Computational Statistics
https://doi.org/10.1007/s00180-024-01457-6

ORIGINAL PAPER

Overlapping coefficient in network-based semi-supervised clustering

Claudio Conversano[1] · **Luca Frigau**[1] · **Giulia Contu**[1]

Received: 3 March 2022 / Accepted: 8 January 2024

Abstract

Network-based Semi-Supervised Clustering (NeSSC) is a semi-supervised approach for clustering in the presence of an outcome variable. It uses a classification or regression model on resampled versions of the original data to produce a proximity matrix that indicates the magnitude of the similarity between pairs of observations measured with respect to the outcome. This matrix is transformed into a complex network on which a community detection algorithm is applied to search for underlying community structures which is a partition of the instances into highly homogeneous clusters to be evaluated in terms of the outcome. In this paper, we focus on the case the outcome variable to be used in NeSSC is numeric and propose an alternative selection criterion of the optimal partition based on a measure of overlapping between density curves as well as a penalization criterion which takes accounts for the number of clusters in a candidate partition. Next, we consider the performance of the proposed method for some artificial datasets and for 20 different real datasets and compare NeSSC with the other three popular methods of semi-supervised clustering with a numeric outcome. Results show that NeSSC with the overlapping criterion works particularly well when a reduced number of clusters are scattered localized.

Keywords Regression tree · Complex networks · Community detection · Walktrap · Louvain · Label propagation · Coefficient of overlapping

✉ Claudio Conversano
conversa@unica.it

Luca Frigau
frigau@unica.it

Giulia Contu
giulia.contu@unica.it

[1] Department of Economics and Business Sciences, University of Cagliari, Viale S. Ignazio da Laconi 17, 09123 Cagliari, Italy

Published online: 19 February 2024

1 Introduction

There has been an increasing interest in semi-supervised clustering in the last decades. Following Aggarwal (2014, Chapt. 20), semi-supervised clustering is concerned with situations where some a-priori information about the assignments of observations to clusters is available and the goal is to incorporate and take advantage of this prior information to improve the quality of the obtained partitions. This improvement consists in finding a feasible solution to a complex clustering problem which is unreachable using unsupervised algorithms or finding a better-quality solution than what could be found using unsupervised methods. Sometimes, data clustering mainly depends on a single outcome variable or is characterized by different levels of this outcome. In these cases, using a clustering algorithm that gives in some way prior importance to this variable might be advantageous, particularly in situations when the different levels of this outcome, and thus the different clusters obtainable from data partitioning, depends in turn from the levels of the other observed variables.

In the following, we deal with semi-supervised clustering associated with an outcome variable and focus on a recently proposed method called Network-based Semi-Supervised Clustering [NeSSC (Frigau et al. 2021)] aimed at finding a partition of the original data into a certain number of disjoint clusters, say K, that are the least possible overlapped with respect to the outcome.

NeSSC works by training several times a classification/regression model on reweighed versions of the original data. This model estimates the outcome based on an ad-hoc selected single predictor. In each trial/iteration, estimated values of the outcome allow us to measure the proximity or degree of similarity between pairs of observations and to feed a proximity matrix. This matrix is subsequently used to define a complex network on which a community detection algorithm is finally applied to obtain a partition of the data into a certain number of disjoint groups. NeSSC's main goal, or most desirable result, is thus the possibility to order the K clusters according to increasing mean values of the outcome with overlapping between density curves reduced as much as possible. Minimum overlapping guarantees, at the same time, that the K clusters are as much as possible internally homogeneous and externally heterogeneous with respect to the outcome. This use of an optimality criterion for data represented within a network structure is quite common in the neural network literature [see, for example, de Jesus Rubio et al. (2022), de Jesus Rubio (2021)].

NeSSC can be used in all situations when data are unlabelled but there is one observed variable that is considered of primary importance in driving the clustering process. One of the main advantages of this method is its flexibility as it can be straightforwardly applied for datasets composed of numerical or categorical variables as well as for those composed of both types of variables. Frigau et al. (2021, Sect. 4.1) a detailed motivating example concerning house renting (Munich Rent Data) has been presented. It is there demonstrated that the monthly net rent is an important driver of the clustering process as using this variable as the outcome in NeSSC leads to distinguishing among five different groups of

houses characterized by different average renting prices as well as allows to identify specific characteristics of houses for each group. Similar results are obtained also for the Boston housing dataset, for a dataset about career statistics of major league baseball players (Hitters) as well as for another dataset including information about characteristics of tourism websites (Conversano et al. 2019).

In this paper, we focus on the case the outcome is numerical and consider an algorithm implemented in NeSSC called Community Detection Tree-based Algorithm (CoDe-Tree). We propose an alternative criterion based on a coefficient of overlapping between density curves to select the optimal partition as well as a penalty term that prevents overestimation of the number of clusters. The effectiveness of the overlapping criterion is assessed through a comparative analysis involving four main methods used in the context of semi-supervised clustering with a numerical outcome, including CoDe-Tree, that are evaluated on artificial data and on 20 real datasets.

A typical situation where NeSSC based on the overlapping criterion is supposed to work well is when the clusters' structure is scattered localized rather than cohesive with a low percentage of overlap between clusters. Specifically, the clusters' structure is considered cohesive when the majority of the observations came from a single multi-dimensional distribution, otherwise scattered localized. The percentage of overlap, instead, concerns the number of observations generated by one multidimensional distribution different from the one characterizing its own cluster. In Fig. 1 we report an illustrative example of three clusters in a two-dimensional space: the top-left panel shows a cohesive structure of the clusters, whilst in the top-right panel they are characterized by scattered localization. The percentage of overlap is represented by the observations generated by one multidimensional distribution different from the one characterizing its own cluster. In this particular case, data are generated by three multivariate gaussian distributions for cohesive structure and six multivariate gaussian distributions (two for each cluster) for scattered localization. The bottom panels, instead, display the distributions of the outcome variable y by clusters, generated by gaussian distributions. The details of the data-generating process for this example are described in Appendix A.

In the case of scattered localization, although the plot shows six data clouds, the presence of the outcome y requires the clustering algorithm to find three groups. Clustering data using NeSSC with the selection of optimal partitions based on overlapping densities (see Sect. 3.2) provides the best results both comparing with the 26 hierarchical and non-hierarchical clustering methods implemented in the R package NbClust (Charrad et al. 2014) as well as with the original NeSSC settings and other semi-supervised benchmarking clustering methods considered (see Sect. 4).

The rest of the paper is organized as follows. We recall the basics of NeSSC in Sect. 2 summarizing the main features of the three steps of the procedure. Next, we focus on the problem of choosing the optimal partition in NeSSC in Sect. 3 and describe the three criteria implemented in NeSSC (Sect. 3.1) before introducing the new overlapping-based criterion (Sect. 3.2), the penalization mechanism used to avoid overestimation of clusters (Sect. 3.3) and discuss convergence, consistency and computational complexity (Sect. 3.4). Section 4 presents the comparative analysis whilst Sect. 5 ends the paper with some concluding remarks.

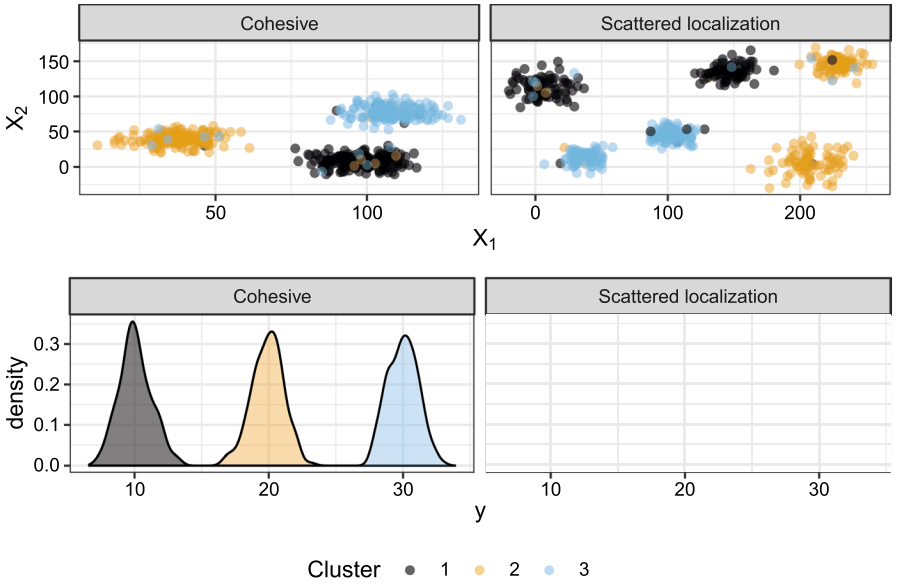

Fig. 1 Illustration with three clusters of the characteristics concerning the cohesion of their structure and the percentage of overlap between them in the control variables' space: the top-left panel shows a cohesive situation, the top-right a situation characterized by a scatter localization. The percentage of overlap is represented by the observations generated by one multidimensional distribution different from the one characterizing its own cluster. The bottom panels, instead, illustrate the distributions of the outcome variable y by clusters

2 Network-based semi-supervised clustering

2.1 Main features

The goal of Network-based Semi-Supervised Clustering (NeSSC) is searching for a partition of data that depends on an outcome variable which is in some way a proxy of the clusters of interest. These clusters are found as those presenting the minimum overlapping in terms of the distribution of the outcome. In the following, we describe the basic steps of NeSCC.

The basic features of NeSSC are represented in Fig. 2. Data consist of p (categorical or numerical) variables $x_1, \ldots, x_p$ and an outcome variable y, both observed for n observations or instances. The final partition is found based on y, which has a priority in determining the clustering structure.

The assessment of the proximity between observations is based on a properly-defined proximity matrix $\mathbf{\Pi}$ derived from simple models $m(\cdot)$ that utilizes y as response and one randomly-selected variable x_j as the predictor. Since x_j could be either numerical or categorical $m(\cdot)$ is specified as a regression or classification model, respectively, and is estimated $b^* \gg 0$ times on re-weighted versions of the original data. A set of weights associated with each observation and each predictor is utilized in each

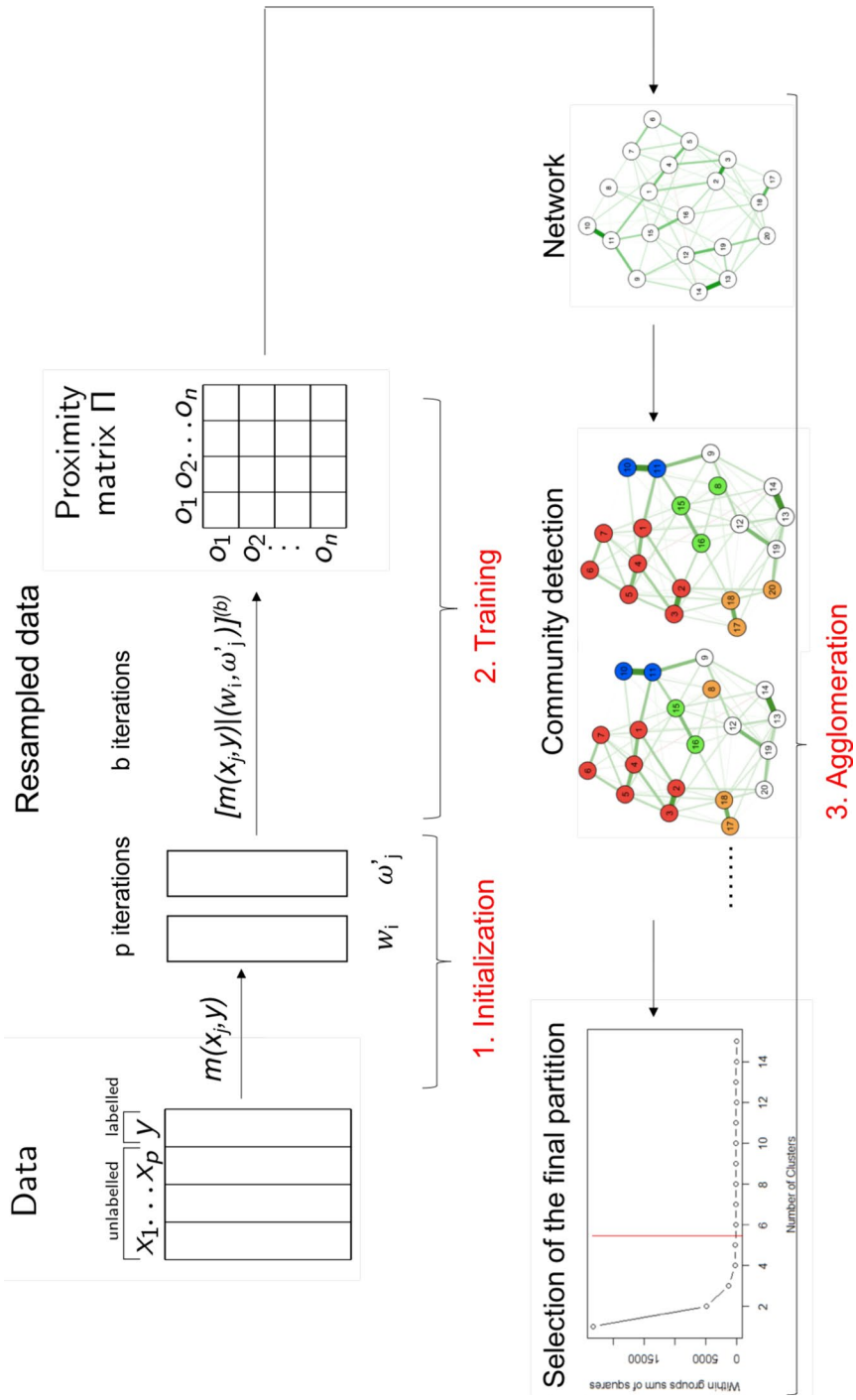

Fig. 2 NeSSC flowchart

iteration to select the observations and the predictor to be used in $m(\cdot)$. Thus, $\mathbf{\Pi}$ is updated based on the proportion of times pairs of observations have been assigned the same estimated value by $m(\cdot)$. Next, data are represented in a network with weights $\mathbf{\Pi}$ and a community detection algorithm is applied to select a final partition.

NeSSC is composed of three steps, described in Sect. 2.1.[1] The first step is *initialization* and it is aimed at assigning a proper system of initial weights to both observations and variables.

Next, a *training* step allows the user to estimate the proximity between pairs of observations. The final *agglomeration* step uses community detection algorithms to find the best partition.

2.2 NeSSC's three steps

Initialization The model $m(\cdot)$ is estimated p times to set initial weights to observations and variables using the predictor x_j ($j = 1, \ldots, p$). For each x_j, two instances i and i' ($i \neq i'$ and $i, i' = 1, \ldots, n$) are considered similar if $\hat{y}_i = \hat{y}_{i'}$. Consequently, the observation weights $w^{(j)}$ are updated by computing the inverse of the average proximity obtained in the previous $j-1$ iterations. Thus, more weight is assigned to cases that are less likely to be paired one with another by $m(\cdot)$. The original weight assigned to the i-th case is $1/n$ and is updated in iteration j (see line 5 in Algorithm 1). The indicator function $I(\cdot)$ equals one if $\hat{y}_i^{(q)} = \hat{y}_{i'}^{(q)}$, and zero otherwise. It checks if the predicted value (label) of y corresponds to the observed value (label) of y in the case y is numerical (categorical).

Algorithm 1: Initialization

1: **for** $j = 1$ to p **do**

2: $\quad \hat{y}^{(j)} = \hat{m}\,(x_j, y)^{(j)}$

3: $\quad \pi_{i,i'} = j^{-1} \sum_{q=1}^{j} I\left(\hat{y}_i^{(q)} = \hat{y}_{i'}^{(q)}\right)_{i,i' \in [1,n], i \neq i'}$

4: $\quad \omega_j^{(j)} = \dfrac{\sum_{q=1}^{j} \mathcal{G}(x_{j'})^{(q)}}{\sum_{q=1}^{j} \mathcal{G}(x_{j'})^{(q)}}$ $\quad (j'=j)$

5: $\quad w_i^{(j)} = \left[1 - \sum_{i'=1}^{n} \left(\dfrac{\sum_{q=1}^{j} I\left(\hat{y}_i^{(q)} = \hat{y}_{i'}^{(q)}\right)}{\sum_{q=1}^{j} \sum_{i'=1}^{n} I\left(\hat{y}_i^{(q)} = \hat{y}_{i'}^{(q)}\right)}\right)^2\right]^{-1}$

6: **end for**

The weighting scheme for variables is based on their predictive ability: the models $m(x_j, y)$ are compared on the basis of their goodness of fit (accuracy) if y is numerical (categorical). For the variable x_j, the weight $\omega_j^{(j)}$ is updated on the basis of the global goodness of fit of $m(x_j, y)^{(j)}$, say $\mathcal{G}(x_j)^{(j)}$. The latter corresponds to the relative decrease in the deviance of the outcome if y is numerical or to accuracy if

[1] See Frigau et al. (2021) for a more detailed description of the NESSC steps.

y is categorical. The initial weight assigned to x_j is $1/p$ and it is updated after each iteration according to the equation reported in line 4 in Algorithm 1. The variable $x_{j'}$ is used to train $m(\cdot)$ at iteration q and the numerator considers all cases when the variable j' sampled in the j-th iteration corresponds to x_j.

The matrix $\mathbf{\Pi}_{n,n}$ derives from the repeated estimation of y_i using reweighed variables and observations. The basic idea is assigning more weight to observations that are more prone to be assigned the same $\hat{y}_i$ and variables that mostly contribute to these assignments. In this respect, a partition of the original data composed of internally homogeneous groups that are as much as possible different one from another is obtained [see Porro and Iacus (2009) for a similar approach in causal inference studies].

Training The model $m(x_j, y)^{(b)}$ is estimated $b^* - p$ times ($b = p+1, \ldots, b^*$, i.e.: from iteration $p+1$ to b^*) to update the observations' weights $w^{(b-1)}$ by selecting, in each iteration, the predictor x_j based on $\omega^{(b-1)}$ and refreshing the proximity matrix $\mathbf{\Pi}_{n,n}$, whose entrance (i, i') in iteration b is defined in line 5 of Algorithm 2.

The matrix $\mathbf{\Pi}_{n,n}$ derives from the repeated estimation of y_i ($i = 1, \ldots, n$) using reweighed variables and observations. The basic idea is assigning more weight to observations that are more prone to be assigned the same $\hat{y}_i$ and variables that mostly contribute to these assignments. This would lead to a partition of the original data composed of internally homogeneous groups that are as much as possible different one from another. A similar approach to the estimation of the proximity matrix has been used in the framework of matching methods for the estimation of treatment effects in causal inference studies (Porro and Iacus 2009).

Algorithm 2: Training

1: Set $\beta = 0$
2: **while** $\beta \neq \gamma \wedge b < B^{\max}$ **do**
3: sample $1 \leqslant j \leqslant p$ according to $\omega^{(b-1)}$
4: $\hat{y}^{(b)} = \hat{m}\left(x_j, y\right)^{(b)}$
5: $\pi_{i,i'} = b^{-1} \sum_{q=1}^{b} I\left(\hat{y}_i^{(q)} = \hat{y}_{i'}^{(q)}\right)_{i,i' \in [1,n], i \neq i'}$
6: $\beta = \sum_{h=1}^{\gamma} \left[I\left(\mathbf{1}^{\intercal} \left| \mathbf{\Pi}^{(b)} - \mathbf{\Pi}^{(b-h)} \right| \mathbf{1} < n(n-1) \cdot \epsilon\right)\right]$
$\frac{\sum_{q=1}^{b} \mathcal{G}(x_{j'})^{(q)}}{(j'=j)}$
7: $\omega_j^{(b)} = \frac{(j'=j)}{\sum_{q=1}^{b} \mathcal{G}(x_{j'})^{(q)}} \qquad j' \in (1, \ldots, p)$
8: $w_i^{(b)} = \left[1 - \sum_{i'=1}^{n} \left(\frac{\sum_{q=1}^{b} I\left(\hat{y}_i^{(q)} = \hat{y}_{i'}^{(q)}\right)}{\sum_{q=1}^{b} \sum_{i'=1}^{n} I\left(\hat{y}_i^{(q)} = \hat{y}_{i'}^{(q)}\right)} \right)^2 \right]^{-1}$
9: **end while**

Agglomeration The matrix $\mathbf{\Pi}^{(b^*)}$ is transformed into a set $\mathcal{N}$ of T complex networks according to a function $\psi : \mathbf{\Pi}^{(b^*)} \to \mathcal{N}$, and a community detection algorithm $\mathcal{A}$ is trained on the nodes of each complex network N_t to find, for each network, a certain number of homogeneous communities corresponding to the ℓ_t disjoint groups defining a proper partition of the original data (Algorithm 3).

Algorithm 3: Agglomeration

if approach is Weighted **then**
  $\psi\left(\mathbf{\Pi}^{(b^*)}\right) = \mathcal{N} = \{N_1\} \implies \lambda^* = \mathcal{A}(N_1)$
else
  $\psi\left(\mathbf{\Pi}^{(b^*)}\right) = \mathcal{N} = \{N_1, \ldots, N_T\}$
  for $N_t \in \mathcal{N}$ **do**
    $\lambda_t = \mathcal{A}(N_t)$
    if criterion is Mean **then**
      $\zeta_t = \left[\sum_h \sum_z \mathcal{E}\left(y_{g(h)}, y_{g(z)}\right)\right] / \binom{\ell_t}{2}$ with $h > z$ and $g(h), g(z) \in \lambda_t$
    end if
    if criterion is Threshold **then**
      $\zeta_t = \left[\sum_h \sum_z I\left[\mathcal{E}\left(y_{g(h)}, y_{g(z)}\right) > \tau\right]\right] / \binom{\ell_t}{2}$ with $h > z$ and $g(h), g(z) \in \lambda_t$
    end if
    if criterion is Overlapping **then**
      $\zeta_t = \mathcal{M}\{\mathcal{O}[g(h), g(z)]\}$ with $h > z$ and $g(h), g(z) \in \lambda_t$
    end if
  end for
  $\lambda^* = \lambda_t : t = \text{argmin}_{t \in \mathcal{T}}\left[\zeta_t \cdot \rho_{\ell_t}\right]$
end if

Each complex network $N_t \in \mathcal{N}$ is defined on the basis of a threshold value t that indicates if, for a generic $\pi_{i,i'}$, an edge exists between observations i and i'. If $\pi_{i,i'} \geqslant t$ an edge between i and i' exists and $\pi_{i,i'} = 0$, otherwise $\pi_{i,i'} = 1$. To generate the set $\mathcal{N}$ composed of T networks N_t $(t = 1, \ldots, T)$, a set of thresholds $\mathcal{T} = \{t_1, \ldots, t_T\}$ is required. To define this set, $\mathbf{\Pi}^{(b^*)}$ is reorganized into a $R \times 2$ matrix $\mathbf{K}$ that reports in each column the number of pairs of observations to which the same estimated value of the outcome ($\hat{y}$) has been assigned and the number of assignments, respectively.[2] $\mathbf{K}$ is ordered with respect to its first column in a non-decreasing way. Next, the structural changes of the empirical fluctuation process of the frequency distribution of the k_r values (first column of $\mathbf{K}$) are detected considering different bandwidths. The breakpoints in which the frequency of each $k_{r,2}$ (the empirical fluctuation process) is different from a linear model with a null slope ($p < 0.05$) are the threshold values $t \in \mathcal{T}$. The empirical process is assumed to follow a moving sums of residuals (MOSUM) process and an OLS-based MOSUM test is performed (Zeileis et al. 2003). If no breakpoints are detected through the empirical fluctuation tests, $\mathcal{T}$ is composed of the deciles of the frequency distribution of the k_r values. The community detection algorithm $\mathcal{A}$ applied to the each network N_t lead to T possible partitions $\lambda_1, \ldots, \lambda_T$ of the original data, of size $\ell_1, \ldots, \ell_T$. As for the choice of the final partition, the four alternative criteria discussed in Sect. 3 can be considered.[3]

[2] In practice, $\mathbf{K}$ corresponds to a frequency table reporting in each row the potential value of an individual entry of $b^* \cdot \mathbf{\Pi}^{(b^*)}$ and its associated frequency. For example, $k_{1,1} = (300, 100)$ indicates that 100 pairs of observations have been assigned the same $\hat{y}$ for 300 times.

[3] In the current implementation of NeSSC, three alternative community detection algorithms can be used: Louvain (Blondel et al. 2008), Walktrap (Pons and Latapy 2005) and Label Propagation (Raghavan et al. 2007).

3 Choosing the optimal partition

3.1 Weighted, mean and threshold criteria

When NeSSC operates on the original cells of $\mathbf{\Pi}^{(b^*)}$ the set $\mathcal{N}$ of possible networks is composed of a single network N_1 only, whose edges are defined by the entries $\pi_{i,i'}$ of $\mathbf{\Pi}^{(b^*)}$. Thus, the community detection algorithm $\mathcal{A}$ is trained on N_1 and a single partition λ^* into ℓ groups is obtained. Since in this case each element $\pi_{i,i'}$ indicates the strength of the connection between i and i', the $\pi_{i,i'}$s weight the edges of the network. This criterion is termed *weighted criterion* and is described in line 2 of Algorithm 3.

Instead, if the set of T thresholds introduced in Sect. 2 is considered, T alternative partitions λ_t are obtained and thus the optimal one has to be chosen. The original NeSSC implementation utilizes a statistical test $\mathcal{E}$ to select the optimal partition. To this purpose, the Tukey's Honestly Significant Difference Test (Tukey 1949) or the Fisher's Exact Test (Mehta and Patel 1983) is used depending on the type of outcome. The former is used when y is numeric to test for significant differences between clusters in terms of mean values of y. The latter is used when y is categorical to test for the independence between the categories of y and the cluster labels. In both cases, the final partition is chosen alternatively as the one minimizing the average p-value arising from pairwise group comparisons. This criterion is named *mean criterion* and is described in lines 8 and 17 of Algorithm 3. Alternatively, with the same tests it is possible to consider the minimum proportion of times the p-values arising from pairwise groups comparisons are higher than a pre-specified threshold τ (usually, $\tau = .01$ or $\tau = .05$). This criterion is named *threshold criterion* and is described in lines 11 and 17 of Algorithm 3.

3.2 Overlapping index for the numerical outcome case

In the following, an alternative criterion is introduced when NeSSC is used in the case of a numerical outcome y. Consistent with the main aim of NeSSC, which is searching for subgroups of cases that are as much as possible not overlapped one with another respect to y, the overlapping index (Inman and Bradley 1989) is considered to select the final partition.

Broadly speaking, the overlapping index η between two probability density functions $f_A(x)$ and $f_B(x)$ is defined as:

$$\eta_{A,B} = \int_{\mathbb{R}} \min \left[f_A(x), f_B(x) \right] dx \tag{1}$$

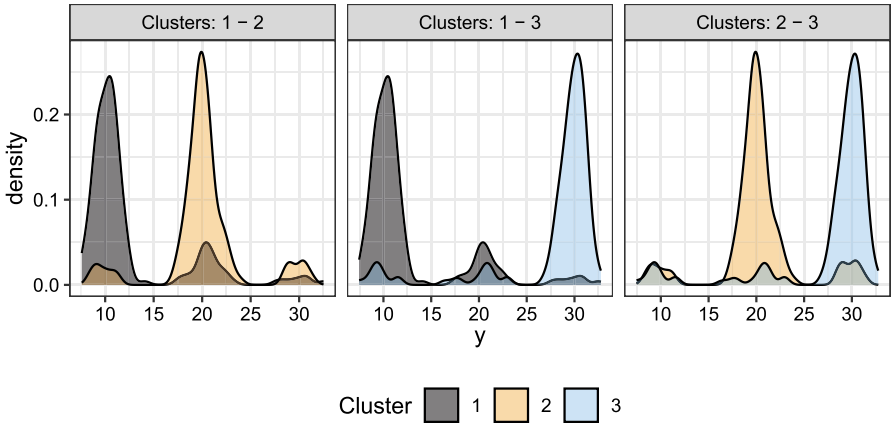

Fig. 3 An example of overlapping distributions concerning the use of NeSSC with a numerical outcome y for data represented in Fig. 1. The method finds a partition in four groups, g_1 to g_3, and each plot shows the empirical distribution of y within pairs of groups and the related value of the overlapping coefficient

where the integral can be replaced by summation in the discrete case. Usually, $\eta_{A,B}$ is normalized to one, thus easily indicating that the two distributions $f_A(x)$ and $f_B(x)$ might be not overlapped ($\eta_{A,B} = 0$) or fully overlapped ($\eta_{A,B} = 1$).

The index introduced in Eq. (1) can be also used as a dissimilarity measure as it can be computed as

$$\eta_{A,B} = 1 - \delta_{A,B} = 1 - \left(\frac{1}{2} \int_{\mathbb{R}} |f_A(x) - f_B(x)| dx \right) \tag{2}$$

In both cases [Eqs. (1), (2)], the overlapping index $\eta_{A,B}$ can be computed either analytically, when the densities $f_A(x)$ and $f_B(x)$ are known, or approximately when researchers have no particular knowledge about the parametric form of $f_A(x)$ and $f_B(x)$.

The $\eta_{A,B}$ index has been applied to several research problems. In psychology, it is the basis of Cohen's U index, as well as McGraw and Wong's CL and Huberty's I degree of non-overlap index measure. All these indexes are based on some distributional assumptions to be properly met (e.g., symmetry of the distributions, unimodality, same parametric family) that guarantee asymptotic properties but may sometimes limit the application of these measures in practice.

The current version of NeSSC implements the distribution-free overlapping index introduced in Pastore and Calcagni (2019). This index is defined in more general terms without resorting to the use of strong distributional assumptions and is appropriate when researchers need to quantify the magnitude of some phenomena like differences, distances, and evidence. Operationally, the distribution-free overlapping index is defined by replacing the unknown densities $f_A(x)$ and $f_B(x)$ introduced in Eqs. (1) and (2) with their kernel density estimates $\hat{f}_A(x)$ and $\hat{f}_B(x)$, obtained once a kernel function and a bandwidth parameter have

been chosen by the user. The effectiveness of this data-driven approach has been positively evaluated in Clemons and Bradley (2000) and in Schmid and Schmidt (2006).

Figure 3 shows an example of the distribution-free overlapping index for the simulated data shown in Sect. 1. Each plot shows the pairwise empirical density of y within the three groups (g_1, g_2, g_3) arising from the final partition found by NeSSC, with the corresponding value of the overlapping index $\mathcal{O}\left[g_h, g_z\right]$.

In NeSSC, the pairwise overlapping indexes are synthesized into a single index ζ_t that summarizes the degree of overlap between the $\binom{\ell_t}{2}$ pairs of groups characterizing a partition λ_t obtained from a complex network N_t with $t \in \mathcal{T}$. Thus, recalling that a community detection algorithm is applied to a proper transformation of the proximity matrix $\mathbf{\Pi}^{(b^*)}$ from which T partitions each one composed of ℓ_t groups are obtained, the degree of overlapping indexes ζ_t are defined as follows (line 14 of Algorithm 3)

$$\zeta_t = \mathcal{M}\left\{\mathcal{O}\left[g(h), g(z)\right]\right\} \tag{3}$$

with $h > z$ and $g(h)$, $g(z)$ being groups of the partition λ_t containing ℓ_t groups in total. The function $\mathcal{M}\{\cdot\}$ computes a central tendency measure of the different pairwise groups overlapping values, such as the mean, the median or the weighted mean. Among all the possible (final) partitions λ_t $(t = 1, \ldots, T)$, the optimal (selected) partition λ^* is that presenting the lowest minimum penalized average overlapping (line 17 of Algorithm 3)

$$\lambda^* = \lambda_t : t = \underset{t \in \mathcal{T}}{\operatorname{argmin}} \left[\zeta_t \cdot \rho_{\ell_t}\right] \tag{4}$$

where ρ_{ℓ_t} is a penalty term used to prevent overestimation of the number of clusters [see Sect. 3.3 and Eq. (5)]. The criterion introduced in Eq. (4) to select the final partition leads to a satisfactory solution if data are not completely dense. It is not appropriate in the case all the pairwise distributions are completely or almost completely overlapped: in these cases, it is very difficult to find a clustering method able to distinguish the different groups.

3.3 Avoiding overestimation of the number of clusters

When choosing one final partition among a set of T alternative partitions, a penalty measure is used to prevent overestimation of the number of clusters. This penalty, called ρ_ℓ, is used either for the mean criterion or for the threshold criterion. It is computed as the ratio between the logarithmic transformation of the possible number of pairs obtainable from a partition ℓ and that of the maximum theoretical number of pairs of clusters obtainable with n cases:

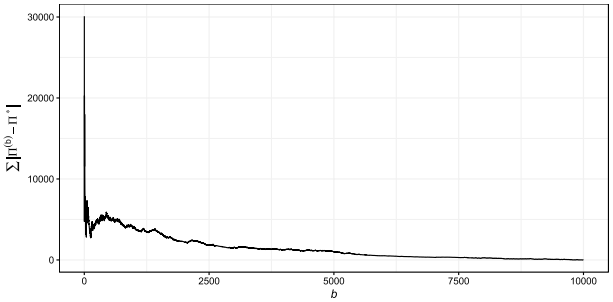

Fig. 4 Convergence of $\mathbf{\Pi}$ to $\mathbf{\Pi}^*$ for the example dataset represented in Figs. 1 and 3

$$\rho_\ell = \frac{\log\left[\binom{\ell}{2} + 1\right]}{\log\left[\binom{n}{2} + 1\right]} \tag{5}$$

ρ_ℓ is defined in (0, 1] and is independent from n. It discourages candidate partitions with an excessive number of clusters, penalizing them logarithmically. When using one of the criteria described in Sects. 3.1 and 3.2 (mean, threshold, overlapping) the index ζ_t, expressing the degree of equivalence between groups, is multiplied by ρ_{ℓ_t} (see line 17 of Algorithm 3).

3.4 Convergence, consistency and computational complexity

Three issues related to the choice of the optimal partition arising from NeSSC are the convergence of the training step (Algorithm 2), the stability of the solution as long as the size of the dataset increases and the computational complexity of the algorithm.

As for the number of iterations required to stabilize the proximity matrix $\mathbf{\Pi}$, NeSSC assumes that as long as the number of iterations increases $\mathbf{\Pi}$ converges to the matrix $\mathbf{\Pi}^*$. The latter expresses the true propensity of (subsets of) observations to behave in the same manner with respect to y. The algorithm convergences when the average element-wise absolute difference $|\mathbf{\Pi}^{(b)} - \mathbf{\Pi}^{(b-h)}|$ is negligible, i.e. lower than a user-defined minimum acceptable value ϵ for γ subsequent iterations (line 6 of Algorithm 2). As long as this condition is met, the final number of iterations is b^* $(p+1, \ldots, b^*, \ldots, B^{\max})$ and the final proximity matrix is $\mathbf{\Pi}^{(b^*)}$. In any case, to avoid overly computational complexity, the user is required to specify a maximum number of iterations $B^{\max}$.

The convergence of $\mathbf{\Pi}$ to $\mathbf{\Pi}^*$ is obtainable since by increasing the number of iterations b the most explanatory variables are repeatedly selected and used in the model $m(\cdot)$, and thus the entries of $\mathbf{\Pi}$ tends to stabilize. As an example, Fig. 4 shows how the average element-wise absolute difference $|\mathbf{\Pi}^{(b)} - \mathbf{\Pi}^{(b-h)}|$ stabilizes when increasing the number of iterations b for the example dataset represented in Figs. 1 and 3.

The difference $|\mathbf{\Pi}^{(b)} - \mathbf{\Pi}^{(b-h)}|$ is null at iteration 10, 000, which corresponds to the user-defined maximum number of iterations.

As for the consistency of the NeSSC's results, we checked empirically if the goodness of the partition obtained by NeSSC depends on the size of the dataset. To this purpose, we generated 231 simulated datasets made up of four clusters with a constant average value of pairwise overlaps (0.05) by varying: the number of the independent variables (from two to eight), the size of the datasets (11 different sizes from 50 to 2000) and the community detection algorithm (Louvain, Walktrap and Label Propagation). The obtained partitions were compared to the true ones through the Adjusted Rand Index (ARI) (Hubert and Arabie 1985) (see Table 2 in Appendix B). The differences among ARI averages by size resulted not statistically significant by ANOVA test (p value = 0.85).

Frigau et al. (2021, Sect. 5.2, pp. 196–198) it has been demonstrated that the computing time required by NeSSC for a dataset composed of 200 observations is always below 10 s and that the computational complexity of the algorithm depends on the threshold ϵ used to define the stopping criterion, the number of instances n and the number of variables p. Specifically, low values of ϵ require more iterations to complete the training step. Nonetheless, setting a reasonable value for $B^{\max}$, the maximum number of iterations, overcomes that issue. Instead, as long as n increases the computational time of both the regression/classification model $m(\cdot)$ and of the community detection algorithm $\mathcal{A}$ increase often more than linearly. The number of variables p affects directly the initialization step, since the number of required iterations equals p, and indirectly the training step since increasing p causes an increase in the number of iterations required to stabilize the weights vector ω and, consequently, $\mathbf{\Pi}$. Last but not least, the computational time of the weighted criterion is lower than those of the unweighted ones, because the latter handle T complex networks instead of just one. All the above-mentioned considerations can be extended to NeSSC based on the overlapping criterion, which belongs to the unweighted criteria since the major change introduced in this case pertains to the criterion used to select the final partition whilst all the other features of the method remain unchanged.

In the following, the effectiveness of the *overlapping criterion* is evaluated by using a proper specification on NeSSC both on artificial data and on 20 different real datasets. In both cases, the overlapping criterion has been compared with the other criteria specified for NeSSC (weighted, mean and threshold) and with three alternative methods used in semi-supervised clustering with a numerical outcome.

4 Comparative analysis

4.1 Methods

In the following, the results of a comparative analysis among different methods for semi-supervised clustering with a numerical outcome are presented. The set of alternative methods includes a tree-based specification of NeSSC as a baseline method and three possible alternatives as the benchmark.

The baseline method is the CoDe-Tree algorithm, which is one of the possible specifications of NeSSC introduced in Frigau et al. (2021). Consistent with the notation introduced in Sect. 2, CoDe-Tree considers classification or regression trees as classification/regression model $m(\cdot)$. Thus, in the initialization step two observations i and i' are considered as similar if they fall into the same terminal node of a tree, and the weights assigned to each variable are computed on the basis of the variable importance obtained from each tree. In view of that, for each tree, $m(x_j, y)$ the function $\mathcal{G}(x_{j'})$ computes the total decrease in nodes impurity generated by the tree using x_j as a unique predictor. In the specific case of a numerical outcome, in the training step of NeSSC b^* regression trees ($b^* = p+1, \ldots, B^{\max}$) are grown on reweighed versions of the original data in order to derive the entries of the proximity matrix $\mathbf{\Pi}^{(b^*)}$ from which the set of T partitions $\lambda_1, \ldots, \lambda_T$ is obtained in the agglomeration step. As for the choice of the optimal partition, we consider the standard approaches based on the *weighted* and *mean* criteria utilized in Frigau et al. (2021) and the newly proposed *overlapping criterion* introduced in Sect. 3.2. For each criterion, we use Louvain as a possible community detection algorithm.

As for the benchmarks, the following three methods are considered:

- Semi-Supervised Clustering, SS-Clust (Bair and Tibshirani 2004): this approach has been introduced in the framework of the analysis of gene expression data and consists of three steps. First, an a-priori selection of the most predictive variables based on a test statistic that differs based on the nature of the predictor itself (numerical, categorical, censored survival time) is carried out. Next, K-means clustering is applied to the subset of most predictive variables in order to assign each observation a class label. Finally, a Nearest Shrunken Centroid classifier is trained on labelled data to predict new observations based on the previously identified K groups. This method requires a priori knowledge of the number of underlying clusters and usually, it does not perform well in situations where there are some overlapping classes.
- Supervised principal components analysis (SPCA) (Bair et al. 2006): this approach can be considered as a variant of SS-Clust to be used when many predictors are available ($p \gg 1$). It computes univariate standard regression coefficients for each predictor and selects those presenting a coefficient larger than a pre-specified threshold estimated by cross-validation. Next, a principal component analysis is performed on the first reduced subset in order to obtain a second reduced subset that includes the first (or first few) principal components only. Lastly, the second reduced subset is the input of a K-means clustering which determines the final partition. Beyond the required prior knowledge of the number of underlying clusters, SPCA is limited by interpretability issues typical of PCA.
- Semi-supervised recursively partitioned mixture models (SS-RPMM) (Koestler et al. 2010): this method first selects, using a scoring function, the subset of variables that are the more associated with the outcome and then applies a recursively partitioned mixture model to estimate the number of clusters in an efficient way. A possible disadvantage of SS-RPMM is that the variables discarded in the first screening step are ignored in the following steps. In addition, the final parti-

tion is not correctly identified if the outcome is originally not strongly associated with the clusters.

Summarizing, the comparative analysis considers six alternative methods to semi-supervised clustering with a numerical outcome (three specifications of CoDe-Tree and three alternative methods). All the compared methods have been implemented in the R environment for statistical computing (R Core Team 2021).

4.2 Simulation study

We simulate different controlled and calibrated scenarios of 500 observations by generating several artificial datasets according to different levels of complexity. The basic data-generating process is the one represented in Fig. 1 and described in Appendix A with some additional design factors. The design factors in the different simulation scenarios are defined according to both the main elements of the clusters [e.g., Van Mechelen et al. (2018)] and two specific characteristics of the proposed method. As for the former, we consider:

(a) the number of control variables: 2, 6 or 10;
(b) the presence of noisy variables: 0% or 33% of the number of variables;
(c) the number of groups ℓ: 2, 3, 5;
(d) the proportion of qualitative variables: 0% or 50%.

The other specific characteristics concern the cohesion of the clusters' structure (Cohesive or Scattered localization) and the percentage of overlap between clusters in the control variables' space (5%, 10% or 15%).

We first consider the four main elements of the clusters, (a)–(d), together with the cohesion of the clusters' structure and geerate 72 artificial datasets with no overlap between clusters. Data are generated from a ℓ-dimensional multivariate normal distribution in case of cohesive structure and two ℓ-dimensional multivariate normal distributions with different mean vectors for each cluster in case of scattered localization. In both cases, the mean vectors are randomly generated and the standard deviations are defined so that there is no overlap between clusters. Next, the overlap is generated by randomly swapping some observations among clusters. For each percentage of overlap, we repeat the random swapping 20 times to obtain 4320 ($72 \times 3 \times 20$) artificial scenarios. Finally, the outcome y is generated as follows: dealing with ℓ groups, y is generated from gaussian distributions with mean $\mu_l = 10 \cdot l, (l = 1, \ldots, \ell)$, and standard deviation $\sigma = 1.2$.

Among the set of possible indexes used to assess cluster validity [see for example Arbelaitz et al. (2013) and Halkidi et al. (2015)] to evaluate the performance of the compared methods, we use ARI to compare the partitions obtained from the algorithms with the true ones. In particular, we compute for each design factor the percentage of times a clustering algorithm obtains the best ARI (ties allowed) over all scenarios where that specific factor is being considered.

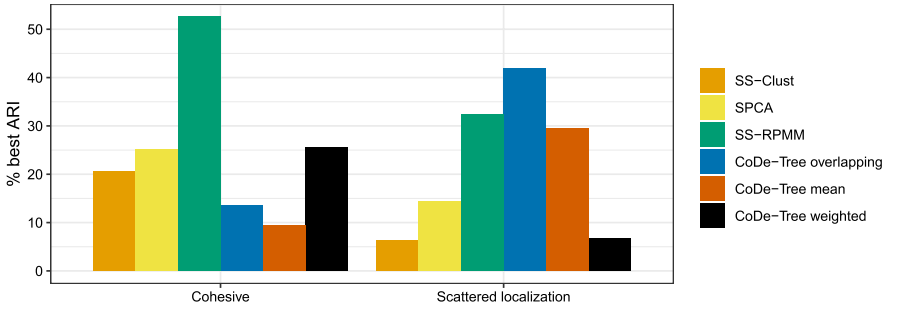

Fig. 5 The percentage of times the six clustering algorithms obtained the best ARI over all scenarios according to the cohesion of the clusters' structure (Cohesive or Scattered localization)

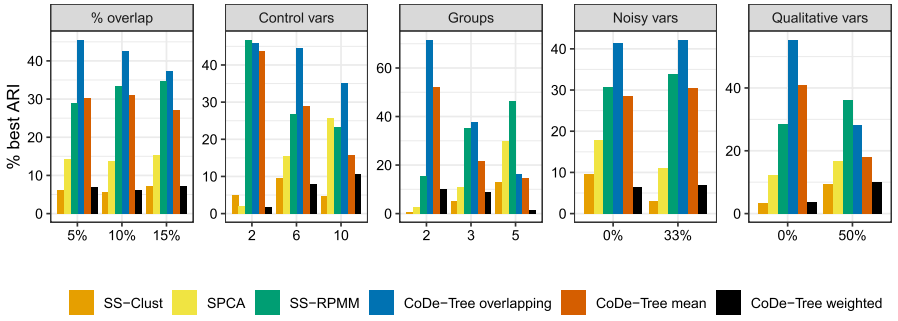

Fig. 6 The percentage of times the six clustering algorithm obtained the best ARI (ties allowed) over all scenarios characterized by a scattered localization of the clusters' structure, respectively according to the percentage of overlap between clusters in the control variables' space, the number of control variables, the number of the groups, the presence of noisy and qualitative variables

A first important result is illustrated in Fig. 5, which points out the best method in case of cohesive structure of the clusters is SS-RPMM in more than 50% of the simulated datasets. Good results, between 20 and 25%, are obtained also from the other two benchmarking semi-supervised clustering as well as the weighted setting of NeSSC. Considering the scattered localization scenarios, CoDe-Tree overlapping is the method that obtained the best ARI the highest number of times ($\approx 40\%$) followed by SS-RPMM and CoDe-Tree mean.

By focusing solely on the scenarios characterized by scattered localization, we obtain the results illustrated in Fig. 6.

In these cases CoDe-Tree overlapping outperforms the other methods for all the considered percentages of overlap between clusters, decreasing its relative performance whilst increasing the degree of overlap. The contrary is, instead, for SS-RPMM, whilst the performance of the other methods are constant. Considering the number of control variables, if these are two CoDe-Tree overlapping performs as good as SS-RPMM and CoDe-Tree mean, whereas with six and ten control variables CoDe-Tree overlapping obtains the best ARI much more often than other methods. Moving to the number of

Table 1 Datasets used in the comparative analysis

R Package	Name	Title	*y*	Rows	Cols (num/cat)
Ecdat	Clothing	Sales of men's fashion stores	Tsales	400	13 (13/0)
Ecdat	Housing	Sales prices of houses in Windsor	Price	546	12 (6/6)
gamlss.data	Rent99	Munich rent of 1999	Rentsqm	3082	10 (6/4)
Matching	Lalonde	Lalonde's NSW demonstration	re78	614	10 (10/0)
vcd	Hitters	Hitters	Salary	263	20 (17/3)
AER	CASchools	California test score	Score	420	11 (9/2)
AER	CPS1985	Determinants of wages (CPS 1985)	Wage	534	11 (4/7)
AER	Fatalities	US traffic fatalities	Frate	336	33 (29/4)
AER	Guns	More guns, less crime?	Violent	1173	13 (11/2)
AER	HousePrices	House prices in Windsor	Price	546	12 (6/6)
AER	Journals	Economics journal subscription	Subs	180	9 (6/3)
AER	MASchools	Massachusetts test score	Score4	220	15 (15/0)
AER	Municipalities	Municipal expenditure	Expenditures	2385	5 (5/0)
AER	PhDPublications	Doctoral publications	Articles	915	6 (4/2)
AER	USSeatBelts	Mandatory seat belt laws in US	Fatalities	765	12 (6/6)
carData	Salaries	Salaries for professors	Salary	397	6 (3/3)
carData	UN98	UN social indicators 1998	tfr	197	13 (12/1)
DAAG	ais	Australian athletes	hg	202	13 (11/2)
DAAG	leafshape17	Leaf shape	Bladelen	286	9 (8/1)
DAAG	nsw74psid1	Labour training evaluation	re78	2344	10 (10/0)

groups, CoDe-Tree overlapping achieves comparatively better results in the case of a reduced number of clusters (2 or 3), but its performance reduces in the case of 5 groups. The presence of noisy variables, instead, does not affect the results inasmuch the distribution in both the cases is roughly the same. In particular, CoDe-Tree overlapping achieves the best ARI the most times followed by SS-RPMM and CoDe-Tree mean. Finally, if none variable is qualitative CoDe-Tree overlapping obtains the best ARI almost in one case over two. Good results are reached also by CoDe-Tree mean and SS-RPMM. Instead, if half of the variables are qualitative, the best method is SS-RPMM.

Summarizing, the results of the simulation study suggest that CoDe-Tree overlapping is a valuable choice in the scattered localization scenarios characterized by the presence of a reduced number of overlapping clusters and qualitative variables. In these scenarios, the possible presence of noisy variables does not degrade the performance of the method.

4.3 Real datasets

We compare the results obtained from the different methods considering 20 real datasets obtained from some R packages. All these datasets include a numeric

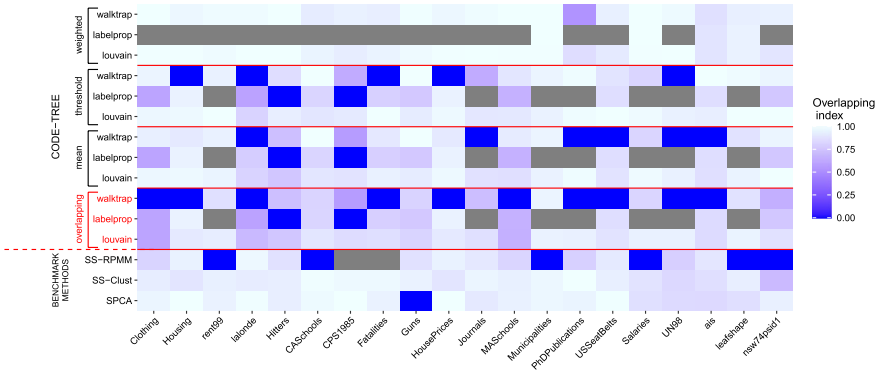

Fig. 7 Heat map of the overlap index values: the colour gradients are relative to columns in order to compare the performance of the 12 different settings of CoDe-Tree and the three semi-supervised methods (the rows) in each dataset (the columns). The color palette ranges from azure (the highest in the column, i.e. the worst) to blue (the lowest in the column, i.e. the best). A grey cell means the method found either a partition composed of one group only or one composed of more than 25 groups: in both cases, the result is considered unsatisfactory and untestable

outcome and a certain number of predictors, either of numeric or categorical type. All the information about the datasets are reported in Table 1. The datasets included in the comparative analysis refer to different types of clustering domains (sales, housing, salaries, students' performance, etc.) although most of them are attributable to socio-economics problems in a broad sense. The size of the considered datasets ranges from those including few observations (as, for example, the Economics Journal Subscription Data composed of 180 cases only) to more large datasets (as, for example, the Munich Rent Data composed of more than 3000 cases), as well as we consider datasets with a reduced number of variables (as, for example, the Municipal Expenditure Data including five predictors only) or others including a relatively large number of predictors (as, for example, the US Traffic Fatalities including 33 predictors). In one case (CPS1985) the number of categorical predictors exceeds that of numerical ones. In three cases (Housing, HousePrices and USSeatBelts) the dataset has the same number of numerical and categorical predictors. In the remaining cases, the number of numerical predictors exceeds that of categorical ones.

4.4 Results

The main results of the comparative analysis are summarized in Figs. 7 and 8. Methods have been compared in terms of the overlapping index introduced in Sect. 3.2 since, consistent with the main goal of semi-supervised clustering with a numerical outcome, we search for the method providing the minimum overlapping among distributions of the outcome in the different groups. Following the previously introduced notation, for each set of T partitions the final partition ℓ_t^* is selected considering the arithmetic mean as a specification of the function $\mathcal{M}$

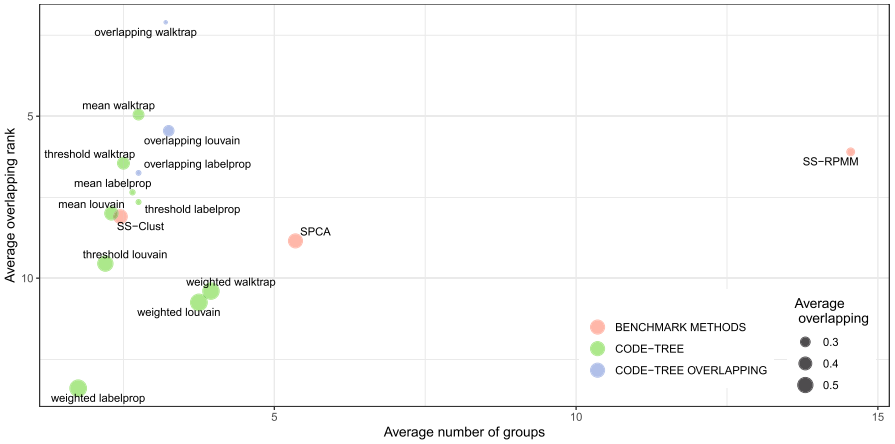

Fig. 8 The relationships between the average number of groups (x-axis) and the average overlapping rank (y-axis) obtained by the 12 different settings of CoDe-Tree and the three semi-supervised methods on the 20 datasets considered. The size of the circles is proportional to the average overlapping, i.e. the smaller size, the better the method (color figure online)

[Eq. (3)]. Unreported results obtained using the weighted mean or the median as possible alternative specifications of $\mathcal{M}$ are very similar to those obtained for the arithmetic mean.

Figure 7 shows a heat map of the values of the overlap index obtained for each method and each dataset. The colour gradients refer to columns in order to compare the performance of the 12 different settings of CoDe-Tree and the three semi-supervised methods (the rows) in each dataset (the columns). The colour palette ranges from azure (the highest in the column, i.e. the worst method) to blue (the lowest in the column, i.e. the best method). Counting the number of blue cells allows us to immediately identify the best-performing method for each dataset.

Overall, we can notice that CoDe-Tree performs better in 13 out of the 20 considered datasets. Interestingly, in 12 cases the best-performing method is the CoDe-Tree using the overlapping criterion introduced in Sect. 3.2 to select the final partition and, within this set of cases, in 10 cases the best-performing method is the CoDe-Tree using the overlapping criterion and the Walktrap community detection algorithm. The latter appears as the most suitable choice for the CoDe-Tree setting as it provides the best performance in 11 of the 13 cases (in the remaining two cases the Label Prop algorithm is the favourite one).

Next, we compare the results obtained from the different methods jointly considering the ability of each method to produce partitions that are as less as possible overlapped with respect to the outcome and the number of groups composing each selected partition. Results are summarized in Fig. 8, which refers to a scatterplot reporting on the horizontal axis the average number of groups provided by each method for the 20 datasets and the average rank (in reverse order) computed with respect to the overlapping index. Each data point, whose size is proportional

to the average overlapping index obtained for the 20 datasets, represents a method and the different colours refer to the implementation of CoDe-Tree with the overlapping criterion (blue), with the other criteria (green) and to the three alternative benchmarking methods (red).

Results reported in Fig. 8 show that CoDe-Tree is the better-ranked method, in particular when it utilizes the overlapping criterion to select the final partition and implements the Walktrap algorithm to detect the community structure. This setting provides on average partitions characterized by a reduced number of groups. As an example, the implementation of CoDe-Tree with the overlapping criterion and Walktrap on the 20 datasets considered in this comparative analysis produces an average ranking of 2.10 (the highest one) and partitions composed of 3.2 groups on average. The reduced number of groups characterizing partitions obtained from this method is consistent with the results obtained in the simulation study. Besides, SS-Clust provides almost the same number of groups (2.45 on average) compared with CoDe-Tree but it is ranked worse on average (the average rank is 8.10). SPCA performs on average even worse compared to SS-Clust (average rank equals 8.85 with partitions sized 5.35 on average), whilst SS-RPMM obtained a better ranking (6.10) but it is very likely that this good rank depends from the very large number of groups (14.55 groups on average) that partitions arising from this method include (increasing the number of groups causes the total overlapping to decrease).

5 Concluding remarks

Traditional cluster analysis methods deal with grouping observations according to their inherent similarities without any supervision. However, the features of each cluster obtained from unsupervised clustering are sometimes hard to distinguish unequivocally. In light of this, semi-supervised clustering attracts lots of interest as it guides the clustering results according to a reference term and improves the clustering performance significantly. Semi-supervised clustering frequently uses the prior information of an outcome variable that drives the clustering process, particularly when no knowledge about the cluster number is available.

In this framework, Network-based Semi Supervised Clustering (NeSSC) has been recently proposed in Frigau et al. (2021) as a novel approach of semi-supervised clustering associated with an outcome variable. It combines an initialization, training and agglomeration phase. In the initialization and training a proximity matrix expressing the pairwise affinity of the instances is estimated by a regression model or a classifier. In the agglomeration phase, the proximity matrix is transformed into a complex network, on which a community detection algorithm searches for an underlying community structure. The final output of NeSSC is a partition of the instances into clusters highly homogeneous in terms of the outcome. Frigau et al. (2021) NeSSC methodology is described in depth by focusing on the basics of each phase, particularly from an algorithmic point of

view and a particular specification of NeSSC called CoDe-Tree is introduced. It uses classification or regression trees as a prediction model or classifier, depending on the type of outcome. Its performance is evaluated on both simulated and real data considering either the case of a categorical outcome or the case of a numerical one.

In this paper, we have focused on the numerical outcome case and we have introduced an alternative criterion for the assessment of the optimal partition of the original data arising from NeSSC together with a penalty term that controls for the number of groups produced by NeSSC. The overlapping criterion is based on the minimum overlapping between density functions concerning the distribution of the outcome in each group. We have implemented this criterion in the CoDe-Tree algorithm whose performance has been evaluated on both artificial data and 20 real datasets by comparing it with three alternative semi-supervised clustering methods and with CoDe-Tree using the original criteria for the search of the optimal partition.

Results of our comparative analysis show that CoDe-Tree based on the minimum overlapping coefficient outperforms other methods (i.e., previous implementations of NeSSC as well as SSClust, SPCA and SS-RPMM) when the clusters' structure is scattered localized rather than cohesive and the percentage of overlap between clusters is limited. Moreover, the CoDe-Tree using the overlapping criterion and implementing the Walktrap community detection algorithm provides in most cases partitions composed of a limited number of groups which are internally homogeneous and not overlapped with respect to the outcome.

Future research will focus on the specification and implementation of additional criteria to improve the performance of NeSSC and CoDe-Tree in the categorical outcome case. Moreover, since the results presented in this paper are obtained on datasets of standard size (up to 5000 observations) another issue that deserves accurate investigation is the assessment of the scalability of NeSSC when dealing with large databases containing millions or even billions of objects, particularly in the Web search scenarios.

A Simulation of data represented in Fig. 1

Data represented in Fig. 1 are generated as follows:

(a) The cohesive example is characterized by 3 groups, each one composed of 168 observations.

  (a1) The outcome variable y is obtained by joining 3 normal distributions of size $n = 168$, with $\mu_1 = 10$, $\mu_2 = 20$ and $\mu_3 = 30$ and standard deviations $\sigma_1 = \sigma_2 = \sigma_3 = 1.2$.
  (a2) The 3 groups are obtained by joining 3 bivariate normal distributions $\mathbf{X}_1, \mathbf{X}_2$ and $\mathbf{X}_3$ of size $n = 168$ with the following parameters:

$\boldsymbol{\mu}_1 = (97, 8)$ and $\boldsymbol{\Sigma}_1 = (57.7, -4.4, -4.4, 66.8)$
$\boldsymbol{\mu}_2 = (38, 39)$ and $\boldsymbol{\Sigma}_2 = (74.0, -3.2, -3.2, 62.2)$
$\boldsymbol{\mu}_3 = (107, 78)$ and $\boldsymbol{\Sigma}_3 = (71.4, -0.3, -0.3, 83.5)$

(a3 For 13 randomly selected pairs of observations belonging to different groups the values of x_1 and x_2 have been swapped.

(b) The scattered localized example is characterized by 6 groups, each one composed of 84 observations.

(b1) The outcome variable y is obtained by joining 3 normal distributions of size $n = 168$, with $\mu_1 = 10$, $\mu_2 = 20$ and $\mu_3 = 30$ and standard deviations $\sigma_1 = \sigma_2 = \sigma_3 = 1.2$.
(b2) The 6 groups are obtained by joining 6 bivariate normal distributions $\mathbf{X}_1, \ldots, \mathbf{X}_6$ of size $n = 84$ with the following parameters:

$\boldsymbol{\mu}_1 = (7, 111)$ and $\boldsymbol{\Sigma}_1 = (153.7, 15.0, 15.0, 210.6)$
$\boldsymbol{\mu}_2 = (147, 134)$ and $\boldsymbol{\Sigma}_2 = (83.2, -7.6, -7.6, 81.5)$
$\boldsymbol{\mu}_3 = (229, 146)$ and $\boldsymbol{\Sigma}_3 = (126.5, -17.6, -17.6, 113.1)$
$\boldsymbol{\mu}_4 = (209, 6)$ and $\boldsymbol{\Sigma}_4 = (199.3, 6.2, 6.2, 256.9)$
$\boldsymbol{\mu}_5 = (103, 45)$ and $\boldsymbol{\Sigma}_5 = (66.7, 7.9, 7.9, 66.9)$
$\boldsymbol{\mu}_6 = (36, 13)$ and $\boldsymbol{\Sigma}_6 = (171.7, -25.3, -25.3, 239.5)$

(b3) For 13 randomly selected pairs of observations belonging to different groups the values of x_1 and x_2 have been swapped.

Data have been simulated using the function `rbinorm` implemented in the R package `VGAM` (Yee 2019).

B Consistency of NeSSC: adjusted rand index (ARI)

See Appendix Table 2.

Table 2 ARI of the comparison between the 231 partitions obtained from simulated datasets and the true ones

	50	100	200	300	500	750	1000	1250	1500	1750	2000
Louvain											
2	0.458	0.253	0.404	0.250	0.287	0.302	0.368	0.480	0.386	0.288	0.341
3	0.225	0.441	0.294	0.216	0.264	0.334	0.343	0.280	0.251	0.256	0.267
4	0.486	0.393	0.226	0.368	0.276	0.298	0.279	0.305	0.230	0.285	0.271
5	0.278	0.260	0.213	0.249	0.175	0.244	0.446	0.058	0.514	0.509	0.448
6	0.276	0.284	0.185	0.000	0.206	0.329	0.286	0.350	0.337	0.359	0.367
7	0.142	0.113	0.130	0.134	0.155	0.352	0.196	0.158	0.145	0.248	0.295
8	0.336	0.366	0.206	0.058	0.272	0.281	0.287	0.263	0.276	0.299	0.298
Label propagation											
2	0.612	0.260	0.404	0.521	0.311	0.302	0.250	0.306	0.390	0.104	0.257
3	0.225	0.000	0.294	0.320	0.000	0.111	0.000	0.000	0.000	0.000	0.170
4	0.486	0.308	0.134	0.324	0.000	0.297	0.275	0.159	0.171	0.128	0.000
5	0.303	0.369	0.000	0.504	0.496	0.387	0.000	0.000	0.000	0.000	0.278
6	0.392	0.000	0.211	0.000	0.183	0.254	0.199	0.162	0.249	0.253	0.237
7	0.000	0.113	0.033	0.134	0.000	0.492	0.276	0.126	0.346	0.119	0.110
8	0.000	0.000	0.306	0.000	0.313	0.250	0.138	0.270	0.349	0.395	0.302
Walktrap											
	50	100	200	300	500	750	1000	1250	1500	1750	2000
2	0.413	0.253	0.426	0.259	0.246	0.302	0.359	0.466	0.282	0.333	0.301
3	0.225	0.266	0.294	0.216	0.264	0.096	0.076	0.046	0.077	0.252	0.254
4	0.486	0.370	0.094	0.324	0.308	0.294	0.194	0.313	0.171	0.279	0.162
5	0.350	0.064	0.249	0.061	0.266	0.188	0.455	0.293	0.484	0.475	0.132
6	0.346	0.160	0.193	0.000	0.069	0.219	0.247	0.291	0.274	0.327	0.016
7	0.142	0.113	0.120	0.134	0.146	0.104	0.073	0.030	0.040	0.116	0.319
8	0.436	0.000	0.306	−0.003	0.256	0.260	0.317	0.001	0.193	0.303	0.293

Each table refers to a community detection algorithm used in NeSSC: the rows and the columns indicate, respectively, the number of covariates and the size. The number of clusters has been set to 4 and the average value of pairwise overlaps to 0.05

Funding Open access funding provided by Università degli Studi di Cagliari within the CRUI-CARE Agreement.

References

Aggarwal CC (2014) Data classification: algorithms and applications. CRC Press

Arbelaitz O, Gurrutxaga I, Muguerza J, Pérez JM, Perona I (2013) An extensive comparative study of cluster validity indices. Pattern Recogn 46(1):243–256
Bair E, Tibshirani R (2004) Semi-supervised methods to predict patient survival from gene expression data. PLoS Biol 2(4):e108
Bair E, Hastie T, Paul D, Tibshirani R (2006) Prediction by supervised principal components. J Am Stat Assoc 101(473):119–137
Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech: Theory Exp 2008(10):P10008
Charrad M, Ghazzali N, Boiteau V, Niknafs A (2014) NbClust: an R package for determining the relevant number of clusters in a data set. J Stat Softw 61(6):1–36
Clemons TE, Bradley EL Jr (2000) A nonparametric measure of the overlapping coefficient. Comput Stat Data Anal 34:51–61
Conversano C, Contu G, Mola F (2019) Online promotion of UNESCO heritage sites in Southern Europe: website information content and managerial implications. Electron J Appl Stat Anal 12(1):108–139
de Jesus Rubio J (2021) Stability analysis of the modified Levenberg–Marquardt algorithm for the artificial neural network training. IEEE Trans Neural Netw Learn Syst 32(8):3510–3524
de Jesus Rubio J, Islas MA, Ochoa G, Cruz DR, Garcia E, Pacheco J (2022) Convergent newton method and neural network for the electric energy usage prediction. Inf Sci 585:89–112
Frigau L, Contu G, Mola F, Conversano C (2021) Network-based semi supervised clustering. Appl Stoch Model Bus Ind 37:182–202
Halkidi M, Vazirgiannis M, Hennig C (2015) Method-independent indices for cluster validation and estimating the number of clusters. In: Hennig C, Meila M, Murtagh F, Rocci R (eds) Handbook of cluster analysis. Chapman and Hall/CRC, pp 595–618
Hubert L, Arabie P (1985) Comparing partitions. J Classif 2(1):193–218
Inman HF, Bradley EL Jr (1989) The overlapping coefficient as a measure of agreement between probability distributions and point estimation of the overlap of two normal densities. Commun Stat: Theory Methods 18:3851–3874
Koestler DC, Marsit CJ, Christensen BC et al (2010) Semi-supervised recursively partitioned mixture models for identifying cancer subtypes. Bioinformatics 26(20):2578–2585
Mehta CR, Patel NR (1983) A network algorithm for performing Fisher's exact test in r× c contingency tables. J Am Stat Assoc 78(382):427–434
Pastore M, Calcagni A (2019) Measuring distribution similarities between samples: a distribution-free overlapping index. Front Psychol 10:1089
Pons P, Latapy M (2005) Computing communities in large networks using random walks. In: Yolum P, Gungor T, Gurgen F, Ozturan C (eds) Computer and information sciences–ISCIS 2005, vol 3733. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg
Porro G, Iacus SM (2009) Random Recursive Partitioning: a matching method for the estimation of the average treatment effect. J Appl Economet 24(1):163–165
R Core Team (2021) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna. https://www.R-project.org
Raghavan UN, Réka A, Kumara S (2007) Near linear time algorithm to detect community structures in large-scale networks. Phys Rev E 76(3):036106
Schmid F, Schmidt A (2006) Nonparametric estimation of the coefficient of overlapping—theory and empirical application. Comput Stat Data Anal 50:1583–1596
Tukey JW (1949) Comparing individual means in the analysis of variance. Biometrics 5(2):99–114
Van Mechelen I, Boulesteix AL, Dangl R, Dean N, Guyon I, Hennig C, Leisch F, Steinley D (2018) Benchmarking in cluster analysis: a white paper. arXiv preprint arXiv:1809.10496
Yee TW (2019) VGAM: vector generalized linear and additive models. R package version 1.1-2. https://CRAN.R-project.org/package=VGAM
Zeileis A, Kleiber C, Krämer W, Hornik K (2003) Testing and dating of structural changes in practice. Comput Stat Data Anal 44:109–123