Applied Network Science

RESEARCH Open Access

Graph-based data clustering via multiscale community detection

Zijing Liu[1] and Mauricio Barahona[1,2*]

*Correspondence:
m.barahona@imperial.ac.uk
[1]Department of Mathematics, Imperial College London, South Kensington Campus, SW7 2AZ London, UK
[2]EPSRC Centre for Mathematics of Precision Healthcare, Imperial College London, SW7 2AZ London, UK

Abstract

We present a graph-theoretical approach to data clustering, which combines the creation of a graph from the data with Markov Stability, a multiscale community detection framework. We show how the multiscale capabilities of the method allow the estimation of the number of clusters, as well as alleviating the sensitivity to the parameters in graph construction. We use both synthetic and benchmark real datasets to compare and evaluate several graph construction methods and clustering algorithms, and show that multiscale graph-based clustering achieves improved performance compared to popular clustering methods without the need to set externally the number of clusters.

Keywords: Multiscale community detection, Clustering, Graph partitioning, Data mining

Introduction

Clustering is a classic task in data mining, whereby input data are organised into groups (or clusters) such that data points within a group are more similar to each other than to those outside the group (Xu and Wunsch 2005). Such a task is distinct from supervised (or semi-supervised) classification, where examples of the different classes are known a priori and are used to train a computational model to assign other objects to the known groups. Instead, clustering aims to find natural, intrinsic sub-classes in the data, without assuming a priori the number or type of clusters. Indeed, a key open issue in this field is the principled determination of the number of clusters in an unsupervised manner, without the assumption of a generative model (Sugar and James 2003; Azran and Ghahramani 2006). The obtained groups can then constitute the basis for a simpler, yet informative, representation of large, complex datasets.

Data clustering has a long history and there exist a myriad of clustering algorithms based on different principles and heuristics (Jain et al. 1999). In their most basic form, many popular clustering techniques (e.g., k-means (MacQueen 1967) and mixture models (Dempster et al. 1977)) are based on the assumption that the data follows an explicit (typically multivariate Gaussian) distribution. Clusters are then defined as the samples most likely generated from the same distribution, and learned by likelihood maximisation. However, in real applications, the model that generates the data is unknown and the

Springer Open

resulting data distribution may be complex. In this case of data-driven analysis, model-based clustering often yields poor results (Shi and Malik 2000; Ng et al. 2001; de Sa 2005; Ye et al. 2016).

An alternative approach is provided by spectral clustering, which uses the eigenvectors of a (normalised) similarity matrix derived from the data to find relevant subgroups in the dataset (Ng et al. 2001; Von Luxburg 2007). Spectral clustering is underpinned by results in matrix analysis (e.g., singular value decomposition), and has strong connections to model reduction, geometric projections and dimensionality reduction (Von Luxburg 2007; Schaub et al. 2019). The choice of similarity measure is a crucial ingredient to the clustering performance but, as long as a similarity matrix can be computed, spectral methods provide an attractive choice for non-vector data or for data sampled from irregular and non-convex data manifolds (Alpert et al. 1999; Dhillon 2001).

From a different perspective, the similarity matrix of a dataset can also be viewed as the adjacency matrix of a fully connected, weighted graph, where the nodes correspond to data points and the edge between two nodes is weighted by their similarity. One can then apply graph-based algorithms for community detection or graph partitioning to the problem of data clustering. Graph-based methods typically operate by searching for balanced graph cuts, sometimes invoking notions from spectral graph theory, i.e., using the spectral decomposition of the adjacency or Laplacian matrices of the graph (Hagen and Kahng 1992; Chung 1997). Spectral clustering can thus be understood as a special case of the broader class of graph-based clustering methods (Schaub et al. 2019). Importantly, graph-based clustering is also able to reveal modular structure in graphs across levels of resolution through multiscale community detection (Lambiotte et al. 2008; 2014; Delvenne et al. 2010). This approach allows for the discovery of natural data clusterings of different coarseness (Altuncu et al. 2019), thus recasting the problem of finding the appropriate number of clusters to the detection of relevant scales in the graph.

Methods for graph construction usually involve a sparsification of the similarity (or distance) matrix under different heuristics (from simple thresholding to sophisticated regularisations) in order to extract a *similarity graph* that preserves key properties of the dataset (Cheng et al. 2010). The representation of data through graphs has attractive characteristics, including the capability of capturing efficiently the local and global properties of the data through graph-theoretical concepts that embody naturally the notions of local neighbourhoods, paths, and global connectivity (Tenenbaum et al. 2000; Beguerisse-Díaz et al. 2013; Lambiotte et al. 2014). The usage of graphs provides a natural links of spectral clustering with other clustering methods and allows for easy generalisation to a semi-supervised setting (Dhillon et al. 2004; Kulis et al. 2009). Graphs also provide a means to capture the geometry of complex manifolds, a feature of interest in realistic datasets (Bronstein et al. 2017). Graph representations not only reduce the computational cost for spectral graph methods, but also allow us to use the techniques developed for complex networks as an alternative to address problems in data clustering. However, it has been shown that both the method of graph construction and the choice of method parameters (i.e., sparsity) have a strong impact on the performance of graph-based clustering methods (Maier et al. 2008; Daitch et al. 2009; Maier et al. 2013; Jebara et al. 2009).

Here, we study the use of multiscale community detection applied to similarity graphs extracted from data for the purpose of unsupervised data clustering. The basic idea of graph-based clustering is shown schematically in Fig. 1. Specifically, we focus on the

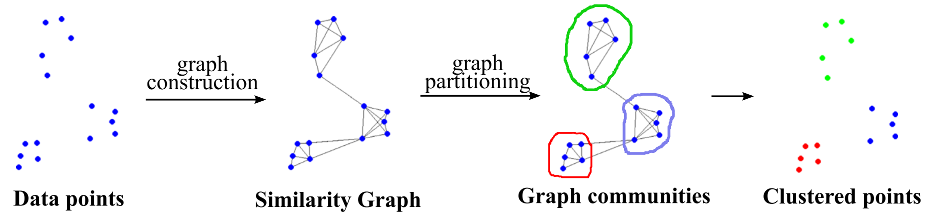

Fig. 1 Schematic diagram of the workflow for graph-based clustering

problem of assessing how to construct graphs that appropriately capture the structure of the dataset with the aim of being used within a multiscale graph-based clustering framework. In particular, we carry out an empirical study of different graph construction methods used in conjunction with Markov Stability (MS), a dynamics-based framework for multiscale community detection (Delvenne et al. 2010; Delvenne et al. 2013). As a dynamics-based framework (Delvenne et al. 2013; Lambiotte et al. 2014; Delvenne et al. 2013), MS provides a unified framework for many multiscale community detection algorithms, such as the RB Potts model (Reichardt and Bornholdt 2006), the constant Potts model (Traag et al. 2011) and the absolute Potts model (Ronhovde and Nussinov 2010), and allows for the unsupervised community detection at different levels of resolution. MS has also been shown to allow for the detection of both clique-like and nonclique-like communities in graphs (Schaub et al. 2012). MS has been applied successfully to a variety of problems, including protein structures (Delmotte et al. 2011; Amor et al. 2014), airport networks (Lambiotte et al. 2014), social networks (Beguerisse-Díaz et al. 2014) and neuronal network analyses (Bacik et al. 2016). Other dynamical processes have been extensively applied in network analysis, such as temporal networks (Petri and Expert 2014), crowded networks (Asllani et al. 2018) and network classification (Tran et al. 2019).

In this paper, we evaluate several geometric graph constructions, from methods that use only local distances to others that balance local and global measures, and find that the recently proposed Continuous k-nearest neighbours (CkNN) graph (Berry and Sauer 2019) performs well for graph-based data clustering via community detection. We then show how the multiscale capabilities of the Markov Stability to scan across scales can be exploited to deliver robust clusterings, reducing the sensitivity to the parameters of the graph construction. In other words, a range of parameters in the graph construction lead to good clustering performance. We validate our graph-based clustering approach on real datasets and compare its performance to several other popular clustering methods, including k-means, mixture models, spectral clustering and hierarchical clustering (Rokach and Maimon 2005).

The rest of the paper is structured as follows. We first introduce several methods for graph construction, apply them to eleven public datasets with ground truths, and evaluate the performance of graph-based data clustering on the ensuing similarity graphs. We then describe briefly the Markov Stability framework for multiscale community detection, and use a synthetic example dataset to illustrate how the multi-resolution clustering reduces the sensitivity to graph construction parameters. Finally, we validate the Markov Stabiity graph-based clustering through comparisons with other clustering methods on real datasets.

Graph construction methods for data clustering

Let us consider a dataset consisting of n samples $\{\mathbf{y}_i\}_{i=1}^{n}$, where each sample $\mathbf{y}_i$ is a d-dimensional vector. We will assume that we can define a measure of pairwise dissimilarity between the samples: $d(i,j) \geq 0$. In some cases, the vectors will be defined in a metric space, and the dissimilarity will be a true distance $d(i,j) = ||\mathbf{y}_i - \mathbf{y}_j|| \geq 0$. Other dissimilarity measures, such as the cosine distance, can also be used depending on the application. In the examples below, we will restrict ourselves to Euclidean distances as the measure of dissimilarity.

The high dimensionality of data usually leads to complex and non-linear geometries associated with datasets, posing challenges to standard clustering methods. The aim of transforming the data into a graph is to capture the complex geometry of the data through the graph topology (Tenenbaum et al. 2000), so as to reveal the structure of the data via graph-theoretical concepts and tools from complex network analysis. There exist a variety of ways to construct a graph from a high-dimensional dataset, invoking different principles. Here we focus on geometric graphs and examine two of the most widely used methods (ϵ-ball graph, k-nearest neighbour (kNN) graph) and three recent methods (continuous k-nearest neighbours (CkNN) graph (Berry and Sauer 2019), perturbed minimum spanning tree (PMST) (Carreira-Perpiñán and Zemel 2004) and relaxed minimum spanning tree (RMST) (Beguerisse-Díaz et al. 2013)). These methods can be broadly ascribed to two categories depending on how they use the geometry of the data, as follows.

Neighbourhood based methods: ϵ-ball, kNN and CkNN graphs

The idea of neighbourhood based methods is to connect two nodes if they are local neighbours, as given by their pairwise distance $d(i,j)$. The two simplest and most popular ways to construct a graph from pairwise distances are the ϵ-ball graph and the k-nearest neighbour graph (kNN): in the ϵ-ball graph, any two points at a distance smaller than ϵ are connected; in the kNN graph, every point is connected to its k-th nearest neighbours. These two methods capture the local information of the data but are highly sensitive to the parameters ϵ or k (Maier et al. 2008). The parameter is usually set according to the density of the data points but in many datasets, the data points are not uniformly distributed.

A recently proposed method that can resolve this problem is the continuous k-nearest neighbours (CkNN) graph (Berry and Sauer 2019). If $d(i,j)$ is the distance between sample i and sample j, and $d^k(i)$ is the distance between sample i and its k-th nearest neighbour, the CkNN graph is constructed by connecting sample i and sample j if

$$d(i,j) < \delta\sqrt{d^k(i)d^k(j)}, \tag{1}$$

where δ is a positive parameter that controls the sparsity of the graph. Through this construction, the topology of the CkNN graph captures the geometric features of the data with the additional consistency that the CkNN graph Laplacian converges to the Laplace-Beltrami operator in the limit of large data (Berry and Sauer 2019). It should be kept in mind that in CkNN the distance $d(i,j)$ must be a metric to ensure geometrical consistency, whereas kNN graphs can be generated from any dissimilarity measure (i.e., one does not need a true distance) since only the ranking of node closeness matters.

Minimum spanning tree based methods: PMST and RMST graphs

A different class of approaches for graph construction attempt to capture the global geometry of the overall dataset by constructing graphs based on measures of global connectivity of the ensuing graph. A popular way to ensure such global connectivity is through the minimum spanning tree (MST) (Cormen et al. 2009), as follows. If we consider the matrix of all pairwise distances $d(i,j)$ as the adjacency matrix of a weighted, fully connected graph, the MST is the subgraph such that all the nodes are path connected and the sum of edge weights is minimised. In other words, the MST provides a graph that connects all the points in the dataset with minimal *global* distance. MST-based approaches can thus capture the geometry of in-homogeneously sampled data points in a high-dimensional space since the MST contains not only local but also global features of the dataset.

In its simplest form, the MST is sometimes added to sparse neighbourhood graphs as a means to guarantee global connectivity of the dataset, i.e., the final graph is the union of the MST and a kNN graph with a small k. These schemes, which are sometimes referred as MST+kNN graphs, are the ones we adopt by default in our neighbourhood constructions. However, the global properties of the MST can be exploited to generate MST-based graphs from data with distinct properties. We have explored here the use of two such MST-based algorithms: the perturbed minimum spanning tree (PMST) (Carreira-Perpiñán and Zemel 2004) and the relaxed minimum spanning tree (RMST) (Beguerisse-Díaz et al. 2013; Vangelov 2014).

In PMST (Carreira-Perpiñán and Zemel 2004), each data point $\mathbf{y}_i$ is perturbed by a small amount of noise of standard deviation $s_i = rd^k(i)$ $(r \in [0,1])$ where $d^k(i)$ is used as an estimation of the local noise and r is a parameter controlling the level of noise. The MST is then computed for each realisation of the perturbed data and the process is performed repeatedly to generate an ensemble of MSTs. The PSMT graph is given by the union of all the perturbed MSTs, plus the original MST. The intuition behind this algorithm is that random perturbations of the points in high-dimensional space will induce changes inhomogeneously in different parts of the MST, depending on how globally important certain edges of the graph are, i.e., globally important edges will be consistently captured across all MSTs in the perturbed ensemble. One limitation of this algorithm is the heavy computational demand, since both the distance matrix and MST need to be computed for each random realisation. The computational burden makes it impractical to sweep over the parameters of the PMST, so we fix $r = 0.5$ and $k = 1$ in this paper.

RMST (Vangelov 2014) proposes a different heuristic for estimating an MST-based graph with lower computational cost. Note that any two nodes are connected by a single path in the MST and we denote the longest edge on this path as $d^{\max}_{\text{path}(i,j)}$. In RMST, the samples $\mathbf{y}_i$ and $\mathbf{y}_j$ are connected if

$$d(i,j) < d^{\max}_{\text{path}(i,j)} + \gamma \left(d^k(i) + d^k(j) \right), \tag{2}$$

where $\gamma > 0$ is a parameter that weights the local density (measured by the average distance to the k-th neighbours of $\mathbf{y}_i$ and $\mathbf{y}_j$) against a global property (the maximum distance found on the MST path linking the samples i and j). Similarly to the parameter δ in CkNN, the value of γ controls the sparsity of the resulting graph. The RMST construction was proposed as a means to reconstruct data that have been inhomogeneously sampled from continuous manifolds, and has been shown to provide good description of datasets when preserving a measure of continuity (due to temporal or parametric changes) is important (Beguerisse-Díaz et al. 2013).

In Fig. 2, we use a synthetic dataset of four groups of points sampled from a geometric structure (a noisy circle and its centre) to illustrate the different graph construction schemes (Ben-Hur et al. 2001). The graph representations allow us to gain intuition about the suitability of the different graph constructions for clustering, and the effect of their parameters. For instance, the sparsity of the RMST graph is controlled by the parameter γ. Note that RMST gives a similar graph to PMST with much lower computational cost. Note also that, for the same number of neighbours k, the CkNN gives a sparser graph than the kNN. To make CkNN and kNN comparable, we thus fix the parameter $\delta = 1$ in CkNN and vary k.

As discussed above, the MST-based methods are not optimised for clustering but aimed at *manifold learning*. It follows from (2) that, in the RMST (and PMST) graphs it is more likely for an edge to appear between node i and node j if the longest edge on the MST path between node i and j is large. This feature makes the geodesic distance on the graph a good approximation to the true distance in the underlying space. Hence both RMST and PMST are closer to *manifold learning* approaches such as Isomap (Tenenbaum et al. 2000)). In contrast, the kNN and CkNN graphs tend to have better modular structures and hence appear as potentially more suitable for graph partitioning. We compare these issues in detail in "Tests on benchmark real datasets" section below.

Markov stability for graph-based clustering

Let us consider a graph representing the dataset. The unweighted and undirected graph with n nodes representing the dataset is encoded by the adjacency matrix A, where $A_{ij} = 1$ if there is an edge connecting node i and j and $A_{ij} = 0$ otherwise. The degree of the nodes is summarised in the degree vector $\mathbf{d}$ where $d_i = \sum_j^n A_{ij}$, and we also define the diagonal degree matrix D where $D_{ii} = d_i$. The total number of edges of the network is $m = \sum_{i,j} A_{ij}/2$. We then apply multiscale community detection to extract relevant subgraphs in an unsupervised manner using the framework of Markov Stability.

Multiscale community detection with Markov Stability

Markov Stability is a quality measure for community detection which adopts a dynamical perspective to unfold relevant structures in the graph at all scales as revealed by a diffusion process (Lambiotte et al. 2008; 2014; Delvenne et al. 2010; Schaub et al. 2012). Consider a continuous-time Markov process on the graph governed by the dynamics $\dot{\mathbf{p}} = -\mathbf{p}(I - M)$ where $\mathbf{p}$ is an n-dimensional row vector defined on the nodes and $M = D^{-1}A$ is the one step random walk transition matrix. For this Markov process, there is a unique stationary distribution $\boldsymbol{\pi} = \mathbf{d}^T/2m$. Let us denote the autocovariance matrix of this process as $B(t) = \Pi P(t) - \boldsymbol{\pi}^T\boldsymbol{\pi}$ where $\Pi = D/2m$ encodes the stationary distribution and $P(t) = \exp(-t(I - M))$ is the transition matrix. Given a partition g of the nodes into c non-overlapping groups denoted by $g = \{g_1, g_2, ..., g_c\}$, the Markov Stability of g is defined as:

$$r(t, g) = \sum_{s=1}^{c} \sum_{i,j \in g_s} B(t)_{ij}. \tag{3}$$

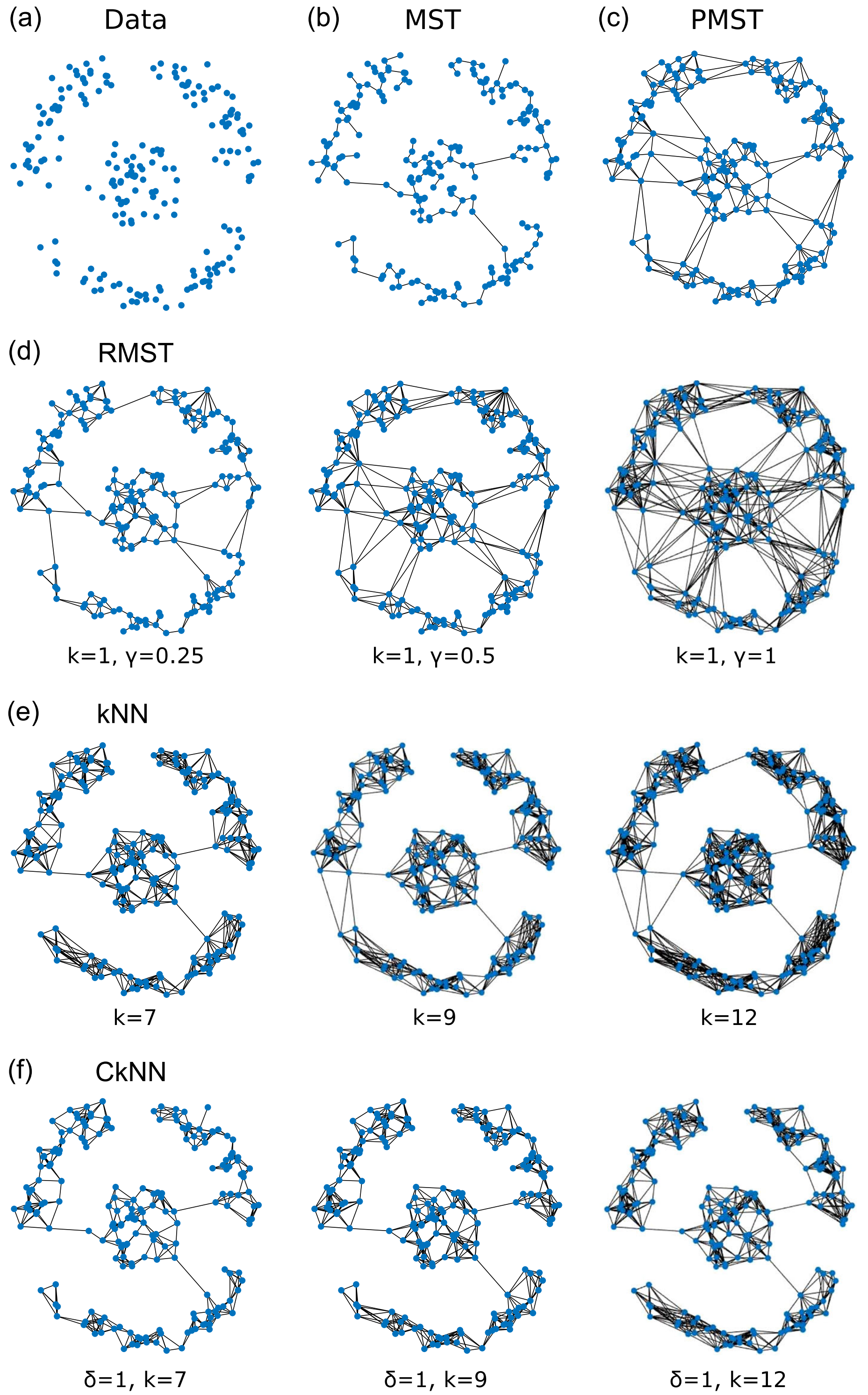

Fig. 2 Illustration of the differences introduced by graph construction methods. **a** Synthetic dataset consisting of four groups of randomly sampled points: three groups are sampled from a circle and one group from its centre. **b** The minimum spanning tree (MST) of the dataset based on Euclidean distances. We also present the graphs constructed with MST-based methods: **c** PMST and **d** RMST, as well as neighbourhood methods: **e** kNN and **f** CkNN. RMST, kNN and CkNN are shown for different parameters that vary the sparsity of the resulting graphs

A partition has a high value of (3) if the probability of finding a random walker at time t within the group where it started at $t = 0$ is higher than that expected by mere chance. In this sense, Markov Stability is a quality function for partitions of a graph and the objective is therefore to find the partitions that achieve high values of the Markov Stability as a function of t:

$$r^*(t) = \max_g r(t, g) \quad \text{achieved by the partition } g^*(t). \tag{4}$$

The (time) parameter t is the so-called Markov time, and can be understood as the resolution parameter that leads to multiscale community detection (Delvenne et al. 2013; Schaub et al. 2012). For small t, the number of detected communities is large and the communities capture the local information of the graph. As t becomes larger, there are fewer communities and the communities are able to capture the global features of the graph. Computationally, the Markov Stability is optimised at different Markov times through a version of the Louvain algorithm (Blondel et al. 2008).

Through the computational maximisation (4), Markov Stability detects optimised partitions $g^*(t)$ at all scales, parameterised by the value of t. However, we are interested in finding robust partitions and robust scales, in the sense that a partition is found to optimise MS over a long interval of Markov time. We thus compute the dissimilarity between the obtained partitions at different times t and t':

$$VI(t, t') = VI(g^*(t), g^*(t')), \tag{5}$$

where we use the variation of information (VI) (Meilă 2003) as the metric of dissimilarity between partitions. If $g^*(t)$ is a robust partition, the partition $g^*(t')$ found at a Markov time t' close to t should be very similar, and hence $VI(t, t')$ will be small. We therefore look for large diagonal blocks of small values in the $VI(t, t')$ matrix. Such blocks correspond to a robust scale with an associated robust partition.

As an additional feature, we look for optimised partitions that are also robust to the Louvain optimisation. Since the Louvain method is a greedy algorithm dependent on the random initialisation, the consistency of the output of the algorithm can be used as an indicator of the robustness of the solution. At each t, we run the Louvain optmisation multiple times and if the Markov time corresponds to a robust scale, the output partition should be always the same. Therefore we expect a low value of the average variation of information of the optimised partitions at time t

$$VI(t) = \frac{1}{n_L(n_L - 1)} \sum_{s=1}^{n_L} \sum_{s'=1}^{n_L} VI(g_s(t), g_{s'}(t)), \tag{6}$$

where the Louvain algorithm is run n_L times. Depending on the structure of the graph, several such robust scales and associated graph partitions might be found, which can then be used as the basis of unsupervised data clustering.

Using Markov stability for data clustering

We illustrate the application of MS to data clustering through the synthetic dataset in Fig. 3. The example dataset has geometric struture and is designed to have two scales, so that it can be divided into 3 big clusters or 9 small clusters. First, we construct an unweighted CkNN graph ($k = 7$, $\delta = 1.8$) and apply MS as described above. We optmise

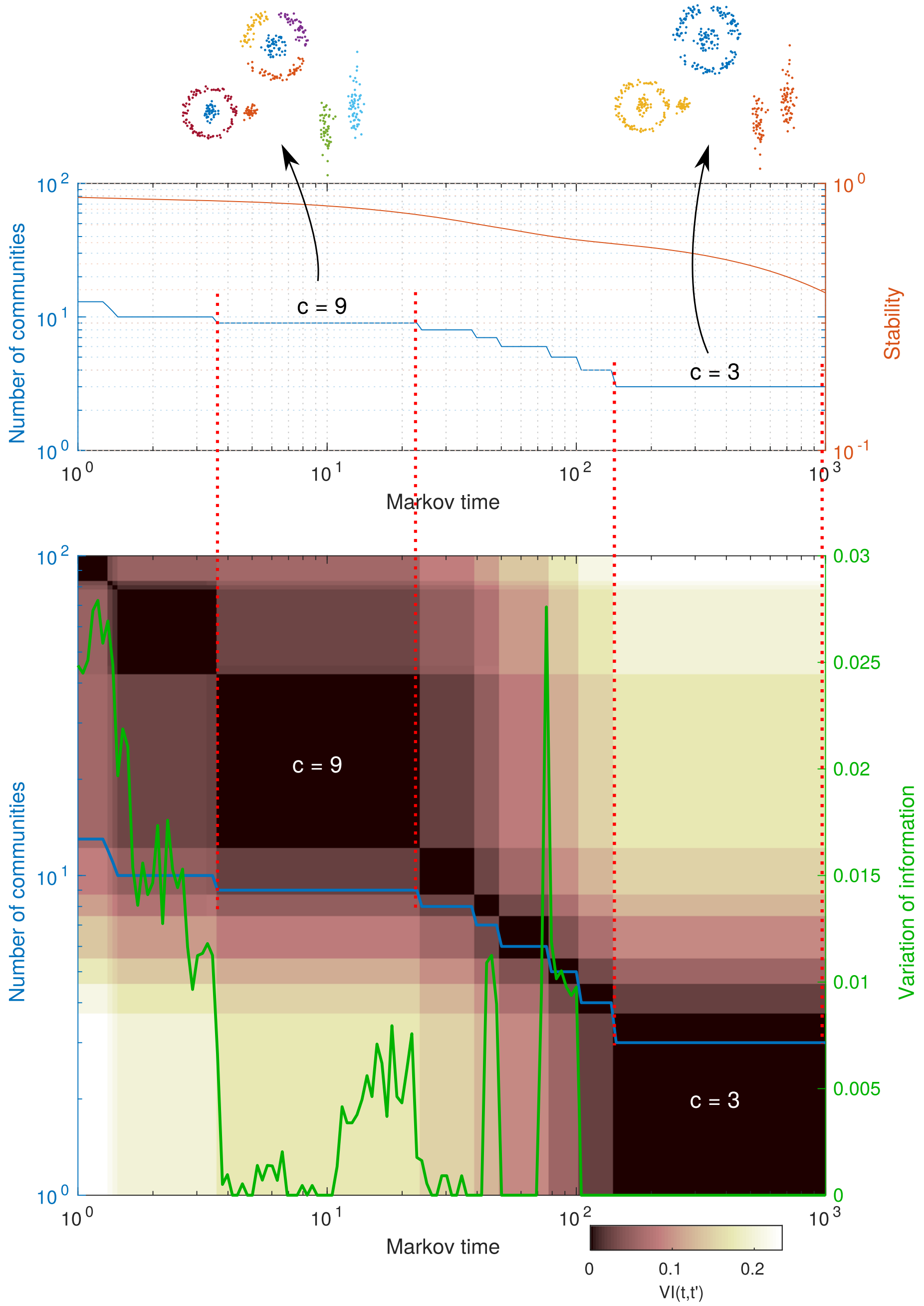

Fig. 3 Graph-based unsupervised analysis using Markov Stability of a synthetic dataset with multiscale geometric structure. We generate a CkNN graph ($k = 7, \delta = 1.8$) and scan across Markov time from 1 to 1000. Two significant scales corresponding to robust partitions are identified based on the zero blocks of the $VI(t, t')$ matrix, the long plateaux in the number of communities, and the low values of $VI(t)$. Note that with Markov Stability, there is a range of δ for the CkNN graph ($k = 7$) that can reveal the multiscale structure of the data (see Fig. 5)

the Markov Stability (3) for n_T Markov times $t \in [1, 1000]$, and at each t, we run the Louvain algorithm $n_L = 500$ times. For each Markov time, we record the partition with the maximal Markov Stability, $g^*(t)$, and the average dissimilarity of the partitions found in the n_L optimisations, $VI(t)$ (6). Once the scan across Markov time is completed, we also compute $VI(t,t')$, the matrix recording the dissimilarity of the optimal partitions found across the scan.

The results are presented in Fig. 3, where, as a function of Markov time t, we plot the number of communities in the optimal partition $g^*(t)$; the optimised Markov Stability $r^*(t)$ (3); the average dissimilarity due to algorithmic variability $VI(t)$; and the dissimilarity of partitions across time given by the $VI(t,t')$ matrix. The diagonal blocks of low values of $VI(t,t')$ (which also correspond to plateaux in the number of communities) and the low values (or dips) of $VI(t)$ suggest that there are two relevant scales in this graph, which correspond to a finer partition into 9 groups (at small t) and a partition into 3 groups (at larger $t > 200$). The inset shows that the partition recovers the planted groups of this synthetic example.

The robustness of the partitions found across scales is further examined in Fig. 4. To understand how the graph partitions evolve with t, we compute the VI metric between all the partitions found across the Markov time scan ($n_L \times n_T$) and project them on a low dimensional space using multidimensional scaling (MDS). In Fig. 4, we use the first MDS coordinate of each of the partitions as a function of t coloured by its number (frequency) of appearances out of the n_L Louvain runs at each Markov time. Several partitions can coexist at a given Markov time, but our numerics show that the two robust partitions ($c = 9$ and $c = 3$) have a long-lived high frequency of appearance when the t matches the corresponding resolution. Between $c = 9$ and $c = 3$, other partitions of lesser robustness appear through mergers of clusters one by one, as shown by the Sankey diagramme in Fig. 4, and only exist for short Markov times until the robust partition of $c = 3$ appears.

One advantage of using community detection for data clustering is the computational efficiency of fast community detection algorithms (Fortunato 2010). Empirically, the Louvain algorithm scales nearly linearly with the size of the dataset (Blondel et al. 2008). In its full form, Markov Stability has been applied to community detection in networks of sizes up to tens of thousands of nodes (Delmotte et al. 2011; Altuncu et al. 2019). To give an indication, running a full MS scan for a dataset of 2310 samples sweeping 100 Markov times with 100 Louvain runs at each Markov time on a desktop [1] requires about 20 min. The space complexity of the graph-based clustering is dominated by the storage of the graph, i.e., similar to spectral clustering if the adjacency matrix is used. For very large datasets(> 20000), the computational complexity is dominated by the matrix exponential in $P(t)$. The use of linearised (approximate) version of MS allows to scale its use up to graphs with hundreds of thousands of nodes with reduced CPU times, at the cost of some reduction in the quality of the clusterings (Delvenne et al. 2013). The linearised versions also allows the storage of the graph in a sparse matrix to reduce the space complexity.

Scanning across Markov time reduces the sensitivity to graph construction

Graph-based clustering performance is sensitive to the parameters of the graph construction method, which modulate sparsity, but there is no easy way to select the best parameter if the ground truth is unknown. In practice, the parameters are usually set empirically with little guidance that the chosen parameter will lead to good clustering results for a particular dataset. Within the Markov Stability framework, we can use the

[1] With a 4-core 3.4GHz CPU and MATLAB implementations

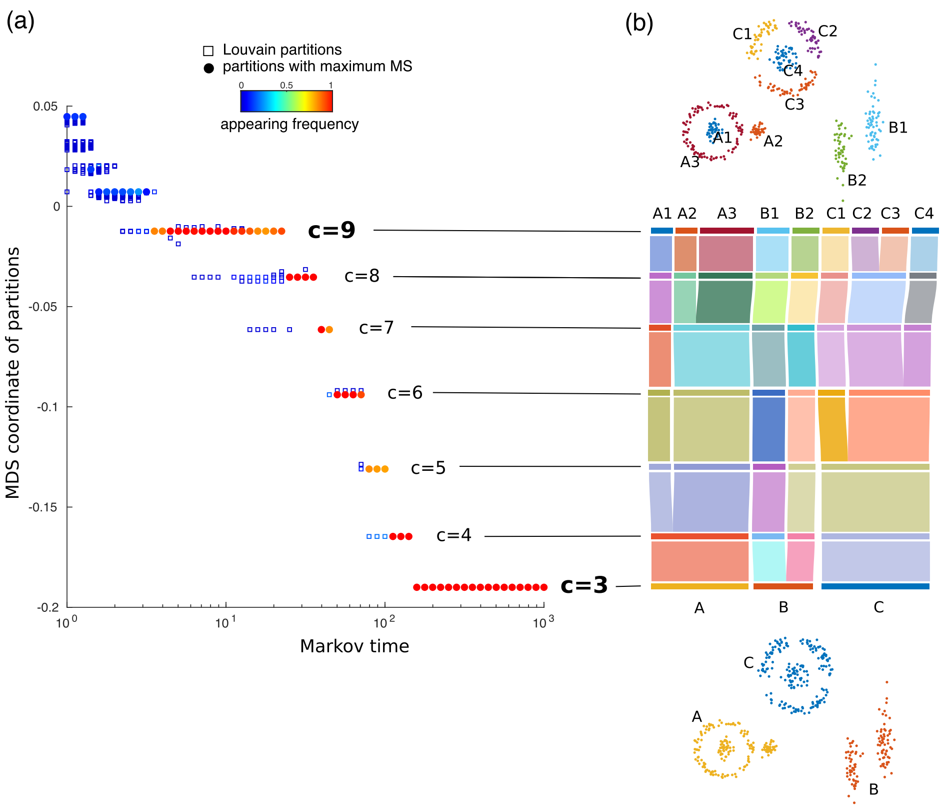

Fig. 4 Robustness of the partitions found through Markov Stability for the CkNN graph with $k = 7$ and $\delta = 1.8$ of the dataset in Fig. 3. **a** We show the one-dimensional MDS embedding (based on the *VI* distance) of all the graph partitions found by the n_L Louvain optimisations for each of the n_T Markov times. The partition with the maximal Markov Stability at each Markov time is labelled by a filled circle, whereas the non-optimal partitions are represented by open squares. The graph partitions are coloured according to the frequency of their appearance in the optimisations. **b** The Sankey diagram represents the relationship between the partitions found by MS across Markov time, from $c = 9$ to $c = 3$. Note the hierarchical structure which is not pre-imposed by the algorithm but emerges from the intrinsic properties of the dataset

robustness provided by scanning across Markov time to reduce the sensitivity to the details of graph construction, thus improving the reliability of the detected clusters.

To illustrate this idea, We use the same dataset in Fig. 3 and construct CkNN ($k = 7$) graphs with different values of $\delta = 1.5, 1.8, 2.4$. Our numerics show that Markov Stability detects the relevant underlying scales of the data ($c = 9$ and $c = 3$) for the different values of δ, as shown by the long diagonal blocks of low $VI(t, t')$ and the low values of $VI(t)$ in Fig. 5. Although the degree of the CkNN graph varies markedly with the parameter δ, the two significant scales are identified by scanning across the Markov time. Hence the scanning across scales inherent to multiscale community detection provides additional robustness to the parameters of the graph construction algorithm. We have also carried out a similar analysis of the clusterings for two real datasets (‘WBDC’ and ‘Control charts’). The clusterings remain robust when varying k in CkNN (see Additional file 1: Figure S1 and Additional file 2: Figure S2).

Tests on benchmark real datasets

We have tested several graph-based clustering approaches (both graph constructions and clustering methods) using eleven benchmark datasets from the UCI repository (see

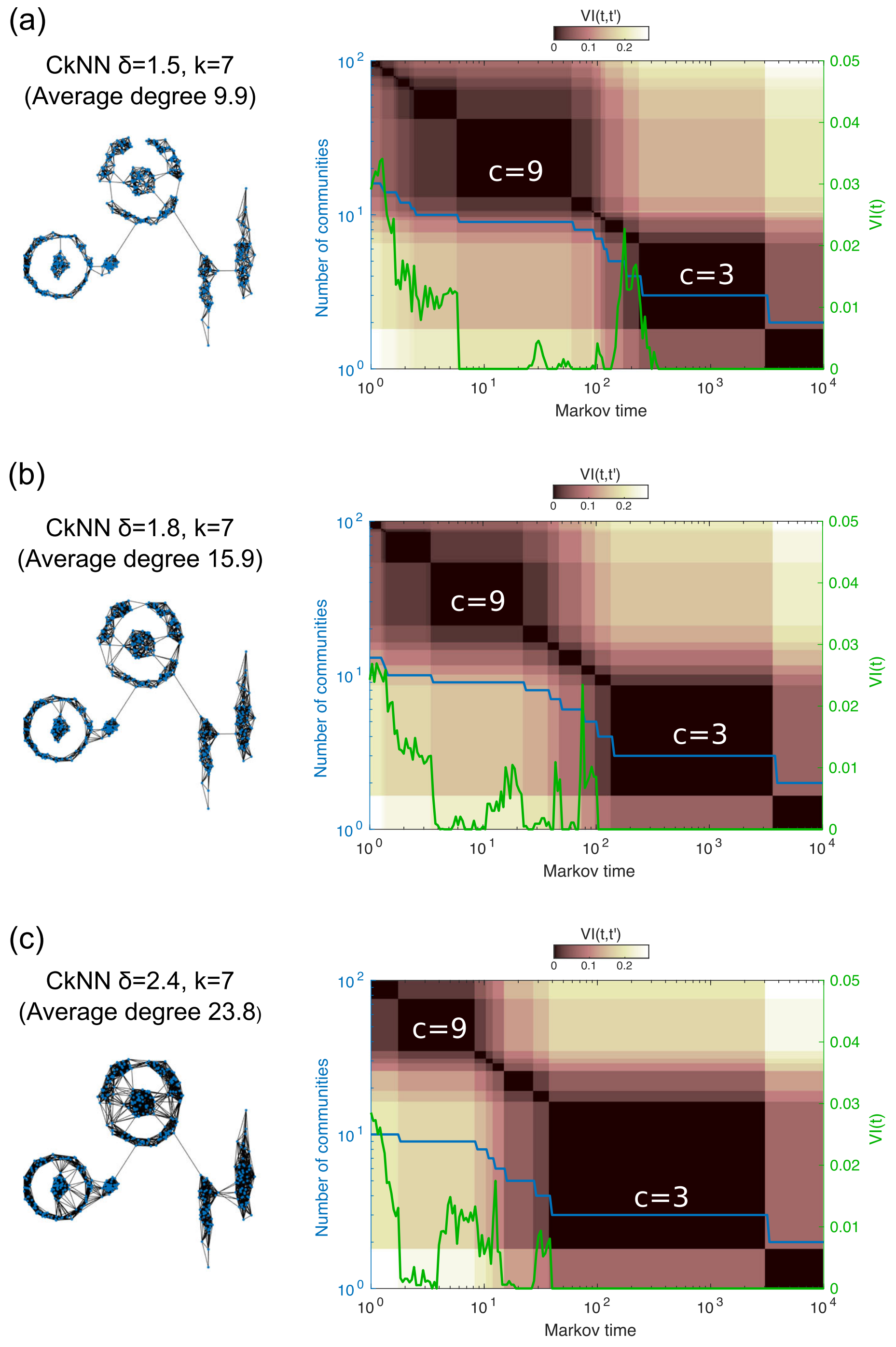

Fig. 5 The multiscale character of Markov Stability alleviates the sensitivity to the parameters of graph construction. Using the same synthetic dataset as in Fig. 3, MS finds the robust partitions and scales in the data ($c = 3$ and $c = 9$) in an unsupervised manner for CkNN graphs constructed with a range sparsities as given by varying the parameters: $k = 7$ and $\delta = 1.5, 1.8, 2.4$ in (**a**), (**b**) and (**c**), respectively. Markov time, as a resolution parameter, allows the community algorithm to reveal the local and global properties of the graph constructed from the data

Table 1 Attributes of the eleven real datasets from the UCI repository used in this study

Dataset	No. of samples, *n*	Dimension, *d*	No. of classes, *c**
Iris	150	4	3
Glass	214	9	6
Wine	178	13	3
WBDC	569	30	2
Control chart	600	60	6
Parkinsons	195	22	2
Vertebral	310	6	3
Breast tissue	106	9	6
Seeds	210	7	3
Image Segmentation	2310	19	7
Yeast	1484	8	10

Table 1 for a summary of attributes) (Dheeru and Karra Taniskidou 2017). All the datasets have ground truth labels, which we use to validate the results of the different methods.

Comparison between graph constructions

Starting from the Euclidean distance $d(i,j) = ||\mathbf{y}_i - \mathbf{y}_j||_2$, we generated geometric graphs from each of the eleven UCI datasets using the five graph construction methods described in "Graph construction methods for data clustering" section. If the constructed graph is disconnected, the MST is added to the graph to ensure a connected graph. Each graph was analysed using Markov Stability to obtain optimised partitions at any scale, and we selected the closest of those partitions to the ground truth, as measured by the normalised mutual information (NMI) (Strehl and Ghosh 2002). The computed NMI is a quality index for the graph construction under clustering. We also compute the adjusted Rand index (ARI) (Hubert and Arabie 1985) as an additional quality index.

The results of the comparison are shown in Table 2. The kNN and ϵ-ball graphs, both widely-used in many machine learning and both based on local neighbourhoods, give good results for a range of k. (Note that ϵ is set to be the average of the distances to the 7-th neighbour, $d^7(i)$.) For the MST-based methods, RMST achieves better performance for sparser graphs (with smaller γ) when the cluster structure is not obscured by the objective of manifold reconstruction (Fig. 2). The same applies to the PMST graph. The empirical tests show that the CkNN graph gives the best average results over the eleven datasets for $k = 7$ and above. Hence we adopt the CkNN graph with k in the range of 7 to 12 as a good choice for graph-based clustering.

The CkNN graph is constructed by using a variable bandwidth diffusion kernel (Berry and Harlim 2016) where the bandwidth is inversely proportional to the sampling density and allows for uniform estimation errors over the underlying manifold, while the graph construction which uses a fixed bandwidth kernel will have large errors in areas of small sampling density. By such a construction, the graph Laplacian of the CkNN graph converges to the Laplacian operator in the manifold. This explains why the CkNN graph shows a better clustering performance than the other graph construction when the graph is partitioned with Markov Stability, which considers a diffusion process on the graph.

Comparison between clustering methods

In this section, we evaluate the performance of graph-based clustering through Markov Stability against several other clustering methods applied to the datasets from the UCI

Table 2 Comparison of graph constructions in terms of clustering performance (NMI and ARI) on eleven UCI datasets

Dataset	PMST	ϵ-ball	RMST $k = 1$			kNN			CkNN $\delta = 1$		
			$\gamma = 0.5$	$\gamma = 0.25$	$\gamma = 0.125$	$k = 3$	$k = 7$	$k = 12$	$k = 3$	$k = 7$	$k = 12$
Normalised Mutual Information (NMI)											
Iris	0.7764	0.8057	0.7106	0.7764	0.7400	0.7627	0.8057	0.8057	**0.8226**	0.7980	0.7777
Glass	0.3886	0.3862	0.3656	0.3646	0.3953	0.3863	0.3927	0.3626	0.3517	0.3941	**0.4170**
Wine	0.8080	0.7972	0.7389	0.8400	**0.8633**	0.7955	0.8336	0.8215	0.7528	0.8347	0.8113
WBDC	0.6042	0.5819	0.5839	0.6042	0.6182	0.6188	0.5944	0.6121	0.5822	**0.7231**	0.6113
Control chart	0.8272	0.8404	0.7602	0.7133	0.8130	0.8266	0.8520	0.852	0.8078	**0.8883**	0.8531
Parkinson	0.2220	0.2065	0.2012	0.2143	0.2811	**0.3175**	0.2815	0.2176	0.2113	0.2973	0.2423
Vertebral	0.5043	0.6273	0.5432	0.5323	0.5999	0.6060	0.5928	0.5424	**0.6450**	0.6018	0.6116
Breast tissue	0.5722	0.5616	0.5298	**0.5820**	0.5461	0.5447	0.5525	0.5243	0.5253	0.5648	0.5601
Seeds	0.7293	0.6943	0.7361	**0.7570**	0.6810	0.7515	0.7458	0.7318	0.6384	0.7381	0.7245
Image Seg.	0.6948	0.6605	0.6762	0.7072	**0.7488**	0.6154	0.6465	0.6667	0.6012	0.6347	0.6541
Yeast	0.2881	0.3051	0.2952	0.2626	0.2563	0.2764	0.2997	**0.3080**	0.2473	0.2959	0.3072
Average	0.5835	0.5879	0.5583	0.5776	0.5945	0.5910	0.5997	0.5859	0.5623	**0.6155**	0.5973
Adjusted Rand Index (ARI)											
Iris	0.7420	0.7592	0.6603	0.7420	0.6957	0.7191	0.7592	0.7592	0.8184	0.7455	0.7445
Glass	0.2323	0.2099	0.1983	0.2029	0.2258	0.2278	0.2231	0.2266	0.2134	0.2398	**0.2496**
Wine	0.8350	0.8072	0.7375	0.8712	**0.8823**	0.8025	0.8498	0.8349	0.7414	0.8471	0.8360
WBDC	0.7193	0.7114	0.7014	0.7193	0.7369	0.7368	0.7010	0.7310	0.6697	**0.8244**	0.7200
Control chart	0.6929	0.7364	0.5694	0.5371	0.6991	0.6748	0.6824	0.7071	0.6902	**0.8280**	0.7227
Parkinson	0.2267	0.2205	0.2038	0.1540	0.2556	0.2176	0.2001	0.2045	0.1165	**0.2667**	0.2101
Vertebral	0.5257	**0.6445**	0.5702	0.5411	0.6015	0.5982	0.5802	0.5330	0.6441	0.6113	0.6302
Breast tissue	**0.4689**	0.4494	0.4100	0.4017	0.3494	0.4012	0.4272	0.4078	0.3631	0.3764	0.4471
Seeds	0.7353	0.7497	0.7458	0.7589	0.6687	0.7876	**0.7889**	0.7761	0.6402	0.7764	0.7655
Image Seg.	0.6060	0.5144	0.6030	**0.6193**	0.5942	0.3800	0.4669	0.5471	0.3791	0.4522	0.5121
Yeast	0.1908	0.2368	0.1827	0.1772	0.1649	0.1755	0.2230	0.2531	0.1797	0.1942	0.2294
Average	0.5432	0.5490	0.5075	0.5204	0.5340	0.5201	0.5365	0.5437	0.4960	**0.5602**	0.5516

A high NMI (and ARI) indicate that the best partition found by Markov Stability is similar to the ground truth, i.e., better clustering. The best performance for each dataset is marked in bold repository. We include a variety of clustering approaches. Model-based methods include k-means and Gaussian mixture (clustering is repeated 50 times and partition with the best objective function is reported). We also apply hierarchical clustering with complete linkage and Euclidean distance is used as the distance measure. Since graph-based clustering is closely related to spectral clustering, we compare to two spectral clustering methods: the multiclass n-cut algorithm (Yu and Shi 2003) and the classic NJW algorithm (Ng et al. 2001). The affinity matrix for these two spectral clustering algorithms is calculated with a local density kernel as described in Zelnik-Manor and Perona (2004). Note that all these algorithms need the number of clusters to be given as an input. Hence we use the number of classes in the ground truth c^* as an input to set the number of clusters. For comparability, in the case of Markov Stability, we construct the CkNN graph with Euclidean distance ($k = 7$ and $\delta = 1$) and find the optimised partition with the number of clusters equal to c^*.

The results are presented in Table 3. Given the distinct features of the datasets, no method is expected to achieve consistently better performance across all datasets. The hierarchical clustering performs worst, as it is easily affected by the noise in real datasets. Model-based methods, such as the Gaussian mixture model, can achieve good performance if the properties of the data fit the assumptions (e.g., the Iris dataset), but can also perform poorly. On datasets that have non-convex geometries, graph-based methods and spectral clustering tend to perform better. On average, the Markov Stability approach achieves the best NMI and ARI scores.

To further validate the quality of the robust partitions found by Markov Stability, we also carried out MS clustering in a fully unsupervised manner, i.e., without providing the number of classes in the ground truth as an input. Using the principles described in "Markov stability for graph-based clustering" section, we identify robust scales and robust partitions in order to establish the number of clusters inherent to the data in an unsupervised manner. The number of clusters detected by MS and the clustering performances in terms of NMI and ARI are presented in Table 4, together with the results obtained in Table 3, where the number of clusters in the ground truth was provided. Although the number of detected clusters differs slightly from the ground truth, the clusters found in the unsupervised MS have comparable ARI values and a higher average NMI value which indicates that the clusters are of good quality and provide more information about the ground truth. This highlights the capability of Markov Stability as an unsupervised data clustering approach in practice where the number of clusters are usually unknown (see also Additional file 3: Table S1).

Conclusion

We have investigated the use of multiscale community detection for graph-based data clustering. The first step in graph-based clustering is to construct a graph from the data, and our empirical study shows that the recently proposed CkNN graph is a good choice for this purpose. In contrast to other neighbourhood-based graph constructions like kNN or ϵ-ball graphs, the CkNN graph is designed to provide a consistent discrete approximation of the diffusion operator on the underlying data manifold. Since many community detection methods are closely related to diffusion or random walks (e.g., Markov Stability and spectral methods), this explains the good performance of CkNN for clustering purposes. Other graph construction methods specifically designed for manifold learning (e.g, RMST) performed well but are not optimised for cluster separation.

Table 3 Comparison of performance of clustering methods on eleven real UCI datasets

Dataset	k-means	Gaussian mixture	Hierarchical	Spectral N-cut	Spectral NJW	MS on CkNN graph
Normalised Mutual Information (NMI) (Number of clusters fixed to ground truth: $c = c^*$)						
Iris ($c^* = 3$)	0.7582	**0.8997**	0.7221	0.7578	0.7665	0.7980
Glass ($c^* = 6$)	0.3163	**0.4033**	0.1548	0.2899	0.3175	0.3932
Wine ($c^* = 3$)	0.8759	0.9120	0.6144	**0.9276**	**0.9276**	0.8347
WBDC ($c^* = 2$)	0.5546	0.5275	0.0102	0.5434	0.5565	**0.7231**
Control chart ($c^* = 6$)	0.6907	0.5348	0.7070	0.7924	0.7891	**0.8489**
Parkinson ($c^* = 2$)	0.0970	0.0387	0.0519	**0.2189**	**0.2189**	0.1334
Vertebral ($c^* = 3$)	0.4120	0.5559	0.1280	0.3969	0.3901	**0.5971**
Breast tissue ($c^* = 6$)	0.5446	**0.5487**	0.4745	0.5481	0.5366	0.5291
Seeds ($c^* = 3$)	0.7279	**0.8111**	0.7010	0.6881	0.6881	0.7142
Image Seg. ($c^* = 7$)	0.5840	0.5697	0.0359	0.5929	**0.6466**	0.5816
Yeast ($c^* = 10$)	**0.2955**	0.2071	0.1846	0.2794	0.2801	0.2909
Average	0.5324	0.5462	0.3440	0.5487	0.5561	**0.5858**
Adjusted Rand Index (ARI) (Number of clusters fixed to ground truth: $c = c^*$)						
Iris ($c^* = 3$)	0.7302	**0.9039**	0.6423	0.7430	0.7570	0.7455
Glass ($c^* = 6$)	0.1722	**0.2649**	0.0350	0.1104	0.1483	0.2380
Wine ($c^* = 3$)	0.8975	0.9325	0.5771	**0.9471**	**0.9471**	0.8471
WBDC ($c^* = 2$)	0.6707	0.6436	0.0048	0.6661	0.6777	**0.8244**
Control chart ($c^* = 6$)	0.5207	0.2699	0.5374	0.6835	0.6803	**0.7056**
Parkinson ($c^* = 2$)	-0.0978	-0.0502	-0.0710	0.1608	0.1608	**0.2659**
Vertebral ($c^* = 3$)	0.3147	0.5256	-0.0327	0.2986	0.2888	**0.6096**
Breast tissue ($c^* = 6$)	0.2876	0.2872	0.1895	**0.3983**	0.3806	0.3659
Seeds ($c^* = 3$)	0.7733	**0.8167**	0.6863	0.7189	0.7189	0.7432
Image Seg. ($c^* = 7$)	0.4605	0.4522	0.0011	0.3360	**0.4849**	0.4150
Yeast ($c^* = 10$)	0.1658	0.0972	0.1083	0.1483	0.1481	**0.1746**
Average	0.4450	0.4676	0.2435	0.4737	0.4902	**0.5395**

The number of clusters is fixed to the number of clusters of the ground truth and given as an input. The best performance for each dataset is indicated in boldface.

Table 4 Graph-based clustering via Markov Stability performs equally well in a totally unsupervised manner, even if the number of clusters is not given a priori

Dataset	Markov Stability (*c* fixed to ground truth *c**)	Markov Stability (*c* found by MS unsupervised)
Normalised Mutual Information (NMI)		
Iris	0.7980 ($c^* = 3$)	**0.7980** ($c = 3$)
Glass	**0.3932** ($c^* = 6$)	0.3687 ($c = 5$)
Wine	0.8347 ($c^* = 3$)	**0.8347** ($c = 3$)
WBDC	0.7231 ($c^* = 2$)	**0.7231** ($c = 2$)
Control chart	0.8489 ($c^* = 6$)	**0.8520** ($c = 4$)
Parkinson	0.1334 ($c^* = 2$)	**0.2973** ($c = 3$)
Vertebral	0.5971 ($c^* = 3$)	**0.6018** ($c = 2$)
Breast tissue	**0.5291** ($c^* = 6$)	0.5104 ($c = 3$)
Seeds	0.7142 ($c^* = 3$)	**0.7142** ($c = 3$)
Image Seg.	0.5816 ($c^* = 7$)	**0.6338** ($c = 13$)
Yeast	0.2909 ($c^* = 10$)	**0.2909** ($c = 10$)
Average	0.5858	**0.6023**
Adjusted Rand Index (ARI)		
Iris	0.7455 ($c^* = 3$)	**0.7455** ($c = 3$)
Glass	**0.2380** ($c^* = 6$)	0.2086 ($c = 5$)
Wine	0.8471 ($c^* = 3$)	**0.8471** ($c = 3$)
WBDC	0.8244 ($c^* = 2$)	**0.8244** ($c = 2$)
Control chart	**0.7056** ($c^* = 6$)	0.6824 ($c = 4$)
Parkinson	0.2659 ($c^* = 2$)	**0.2667** ($c = 3$)
Vertebral	0.6096 ($c^* = 3$)	**0.6113** ($c = 2$)
Breast tissue	0.3659 ($c^* = 6$)	**0.3764** ($c = 3$)
Seeds	0.7432 ($c^* = 3$)	**0.7432** ($c = 3$)
Image Seg.	0.4150 ($c^* = 7$)	**0.4516** ($c = 13$)
Yeast	0.1746 ($c^* = 10$)	**0.1746** ($c = 10$)
Average	**0.5395**	0.5393

The best performance for each dataset is indicated in boldface

Our work has also examined the suitability of multiscale community detection as a means for unsupervised data clustering. Specifically, we have used the Markov Stability framework, which employs a diffusion process on the graph to detect the presence of relevant subgraphs at all scales. The time of the diffusion process acts as a resolution parameter and a cost function for graph partitioning is optimised at different scales by scanning time. Robust partitions and robust scales can be identified by analysing the consistency of the ensemble of optimised partitions found by the Louvain algorithm. Our numerics show that the Markov Stability framework is able to determine the number of clusters and reveal the multiscale structure in data. Further, by scanning Markov time, the MS analysis can reduce the sensitivity to the parameters in the graph construction step, thus improving the robustness of graph-based clustering.

We have validated our graph-based clustering approach on several real datasets by comparing with other popular clustering methods, including k-means, Gaussian mixture model, hierarchical clustering, and two spectral clustering algorithms. The graph-based clustering method achieves the best NMI and ARI values on average across the datasets. Importantly, we show that the clustering can be done in a completely unsupervised way (without assuming a knowledge of the number of clusters), whereas for the other standard methods the number of clusters needs to be given as an input.

Our study also suggests several directions of future work. Here we showed that the CkNN graph is a good choice for graph-based clustering, but it will be interesting to establish the performance of CkNN in other data mining problems, such as manifold learning where graphs also play a important role (Yan et al. 2007). We showed that the variation of information of partitions and the ensemble of partitions found by greedy optimisation can be used to guide the identification of robust partitions. However, a quantitative, statistically sound process to choose the significant partitions automatically would be desirable and useful in practice. Another interesting direction is the potential study of the outputs of MS clustering using methods from topological data analysis (TDA). Scanning across Markov time results in an ensemble of time-dependent weighted graphs with adjacency matrices $DP(t)$. It would the be possible to use methods from TDA to characterise the persistent homology of the graphs as a function of the Markov time t to detect robust structures in the data across scales and its relationship with the observed hierarchy of clusters of increasing coarseness.

Supplementary information

Supplementary information accompanies this paper at https://doi.org/10.1007/s41109-019-0248-7.

Additional file 1: Supplementary Figure: Multiscale Markov Stability analysis of the Control Chart dataset.

Additional file 2: Supplementary Figure: Multiscale Markov Stability analysis of the WBDC dataset.

Additional file 3: Supplementary Table: Performance of clustering methods measured by Purity.

Acknowledgements

This work was supported by the European Commission [European Union 7th Framework Programme for research, technological development and demonstration under grant agreement no. 607466], and the Engineering and Physical Sciences Research Council (EPSRC) through grant EP/N014529/1 to M.B..

Authors' contributions

ZL and MB conceived of the idea of the study. ZL implemented the methods and ran the numerical experiments. Both authors wrote, reviewed and approved the manuscript.

Competing interests

The authors declare that they have no competing interests.

Received: 20 August 2019 Accepted: 16 December 2019
Published online: 08 January 2020

References

Alpert CJ, Kahng AB, Yao S-Z (1999) Spectral partitioning with multiple eigenvectors. Discret Appl Math 90(1):3–26

Altuncu MT, Mayer E, Yaliraki SN, Barahona M (2019) From free text to clusters of content in health records: an unsupervised graph partitioning approach. Appl Netw Sci 4(1):2. https://doi.org/10.1007/s41109-018-0109-9

Amor B, Yaliraki S, Woscholski R, Barahona M (2014) Uncovering allosteric pathways in caspase-1 using markov transient analysis and multiscale community detection. Mol Biosyst 10(8):2247–2258

Asllani M, Carletti T, Di Patti F, Fanelli D, Piazza F (2018) Hopping in the crowd to unveil network topology. Phys Rev Lett 120(15):158301

Azran A, Ghahramani Z (2006) Spectral methods for automatic multiscale data clustering. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06). IEEE. pp 190–197. https://doi.org/10.1109%2Fcvpr.2006.289

Bacik KA, Schaub MT, Beguerisse-Díaz M, Billeh YN, Barahona M (2016) Flow-based network analysis of the Caenorhabditis elegans connectome. PLoS Comput Biol 12(8):1005055

Beguerisse-Díaz M, Garduno-Hernández G, Vangelov B, Yaliraki SN, Barahona M (2014) Interest communities and flow roles in directed networks: the Twitter network of the UK riots. J R Soc Interface 11(101):20140940

Beguerisse-Díaz M, Vangelov B, Barahona M (2013) Finding role communities in directed networks using Role-Based Similarity, Markov Stability and the Relaxed Minimum Spanning Tree. In: 2013 IEEE Global Conference on Signal and Information Processing (GlobalSIP). IEEE, Austin. pp 937–940

Berry T, Harlim J (2016) Variable bandwidth diffusion kernels. Appl Comput Harmon Anal 40(1):68–96

Berry T, Sauer T (2019) Consistent manifold representation for topological data analysis. Found Data Sci 1(1):1–38

Ben-Hur A, Horn D, Siegelmann HT, Vapnik V (2001) Support vector clustering. J Mach Learn Res 2:125–137

Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech Theory Exp 2008(10):10008

Bronstein MM, Bruna J, LeCun Y, Szlam A, Vandergheynst P (2017) Geometric deep learning: Going beyond euclidean data. IEEE Sign Process Mag 34(4):18–42. https://doi.org/10.1109/MSP.2017.2693418

Carreira-Perpiñán MA, Zemel RS (2004) Proximity graphs for clustering and manifold learning. In: Proceedings of the 17th International Conference on Neural Information Processing Systems (NIPS'04). MIT Press, Cambridge, MA. pp 225–232

Cheng B, Yang J, Yan S, Fu Y, Huang TS (2010) Learning with ℓ^1-graph for image analysis. IEEE Trans Image Process 19(4):858–866. https://doi.org/10.1109/TIP.2009.2038764

Chung FRK (1997) Spectral Graph Theory. Regional Conference Series in Math. CBMS, Amer. Math. Soc. 1997

Cormen TH, Leiserson CE, Rivest RL, Stein C (2009) Introduction to Algorithms, 3rd ed. The MIT Press, Cambridge, MA

Daitch SI, Kelner JA, Spielman DA (2009) Fitting a graph to vector data. In: Proceedings of the 26th Annual International Conference on Machine Learning. ACM, New York. pp 201–208

de Sa VR (2005) Spectral clustering with two views. In: Proceedings of ICML 2005 workshop on learning with multiple views, Bonn. pp 20–27

Delmotte A, Tate EW, Yaliraki SN, Barahona M (2011) Protein multi-scale organization through graph partitioning and robustness analysis: application to the myosin–myosin light chain interaction. Phys Biol 8(5):055010

Delvenne J-C, Schaub MT, Yaliraki SN, Barahona M (2013) The stability of a graph partition: A dynamics-based framework for community detection. In: Mukherjee A, Choudhury M, Peruani F, Ganguly N, Mitra B (eds). Dynamics On and Of Complex Networks, Volume 2: Applications to Time-Varying Dynamical Systems. Springer, New York. pp 221–242

Delvenne J-C, Yaliraki SN, Barahona M (2010) Stability of graph communities across time scales. Proc Natl Acad Sci 107(29):12755–12760

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B (Method) 39(1):1–38

Dheeru D, Karra Taniskidou E (2017) UCI Machine Learning Repository. Irvine. http://archive.ics.uci.edu/ml. Accessed 22 Dec 2019

Dhillon IS (2001) Co-clustering documents and words using bipartite spectral graph partitioning. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, New York. pp 269–274

Dhillon IS, Guan Y, Kulis B (2004) Kernel k-means: spectral clustering and normalized cuts. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, New York. pp 551–556

Fortunato S (2010) Community detection in graphs. Phys Rep 486(3):75–174

Hagen L, Kahng AB (1992) IEEE Trans Comput-aided Des Integr Circ Syst 11(9):1074–1085

Hubert L, Arabie P (1985) Comparing partitions. J Classif 2(1):193–218

Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv (CSUR) 31(3):264–323

Jebara T, Wang J, Chang S-F (2009) Graph construction and b-matching for semi-supervised learning. In: Proceedings of the 26th Annual International Conference on Machine Learning. ACM, New York. pp 441–448

Kulis B, Basu S, Dhillon I, Mooney R (2009) Semi-supervised graph clustering: a kernel approach. Mach Learn 74(1):1–22

Lambiotte R, Delvenne J-C, Barahona M (2008) Laplacian Dynamics and Multiscale Modular Structure in Networks. arXiv:0812.1770v3. Accessed 22 Dec 2019

Lambiotte, R, Delvenne J-C, Barahona M (2014) Random walks, Markov processes and the multiscale modular organization of complex networks. IEEE Trans Netw Sci Eng 1(2):76–90

MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics. University of California Press, Berkeley. pp 281–297. https://projecteuclid.org/euclid.bsmsp/1200512992

Maier M, Luxburg UV, Hein M (2008) Influence of graph construction on graph-based clustering measures. In: Proceedings of the 21st International Conference on Neural Information Processing Systems (NIPS'08). Curran Associates Inc., USA. pp 1025–1032

Maier M, Von Luxburg U, Hein M (2013) How the result of graph clustering methods depends on the construction of the graph. ESAIM Probab Stat 17:370–418

Meilă M (2003) Comparing clusterings by the variation of information. In: Schölkopf B, Warmuth MK (eds). Learning Theory and Kernel Machines. Springer, Berlin, Heidelberg. pp 173–187

Ng AY, Jordan MI, Weiss Y (2001) On spectral clustering: Analysis and an algorithm. In: Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic (NIPS'01). MIT Press, Cambridge, MA. pp 849–856

Petri G, Expert P (2014) Temporal stability of network partitions. Phys Rev E 90(2):022813

Reichardt J, Bornholdt S (2006) Statistical mechanics of community detection. Phys Rev E 74(1):016110

Rokach L, Maimon O (2005) Clustering methods. In: Data Mining and Knowledge Discovery Handbook. Springer, Boston, MA. pp 321–352

Ronhovde P, Nussinov Z (2010) Local resolution-limit-free potts model for community detection. Phys Rev E 81(4):046114

Schaub MT, Delvenne J-C, Lambiotte R, Barahona M (2019) Multiscale dynamical embeddings of complex networks. Phys Rev E 99:062308. https://doi.org/10.1103/PhysRevE.99.062308

Schaub MT, Delvenne J-C, Yaliraki SN, Barahona M (2012) Markov dynamics as a zooming lens for multiscale community detection: non clique-like communities and the field-of-view limit. PloS ONE 7(2):32210

Shi J, Malik J (2000) Normalized cuts and image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on 22(8):888–905

Strehl A, Ghosh J (2002) Cluster ensembles—a knowledge reuse framework for combining multiple partitions. J Mach Learn Res 3:583–617

Sugar CA, James GM (2003) Finding the number of clusters in a dataset: An information-theoretic approach. J Am Soc Stat Assoc 98(463):750–763

Tran QH, Hasegawa Y, et al (2019) Scale-variant topological information for characterizing the structure of complex networks. Phys Rev E 100(3):032308

Tenenbaum JB, De Silva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290(5500):2319–2323

Traag VA, Van Dooren P, Nesterov Y (2011) Narrow scope for resolution-limit-free community detection. Phys Rev E 84(1):016114

Vangelov B (2014) Unravelling Biological Processes using Graph Theoretical Algorithms and Probabilistic Models. PhD thesis, Imperial College London, London

Von Luxburg U (2007) A tutorial on spectral clustering. Stat Comput 17(4):395–416

Xu R, Wunsch D (2005) Survey of clustering algorithms. IEEE Trans Neural Netw 16(3):645–678

Yan S, Xu D, Zhang B, Zhang H-J, Yang Q, Lin S (2007) Graph embedding and extensions: A general framework for dimensionality reduction. IEEE Trans Pattern Anal Mach Intell 29(1):40–51

Ye W, Goebl S, Plant C, Böhm C (2016) Fuse: Full spectral clustering. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, New York. pp 1985–1994

Yu SX, Shi J (2003) Multiclass spectral clustering. In: Proceedings Ninth IEEE International Conference on Computer Vision. pp 313–319. https://doi.org/10.1109/ICCV.2003.1238361

Zelnik-Manor L, Perona P (2004) Self-tuning spectral clustering. In: Proceedings of the 17th International Conference on Neural Information Processing Systems (NIPS'04). MIT Press, Cambridge, MA. pp 1601–1608

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.