communications physics

Article

https://doi.org/10.1038/s42005-024-01635-4

Local dominance unveils clusters in networks

Check for updates

Dingyi Shi[1,9], Fan Shang[1,9], Bingsheng Chen[1,2,3] ✉, Paul Expert[4], Linyuan Lü[5] ✉, H. Eugene Stanley[6], Renaud Lambiotte[7,8] ✉, Tim S. Evans[2] & Ruiqi Li[1] ✉

Clusters or communities can provide a coarse-grained description of complex systems at multiple scales, but their detection remains challenging in practice. Community detection methods often define communities as dense subgraphs, or subgraphs with few connections in-between, via concepts such as the cut, conductance, or modularity. Here we consider another perspective built on the notion of local dominance, where low-degree nodes are assigned to the basin of influence of high-degree nodes, and design an efficient algorithm based on local information. Local dominance gives rises to community centers, and uncovers local hierarchies in the network. Community centers have a larger degree than their neighbors and are sufficiently distant from other centers. The strength of our framework is demonstrated on synthesized and empirical networks with ground-truth community labels. The notion of local dominance and the associated asymmetric relations between nodes are not restricted to community detection, and can be utilised in clustering problems, as we illustrate on networks derived from vector data.

Many real-world datasets can be viewed as a collection of objects embedded into a global metric space, thereby providing a vector representation[1]. Alternatively, networks have become another fundamental way to model complex systems with a focus on direct pairwise interactions between constituents[2–4]. In the case of social systems, for instance, these complementary representations may correspond to a set of socio-demographic variables for each individual, e.g., in a Blau space[5], or to a social network of interactions between individuals, e.g., via a mobile communication network[6] or spatio-temporal co-occurrence interactions[7]. In each representation, real-world systems tend to exhibit groups: regions of high density in the spatial representation, known as clusters, or high-density subgraphs in the network, known as communities. Such cluster or community structure provides a coarse-grained representation of the underlying complex system[8–11], often associated to different functions and impacting its collective behaviors[12–14], and their unsupervised detection is thus essential in different areas of data science[1,10].

In the vector representation, the introduction of a dissimilarity function and ideally of a distance in a metric space, provides a natural way to identify the center of a cluster, e.g., the medoid in a general metric space[15,16], and a hierarchy would form within a cluster between central and other more peripheral nodes, implying an asymmetric relationship between them. On the other hand, in the case of asymmetric pairwise interactions, which can be associated to an implicit hierarchy[17] and have long been recognized[18–22] in various network systems, community detection methods for networks place much less emphasis on the concept of community center and hierarchy within communities. We can always use network centrality measures on the subgraphs identified as communities to identify core and peripheral nodes a posteriori, but these roles are not central to community detection[23,24], in stark contrast to clustering methods based on embedding the data in a metric space.

In this paper, we propose a community detection algorithm in networks, Local Search (LS), that explicitly uses the notion of local dominance and identifies community centers based on local information. In our method, every node is given at most one parent node deemed to be higher up in a partial ranking. Nodes that have a dominant position in their immediate neighborhood[18] or even beyond are identified as local leaders[18]. This defines a rooted tree that spans the network and gives rise to community centers that are local

[1]UrbanNet Lab, College of Information Science and Technology, Beijing University of Chemical Technology, 100029 Beijing, China. [2]Centre for Complexity Science, Imperial College London, London SW7 2AZ, UK. [3]Network Science Institute and Department of Physics, Northeastern University, Boston, MA 02115, USA. [4]Global Business School for Health, University College London, London WC1E 6BT, UK. [5]School of Cyberspace Security, University of Science and Technology of China, Hefei 230026, China. [6]Center for Polymer Studies and Physics Department, Boston University, Boston, MA 02215, USA. [7]Mathematical Institute, University of Oxford, Oxford OX2 6GG, UK. [8]Turing Institute, London NW1 2DB, UK. [9]These authors contributed equally: Dingyi Shi, Fan Shang. ✉e-mail: bi.chen@northeastern.edu; linyuan.lv@ustc.edu.cn; renaud.lambiotte@maths.ox.ac.uk; lir@buct.edu.cn

leaders[18] with both a larger degree than the nodes in their basin of attraction and a relatively long distance to other local leaders higher up in the ranking. Our approach possesses several interesting properties. Firstly, to the best of our knowledge, it provides a new perspective on community detection and delivers community centers and a hierarchy within the community and even a hierarchy among communities as an explicit part of our algorithm, and so mimics advantageous features of the methods based on embedding data in a metric space. Secondly, the identification of communities through local dominance is highly efficient, as it uses purely local topological information and breadth-first search, and runs in linear time. The method does not require the heuristic optimization of an objective function that relies on a global null model[9,25–29] or computationally costly spreading dynamics[30–32]. Also, our method does not rely on a similarity measure for which there is a wide choice, with an associated uncertainty and variability in results, such as is found in hierarchical clustering based methods[8,14,33,34]. Finally, LS is not as susceptible to noise as most methods[10,35], and is less therefore susceptible to finding spurious communities in random graph model realisations[36].

We demonstrate the strength of LS on several classical but challenging synthetic benchmarks and on standard empirical networks with known ground-truth community labels. Our numerical evaluation also includes network representations derived from vector data. As the LS method naturally provides community centers and local hierarchies, it creates an explicit analogy with the notion of cluster centers and distances within clusters that are found in vector clustering methods. Moreover, we also show that applying LS on discretised version of data cloud points outperforms classical unsupervised vector data clustering methods on benchmarks[16].

Results

Local search algorithm

Cluster analysis and community detection share many conceptual similarities, but often have a contrasting focus. Cluster analysis puts emphasis on the center of a cluster[15,16], while community boundaries often play a more predominant role in community detection[37]. Community centers can be inferred from some community detection algorithm outputs, for example, the nodes associated to the largest absolute weights of the leading eigenvector of the modularity matrix, or exhibiting a higher density of connections inside the communities, are deemed to be community centers, core members or provincial hubs[23,38]. But centers are only a by-product of those algorithms, rather than at their core of methodologies. The approach that we propose here is explicitly focusing on community centers to identify clusters, which is motivated by the existence of underlying asymmetries between nodes[19–21], the concept of local leaders[18] in networks and borrows ideas from density and distance based clustering algorithms on vector data[16]. In our local search (LS) algorithm, the local dominance refers to a leader-follower relation, and we pose a further restriction that each node has eventually at most one out-going link pointing to its leader. We hypothesise that communities are organized around centers that are nodes with both a dominant position in its neighborhood (e.g., has a larger degree, or other centrality measures, than its neighbors) and distant enough from other potential centers. Then based on community centers, partition is naturally ensuing. The process of our LS algorithm involves four steps:

Firstly, we calculate the degree k_u of each node (see digits in Fig. 1a). Secondly, we point each node u to its largest-degree-neighbor v if this neighbor is no smaller than itself on degree (i.e., $k_v \geq k_u$ and $k_v = \max\{k_j | j \in V(u)\}$, where $V(u)$ is the set of neighboring nodes of u). Nodes with in-going edge(s) and no out-going edge are local leaders[18] that dominates its vicinity (see nodes f, m, and p in Fig. 1b). Such local leaders are

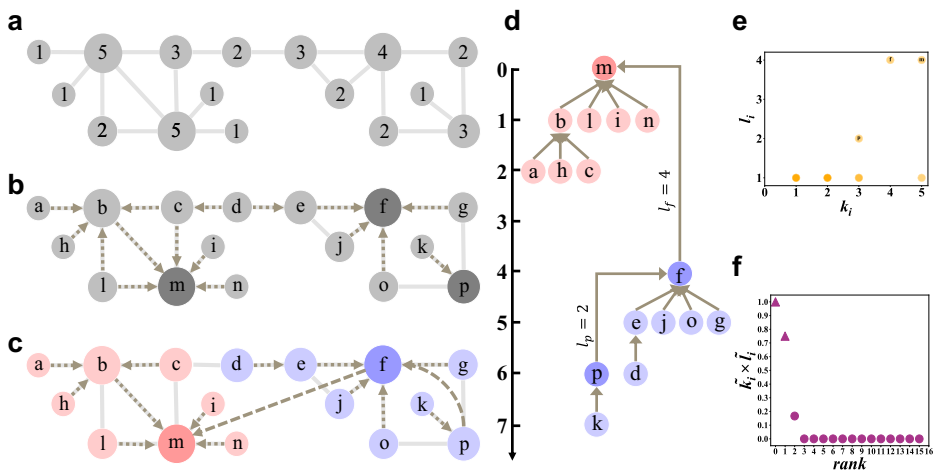

Fig. 1 | Schematic illustration of the Local Search (LS) algorithm. a An example network where digits on nodes and size of nodes indicate the degree. **b** The identification of local leaders based on local dominance by creating a forest of directed acyclic graphs (DAGs) as indicated by short dashed directed edges. For each node u, it points to any adjacent neighbor v with $k_v \geq k_u$ and $k_v = \max\{k_z | z \in \mathbf{V}(u)\}$, where $\mathbf{V}(u)$ is the set of neighboring nodes. In this example, nodes are traversed by their lexicographical order, when node b is traversed, it points to m as $k_m = \max\{k_z | z \in \mathbf{V}(b)\} \geq k_b$; later, when m is traversed, it has no out-going link, and so m is identified as a local leader: it does not point to any of its followers and its remaining neighbors all have smaller degrees. When there are more than one neighbor with the same largest degree, more than one directed edge is temporarily added, e.g., node c points to both b and m as $k_b = k_m = \max\{k_z | z \in \mathbf{V}(c)\} \geq k_c$; nodes d and l also have more than one outgoing link. The local leaders, which are potential community centers, are f, m, and p (indicated by dark gray color). **c** Each node randomly retains just one out-going edge shown as a short dashed directed edge (e.g., c can point to b or m with an equal probability, similarly for l and d). Then, for each local leader u, a local-BFS is performed to find its nearest local leader with $k_v \geq k_u$, and the shortest path length on network d_{uv}, $\forall$ v is designated by l_u. Here, $p \rightarrow f$ with $l_p = 2$, and $f \rightarrow m$ with $l_f = 4$. In (**c**), short-dash arrows and long-dash arrows correspond to pure followers (whose $l_u = 1$) and local leaders (whose $l_u \geq 2$), respectively. Each node has at most one out-going link ($u \rightarrow v$), which can go beyond direct connections. The local leader(s) with the maximal degree has no out-going link (here node m). **d** The corresponding tree structure formed by local dominance. The scale on the left is a visual aid for calculating l_i between connected nodes in the DAG. **e** The scatter plot of k_i and l_i for all nodes. Community centers are of both a larger degree k_i and a longer l_i. **f** The decision graph for quantitatively determining community centers (indicated by triangles) based on the product of rescaled degree $\tilde{k}_i$ and rescaled distance $\tilde{l}_i$ (see more details in Supplementary Note 1.2). Community centers can be detected by a visual inspection for obvious gaps or sophisticated automatic detection methods. Here, two centers, nodes m and f, are identified. The color of nodes in (**c**) and (**d**) represents the community partition, and community centers are highlighted by a darker hue of the same color.

like rich-among-poor and are potential community centers. Thirdly, for each local leader u, we use a local breath-first searching (LBFS) to find it a nearest local leader v with $k_v \geq k_u$ and record its shortest path length to node v as $l_u = d_{uv}$, which is larger than one (see long-dash arrows in Fig. 1c, and see Fig. 1d for a better extracted local dominance relation, which is in a reverse direction of arrows). The LBFS process stops after finding such a local leader v, which is the reason why we call it local BFS, so it generally searches a small region and does not traverse the whole network. For local leader(s) with the maximal degree, we do not perform such a LBFS, and directly assign the maximal l_u of other local leaders (see mathematical descriptions in Methods).

After performing LBFS to all local leaders except the maximal one(s), we can determine community centers according to degree k_u and distance along the local dominance relation l_u (in the network in Fig. 1, nodes f and m stands out as centers, which have both a large k_u and l_u, see Fig. 1e). Note that for nodes except local leaders, their $l_u = 1$. By multiplying normalized k_u and normalized l_u, we can better quantitatively identify community centers via a notable gap between candidates (see Fig. 1f, and details in Methods and Supplementary Note 1). Lastly, after the identification of community centers, the group label can be assigned to its followers along the local dominance relation (i.e., the reverse direction of arrows in Fig. 1c) in one single step.

We name our framework as local search (LS) algorithm, since it only require local information of nodes and rely on efficient LBFS processes for local leaders, which takes up a very small fraction of the whole network (see Supplementary Table 1). The identification of local dominance relation is quite resilient to missing and noisy links. Our LS algorithm is of a linear time complexity in terms of the number of edges (see Methods and Supplementary Table 1) and is in no need of iteratively optimizing an objective function that relies on a global null model in other state-of-the-art methods[9,26–29] or resorting to spreading dynamics[30,31]. In addition, our LS algorithm is also capable of identifying multiscale communities structure, as local dominance relation also provides us hierarchies between communities via asymmetric relationship between communities centers. The strength of our framework is demonstrated on several classical challenging synthesized test cases and empirical network datasets with ground-truth community labels. Finally, we also show how it provides a connection to clusters in a metric space, and our LS algorithm outperforms current state-of-the-art unsupervised both clustering and community detection methods when applied to discretised vector data clouds.

As the implementation of our algorithm was done in Python, we use the NetworkX package implementation of the Louvain algorithm, our main point of comparison, as they are both of a linear time complexity, to obtain fair comparison for running time. We also compare with a broader range of popular community detection algorithms on partition performance, some of which are slower but more accurate ones. Our LS algorithm still ranks first or second on the partition performance for five out of seven networks. More details of our LS algorithm can be found in Methods and Supplementary Note 1.

Synthetic networks

Here, we use well-known benchmark networks to illustrate how the LS method functions and in which situations it performs well. For illustration, we mainly contrast the results obtained by the LS method to those obtained by the Louvain method[9], which is widely applied due to its good performance and high efficiency. In addition, both algorithms have a linear time complexity, and thus the comparison on performance between them are more meaningful. We first look at a circular regular network, where all nodes are equivalent and thus no community structure should be discovered. LS correctly identifies a single community (Fig. 2a), by contrast, modularity forces community structure to exist and finds five communities (Fig. 2d). Let us look in detail at the reason why LS finds a single community. First, each node will point to all its adjacent neighbors as they all have the same degree, and since node are sequentially traversed and they will not point to their followers, loops cannot be formed, see Supplementary Fig. 1c and Supplementary Note 1.1.1 for a proof. After all nodes have been considered, each node will only keep one outgoing link with an equal probability, and eventually a tree structure will be formed. Because of the homogeneity of the graph, the tree only allows the identification a single community center and therefore of a single community. Because all nodes are equivalent, the labeling and thus order in which they are visited, is irrelevant. We note that in the case of a clique, an extreme case of regular

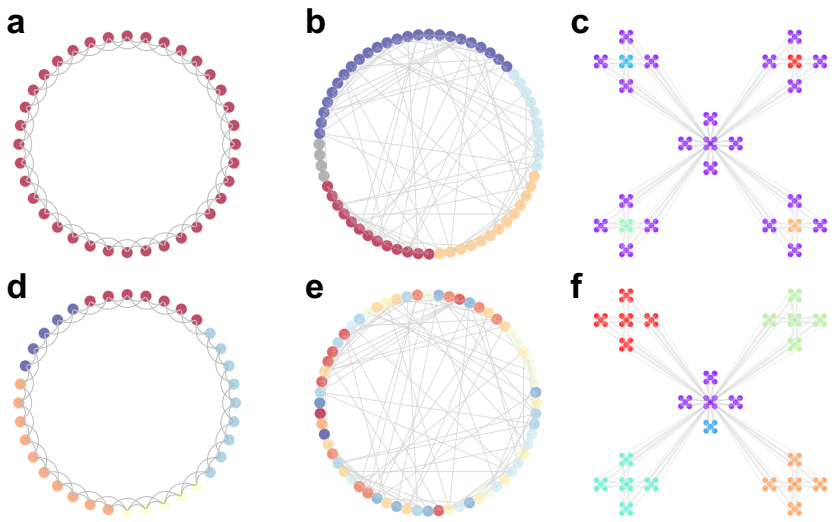

Fig. 2 | Community partitions by the local search (LS) and Louvain algorithms on synthesized networks with different strength of heterogeneity. The heterogeneity increases from left to right. The color of nodes denotes the community membership. **a** In a strict homogeneous regular network ($N = 36$, $\langle k \rangle = 4$), all nodes are identical, only one community is detected by the LS algorithm (see Supplementary Fig. 1 for more details); In an Erdős-Rényi (ER) random network ($N = 64$, $\langle k \rangle = 4$), there may exist some communities due to randomness[36], **b** the LS algorithm detects only a few communities. In a Ravasz-Barabási network[43] that displays stronger heterogeneity, **c** the LS algorithm groups all first-level nodes and all sixteen second-level peripheral clusters into one community, and four small communities emerge (see Supplementary Fig. 4 for more details). **d** By contrast, the Louvain algorithm detects five communities by optimizing modularity in the same strictly regular network as in (**a**). **e** And the Louvain algorithm detects a lot more communities in the same ER network as in (**b**). **f** In the same Ravasz-Barabási network as in (**c**), the Louvain algorithm partitions each second-level branching as a separate community and misclassifies a first-level peripheral cluster into its own community, a result of traversal order and modularity optimization process in the Louvain algorithm.

network, the mapping of local hierarchy can yield a range of structure from a chain to a star structure, see Supplementary Fig. 1b for more details. In all cases, only one center is identified. By contrast, the Louvain method would partition a homogeneous regular network into several communities by optimizing modularity (see Fig. 2d).

Our second application focuses on Erdős-Rényi (ER) random graphs, which is still relatively homogeneous though not strictly homogeneous. While in the limit of an infinite random graph no community structure exists, in finite-size ER graphs, fluctuations may create spurious community structures[11,36], as well as weak or spurious hierarchies between nodes. In this example, the LS method detects fewer communities than the Louvain algorithm, see Fig. 2b, e. In ER random networks, the degree distribution is relatively restricted around its average, but the system nonetheless exhibits fluctuations in the degrees. Large degree nodes are more likely to connect to each other, as the connection probability between them, $k_ik_j/2E$, are among the highest ones. When two large nodes are connected, there will be a directed out-going link pointing from one node to the other, making one of them a follower. Thus the LS method detects fewer communities. On the other hand, when we fix the size of the network and increase the connection probability p, the number of communities detected by the Louvain algorithm also decreases but it consistently finds more communities than the LS method (see Supplementary Fig. 2). In addition, we are able to detect isolated nodes as noise (see gray nodes in Fig. 2b), as these nodes are of a small degree but infinite l_i.

We also consider an extension of the ER random graph model, the stochastic block models (SBM) shown in Supplementary Fig. 3 and discussed in Supplementary Note 1.1.2. For random networks generated by SBM[39–41] with two blocks, when the inter-connection probability is zero, $c_{out} = 0$, the Louvain algorithm detects two communities that align with ground truth, but this is a reflection of the resolution limit[42], as when analyzing each community (each ER graph), it may partition it into more than ten communities (see Supplementary Fig. 2). By contrast, the LS algorithm still detects as many community centers as when looking at each individual random network. This result can be understood by the local nature of the algorithm, where the structure of one disconnected cluster does not affect the communities found in the other and thus LS algorithm does not suffer from resolution limit. And when fixing the intra-connection probability c_{in} and increasing c_{out}, the boundary of the two communities becomes blurred. We find that when slightly increasing c_{out}, the number of community partitions given by the Louvain algorithm increases drastically (see Supplementary Fig. 3b). By contrast, the F_1-score of the LS algorithm is relatively stable, though not too high, and outperforms Louvain when c_{out} is larger (see Supplementary Fig. 3).

Finally, we consider a hierarchical benchmark, the Ravasz-Barabási network model[43] with two layers, which naturally provides a model with a hierarchy between the center and peripheral nodes. The clustering proposed by LS method groups explicitly reflects the hierarchical nature of the model by grouping first-level nodes and all sixteen second-level peripheral clusters into one community that centers at the original seed node, as it dominates their neighborhood; and four small communities emerge due to the existence of four local leaders (see Fig. 2c), which have a degree larger than their neighbors and a longer path length to the original seed node, i.e., $l_i > 1$, see Supplementary Fig. 4c for the decision graph that identifies community centers. The Louvain algorithm offers an alternative partitioning that ignores the hierarchical nature of the model and finds five communities of roughly equal size, and misclassifies a peripheral cluster as a separate small community (see Fig. 2f). This example is interesting in that the clustering provided by Louvain here provides a reasonable, yet alternative, answer that ignores one aspect of the data. This reminds us that different clustering methods rely on different underlying mechanisms and, as often occurs when using unsupervised methods, the outputs are rarely strictly right or wrong. The outputs should be understood not only in terms of the data, but of the methods as well. Still, it is worth noting that the Louvain algorithm misclassifies a first-level peripheral cluster into another community (see the blue cluster in Fig. 2f), due to the traversal order used by the algorithm and modularity optimization process (see Supplementary Note 1.1.3 for more details). When we further modify the network generated by the Ravasz-Barabási model by adding a third-level branching to one of the second-level central cluster, and add noise in the connectivity to other second-level central clusters, the LS method still detects meaningful hierarchical structure, see Supplementary Fig. 4b and Supplementary Fig. 4d.

Detection of multiscale community structure. As partially reflected in the decision graph of the LS algorithm for the Ravasz-Barabási network (see Supplementary Fig. 4c, d), the reliance on local dominance of our method to identify local leaders naturally lends itself to detect multiscale community structure[14,34,44]. To illustrate this point, we generate a multiscale network made of two levels: four top-level communities with 400 nodes each and inter-connection probability $p_1 = 0.0002$, each top level community contains four second-level communities with 100 nodes each and $p_2 = 0.035$[14,34]. Each second-level community is generated by the standard Barabási-Albert model[45] with $m = 7$ that yields $\langle k \rangle = 14$ (see Fig. 3a). The LS method correctly identifies two levels of community structure with a notable gap between first four top-level centers, which have similar $\tilde{k}_i \times \tilde{l}_i$, and other potential centers, as shown in Fig. 3b. Then taking the twelve subsequent centers, these sixteen centers together correspond to the sixteen second-level communities, and their affiliation within each top-level communities are correct (see the tree structure for local leaders in Fig. 3c). As all sixteen second-level communities are statistically equivalent, the directionality of community centers (Fig. 3c) is determined by fluctuations in the network generating mechanism. The partition obtained by the LS method has an F_1-score of 0.99 at the top level and of $F_1 = 0.56$ at the second level. Misclassifications at the second level mainly come from a relatively large inter-connection probability p_2, which blurs the boundary between communities. In comparison, the Louvain algorithm only detects four large communities that correspond to the top-level ones with F_1-score equals 1, but it cannot detect second-level smaller communities due to the resolution limit[42]. This demonstrates the strength of the LS method on detecting smaller scale community structure.

One reason that LS works on detecting multiscale structure resides in the fact that the average path length between nodes is governed by the connection probability[46]. The distance between nodes from different second-level communities within the same top-level community is on average shorter than the distance between nodes from different top-level communities, and thus the hierarchical structure is uncovered by the LS method. Another reason is the intrinsic heterogeneity in each second-level community.

By contrast, when keeping the average degree and inter-connection probability (p_1 and p_2) the same, and replacing the second-level communities by ER random networks with $p = 0.14$, which also yields $\langle k \rangle = 14$ (see Fig. 3d), the whole network becomes more homogeneous (see Supplementary Fig. 5). In this case, the LS method can still detect four top-level communities (see Fig. 3e) but mis-identify some second-level communities (e.g., communities *c2* and *d1* are missing in this example, see Fig. 3f) and detect more smaller communities (29 second-level communities are detected instead of 16). The mis-identification of some second-level communities is due to the largest degree node u in those ground-truth communities being directly connected to a node v in other communities with $k_v \geq k_u$, and thus u is considered as followers. This is more common in such a random setting, as there are more nodes with a relatively large degree beyond the reference value (i.e., the smallest degree of all of the largest node in each ground-truth second-level communities, see Supplementary Fig. 5 for more details). By contrast, in the scale-free case, there are fewer nodes beyond the reference value. For example, in the random multiscale network in Fig. 3d, the reference value is 34, and there are 60 nodes beyond it; in comparison, in the scale-free one, there are only 31 nodes beyond its reference value. The homogeneity makes the detection of such communities harder, if this minimum value become only slightly smaller, there will be much more nodes beyond the reference value in the random setting (see

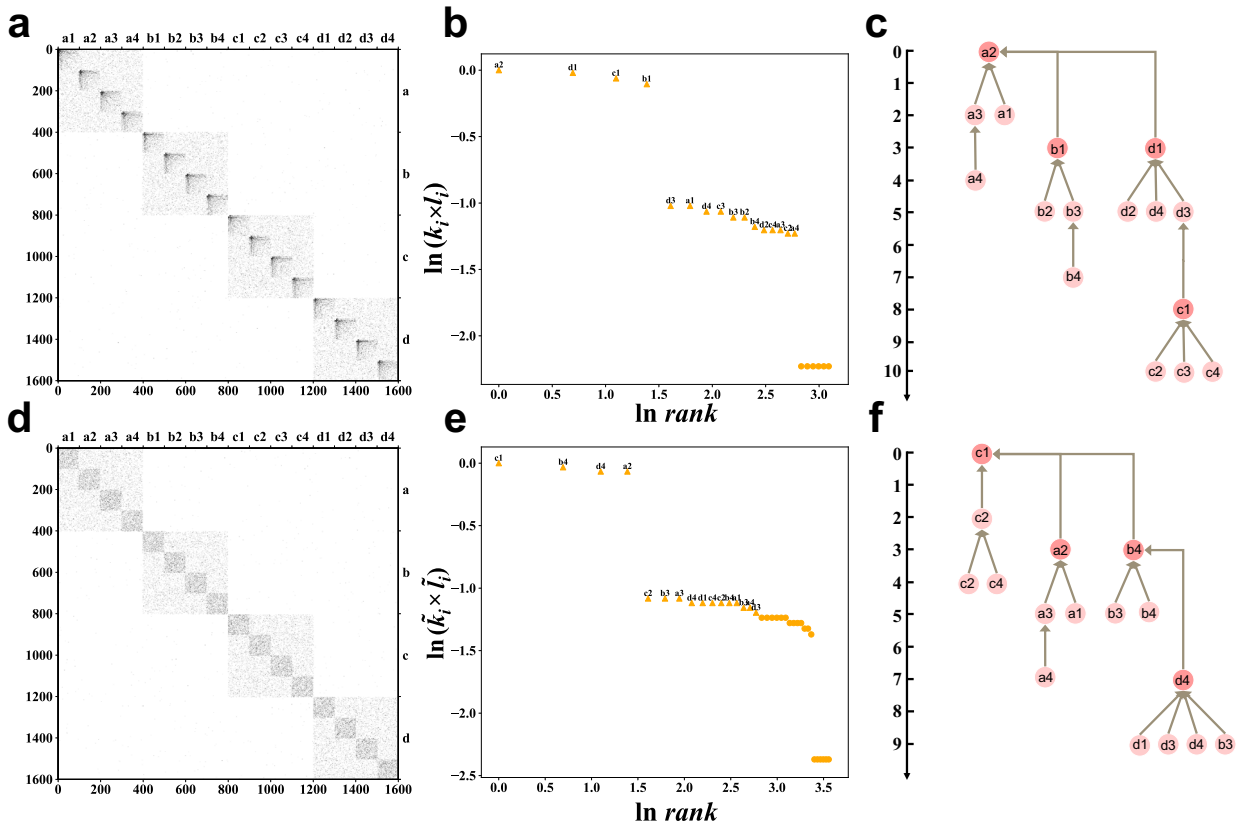

Fig. 3 | Detection of multiscale community structure with different heterogeneity. a This network comprise four top-level communities (labeled as a, b, c, and d) with 400 nodes each and an inter-connection probability $p_1 = 0.0002$, each of which further comprises four second-level communities with 100 nodes and $p_2 = 0.035$ (e.g., community c comprises c1, c2, c3, and c4). The second-level communities are generated by the Barabási-Albert model[45] with $m = 7$, which leads to an average degree $\langle k \rangle = 14$. **b** shows the decision graph for the LS method when analyzing the network in (**a**). **c** displays the hierarchical structure formed by the local dominance between identified centers of each community. For better clarity, community centers are named by the community label instead of the real index of the node, and we only show the tree structure of these centers. The height difference indicates the l_i of the lower node. (**d**)-(**f**) is the same as (**a**)-(**c**), with only changing the generation process of second-level communities to the Erdős-Rényi random network with a connection probability $p = 0.14$, which still leads to the same average degree $\langle k \rangle = 14$. In such a setting, similar to stochastic block models, nodes in the network are again relatively homogeneous. For better clarity, in (**e**) and (**f**) only top sixteen centers are labeled and their affiliation relation are visualized, and in total, LS detects 29 centers at the second-level for this network. For the multiscale network in (**a**), the LS method detects four top-level communities with $F_1 = 0.99$ and 16 second-level communities with $F_1 = 0.56$. For the network in **d**, the LS method detects four top-level communities with $F_1 = 0.89$ and 29 second-level communities with $F_1 = 0.29$. In both cases, the Louvain algorithm only obtain four communities, which corresponds to the first-level ones, with F_1 equals 1, however, it cannot detect second-level partitions. By comparing results in (**a**)-(**c**) and in (**d**)-(**f**), we can find that our LS algorithm works well on networks with stronger heterogeneity. Results shown here correspond to just one realization, in multiple realizations, as every first- and second-level communities are equivalent, the label sequence in (**b**) and (**e**) and the tree structure in (**c**) and (**f**) may vary but have a consistent structure.

Supplementary Fig. 5b). Mis-affiliation, i.e., one local leader in community *b3* follows the center of *d4* instead of other centers in community *d*, is also partially due to a similar reason and partially due to randomness. The discussion above also imply that the LS method would be vulnerable to targeted failure – connecting two community centers would diminish one center as a follower and their corresponding communities merge as one (see Supplementary Fig. 24). In addition, due to randomness, two or more local leaders might emerge in the same second-level communities, which will lead to split of the community (e.g., there are two local leaders in communities *c2*). These would constitute cases where the LS method is not appropriate.

Real-world benchmark networks

We now test the LS algorithm and demonstrate its strength on several empirical benchmark networks (see Table 1) with known ground-truth community labels, see Table 2. We chose to compare with the Louvain algorithm in Table 2 because both algorithms are linear, making them well-suited for large-scale networks and facilitating a more meaningful comparison, and the Louvain method is the most widely used community detection algorithm implemented in most network packages. LS is faster than Louvain for 7 of the 8 benchmarks. The speed advantage becomes more noticeable as the networks get larger (see Table 2). For example, for the DBLP (Digital Bibliography & Library Project) network[47] with 317,080 nodes and 1,049,866 edges, our LS method takes 45 s, while Louvain takes 256 s.

The LS method is not only faster, but also classifies better than the Louvain algorithm measured by the F_1-score for 5 out of 7 examples with ground-truth community labels (see Supplementary Fig. 9 and Supplementary Note 2 for more details and discussions on the evaluation by F_1-score, and we also make comparisons between algorithms on performance evaluated by conductance[47], see Supplementary Table 4). In Table 3, we extend our comparison of the LS algorithm to include other popular algorithms with different perspectives on community detection, some of which are with greater accuracy albeit slower in implementation. For example, the geodesic density gradient (GDG) algorithm[48] first embeds the network into vector space based on shortest path length between nodes and then applies an iterative clustering algorithm, which is similar to mean-shift algorithm[49], both of which are costly in computation, to obtain partitions of communities. GDG algorithm achieves the best performance in two out of seven networks, and has an obvious advantage over other methods on Citeseer (see Table 3 and Supplementary Table 2 for more details). Another important type of method is inferential ones, which provides powerful tools

without arbitrariness and further advances our understanding of community structure of networks[50]. Inferential algorithms, which generally rely on SBM as generative models, can explain the inability to detect communities in the very sparse limit[51], help eliminate the resolution limit in Louvain algorithm and detect hierarchical structure of complex networks[52]. Inferential methods can be adapted to different types of networks ranging from weighted networks[53] to directed ones[54] and hypergraphs[55]. However, inferential methods are generally computationally expensive[52,56]. The inferential algorithm described in Refs. 52,56 has a much better performance than the LS on the Football network (see Supplementary Table 3), whose degree distribution is quite homogeneous. Other algorithms typically only attain the first position in one out of seven networks. While our LS algorithm takes the lead in two out of seven networks and secures second place in an additional two. We note that the best performing algorithms are well distributed among the benchmarks, which reflects that real networks have generally different generating mechanisms that are better captured by some algorithm than others[26,27]. This suggests that achieving optimal performance across all scenarios is highly improbable[57], aligning with the No Free Lunch theorem[27]. It is, however, interesting that LS consistently ranks first or second in the four out of seven benchmark networks, and is overall the best classifier, suggesting that the notions of local dominance, hierarchy and community centers are pervasive in real networks, whose degree distributions are generally heterogeneous. It is also instructive to understand why LS does not perform well on the Football network[8,58,59]. It is due to the fact that the Football network is fairly homogeneous, and we have already explained why the LS method does not perform well in this situation, see the subsection on multiscale community detection. There is also significant connectivity between the largest degree nodes in the ground-truth communities, thus some of them become direct followers to others and their

Table 1 | Basic statistics of networks

	N	E	$\langle k \rangle$	$\langle d \rangle$	$\langle CC \rangle$	ρ	α
Karate	34	78	4.59	2.443	0.256	−0.476	1.781
Football[59]	115	613	10.66	2.397	0.407	0.162	–
Polbooks	105	441	8.40	2.841	0.348	−0.128	1.791
Polblogs	1222	16,717	27.36	2.747	0.226	−0.221	1.415
Cora	2485	5609	4.08	5.738	0.117	−0.055	1.645
Citeseers	2110	3668	3.48	10.257	0.171	−0.024	2.074
PubMed	19,717	44,327	4.50	2.764	0.060	−0.044	2.227

N is the number of nodes in the network, E is the number of edges, $\langle k \rangle$ is the average degree of the network, $\langle d \rangle$ refers to the average shortest path length between all node pairs, $\langle CC \rangle$ refers to the average clustering coefficient, ρ refers to assortativity, and α refers to the power-exponent of the degree distribution if it can be reasonably well fitted by a power law.

Table 2 | Comparison between the LS and Louvain algorithms on networks with ground-truth community labels

	N	E	N_c	Louvain			LS			Δt (ms)
				F_1	N_c	t (ms)	F_1	N_c	t (ms)	
Karate	34	78	2	0.63	4	8	**0.83**	**2**	**6**	2
Football[59]	115	613	10	**0.87**	**10**	**18**	0.35	6	20	−2
Polbooks	105	441	3	0.70	5	13	**0.80**	**2**	**8**	5
Polblogs	1490	19,090	2	**0.85**	9	328	0.69	**3**	**212**	116
Cora	2708	5429	7	0.32	28	380	**0.33**	**7**	**139**	241
Citeseers	3264	9072	6	0.27	35	384	**0.45**	**7**	**131**	253
PubMed	19,717	44,327	3	0.20	43	8745	**0.46**	**8**	**2298**	6447
DBLP	317,080	1,049,866	–	–	220	256,000	–	8; 1859	**45,000**	211,000

N_c denotes the number of ground-truth communities in the network or identified by different methods, and F_1-score is a common performance measure in machine learning between predictions and ground-truth labels (see more details in Supplementary Note 2), and t (ms) is the running time of the algorithm when implemented in Python. As there is no ground truth labels but only meta data for DBLP[47] (see Supplementary Note 2 for more discussions), we are unable to report F_1-score. As LS is able to detect multiscale structure, we report the number of communities detected with notable gaps: 8 large communities, 1859 smaller communities. Both the Louvain and LS algorithm are of linear complexity in time, and our LS method is faster. In addition, the LS method performs better in most cases. The algorithm with a better performance is highlighted in bold. Comparisons with a broader range of classical community detection algorithms are shown in Table 3.

Table 3 | Comparisons with classical community detection algorithms on real networks with ground-truth community labels

	Karate	Football[59]	Polbooks	Polblogs	Cora	Citeseers	Pubmed	Time Complexity
Spin glass[89]	0.61	0.92	0.62	0.88	0.33	0.22	0.21	NA
GN[8]	0.59	0.84	**0.80**	0.74	0.32	0.23	0.18	$O(N^3)$
GDG[48]	**0.91**	0.60	0.75	0.66	0.29	**0.63**	0.19	$O(dtN^2)$
Walktrap[90]	0.51	0.88	0.79	0.88	0.29	0.14	0.16	$O(N^2 \log N)$
Spectral[23]	0.62	0.54	0.70	0.89	0.32	0.25	**0.46**	$O(N^2 \log N)$
Inferential[52,56]	0.65	0.87	0.77	0.32	0.31	0.40	0.07	$O(N^2 \log N)$
Fastgreedy[25]	0.75	0.56	0.78	0.89	**0.39**	0.28	0.32	$O(N \log N)$
Infomap[26]	0.76	**0.96**	0.69	0.80	0.07	0.04	0.01	$O(N \log N)$
LPA[30]	0.88	0.79	0.69	**0.91**	0.22	0.11	0.18	$\sim O(E)$
Louvain[9]	0.63	0.87	0.70	0.85	0.32	0.27	0.20	$\sim O(E)$
LS	0.83	0.35	**0.80**	0.69	0.33	0.45	**0.46**	$O(E)$

The algorithm with the highest F_1 score is highlighted in bold, and the second highest one is highlighted by underline. Overall, our LS algorithm have a pretty good performance, it is ranked first in two out of seven networks and ranked second in another two when compared to other popular algorithms, some of which are slower but more accurate ones. And our algorithm is the fastest one. For the GDG algorithm, d is the dimension of embedding space, and t is the number of iterations.

communities are merged. If a portion of links between the largest degree nodes were removed, the partition given by the LS method would be much closer to the ground truth.

Targeted link removal and addition can significantly change the structure of a network and the outcome of community detection algorithms. The LS algorithm is not immune to that effect, as it relies on local leaders to separate communities, therefore, intentional targeted link addition between two community centers would make one of them a follower and lead to just a single community, which will dramatically reduce the performance of the classification. For example, if we connected the president and instructor in the Zachary Karate Club network, then the LS method only yields a single community (Supplementary Fig. 24), which is the case before the split[60]. This also lends us a way to identify critical links for merging or splitting communities[61]. The identification of local hierarchy is, on the other hand, more robust against links missing or adding at random, see Supplementary Figs. 22-23.

In addition, the number of communities detected by LS is also closer to the ground truth, see Table 2. For example, for the Zachary Karate Club network, Louvain detects four communities, while LS detects two, which is consistent with reality. As usually more potential centers can be detected in real networks, see Supplementary Fig. 6, and might correspond to meaningful multiscale structure. As for the Polblogs network, where LS finds three instead of two communities indicated by current labeling, and there is debate whether three groups should be considered as the ground truth (i.e., apart from liberal and conservative, there is a neutral community)[62]. This partially explains why the LS does not work that well on this example. This also reflects the importance and difficulty of obtaining ground-truth labels, if there are any[27]. Although the evaluation of the classification performance of an algorithm with a ground truth is standard practice[63], establishing the ground truth for community assignment usually require detailed survey, which can be difficult for very large networks[40,63], and is usually regarded as distinct from metadata available[27,40]. The choice(s) of the ground truth(s) is crucial and there might be alternative ground truth that emerge from unsupervised clustering analysis and are validated a posteriori. The notion of alignment between ground truth and structure is indeed crucial to obtain good clusters[64]. For example, in the well known Zachary Karate Club network[60], the metadata of nodes can also be their gender, age, major, ethnicity, however, most of which are irrelevant to the community structure when interested in understanding the split of the club[27,40], but might be relevant to understand other type of community structure. Apart from evaluations based on ground-truth labels, various evaluation criteria purely based on network structure (e.g., optimizing modularity, conductance, cut) have been proposed, however, they may deviate from the real generating process of networks and will not be suitable for all scenarios. For example, maximizing modularity cannot generate good partitions in ecological networks, as herbivores in the same community will not prey on each other, thus there are no dense connections within the same ecological community. In this sense, if there can be some ground-truth labels, using F_1-score is a more objective evaluation.

Applications to urban systems

Our final example of real-world networks is to uncover the structure of spatial interactions in cities. It also showcases the capacity of LS to adapt to weighted networks, with node degree replaced by the node strength and the least weighted shortest path, where the distance between two adjacent nodes is the reverse of the volume of mobility flow. Many cities have or will evolve from a monocentric to a polycentric structure[65], which can be inferred from the patterns induced in human mobility data. We use human mobility flow networks derived from massive cellphone data at the cellphone tower resolution with careful noise filtering and stay location detection[66–68] for three cities in different continents: Dakar[69], Abidjan[7,70], and Beijing[71,72] (see references therein and Supplementary Material of Liu et al.'s work[7] for more details on obtaining the mobility flow network from cellphone data). The LS algorithm can detect both communities with strong internal interactions and meaningful community centers, see Supplementary Fig. 8 for the decision graph. We find that for the smaller cities Dakar and Abidjan, communities are more spatially compact, while in the larger city, Beijing, they are more spatially mixed, see Fig. 4. This indicates that in Beijing, interactions are less constrained by geometric distance, which might be due to a more advanced transportation infrastructure and a superlinearly stronger and diversified interactions tendency in larger cities[7,73,74]. In addition, the identified community centers correspond to important interaction spaces in cities, see Fig. 4. For example, in Beijing, the top three centers are The China World Trade Center in Chaoyang District, the Zhongguancun Plaza Shopping Mall in Haidian District, and Beijing Economic and Technological Development Zone in Daxing District. In Abidjan, LS detects the Digital Zone, local mosques, and markets as centers. In Dakar, a university and some mosques are detected.

Clustering vector data via the LS algorithm

Community detection and vector data clustering share many similarities, but are often considered separately and having contrasting focus. Our use of local leaders identified by local dominance was directly inspired by the concept of the center of a cluster, which is characterised by a higher centrality measure in its vicinity/neighborhood (e.g., density or degree) and a relatively long distance (i.e., a large l_i) to the nearest object with a large centrality. Local dominance concretely and explicitly identifies fundamental asymmetric leader-follower relation between objects, which naturally give rises to centers. This creates a direct link between the two viewpoints of network science and data science. It is therefore natural to ask whether LS would perform well, or even better, than vector data clustering methods on a discretised version of a data cloud.

To cluster vector data with the LS method, we first need to discretise it into a network. Many methods exist to perform this task, including ϵ-ball, k-

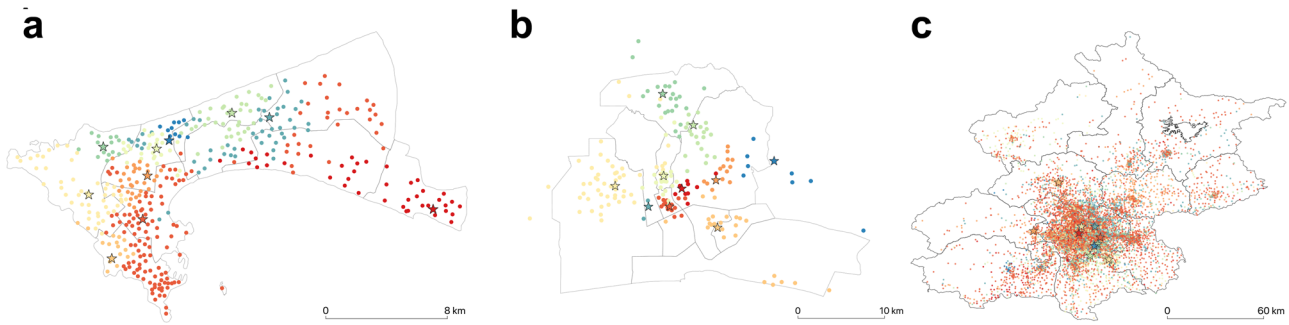

Fig. 4 | The community structure detected by our local search (LS) algorithm on mobility flow networks in three diversified cities across continents. a Dakar in Senegal, Africa. **b** Abidjan in Côte d'Ivoire, Africa. **c** Beijing in China, Asia. Each dot represents a location, which corresponds to a region by Voronoi tessellation according to cellphone towers. Communities are indicated by different colors, and their centers are marked as stars. The decision graphs are shown in Supplementary Fig. 8.

nearest-neighbors (kNN) and its variants (such as mutual kNN, continuous kNN), relaxed maximum spanning tree[75], percolation or threshold related methods[35,76], and more sophisticated ones[77]. Here, we employ the commonly used ϵ-ball method that sets a distance threshold ϵ and connects vectors, which become nodes, whose ϵ-balls overlap, see Fig. 5a and inset. This process can be accelerated by using R-trees and are implemented in a time complexity of $O(N \log N)$[74,78] (see Supplementary Note 1.4). After traversing all nodes, a network encoding a geometric closeness within ϵ between nodes is obtained, see Fig. 5b. The ϵ-ball method preserves spatially local information, e.g., the vector density in the metric space can be interpreted as degree in the constructed network, and coarse-grains continuous distance between objects into discrete values. This makes the determination of centers clearer (see Fig. 5c, d). The choice of ϵ influences greatly the structure of the network obtained, here we chose ϵ to be near the network percolation value to ensure a minimally connected graph[79–81], more details on determining ϵ can be found in Supplementary Note 3.1.

Applying the LS algorithm on the constructed network for a series of well-known two dimensional benchmark data (Fig. 6a, e and Supplementary Fig. 10 for more cases), yields the expected clusters (Fig. 6b, f and Supplementary Fig. 11). By contrast, the Louvain algorithm generally obtains more and smaller clusters in a relatively fragmented way (Fig. 6c, g and Supplementary Fig. 13 for more examples) on the same networks. The reason is that the Louvain algorithm overlook the transitivity of local relations[82]. The state-of-the-art unsupervised clustering algorithm density and distance based (DDB)[16] applied to the original vector data yields expected clusters in most cases, see Fig. 6d and the original work that introduces DDB[16] for other examples. This confirms the universality of local hierarchy between objects and the analogy between our community centers and cluster centers. However, the DDB algorithm fails in the test case[83] in Fig. 6h due to a mixture of local and global metrics in this associate rule[83], which do not affect the LS method (see Fig. 6f). From a network perspective, certain dynamics can give rise to meaningful clusters with arbitrary shapes in metric space (e.g., synchronization or spreading dynamics are usually only possible along the manifold via local interactions but not through global ones). For example, different clusters in Fig. 6e or Supplementary Fig. 11c, g, h might correspond to groups of fireflies that are only able to synchronize within the group rather than between groups, as their interaction range is usually limited. In the situations above, the distance measured by the local metric is more appropriate than the one measured by the global metric, see a more in depth discussion in Supplementary Note 3.2. The good performance of the LS algorithm on vector data resides in the correct identification of the local dominance, i.e., finding the centers, from the local metric.

In addition, we show that the LS method is robust against noisy data in different scenarios, see Supplementary Note 4 and Supplementary Figs. 23-24. Though less common when considering vector data, targeted addition of edges in a network that connect two cluster centers, explicitly brings two cluster centers closer to each other in the metric space and will distort the space, whereas, conversely, the removal of links increases the distances between two objects.

The advantage of building networks for high dimensional vector data. We now show the advantage of combining the ϵ-ball discretisation and community detection methods on clustering high-dimensional data sets. Here, we use well-known benchmark datasets with very high dimensions: the MNIST (Modified National Institute of Standards and Technology) of hand written digits[84], and Olivetti of human faces[85], and show that our simple framework outperforms the-state-of-the-art DDB clustering algorithm[16], see Table 4. Let us consider, for example, the Olivetti human face dataset, a challenging high dimensional dataset with small sample size. Each cluster obtained by the LS algorithm only contain images from a single individual, see Supplementary Figs. 17-19, simply based on Euclidean distance between images and without resorting to using complex image similarity measure. Moreover, it obtains a higher F_1-score than the DDB method. We note that for MNIST and Olivetti datasets, the Louvain algorithm has a higher F_1-score than LS, but identifies an inappropriately large number of clusters. The better performance of the Louvain algorithm lies in some subtle differences from clustering results obtained by the LS method (see comparisons between Supplementary Fig. 19a and Supplementary Fig. 19b for the Olivetti dataset with 100 images. The Louvain algorithm detects all images of the eighth person as one cluster, but the LS method classifies four images of the eighth person as another cluster).

We conjecture that the conversion from vector data to a network is not merely a translation of the data, but a fundamental information filtering process that accentuates the prominence of local leaders and thus increases the strength of local hierarchy, which in practice turns out to be a great advantage of our framework for handling vector data with high dimensions. Constructing the network via ϵ-balls is similar to a coarse-graining process: as long as two objects are close enough, the small differences in distances within ϵ are neglected. In addition, such a process also corresponds to subtracting irrelevant global information and puts the focus on similarity based on a local metric. Though there will be some information loss during the conversion from vector data to topological data, purely local information is enough to identify local dominance in the data. Not all information embedded in the vector data needs be utilized[75], sometimes too much information might complicate the process. Although admitting asymmetric relations between objects would violate certain formal metric properties (e.g., distances are symmetric), it turns out to be an advantage for cluster analysis (see more discussions in Supplementary Note 3).

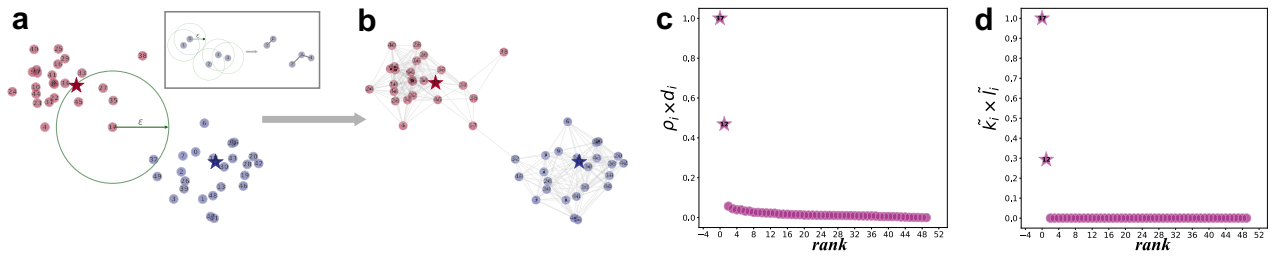

Fig. 5 | Conversion from vector data to a network via the ϵ-ball method and the analogy between the community centers of networks and the cluster centers of vector data. a An example of data cloud and **b** its dicretised network representation by (**Inset**) the ϵ-ball method. **c** The decision graph by the density and distance based (DDB) algorithm[16]. **d** The decision graph by the local search (LS) method. Cluster centers are data points of both a higher density ρ_i than its neighbors and relatively far from other points with a larger density (i.e., a large d_i)[16]. The density ρ_i of a data point i is simply the number of nodes within a certain radius ϵ, and it is equivalent to the degree of node i in the corresponding network (i.e., $k_i = \rho_i$). The network constructing process is a coarse-graining and discretization process, where the absolute distance value is not preserved (e.g., in the Inset, $d_{32} > d_{34}$ for the original vector data, but $l_{32} = l_{34} = 1$ in the network). The Euclidean distance between any data points is based on a global metric, but the topological path length between two nodes are based on a local metric. For example, d_{24} is only slightly larger than d_{34}, but in the network, $l_{24} = 2$ and $l_{23} = 1$ (see the Inset); though $d_{21} \approx 2d_{23}$ according to global metric, node 2 and node 1 are not reachable in the network based on the local metric. Cluster centers identified by the DDB algorithm matches community centers identified by the LS method, which are all marked as stars.

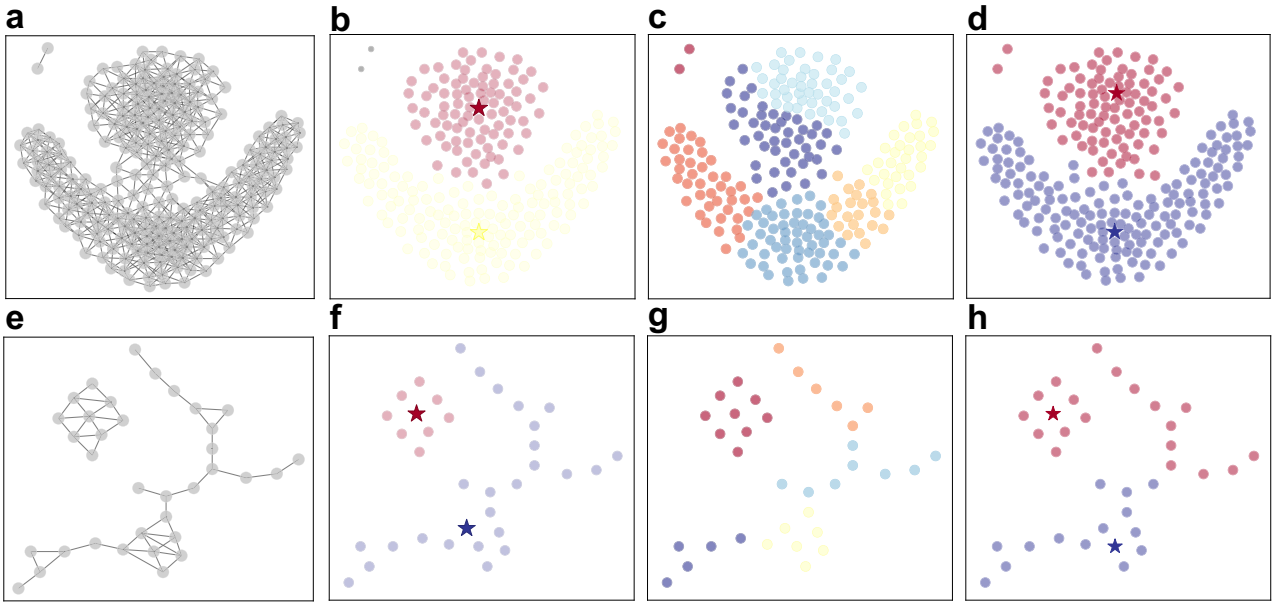

Fig. 6 | Comparisons on the clustering performance between the local search (LS), Louvain, and density and distance based (DDB) algorithms for two dimensional benchmark vector data. a represent the network constructed from vector data using the ϵ-ball method (see Supplementary Note 3.1 and Supplementary Fig. 10 for details on the network constructions). **b** shows the result of the LS method that correctly identify clusters that align with common consensus (see Supplementary Fig. 11 for more cases). In addition, LS can detect noisy points (marked in gray) that are of low degrees but long l_i. **c** shows the partitions obtained from the Louvain method, which are more fragmented than the LS result (see Supplementary Fig. 13 for more cases). **d** shows the result obtained from the DDB method which provides correct partitions to most benchmark data, see the original work that introduces DDB[16] for other cases. **e**–**h** are the same as (**a**–**d**) for another vector dataset, where both a low density manifold and a high density cluster exist. In (**h**), DDB algorithm fails detecting correct clusters due to its local association rule[83] being affected by a mixture of local and global metrics. LS and Louvain methods are performed on the constructed networks shown in (**a**) and (**e**), and the DDB algorithm is performed on the original vector data.

Discussion

Community detection and cluster analysis are analogous as both aim to group objects into categories based on some notion of similarity. In this work, we develop a fast and scalable community detection method based on the notion of a community center that echoes the commonly used concept of a cluster center. The identification of community and cluster structures requires a heterogeneous system: uniformly distributed data points and strictly regular networks do not possess meaningful mesoscopic cluster structure. Heterogeneity leads to the emergence of more important loci in a data space, or central nodes in a network. The notion of center is pervasive in cluster analysis, but underused in community detection. We define community centers as local leaders that are both of a high degree, corresponding to a high density in cluster analysis, and relatively distant from other local leaders, corresponding to cluster separability. The nodes belonging to each community defined by their center are identified by basins of attraction[34] based on the local dominance existing between nodes, which indicates the asymmetric leader-follower relationship and defines a local hierarchy. While dominance is an explicit characteristic of edges in a directed network, it can be seen as an intrinsic hidden higher-order directionality between nodes even in undirected networks. The resulting local hierarchy reflects asymmetric interactions between objects inferred from the local connectivity of nodes that then naturally defines leaders and community affiliations, as well as hierarchies among communities. In addition, the position of local leaders and distribution of shortest path length l_u between local leaders can be developed into some indicators for depicting network structure. And with the concept of local leaders and corresponding local hierarchy, automated discovery[24] and evolution dynamics of communities[61] can be ensuing studies.

The local hierarchy structure is quite robust against random noise, and is identified based on local information. Moreover, in contrast to most state-of-the-art clustering and community detection methods, the LS method does not depend on the structure of the entire network as of most existing methods[24]. We are able to detect communities in a small region and avoid the computational burden of analysing the whole network[24]. In cluster analysis, approximating similarity relations between objects by a distance matrix actually assumes that every object is in a direct relation with all others, which is also the case for modularity optimization algorithms that utilize a random null model, which also assumes that each node has a probability to interact with every other node[10]. In addition, community detection methods also generally assume a mutual relation between objects, which is an important formal metric property and an implicit feature of an undirected connectivity matrix. Local hierarchy implicitly violates such an assumption, but it turns out that abandoning such a restriction gives better flexibility to the clustering method (see Supplementary Note 3 for more details). Finally, our LS algorithm is fast and scalable with a linear time

Table 4 | Comparisons on the clustering performance between the LS, Louvain, and DDB algorithms for high dimensional vector data

	D	N	N_c	LS		DDB		Louvain	
				F_1	N_c	F_1	N_c	F_1	N_c
Iris	4	50	3	0.73	2	**0.82**	**3**	0.70	8
Wine	13	178	3	**0.57**	**3**	**0.57**	**3**	0.41	7
MNIST	784	1,000	10	0.32	**21**	0.26	(10)	**0.45**	247
Olivetti	10,304	100	10	0.74	**14**	0.64	(10)	**0.78**	32
Olivetti	10,304	400	40	0.59	**64**	0.49	(40)	**0.68**	112

D denotes the dimension of the dataset, N denotes the number of objects, and N_c denotes the number of clusters from the ground-truth or identified from algorithms. The hand-written figures in MNIST is of dimension $28 \times 28 = 784$ pixels; and in Olivetti, the face image is of dimension $92 \times 112 = 10{,}304$ pixels. The Olivetti dataset with $N = 100$ comprises the first 100 images of 10 people from the original data set. The original dataset comprises 400 images of 40 different people. Our LS algorithm outperforms DDB in all high-dimensional and large-scale data sets except in Iris, whose dimension is quite low. Note that as the DDB algorithm does not have a clear recognition of the number of clusters (i.e., no clear gaps between centers in the decision graph)[16] for MNIST and Olivetti, the number of clusters identified by DDB are putative based on the ground truth (i.e., selecting the top ten or top forty nodes in the decision graph), which is marked in brackets. The algorithm with the best performance is highlighted in bold.

complexity, which is crucial for analyzing large scale networks, and also performs well on most benchmarks, except the ones that do not possess the type of heterogeneity (e.g., football network[59]) exploited by the LS method.

Overall, the performance of the LS method is particularly good given its simplicity. On benchmark network models, it outperforms the currently most widely used community detection method, the Louvain modularity optimisation algorithm. The LS method consistently ranks higher than any other methods when the performance is averaged over several data sets, see Table 3. We have also shown that the LS method is naturally able to detect multiscale structure of communities in complex networks. This implies that while not necessarily identifying the partition defined by some existing ground truth, it finds a good approximation of it and the output can then be used as starting point for other slower but more accurate and dedicated community detection methods, offering a significant speed up.

Given the similarity in spirit between LS and clustering methods, we applied LS to ϵ-ball discretised version of benchmark vector data, both low and high dimensional. For low-dimensional data, we find it provides the expected clusters and outperforms Louvain modularity optimisation algorithm ran on the discretised data, which generally yields too many communities and performs poorly. LS also outperforms DDB, a state-of-the-art unsupervised clustering method, on some challenging cases in the presence of low-density manifolds. For high-dimensional data, LS still outperforms DDB, but not Louvain, although on closer inspection, Louvain obtains a better F_1-score, but suffers again from providing too many communities, outbalancing the advantage in F_1-score.

We hypothesise that the discretisation step of creating a network from vector data acts as a topological filter, which enhances the key property of the data that makes cluster detection work: the existence of well defined cluster centers and a clearer identification of local hierarchy. The performance of any community detection algorithm is going to be influenced by the discretisation method used, and more work is needed to understand the relationship between topological denoising and the performance of the community detection algorithms, as different community detection methods might respond differently to different discretisation schemes.

Another area for future work is to adapt LS to find halo nodes residing at the boundary of two or more communities (e.g., node d in Fig. 1), detect overlapping communities[13] potentially by producing line graphs[86–88] or clique graphs[58], and identify critical link responsible for the merging or splitting dynamics of communities[61]. Another point that could be improved is when two or more local leaders are equivalent on both degree and distance to a node. We currently assign it to a local leaders at random but we could look at other options.

Finally, another possible direction for future research concerns the definition of dominance itself. In this article, it was built on a specific network property, the degrees of the nodes. For a weighted network, it would be appropriate to use strength rather than degree and we would retain all the benefits of the LS method. Extending LS algorithm to directed networks is worth closer investigations in the future. In directed networks, two types of local leaders, the integrators (determined by in-degree) and the influencers (by out-degree), might be needed, which can lead to two types of clustering. The influence of edge directionality should be closely examined, as influence may propagate in the reverse direction of the directed edge. For example, on Twitter, information often flows from a user to their followers. Additionally, directionality affects the calculation of path lengths between nodes. Apart from using degree, dominance could also be based on other node centrality measures but most of these require global network calculations, which would slow the algorithm considerably. If dominance was based on non-structural properties, such as numerical attributes for nodes already defined in the data, then the LS approach would still work well.

Methods

The local search (LS) algorithm

Cluster analysis and community detection share many conceptual similarities, but often have a contrasting focus. Cluster analysis puts emphasis on the center of a cluster[15,16], while community boundaries often play a more predominant role in community detection[37]. Community centers can be inferred from some community detection algorithm outputs, for example, the nodes associated to the largest absolute weights of the leading eigenvector of the modularity matrix, or exhibiting a higher density of connections inside the communities, are deemed to be community centers, core members or provincial hubs[23,38]. But centers are only a by-product of the algorithm, rather than at their core of methodologies. The approach that we propose here is explicitly focusing on community centers to identify clusters, which is motivated by the existence of underlying asymmetries between nodes[19–21], the concept of local leaders[18] in networks and borrows ideas from density and distance based clustering algorithms on vector data[16]. We hypothesise that a community center is a local leader that is comparatively of a larger degree than its neighbors, thus dominating them, and is of a relatively long shortest-path distance to other local leaders.

Our algorithm consists of four steps that we now detail. We start with an undirected network with N nodes and E edges, for example see Fig. 1a. For better clarity, nodes are also labeled and traversed in lexicographical order (see Fig. 1b).

Step 1 First, we calculate the degree k_u of each node u (see digits in Fig. 1a), which is an operation of linear time complexity O(E). Our algorithm neglects self-loops in default, but if self-loops are meaningful for calculating degree of nodes, setting the input parameter self_loop of the algorithm as True will increase the degree of nodes accordingly, and nodes with self-loops will not be considered as neighbors of themselves.

Step 2 Second, we traverse each node u and point u to any adjacent node v with $k_v \geq k_u$ and $k_v = \max\{k_z | z \in \mathbf{V}(u)\}$ (i.e., v has the largest degree in the neighborhood of u). For example, in Fig. 1b node g will point to f instead of p as $k_f > k_p > k_g$; and c temporarily points to both b and m as $k_b = k_m = \max\{k_z | z \in \mathbf{V}(c)\} > k_c$. Note that a node cannot point to its follower, and since nodes are traversed in lexicographical order, when node b is traversed, it will point to m as $k_m = \max\{k_z | z \in \mathbf{V}(b)\} \geq k_b$. When m is traversed, it will not point to any of its followers (e.g., b). This process naturally avoids the creation of loops and ensure we only obtain directed acyclic graphs (DAGs), see Supplementary Fig. 1 and proof in Supplementary Note 1.1.1 for more details. If such a v does not exist, u will not have any out-going edge and will be identified as a local leader (see dark gray nodes f, p, and m in Fig. 1b). We denote the set of local leaders as $\mathbf{C}$.

After traversing all nodes, for nodes with multiple out-going links, we randomly retain one (see only short-dash arrows in Fig. 1c for a possible mapping). Mathematically, we have obtained a forest of trees, where the root of each tree is a local leader, and is also a potential community center. For most nodes, except local leaders, this process identifies a local hierarchy (indicated by dash arrows), with an asymmetric leader-follower relation (see short-dash arrows in Fig. 1b). This step is completed in O(E).

Step 3 Third, to identify the upper level for local leaders along the hierarchy, we use a local breadth-first search (LBFS) starting from each local leader u and stop the search when encountering the first local leader v with $k_v \geq k_u$ and assign the shortest path length on the original network d_{uv} to l_u, which is the length of the out-going link of node u. Note that $l_u \geq 2$ for all local leaders, and all pure followers have $l_u = 1$. For example, node p is a local leader, in the second iteration of the LBFS, it encounters another local leader f with $k_f > k_p$. We then stop the LBFS and point p to f, and $l_p = d_{pf} = 2$. Similarly, $f \rightarrow m$ and $d_{mf} = 4$. The out-going link of local leaders goes beyond the direct connections in the original network (see long-dash arrows in Fig. 1c).

When there are several local leaders that have a no smaller degree than the local leader u in the l_u^{th} iteration, the largest one is chosen; if multiple local leaders in this iteration have the same largest degree, one is picked at random uniformly. For local leader(s) with the maximal degree in the whole network, denoted as $\mathbf{M}$, a subset of $\mathbf{C}$, there is no need to perform the LBFS, and we directly assign $l_{x \in \mathbf{M}} = \max_{u \in \mathbf{C} \setminus \mathbf{M}}(l_u)$.

Community centers can be easily identified as local leaders with both a large k_u and a long l_u (see Fig. 1e), and naturally emerges from the rooted tree revealed by local dominance (see all dash arrows in Fig. 1c and the explicit tree structure in Fig. 1d). We use the product of rescaled degree $\tilde{k}_i$ and rescaled distance $\tilde{l}_i$ to quantitatively measure the centerness of each node (see more details and discussions in Supplementary Note 1.2). Community centers can be determined via visual inspection for obvious gaps or by, possibly, sophisticated automated detection methods for gaps in the future (see Fig. 1f). For example, the community centers identified by the LS method in Fig. 1 are nodes *f* and *m*. In the Zachary Karate Club network, the identified community centers correspond to the president and the instructor, which is consistent with reality[60] (see Supplementary Fig. 6).

Theoretically, this third step takes $\mathrm{O}\left((|\mathbf{C}|-|\mathbf{M}|)\langle k\rangle^{\langle l\rangle_{\mathbf{C}\setminus\mathbf{M}}}\right)=\mathrm{O}((|\mathbf{C}|-|\mathbf{M}|)E)$, where $\langle k\rangle$ is the average degree of the network, $\langle l\rangle_{\mathbf{C}\setminus\mathbf{M}}=\sum_{u\in\mathbf{C}\setminus\mathbf{M}} l_u/(|\mathbf{C}|-|\mathbf{M}|)$, and the size of the set of potential centers $|\mathbf{C}|$ is usually much smaller than N (see Supplementary Table 1). In practice, $\langle k\rangle^{\langle l\rangle_{\mathbf{C}\setminus\mathbf{M}}}$ is bounded to be smaller than E as it mimics a LBFS process. As indicated by numerical results, even $(|\mathbf{C}|-|\mathbf{M}|)\langle k\rangle^{\langle l\rangle_{\mathbf{C}\setminus\mathbf{M}}}$ is usually smaller than E (see Supplementary Table 1). In addition, the LBFS process can be simultaneously implemented for all local leaders in parallel to further speed up the algorithm in practice.

Step 4 Finally, for all identified community centers, we remove their outgoing links, if any. Community labels are then assigned along the reverse direction of directionality $u \leftarrow v$ from community centers. This step takes again a linear time $\mathrm{O}(N)$.

Taken together, the time complexity of our LS algorithm is linear in the number of edges: $\mathrm{O}\left(E+(|\mathbf{C}|-|\mathbf{M}|)\langle k\rangle^{\langle l\rangle_{\mathbf{C}\setminus\mathbf{M}}}+N\right)=\Theta(E)$, which is among the fastest community detection algorithms. To the best of our knowledge, our framework provides a new perspective on community detection methods. It only relies on the notion of local dominance, which is identified solely from local information from the topology. It does not need to iteratively optimize an objective function[9,26–29] based on a global randomized null model[9,23,27] or resorting to iterative spreading dynamics[30,31] as other state-of-the-art algorithms. It is important to emphasise that the communities that are uncovered by LS are not necessarily associated to a high density of links, as in modularity optimisation, or specific patterns of connectivity inside versus across groups, as in methods based on stochastic block models[39–41], but are instead obtained as a group of nodes that are dominated by the same leader.

Data availability

All network and vector datasets needed to evaluate the conclusions in the paper are publicly available and present in the paper and/or the Supplementary Materials. The original cellphone datasets of Dakar and Abidjan are accessed through the D4D challenge, and the Beijing dataset is obtained from a Chinese telecommunication operator and the original dataset is not publicly available.

Code availability

Computer code for the LS algorithm and code implementing the analysis described in this paper and other information is online at https://github.com/UrbanNet-Lab/LS_for_CommunityDetection_and_Clustering.

Received: 3 July 2023; Accepted: 17 April 2024;
Published online: 31 May 2024

References

1. Manning, C. D., Raghavan, P. & Schütze, H. *Introduction to Information Retrieval* (Cambridge University Press, 2008).
2. Albert, R. & Barabási, A.-L. Statistical mechanics of complex networks. *Rev. Mod. Phys.* **74**, 47 (2002).
3. Barabási, A.-L. *Network Science* (Cambridge University Press, 2016).
4. Newman, M.*Networks* (Oxford University Press, 2018).
5. McPherson, M. An ecology of affiliation. *Am. Sociol. Rev.* **48**, 519–532 (1983).
6. Blondel, V. D., Decuyper, A. & Krings, G. A survey of results on mobile phone datasets analysis. *EPJ Data Sci.* **4**, 10 (2015).
7. Liu, C. et al. Revealing spatio-temporal interacting patterns behind complex cities. *Chaos* **32**, 081105 (2022).
8. Girvan, M. & Newman, M. E. Community structure in social and biological networks. *Proc. Natl. Acad. Sci.* **99**, 7821–7826 (2002).
9. Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, P10008 (2008).
10. Fortunato, S. Community detection in graphs. *Phys. Rep.* **486**, 75–174 (2010).
11. Fortunato, S. & Hric, D. Community detection in networks: A user guide. *Phys. Rep.* **659**, 1–44 (2016).
12. Holme, P., Huss, M. & Jeong, H. Subnetwork hierarchies of biochemical pathways. *Bioinformatics* **19**, 532–538 (2003).
13. Palla, G., Derényi, I., Farkas, I. & Vicsek, T. Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**, 814–818 (2005).
14. Clauset, A., Moore, C. & Newman, M. E. Hierarchical structure and the prediction of missing links in networks. *Nature* **453**, 98–101 (2008).
15. Kaufman, L. & Rousseeuw, P. J. *Finding groups in data: an introduction to cluster analysis* (John Wiley & Sons, 2009).
16. Rodriguez, A. & Laio, A. Clustering by fast search and find of density peaks. *Science* **344**, 1492–1496 (2014).
17. Li, H.-J. & Daniels, J. J. Social significance of community structure: Statistical view. *Phys. Rev. E* **91**, 012801 (2015).
18. Blondel, V. D., Guillaume, J.-L., Hendrickx, J. M., de Kerchove, C. & Lambiotte, R. Local leaders in random networks. *Phys. Rev. E* **77**, 036114 (2008).
19. Serrano, M. Á., Boguná, M. & Vespignani, A. Extracting the multiscale backbone of complex weighted networks. *Proc. Natl Acad. Sci.* **106**, 6483–6488 (2009).
20. Stanoev, A., Smilkov, D. & Kocarev, L. Identifying communities by influence dynamics in social networks. *Phys. Rev. E* **84**, 046102 (2011).
21. Lee, M. J. et al. Uncovering hidden dependency in weighted networks via information entropy. *Phys. Rev. Res.* **3**, 043136 (2021).
22. Li, R. et al. Gravity model in dockless bike-sharing systems within cities. *Phys. Rev. E* **103**, 012312 (2021).
23. Newman, M. E. Modularity and community structure in networks. *Proc. Natl. Acad. Sci.* **103**, 8577–8582 (2006).
24. Fortunato, S. & Newman, M. E. 20 years of network community detection. *Nat. Phys.* **18**, 848–850 (2022).
25. Clauset, A., Newman, M. E. & Moore, C. Finding community structure in very large networks. *Phys. Rev. E* **70**, 066111 (2004).
26. Rosvall, M. & Bergstrom, C. T. Maps of random walks on complex networks reveal community structure. *Proc. Natl Acad. Sci.* **105**, 1118–1123 (2008).
27. Peel, L., Larremore, D. B. & Clauset, A. The ground truth about metadata and community detection in networks. *Sci. Adv.* **3**, e1602548 (2017).
28. Duch, J. & Arenas, A. Community detection in complex networks using extremal optimization. *Phys. Rev. E* **72**, 027104 (2005).
29. McLachlan, G. J. & Krishnan, T.*The EM algorithm and extensions*, vol. 382 (John Wiley & Sons, 2007).
30. Raghavan, U. N., Albert, R. & Kumara, S. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E* **76**, 036106 (2007).
31. Frey, B. J. & Dueck, D. Clustering by passing messages between data points. *Science* **315**, 972–976 (2007).

32. Delvenne, J.-C., Yaliraki, S. N. & Barahona, M. Stability of graph communities across time scales. *Proc. Natl. Acad. Sci.* **107**, 12755–12760 (2010).
33. Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N. & Barabási, A.-L. Hierarchical organization of modularity in metabolic networks. *Science* **297**, 1551–1555 (2002).
34. Sales-Pardo, M., Guimera, R., Moreira, A. A. & Amaral, L. A. N. Extracting the hierarchical organization of complex systems. *Proc. Natl. Acad. Sci.* **104**, 15224–15229 (2007).
35. Hoffmann, T., Peel, L., Lambiotte, R. & Jones, N. S. Community detection in networks without observing edges. *Sci. Adv.* **6**, eaav1478 (2020).
36. Reichardt, J. & Bornholdt, S. When are networks truly modular? *Phys. D.* **224**, 20–26 (2006).
37. Zitnik, M., Sosič, R. & Leskovec, J. Prioritizing network communities. *Nat. Commun.* **9**, 2544 (2018).
38. Guimera, R. & Nunes Amaral, L. A. Functional cartography of complex metabolic networks. *Nature* **433**, 895–900 (2005).
39. Holland, P. W., Laskey, K. B. & Leinhardt, S. Stochastic blockmodels: First steps. *Soc. Netw.* **5**, 109–137 (1983).
40. Newman, M. E. & Clauset, A. Structure and inference in annotated networks. *Nat. Commun.* **7**, 11863 (2016).
41. Schaub, M. T., Li, J. & Peel, L. Hierarchical community structure in networks. *Phys. Rev. E* **107**, 054305 (2023).
42. Fortunato, S. & Barthelemy, M. Resolution limit in community detection. *Proc. Natl. Acad. Sci.* **104**, 36–41 (2007).
43. Ravasz, E. & Barabási, A.-L. Hierarchical organization in complex networks. *Phys. Rev. E* **67**, 026112 (2003).
44. Mucha, P. J., Richardson, T., Macon, K., Porter, M. A. & Onnela, J.-P. Community structure in time-dependent, multiscale, and multiplex networks. *Science* **328**, 876–878 (2010).
45. Barabási, A.-L. & Albert, R. Emergence of scaling in random networks. *Science* **286**, 509–512 (1999).
46. Evans, T. S. & Chen, B. Linking the network centrality measures closeness and degree. *Commun. Phys.* **5**, 172 (2022).
47. Yang, J. & Leskovec, J. Defining and evaluating network communities based on ground-truth. *Knowl. Inf. Syst.* **42**, 181–213 (2013).
48. Mahmood, A., Small, M., Al-Maadeed, S. A. & Rajpoot, N. Using geodesic space density gradients for network community detection. *IEEE Trans. Knowl. Data Eng.* **29**, 921–935 (2016).
49. Comaniciu, D. & Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**, 603–619 (2002).
50. Peixoto, T. P. *Descriptive vs. inferential community detection in networks: Pitfalls, myths and half-truths* (Cambridge University Press, 2023).
51. Decelle, A., Krzakala, F., Moore, C. & Zdeborová, L. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Phys. Rev. E* **84**, 066106 (2011).
52. Peixoto, T. P. Hierarchical block structures and high-resolution model selection in large networks. *Phys. Rev. X* **4**, 011047 (2014).
53. Aicher, C., Jacobs, A. Z. & Clauset, A. Learning latent block structure in weighted networks. *J. Complex Netw.* **3**, 221–248 (2015).
54. Larremore, D. B., Clauset, A. & Jacobs, A. Z. Efficiently inferring community structure in bipartite networks. *Phys. Rev. E* **90**, 012805 (2014).
55. Contisciani, M., Battiston, F. & De Bacco, C. Inference of hyperedges and overlapping communities in hypergraphs. *Nat. Commun.* **13**, 7229 (2022).
56. Peixoto, T. P. Bayesian stochastic blockmodeling. In *Advances in network clustering and blockmodeling*, vol. 11, 289–332 (Wiley Online Library, 2019).
57. Peixoto, T. P. & Kirkley, A. Implicit models, latent compression, intrinsic biases, and cheap lunches in community detection. *Phys. Rev. E* **108**, 024309 (2023).
58. Evans, T. S. Clique graphs and overlapping communities. *J. Stat. Mech. Theory Exp.* **2010**, P12037 (2010).
59. Evans, T. S. American college football network files, https://doi.org/10.6084/M9.FIGSHARE.93179.V2
60. Zachary, W. W. An information flow model for conflict and fission in small groups. *J. Anthropol. Res.* **33**, 452–473 (1977).
61. Palla, G., Barabási, A.-L. & Vicsek, T. Quantifying social group evolution. *Nature* **446**, 664–667 (2007).
62. Adamic, L. A. & Glance, N. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery*, 36–43 (2005).
63. Hric, D., Darst, R. K. & Fortunato, S. Community detection in networks: structural clusters versus ground truth. *Phys. Rev. E* **90**, 062805 (2014).
64. Qian, Y., Expert, P., Rieu, T., Panzarasa, P. & Barahona, M. Quantifying the alignment of graph and features in deep learning. *IEEE Trans. Neural Netw. Learn. Syst.* **33**, 1663–1672 (2021).
65. Louf, R. & Barthelemy, M. Modeling the polycentric transition of cities. *Phys. Rev. Lett.* **111**, 198702 (2013).
66. Alexander, L., Jiang, S., Murga, M. & González, M. C. Origin–destination trips by purpose and time of day inferred from mobile phone data. *Transp. Res. Part C: Emerg. Technol.* **58**, 240–250 (2015).
67. Çolak, S., Alexander, L. P., Alvim, B. G., Mehndiratta, S. R. & González, M. C. Analyzing cell phone location data for urban travel: current methods, limitations, and opportunities. *Transp. Res. Rec.* **2526**, 126–135 (2015).
68. Çolak, S., Lima, A. & González, M. C. Understanding congested travel in urban areas. *Nat. Commun.* **7**, 10793 (2016).
69. Dong, L., Li, R., Zhang, J. & Di, Z. Population-weighted efficiency in transportation networks. *Sci. Rep.* **6**, 26377 (2016).
70. Li, R., Wang, W. & Di, Z. Effects of human dynamics on epidemic spreading in Côte d'Ivoire. *Phys. A* **467**, 30–40 (2017).
71. Xu, Y., Li, R., Jiang, S., Zhang, J. & González, M. C. Clearer skies in Beijing — revealing the impacts of traffic on the modeling of air quality. In *The 96th Annual Meeting of Transportation Research Board, Washington DC*, 17-05211 (2017).
72. Xu, Y. et al. Unraveling environmental justice in ambient pm2.5 exposure in Beijing: A big data approach. *Comput. Environ. Urban Syst.* **75**, 12–21 (2019).
73. Schläpfer, M. et al. The scaling of human interactions with city size. *J. R. Soc. Interface* **11**, 20130789 (2014).
74. Li, R. et al. Simple spatial scaling rules behind complex cities. *Nat. Commun.* **8**, 1841 (2017).
75. Qian, Y., Expert, P., Panzarasa, P. & Barahona, M. Geometric graphs from data to aid classification tasks with graph convolutional networks. *Patterns* **2**, 100237 (2021).
76. Bullmore, E. T. & Bassett, D. S. Brain graphs: graphical models of the human brain connectome. *Annu. Rev. Clin. Psychol.* **7**, 113–140 (2011).
77. Berenhaut, K. S., Moore, K. E. & Melvin, R. L. A social perspective on perceived distances reveals deep community structure. *Proc. Natl. Acad. Sci.* **119**, e2003634119 (2022).
78. Guttman, A. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, 47–57 (1984).
79. Bollobás, B. & Béla, B. *Random graphs*. 73 (Cambridge University Press, 2001).
80. Bunde, A. & Havlin, S. *Fractals and disordered systems* (Springer Science & Business Media, 2012).
81. Li, D. et al. Percolation transition in dynamical traffic network with evolving critical bottlenecks. *Proc. Natl. Acad. Sci.* **112**, 669–672 (2015).
82. Traag, V. A., Waltman, L. & Van Eck, N. J. From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep.* **9**, 5233 (2019).

83. Pizzagalli, D. U., Gonzalez, S. F. & Krause, R. A trainable clustering algorithm based on shortest paths from density peaks. *Sci. Adv.* **5**, eaax3770 (2019).
84. Chatfield, K., Lempitsky, V. S., Vedaldi, A. & Zisserman, A. The devil is in the details: an evaluation of recent feature encoding methods. In *Proceedings of the British Machine Vision Conference*, 76.1–76.12 (2011).
85. Samaria, F. S. & Harter, A. C. Parameterisation of a stochastic model for human face identification. In *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, 138–142 (IEEE, 1994).
86. Evans, T. S. & Lambiotte, R. Line graphs, link partitions and overlapping communities. *Phys. Rev. E* **80**, 016105 (2009).
87. Evans, T. S. & Lambiotte, R. Line graphs of weighted networks for overlapping communities. *Eur. Phys. J. B* **77**, 265–272 (2010).
88. Ahn, Y.-Y., Bagrow, J. P. & Lehmann, S. Link communities reveal multiscale complexity in networks. *Nature* **466**, 761–764 (2010).
89. Reichardt, J. & Bornholdt, S. Statistical mechanics of community detection. *Phys. Rev. E* **74**, 016110 (2006).
90. Pons, P. & Latapy, M. Computing communities in large networks using random walks. In *International symposium on computer and information sciences*, 284–293 (Springer, 2005).

Acknowledgements

R.Li acknowledges helpful discussions with Dr. Gezhi Xiu and Dr. Wenyi Fang from Peking University. F.S. acknowledges technical help from Mr. Ankang Luo and Ms. Chenxin Liu from UrbanNet Lab. This work receives financial supports from the National Natural Science Foundation of China (Grant No. 72371014, 61903020). L.L. acknowledges financial supports from the National Natural Science Foundation of China (Grant No. T2293771), STI 2030–Major Projects (Grant No. 2022ZD0211400), and the New Cornerstone Science Foundation through the XPLORER PRIZE. R.Lambiotte acknowledges support from the EPSRC Grants EP/V013068/1 and EP/V03474X/1.

Author contributions

R.Li conceived the research. R.Li, B.C., H.E.S., L.L., P.E., T.E, and R.Lambiotte designed the research. R.Li and F.S. designed the first version of the LS algorithm, T.E. and D.S. further refined and improved the algorithm. F.S. and D.S. implemented the LS algorithm and conducted experiments. R.Li, F.S., D.S., B.C., P.E., L.L., T.E. and R.Lambiotte discussed the results and wrote the manuscript. R.Li was the lead writer of the manuscript. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42005-024-01635-4.

Correspondence and requests for materials should be addressed to Bingsheng Chen, Linyuan Lü, Renaud Lambiotte or Ruiqi Li.

Peer review information *Communications Physics* thanks the anonymous reviewers for their contribution to the peer review of this work. A peer review file is available.

Reprints and permissions information is available at http://www.nature.com/reprints

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

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

© The Author(s) 2024