cut.ratio.	0.23	0.25	-6.87%	0.24	0.25	-3.74%
conductance	0.89	0.95	-6.24%	0.87	0.95	-8.53%
norm.cut	0.91	0.97	-5.37%	0.89	0.96	-7.32%
max.ODF	0.94	0.97	-3.84%	0.92	0.97	-5.41%
average.ODF	0.94	0.97	-3.84%	0.92	0.97	-5.40%
clustering.coef	0.89	0.75	17.82%	0.91	0.83	9.72 %
hamiltonian	-79.49	-18.24	335.79%	-259.35	-57.89	348.02 %

6 Conclusions

The clustering analysis in FX networks with appropriate scoring functions allow us to conclude that the community structure formed by the modularity maximisation algorithm is statistically significant as it is not present in random networks.

As for the comparison between clusterings relying on the distance or the Pearson correlations, the results obtained here back up the soundness of both methods, but there are slight quantitative differences worth mentioning. The distance correlation does offer some improvements in the clustering coefficient, one of the most relevant scoring functions, and the values of its Hamiltonian achieved by the optimisation algorithm, while similar on average to those of the Pearson correlation, are more consistent. Additionally, the fact that it runs the optimisation algorithm on networks of half the number of nodes greatly reduces the computation time.

It is also worth noting that while this work focused on financial networks, the methods proposed here are valid for evaluating the results of clustering algorithms on weighted networks in general.

Acknowledgments. A. Arratia acknowledge support of MINECO project APCOM (TIN2014-57226-P), and Gen. Cat. SGR2014-890 (MACDA).

References

1. Barrat, A., Barthélemy, M., Pastor-Satorras, R., Vespignani, A.: The architecture of complex weighted networks. Proceedings of the National Academy of Sciences of the United States of America 101(11), 3747–3752 (2004)
2. Blondel, V.D., Guillaume, J., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. of Stats. Mechanics: Theory and Experiment (2008)
3. Fenn, D.J., Porter, M.A., Mucha, P.J., McDonald, M., Williams, S., Johnson, N.F., Jones, N.S.: Dynamical clustering of exchange rates. Quantitative Finance 12(10), 1493–1520 (2012)
4. G. J. Székely, M. L. Rizzo, N.K.B.: Measuring and testing dependency by correlation of distances. The Annals of Statistics 35(6), 2769–2794 (2007)
5. Mantegna, R.N.: Hierarchical structure in financial markets. Eur. Phys. J. B. 11, 193–197 (1999)
6. McAssey, M.P., Bijma, F.: A clustering coefficient for complete weighted networks. Network Science 3(2), 183–195 (2015)
7. Meilă, M.: Comparing clusterings - an information based distance. Journal of Multivariate Analysis 98(5), 873 – 895 (2007)
8. Milo, R., Kashtan, N., Itzkovitz, S., Newman, M.E.J., Alon, U.: On the uniform generation of random graphs with prescribed degree sequences. Arxiv preprint cond-mat/0312028 (2003)
9. Mucha, P.J., Richardson, T., Macon, K., Porter, M.A., Onnela, J.P.: Community structure in time-dependent, multiscale, and multiplex networks. Science 328(5980), 876–878 (2010)
10. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E 69, 026113 (2004)
11. Onnela, J.P., Kaski, K., Kertész, J.: Clustering and information in correlation based financial networks. The European Physical Journal B-Condensed Matter and Complex Systems 38(2), 353–362 (2004)
12. Otranto, E.: Clustering heteroskedastic time series by model-based procedures. Computational Statistics & Data Analysis 52(10), 4685–4698 (2008)
13. Renedo, M.: On methods to assess the significance of community structure in networks of financial time series. Master's thesis, FME, Universitat Politècnica de Catalunya (2017), http://www.lsi.upc.edu/∼argimiro/mystudents/TFM_MartiRenedo.pdf
14. Renedo, M., Arratia, A.: Clustering of exchange rates and their dynamics under different dependence measures. In: Proc. of the First Workshop on MIning DAta for financial applicationS (MIDAS 2016), collocated ECML-PKDD. pp. 17–28 (2016)
15. Yang, J., Leskovec, J.: Defining and evaluating network communities based on ground-truth. Knowledge and Information Systems 42(1), 181–213 (2015)