Semi-Supervised Clustering via Information-Theoretic Markov Chain Aggregation

SOPHIE STEGER, Signal Processing and Speech Communication Laboratory, Graz University of Technology, Austria
BERNHARD C. GEIGER, Know-Center GmbH, Austria
MAREK ŚMIEJA, Faculty of Mathematics and Computer Science, Jagiellonian University, Poland

We connect the problem of semi-supervised clustering to constrained Markov aggregation, i.e., the task of partitioning the state space of a Markov chain. We achieve this connection by considering every data point in the dataset as an element of the Markov chain's state space, by defining the transition probabilities between states via similarities between corresponding data points, and by incorporating semi-supervision information as hard constraints in a Hartigan-style algorithm. The introduced Constrained Markov Clustering (CoMaC) is an extension of a recent information-theoretic framework for (unsupervised) Markov aggregation to the semi-supervised case. Instantiating CoMaC for certain parameter settings further generalizes two previous information-theoretic objectives for unsupervised clustering. Our results indicate that CoMaC is competitive with the state-of-the-art.

CCS Concepts: • **Computing methodologies** → **Semi-supervised learning settings**; • **Mathematics of computing** → *Information theory*; • **Information systems** → **Clustering**;

Additional Key Words and Phrases: semi-supervised clustering, Markov aggregation

1 INTRODUCTION

A popular approach to clustering, especially if only pairwise similarities between data points are available, is to view the problem from the perspective of random walks. From this perspective, each data point is represented by a state in the state space of a Markov chain, whose transition probabilities are determined by the pairwise similarities between the corresponding data points. The clustering problem can then be solved via aggregating the state space of the thus defined Markov chain. In this paper, we focus on the unifying framework from [2], which captures previous information-theoretic approaches to Markov aggregation [3, 5, 15, 17] and, if instantiated appropriately, clustering [1, 14] as special cases (see Section 2).

Although clustering via Markov aggregation has a solid theoretical basis, allows for creating non-linear decision boundaries, and was shown to achieve competitive performance [1], it appears to be highly sensitive on a careful selection of hyperparameters or optimization procedures. In random walk-based clustering, even representing a dataset as a Markov chain requires appropriately selecting hyperparameters, cf. (3) below. Tuning these hyperparameters to individual datasets is cumbersome and severely limits the practical applicability of the respective clustering method. Moreover, there are no clear rules and objective evaluation measures for their selection because of the unsupervised nature of clustering.

In this paper, we propose Constrained Markov Clustering (CoMaC), the extension of clustering via Markov aggregation [2] to the semi-supervised setting, where the side information is given in the form of partition-level information

Authors' addresses: Sophie Steger, Signal Processing and Speech Communication Laboratory, Graz University of Technology, Inffeldgasse 16c, Graz, Austria, 8010, sophie.steger@student.tugraz.at; Bernhard C. Geiger, Know-Center GmbH, Inffeldgasse 13, Graz, Austria, 8010, geiger@ieee.org; Marek Śmieja, Faculty of Mathematics and Computer Science, Jagiellonian University, Łojasiewicza 6, Krakow, Poland, marek.smieja@uj.edu.pl.

2022. Manuscript submitted to ACM

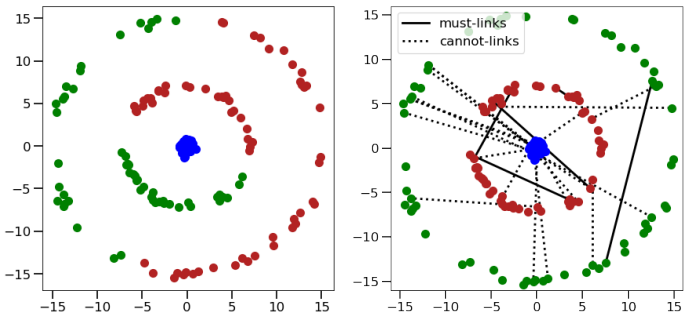


Fig. 1. Circles dataset with three concentric circles consisting of 60 data points each. Result for (unsupervised) Markov aggregation clustering (a) and the proposed semi-supervised CoMaC with 30 constraints (b). While also the unsupervised approach can learn non-linear boundaries between clusters, the addition of pairwise constraints helps avoiding bad local optima.

[8] (some data points are labeled by their cluster index) or pairwise constraints [16] (for some pairs of data points we know whether they belong to the same or to different clusters, see Figure 1).

Experimental results confirm that the proposed adaptions to the Hartigan-style clustering algorithm of [2] achieve performance on common benchmarks that is competitive with the state-of-the-art in semi-supervised clustering. Furthermore, there are indications that introducing side information makes the algorithm more robust to the selection of some of its hyperparameters. Reducing the sensitivity to hyperparameters is important for (semi-supervised) clustering because the limited available information about class labels typically precludes performing proper validation.

2 PROBLEM STATEMENT AND RELATED WORK

We address the (semi-supervised) clustering problem via transforming it into a Markov chain aggregation problem. In this section, we thus give a statement of these problems and review the relevant literature to prepare the stage for our method in Section 3.

Clustering refers to the task of grouping the elements of a dataset $\mathcal{X} = (x_1, \ldots, x_N)$, $x_i \in \mathbb{R}^n$, such that the data points within each group have a higher similarity with each other than with those of different groups, where similarity has to be defined appropriately. If this grouping is deterministic, then there exists a clustering function $g\colon \mathcal{X} \to \{1, \ldots, K\}$ that maps each element of $\mathcal{X}$ to the index of one of the K clusters. Clustering is successful if the candidate clustering function g is close (in a well-defined sense) to the function $g^\bullet\colon \mathcal{X} \to \{1, \ldots, K^\bullet\}$ determining the true partition.

Semi-supervised clustering simplifies the task by providing additional information in one of two flavors: First, partition-level side information refers to a subset $\mathcal{X}'$ of $\mathcal{X}$ for which the true cluster indices are known, i.e., $\{(x, g^\bullet(x)) \mid x \in \mathcal{X}' \subset \mathcal{X}\}$. Such partition-level side information was proposed for k-means [8], fuzzy c-means [9, 10], Gaussian Mixture Models (GMMs) [7], or cross-entropy clustering with information bottleneck regularization (CEC-IB) [13]. The second option are pairwise constraints, which indicate which pairs of data points of $\mathcal{X}$ must or must not be put in the same cluster; this setting is often referred to as constrained clustering. Pairwise constraints are given as

$$\mathcal{M} = \{ (x, x') \mid g^\bullet(x) = g^\bullet(x') \} \tag{1a}$$

$$\mathcal{N} = \{ (x, x') \mid g^\bullet(x) \neq g^\bullet(x') \} \tag{1b}$$

for a (small) subset of pairs $(\mathcal{M} \cup \mathcal{N}) \subset \mathcal{X}^2$. Pairwise constraints have been utilized for discriminative clustering with graph regularization [18], GMMs [12], or spectral clustering [11].

A special instance of clustering is the problem of Markov aggregation, i.e., the problem of clustering states of a Markov chain. Mathematically, if the stochastic process $X = (X_1, X_2, \dots)$ is an aperiodic, irreducible, and stationary Markov chain (see [6] for terminology) with finite state space $\mathcal{Z}$, then the task is to find an aggregation function $h\colon \mathcal{Z} \to \{1, \dots, K\}$ such that the aggregated process $Y = (h(X_1), h(X_2), \dots)$ satisfies certain properties. Several information-theoretic cost functions have been proposed for this problem. For example, the authors of [3, 15] aimed at a maximally predictable process Y by maximizing the mutual information $I(Y_1; Y_2)$, while the authors of [5] selected h such that Y is as Markov as possible as measured via the Kullback-Leibler divergence rate. Recently, an information-theoretic framework for Markov aggregation has been proposed, which aims at finding a minimizer of [2]

$$C_\beta(X, h) = (1 - 2\beta)\,(H(Y_2|Y_1) - H(Y_2|X_1)) - \beta I(Y_1; Y_2) \tag{2}$$

where H denotes the entropy, where the minimum is taken over all functions $h : \mathcal{Z} \to \{1, \dots, K\}$, and where the first two and the third terms represent the operational goals of preserving the Markov property and the temporal dependence structure of X, respectively. It can be shown that this framework covers the cost functions of [5], [3, 15], and [17] as special cases for β being equal to 0, 0.5, and 1, respectively.

Markov aggregation is thus presented as a clustering problem. Conversely, by identifying each element of a dataset $\mathcal{X}$ with a state of a Markov chain and by parameterizing the transition probability between states via the similarity of corresponding data points, the clustering problem can be formulated as a Markov aggregation problem. For example, if $d\colon \mathcal{X}^2 \to [0, \infty)$ is a measure of dissimilarity between data points, then $\mathcal{X}$ can be clustered via aggregating the Markov chain $X = (X_1, X_2, \dots)$ with state space $\mathcal{X}$ and transition probability matrix $\mathbb{P} = [P_{i,j}]$,

$$P_{i,j} \propto \mathrm{e}^{-\frac{d(x_i, x_j)}{\sigma^2}} \tag{3}$$

where σ^2 is a scaling factor. The candidate aggregation function h obtained by solving (2) can then be interpreted as the candidate clustering function g. Indeed, the authors of [1] proposed maximizing $I(Y_1; Y_2)$, where $d(x_i, x_j)$ is 0 if x_i and x_j are k-nearest neighbors of each other and ∞ otherwise. Furthermore, in [14] d was chosen as the Euclidean distance, σ^2 as the k-nearest neighbor distance, and the authors proposed to minimize

$$I(Y_1; X_1) - \beta I(Y_1; X_{T+1}) \tag{4}$$

where T is selected such that the Markov chain X has relaxed to a meta-stable state. These two approaches of [1] and [14] (for $T = 1$ and symmetric dissimilarity measures d) thus correspond to solving (2) for $\beta = 0.5$ and, for, $\beta = 1$, respectively.[1] The main differences between [1, 14] and minimizing (2) rely on the definition of $\mathbb{P}$ in (3) and, potentially, the relaxation time T proposed in [14].

Preliminary experiments suggest that good clustering performance can only be achieved by the methods in [1, 14] if the parameters σ^2 (or its proxy k) and T are carefully set. Our research hypothesis, which will be confirmed in this paper, is that the provision of pairwise constraints makes the proposed methods less sensitive to these parameter settings.

[1] For symmetric dissimilarity measures, the transition probability matrix resulting from (3) is reversible, i.e., $I(Y_1; X_2) = I(Y_2; X_1)$, cf. [2, Sec. IV.C].

3 SEMI-SUPERVISED CLUSTERING VIA MARKOV AGGREGATION

In this section, we introduce our approach to semi-supervised clustering based on Markov aggregation (CoMaC). To this end, we utilize the Markov aggregation problem (2) proposed in [2] and apply it to a Markov chain X with a transition probability matrix $\mathbb{P}$ depending on the clustering dataset $\mathcal{X}$ via (3). Specifically, we choose d to be the squared Euclidean distance and σ_k^2 as the average squared Euclidean distance between the data point and its k nearest neighbors (averaged over all data points), i.e.,

$$P_{i,j} \propto \mathrm{e}^{-\frac{\|x_i - x_j\|_2^2}{\sigma_k}}. \tag{5}$$

The authors of [2] proposed a Hartigan-style algorithm for solving the optimization problem in (2) for a deterministic clustering function $g\colon \mathcal{X} \to \{1, \dots, K\}$. Starting from an initial clustering of $\mathcal{X}$ into K clusters, each data point x is mapped to every aggregate state $y \in \{1, \dots, K\}$ and the cost function is evaluated. The data point is then assigned to the aggregate state that minimizes the cost function. Since this algorithm tends to get stuck in poor local optima for small values of β, an additional annealing procedure was introduced in [2] that provides clustering functions g obtained for higher values of β as initial clusterings for lower values of β.

In this work, the algorithm of [2] is extended in order to accept pairwise constraints $\mathcal{M}$ and $\mathcal{N}$ as given in (1). Since partition-level side information can easily be converted to pairwise constraints (but not vice-versa), the resulting algorithm can handle both types of side information. Below we describe the initialization, iteration, and annealing procedures of CoMaC.

Initialization. First, the candidate partition function g is initialized such that all pairwise constraints are satisfied. This is done via solving a graph coloring problem, where no adjacent vertices of a graph are allowed to be of the same color. In our procedure, each vertex of this graph either corresponds to an individual data point not involved in any must-link constraint, or to a set of data points that are connected via must-link constraints, while each edge of this graph corresponds to a cannot-link constraint. The initial coloring of the graph is performed by a greedy algorithm (see Algorithm 1) where each vertex is assigned the first color available in sequence. To avoid the algorithm getting stuck in bad local minima, vertices with no cannot-link constraints are assigned a random color.

Algorithm 1 Greedy coloring algorithm.

```
1:  function g = greedyColoring(must-link constraints M, cannot-link constraints N, K)
2:      for all elements x ∈ X do
3:          M_x = funcMust(M, N, x)
4:          N_x = funcCannot(M, N, x)
5:          if N_x is empty then
6:              g(M_x) ← random value out of K colors
7:          else
8:              g(M_x) ← firstColorAvailable(N_x)
9:          end if
10:     end for
11: end function
```

Iteration. Once the initial partition function is defined, the sequential algorithm minimizes the cost function in (2) iteratively. Cannot-link constraints are incorporated by restricting the possible states of the aggregation function in line 14 of Algorithm 2. Data points connected by must-link constraints are assigned to an aggregate state simultaneously.

Erroneous or noisy pairwise constraints can lead to the case where a data point cannot be assigned to any aggregate state. Then, the aggregate state with the least occurrences of cannot-link constraints is selected. Algorithm 2 contains the sequential algorithm in [2] as special case if $\mathcal{M} = \mathcal{N} = \emptyset$ and if the propagation functions are such that $\mathcal{M}_x = \{x\}$ and $\mathcal{N}_x = \emptyset$.

Algorithm 2 Sequential Generalized Information-Theoretic Markov Aggregation with Pairwise Constraints.

function g =CoMaC-seq($\mathbb{P}$, β, K, #iter$_{\max}$, *optional*: initial aggregation function g_{init}, must-link constraints $\mathcal{M}$, cannot-link constraints $\mathcal{N}$)
  if g_{init} is empty **then** ▷ Initialization
    g = greedyColoring($\mathcal{M}$, $\mathcal{N}$, K)
  else
    $g \leftarrow g_{\text{init}}$
  end if
  #iter $\leftarrow 0$
  while #iter < #iter$_{\max}$ **do** ▷ Main Loop
    for all elements $x \in \mathcal{X}$ **do** ▷ Optimizing g
      $\mathcal{M}_x$ = funcMust($\mathcal{M}, \mathcal{N}, x$)
      $\mathcal{N}_x$ = funcCannot($\mathcal{M}, \mathcal{N}, x$)
      $\mathcal{Y}_{pos} = \{1, \ldots, K\} \setminus g(\mathcal{N}_x)$ ▷ Possible aggregate states
      if $\mathcal{Y}_{pos}$ is empty **then**
        $\mathcal{Y}_{pos} = \arg\min_y |y \in g(\mathcal{N}_x)|$ ▷ Select state with the least occurrences
      end if
      for all possible aggregate states $y \in \mathcal{Y}_{pos}$ **do**
        $g_y(x') = \begin{cases} g(x'), & x' \notin \mathcal{M}_x \\ y, & x' \in \mathcal{M}_x \end{cases}$
        $C_{g_y} = C_\beta(X, g_y)$
      end for
      $g = \arg\min_{g_y} C_{g_y}$ ▷ (break ties)
    end for
    #iter $\leftarrow$ #iter + 1
  end while
end function

Annealing. The annealing procedure for β was introduced in [2] to avoid the sequential algorithm getting stuck in poor local minima for small values of β. Annealing is initialized with $\beta = 1$, and the resulting aggregation functions are iteratively used as initialization for the sequential algorithm with reduced β. The annealing procedure itself was not adapted and is shown in Algorithm 3.

Constraint Propagation. Usually, the sets $\mathcal{M}$ and $\mathcal{N}$ of pairwise constraints are not exhaustive. For example, if $(x, x') \in \mathcal{M}$ and $(x, x'') \in \mathcal{M}$, then also x' and x'' must link, even though such relation does not appear explicitly in the side information. If the sets of pairwise constraints are exhaustive, vertices connected by must-link constraints form graph cliques such that every pair of vertices within the clique is connected by a must-link constraint. In the case of sparse pairwise constraints, must-link constraints do not necessarily form cliques. To account for this, constraints in $\mathcal{M}$ can be *propagated*. However, propagating these constraints is a non-trivial problem, especially if the pairwise constraints are conflicting (e.g., due to labeling errors). In this work we assume non-contradictory constraints. We look

Algorithm 3 β-Annealing Information-Theoretic Markov Aggregation with Pairwise Constraints.

function g = CoMaC-ann($\mathbb{P}$, β_{target}, K, #iter$_{\max}$, Δ, *optional*: must-link constraints $\mathcal{M}$, cannot-link constraints $\mathcal{N}$)
  g = CoMaC-seq($\mathbb{P}$, 1, K, #iter$_{\max}$, *optional*: $\mathcal{M}$, $\mathcal{N}$) ▷ Inizialization
  while $\beta > \beta_{\text{target}}$ **do**
    $\beta \leftarrow \max\{\beta - \Delta, \beta_{\text{target}}\}$
    g = CoMaC-seq($\mathbb{P}$, β, K, #iter$_{\max}$, g, *optional*: $\mathcal{M}$, $\mathcal{N}$)
  end while
end function

for all connected components in the graph given by $\mathcal{M}$ using a depth-first search (DFS) algorithm. Initially, all vertices of the graph are marked as unvisited. Starting at an arbitrary vertex, we determine all connected vertices and mark them as visited. Iteratively, the procedure is repeated for those vertices until we reach a vertex that has been visited before. Using these connected components, the set $\mathcal{M}$ can be extended, such that it describes a graph consisting of independent cliques only. In Algorithm 1, this is done in the function funcMust. The function returns all x' that must be in the same clusters as x given by the DFS algorithm on $\mathcal{M}$. Additionally, the function funcCannot considers the case where two data points from different cliques are connected by a cannot-link constraint. Then, all elements of the two respective cliques are connected by cannot-link constraints, and thus are not allowed to be in the same cluster. E.g., if $(x, x') \in \mathcal{M}$ and $(x, x'') \in \mathcal{N}$, then also x' and x'' should not link, despite (x', x'') not being a labeled cannot-link constraint. Our function funcCannot thus returns $\mathcal{N}_x = \{x', x''\}$.

4 EXPERIMENTS

In this section, we experimentally evaluate the performance of CoMaC[2]. First, we verify the research hypothesis that the introduction of pairwise constraints makes Markov aggregation-based clustering less sensitive to the choice of hyperparameters. Next, we compare its performance with the state-of-the-art semi-supervised clustering techniques following the experimental setup of [13]. Finally, we demonstrate and discuss the current limitations of CoMaC.

We measure the accuracy of the obtained clusterings with the Normalized Mutual Information (NMI). It is defined by the mutual information between the true partition $g^\bullet$ and estimated partition g normalized by sum of the entropy of the partitions, i.e.,

$$\text{NMI}(g^\bullet(U), g(U)) = \frac{2I(g^\bullet(U); g(U))}{H(g^\bullet(U)) + H(g(U))} \tag{6}$$

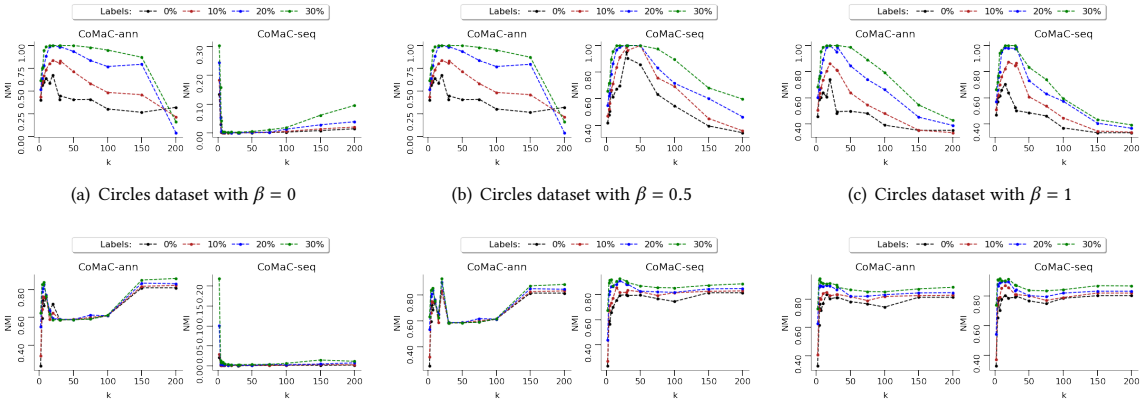
where U is a random variable uniformly distributed on the elements of $\mathcal{X}$. Thus, the NMI returns the similarity between the estimated and true partition. It has a lower bound of 0 (independent partitions) and an upper bound of 1 (for identical partitions). To avoid the impact of random initialization on the results, we average NMI values over 10 randomized runs.

4.1 Sensitivity Analysis of CoMaC

In this part, we investigate the effect of the hyperparameters selection on the clustering results produced by CoMaC. To be consistent with [13], we generate pairwise constraints from randomly sampled partition-level side information.

Influence of parameter k. Both for the sequential and annealing algorithm (referred to as CoMaC-ann and CoMaC-seq) one can observe an influence of the hyperparameter k on the clustering accuracy. Ideally, k is chosen such that the transition probability matrix is nearly completely decomposable, which strongly depends on the chosen dataset. In

[2]The data and code used for these experiments is publicly available at https://github.com/stegsoph/Constrained-Markov-Clustering



(a) Circles dataset with $\beta = 0$ (b) Circles dataset with $\beta = 0.5$ (c) Circles dataset with $\beta = 1$

(d) Iris dataset with $\beta = 0$ (e) Iris dataset with $\beta = 0.5$ (f) Iris dataset with $\beta = 1$

Fig. 2. Dependence of the clustering accuracy on the parameter k for the circles dataset (a–c) and the Iris dataset (d–f).

this subsection, we analyse the influence of k on the three circles dataset shown in Figure 1. Data points are placed uniformly distributed at radii $\{0.5, 7, 15\}$ and corrupted by spherical Gaussian noise with standard deviation of 0.3.

We analyse the performance of CoMaC-ann and CoMaC-seq with $\beta = \{0,\ 0.5,\ 1\}$ for semi-supervised clustering where 0%, 10%, 20%, 30% of data points are labeled while k is varied (see Figure 2, top). On the one hand, the experiment shows that CoMaC-ann performs more robustly than CoMaC-seq w.r.t. the hyperparameter β. On the other hand, we observe that the additional side information makes CoMaC more robust to the selection of the hyperparameter k, at least for this dataset. Clustering accuracy degrades for increasing values of k, and the degradation is less severe the more data points are labeled.

The same experiment is repeated for the Iris dataset (see Figure 2, bottom). As it can be seen, the NMI as a function of k shows less variations than for the three concentric circles. As expected, the optimal value of k depends on the dataset. However, for $k < 50$, the performance is quite stable for both datasets and all considered levels of side information. Thus, for all subsequent experiments we set $k = 20$ rather than optimize it for each dataset.

Influence of parameter β. We next analyse the influence of the parameter β for a constant setting of $k = 20$. The results of the experiment for the unsupervised case and for a semi-supervised case where 20% of the data points are labeled and used to generate the pairwise constraints are shown in Figure 3.

The sequential algorithm performs particularly badly for small β values as it is prone to getting stuck in bad local minima. This parallels the behavior of the Markov aggregation method proposed in [2]. Since these bad minima for small values of β cannot be escaped by introducing additional side-information, the annealing scheme was proposed, reducing β iteratively. The annealing algorithm was introduced to ensure a convergence to a good local minimum close to the global minimum. When varying the parameter β in smaller step sizes, we can observe that the annealing algorithm returns stable results for the accuracy (see Figure 3). For most datasets the additional side-information adds more stability and increased accuracy to the annealing algorithm with respect to variations of β. For all following experiments, we choose $\beta = 0.5$, which results in the same cost function as in [1].

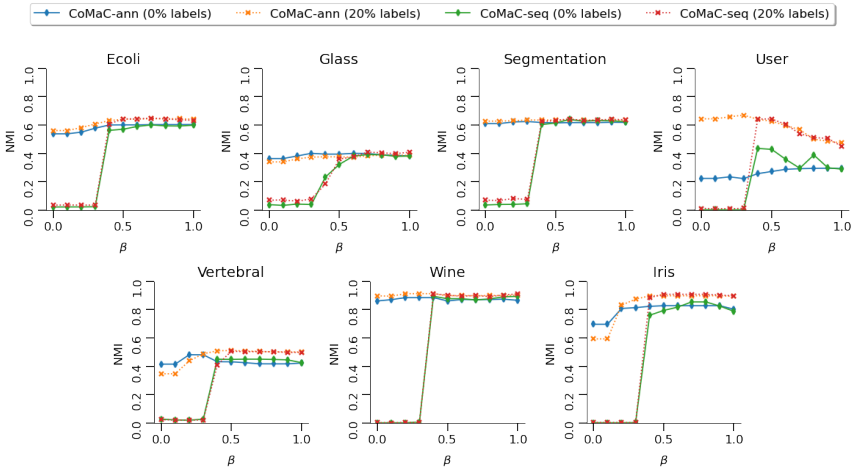


Fig. 3. Influence of the parameter β on the clustering accuracy in the unsupervised and semi-supervised setting (fraction of labeled data = 20%).

Table 1. Description of the datasets used in our experiments. For datasets with +, principal components analysis was used to reduce dimensionality. For Wine dataset we normalize the attributes for CoMaC.

dataset	# Instances	# Features	# Classes
Ecoli$^+$	327	5	5
Glass	214	9	6
Iris	150	4	3
Segmentation$^+$	210	5	7
User Modeling	403	5	4
Vertebral	310	6	3
Wine*	178	13	3

4.2 Evaluation

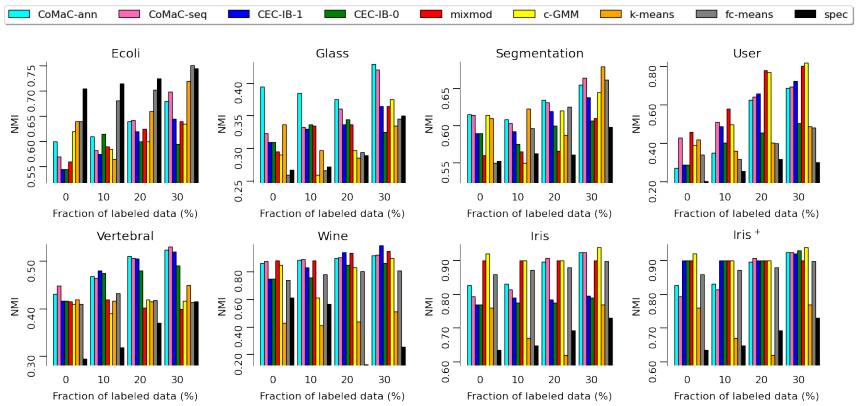
Next, we compare CoMaC with the state-of-the-art semi-supervised clustering techniques on several UCI datasets [4] as described in Table 1. For a fair comparison, we follow the experimental setup in [13] and directly use the clustering results of comparative methods reported there. We assume that the number of clusters is known.

Experimental setup. We consider CoMaC-ann and CoMaC-seq with $k = 20$ and $\beta = 0.5$ throughout all experiments. The latter parameter setting corresponds to the clustering method proposed in [1], albeit for a different transition probability matrix $\mathbb{P}$.

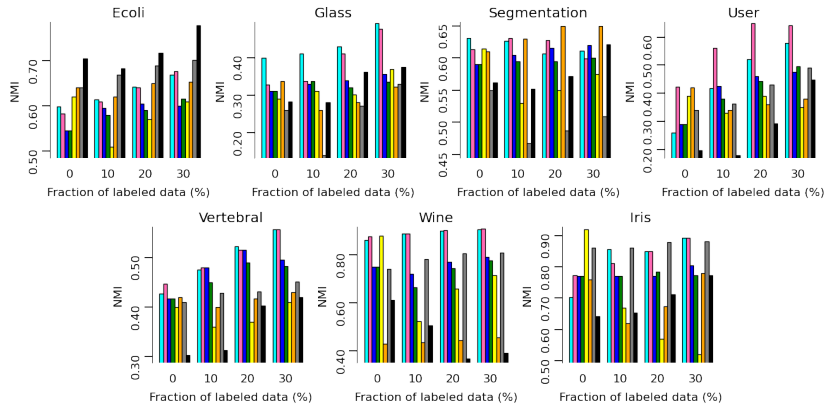
The following baselines are considered (see [13] for a description of hyperparameters selection):

- CEC-IB [13]: model-based clustering based on cross-entropy and information bottleneck using partition-level side information. We considered two values of a hyperparameter, denoted as CEC-IB$_1$ and CEC-IB$_0$. This is the only method that is initialized with twice the correct number of clusters since it identifies the optimal number automatically.
- mixmod: GMM with a partition-level side information implemented in R package Rmixmod [7].
- c-GMM [12]: GMM that uses pairwise constraints.

- k-means [8]: the extension of k-means that supports partition-level side information.
- fc-means [9, 10]: a fuzzy c-means using partition-level side information.
- spec [11]: a spectral clustering algorithm that incorporates pairwise constraints



(a) The side information covers the elements of all classes



(b) The side information covers the elements of only two classes

Fig. 4. Normalized mutual information computed on UCI datasets with noise-free side information from all classes (a) and two classes only (b). The parameters of CoMaC are set to $\beta = 0.5$ and $k = 20$.

Since some of the comparison methods reported in [13] only accept partition-level side information, those methods are provided with the ground truth clusters $g^\bullet(x)$ for a subset $\mathcal{X}'$ of data points. More precisely, the partition-level side information is generated by choosing 0%, 10%, 20%, and 30% of the data points and labeling them according to their class. This partition-level side information is subsequently converted to pairwise constraints and incorporated to the remaining methods. To allow for a fair comparison, the constraint sets $\mathcal{M}$ and $\mathcal{N}$ are exhaustive, i.e., they contain all pairwise constraints that are implied by partition-level side information. Specifically, if $|\mathcal{X}'| = m$, then $|\mathcal{M}| + |\mathcal{N}| = m(m-1)/2$.

Clustering with side information from all classes. First, we consider a typical case, where the partition-level side information covers elements of all classes. Figure 4(a) shows the accuracy of the clustering results for each algorithm and dataset for different fractions of labeled data points. Overall, we can observe that CoMaC clearly benefits from labeled data, at least for $k = 20$ and $\beta = 0.5$, as NMI increases with increasing amounts of labeled data points.

The improvement of CoMaC performance due to side information in comparison to the other techniques is most notable on the Iris dataset. Only 20% of labeled data points noticeably improve the accuracy of CoMaC while the other techniques do not benefit as much from additional side information. CoMaC furthermore achieves superior performance on the Glass, Segmentation, Vertebral and Wine datasets. Both k-means and spec are sensitive to the scale of attributes, which may partially explain why these methods perform worse on the Wine dataset (CoMaC was run on the normalized Wine dataset). Interestingly, on the Vertebral dataset, all algorithms perform equally well in the unsupervised case. However, when incorporating labeled data, both CoMaC and CEC-IB outperform all other methods.

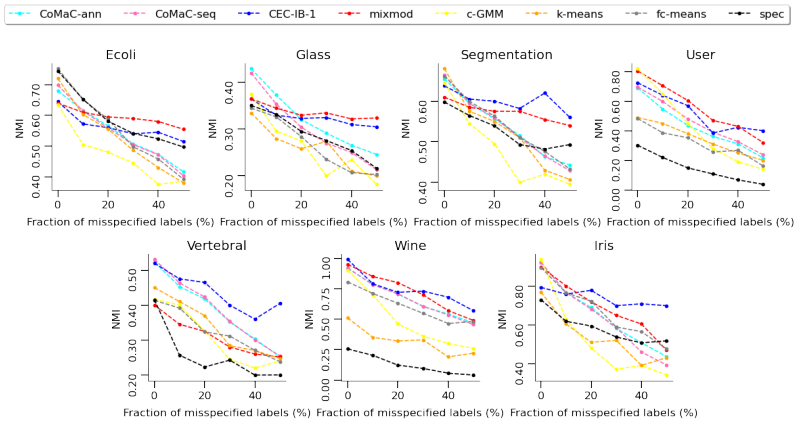


Fig. 5. Normalized mutual information computed on UCI datasets with noisy partition-level side information from all classes. The parameters of CoMaC are set to $\beta = 0.5$ and $k = 20$.

Clustering with side information from a subset of classes. Next, we investigate the case where the side information does not cover all classes. Now, a certain percentage (0%, 10%, 20%, 30%) of data points from only two classes is selected and used for labeling. Following the work of [13], the two classes must cover at least 30% of the total data. The goal is to determine the ability of our clustering algorithm to correctly identify all classes, although it is given information about only two of them.

The results reported in Figure 4(b) show that CoMaC is robust against missing labels from other classes. The advantage of CoMaC is especially evident in the case of Vertebral, Wine and User dataset, but it also performs well on Glass and Segmentation datasets. Interestingly, CoMaC-seq returns significantly better scores than CoMaC-ann on the User data.

4.3 Limitations

In this last part, we investigate current limitations of CoMaC. Specifically, we examine the effects of erroneous side information and of how pairwise constraints are propagated before being utilized in Algorithm 2. Furthermore, we

discuss an inherent shortcoming of the greedy coloring initialization in Algorithm 1 concerning cannot-link constraints. The insights of this section thus suggest interesting avenues for future research.

Erroneous partition-level side information. We first examine the robustness of our CoMaC algorithm against erroneous partition-level side information. In this case, 30% of the data points are labeled, for which a certain percentage (0%, 10%, 20%, 30%, 40%, 50%) of labels are substituted with random incorrect labels. Note that the pairwise constraints generated from the noisy partition-level side information are not contradictory, but only are inconsistent with the ground-truth.

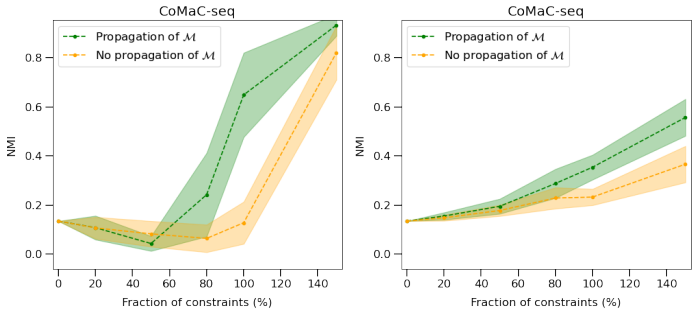
When compared to other clustering techniques (see Figure 5), the performance of CoMaC is on par with c-GMM, k-means, fc-means and spec, which are all highly sensitive to noise: For four datasets, the semi-supervised setting with only 10% of erroneous labels is even outperformed by the unsupervised CoMaC algorithm. Besides CoMaC, c-GMM and spec incorporate side information using pairwise constraints, and both exhibit a steeply falling accuracy as a function of the level of noisy side information.

If not generated from partition-level side information, erroneous pairwise constraints can be conflicting. Consolidating these conflicts either using separate algorithms or via specifically designed cost functions may lead to a degradation in accuracy that is less substantial than for erroneous partition-level side information. Verifying this conjecture will require further experiments and is out of the scope of the current work.

Propagation of pairwise constraints. As discussed in Section 3, the pairwise constraint sets $\mathcal{M}$ and $\mathcal{N}$ may not be exhaustive. When dealing with noise-free pairwise constraints, we may encounter the following two cases: if $(x, x') \in \mathcal{M}$ and $(x, x'') \in \mathcal{N}$, then also x' and x'' should not link; and if $(x, x') \in \mathcal{M}$ and $(x, x'') \in \mathcal{M}$, then also x' and x'' should link. The first case is accounted for in the function FUNCCANNOT$(\mathcal{M}, \mathcal{N}, x)$. To study the influence of the propagation of must-link constraints, we compare the propagation function FUNCMUST$(\mathcal{M}, \mathcal{N}, x)$ as described in Section 3 with a more primitive version that only returns as $\mathcal{M}_x$ all x' that are linked with x explicitly in $\mathcal{M}$.

We performed this experiment on four UCI datasets (Ionosphere, Iris, User, Wine) for a given fraction of pairwise constraints $(|\mathcal{M}| + |\mathcal{N}|)/|X| = \{0\%, 20\%, 50\%, 100\%, 150\%\}$. The influence on accuracy due to the propagation of must-link constraints depends on the datasets and could be observed most prominently on the Ionosphere dataset (see Figure 6(a)). For a small number of pairwise constraints, propagation has no noticeable effect as the randomly sampled must-link constraints rarely overlap. Only after a certain fraction of constraints is added, accuracy starts to increase while simultaneously the influence of propagation on accuracy is visible. However, these effects are highly dependent on the individual datasets. For Iris, User, and Wine, propagation of must-link constraints did not increase accuracy significantly.

Negative influence of cannot-link constraints. Interestingly, in Figure 6(a) we could observe an initial drop in accuracy after including 50% of constraints. There, the accuracy of the clustering on the Ionosphere dataset degrades towards zero, indicating random partitioning. When initializing the graph via a greedy coloring algorithm, vertices connected by cannot-link constraints are assigned to the first cluster available. As the Ionosphere dataset consist of two classes only, those constrained data pairs are forced to stay in place during the rest of the algorithm as there are no further free classes available. Excluding all cannot-link constraints avoids the initial drop in accuracy and leads to a monotonically increasing accuracy (see Figure 6(b)). Additionally, we could observe that the number of constraints where the propagation of must-link constraints starts to improve clustering accuracy seems to corresponds to the point where cannot-link constraints stop harming the performance. However, at least for Ionosphere and a larger amount of constraints, the performance without cannot-link constraints was generally lower than by considering them.



(a) With cannot-link constraints (b) Without cannot-link constraints

Fig. 6. Influence of propagation of must-link constraints on the accuracy for the Ionosphere dataset (351 samples, 34 features, 2 classes)

Again, these effects depended on the individual datasets. To gain further into the reason behind these effects and possible mitigation by adapting the algorithms, further investigation is required.

5 DISCUSSION AND CONCLUSION

In this work we have extended the unsupervised optimization algorithm for clustering via Markov aggregation of [2] to accept pairwise constraints. We showed that that the use of pairwise constraints successfully lowers the algorithm's sensitivity to hyperparameter settings. Extensive experiments using pairwise constraints from partition-level side information confirmed that our algorithm can learn non-linear decision boundaries between clusters and competes with state-of-the-art semi-supervised clustering techniques. Especially when provided with side information covering only a subset of all classes, our method can achieve better results than existing techniques.

Finally, a number of potential limitations require further work beyond the scope of this paper. First, our method reacts sensitively to noisy constraints generated from erroneous partition-level side information. Second, presented non-exhaustive pairwise constraint sets, propagating the constraints over the whole dataset can improve performance. However, this task becomes non-trivial when pairwise constraints are inconsistent due to noise. Finally, when dealing with sparse pairwise constraints, we could observe that the greedy graph initialization hampers clustering performance for low to moderate numbers of cannot-link constraints. Future work shall determine possible explanations for these limiting factors and subsequently adapt the algorithm to mitigate said problems.

ACKNOWLEDGMENTS

The work of Sophie Steger has been supported by iDev40. The iDev40 project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 783163. The JU receives support from the European Union's Horizon 2020 research and innovation programme. It is co-funded by the consortium members, grants from Austria, Germany, Belgium, Italy, Spain and Romania.

The work of Bernhard C. Geiger has been supported by the HiDALGO project and has been funded by the European Commission's ICT activity of the H2020 Programme under grant agreement number 824115.

The Know-Center is funded within the Austrian COMET Program - Competence Centers for Excellent Technologies - under the auspices of the Austrian Federal Ministry of Climate Action, Environment, Energy, Mobility, Innovation

and Technology, the Austrian Federal Ministry of Digital and Economic Affairs, and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

REFERENCES

[1] A. Alush, A. Friedman, and J. Goldberger. 2016. Pairwise clustering based on the mutual-information criterion. *Neurocomputing* 182 (2016), 284–293. https://doi.org/10.1016/j.neucom.2015.12.025

[2] R. A. Amjad, C. Blochl, and B. C. Geiger. 2020. A Generalized Framework For Kullback–Leibler Markov Aggregation. *IEEE Trans. Automat. Control* 65, 7 (Jul 2020), 3068–3075. https://doi.org/10.1109/tac.2019.2945891

[3] K. Deng, P. G. Mehta, and S. P. Meyn. 2011. Optimal Kullback-Leibler Aggregation via Spectral Theory of Markov Chains. *IEEE Trans. Automat. Control* 56, 12 (2011), 2793–2808. https://doi.org/10.1109/TAC.2011.2141350

[4] D. Dua and C. Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[5] B. C. Geiger, T. Petrov, G. Kubin, and H. Koeppl. 2015. Optimal Kullback–Leibler Aggregation via Information Bottleneck. *IEEE Trans. Automat. Control* 60, 4 (Apr 2015), 1010–1022. https://doi.org/10.1109/tac.2014.2364971

[6] John G. Kemeny and James Laurie Snell. 1976. *Finite Markov Chains* (2 ed.). Springer.

[7] R. Lebret, S. Iovleff, F. Langrognet, C. Biernacki, G. Celeux, and G. Govaert. 2014. Rmixmod: The R Package of the Model-Based Unsupervised, Supervised and Semi-Supervised Classification Mixmod Library. *Journal of Statistical Software* 67 (12 2014). https://doi.org/10.18637/jss.v067.i06

[8] H. Liu and Y. Fu. 2015. Clustering with Partition Level Side Information. In *Proc. IEEE Int. Conf. on Data Mining (ICDM)*. IEEE, Atlantic City, NJ, 877–882. https://doi.org/10.1109/ICDM.2015.18

[9] W. Pedrycz, A. Amato, V. Di Lecce, and V. Piuri. 2008. Fuzzy Clustering With Partial Supervision in Organization and Classification of Digital Images. *IEEE Transactions on Fuzzy Systems* 16, 4 (2008), 1008–1026. https://doi.org/10.1109/TFUZZ.2008.917287

[10] W. Pedrycz and J. Waletzky. 1997. Fuzzy clustering with partial supervision. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 27, 5 (1997), 787–795. https://doi.org/10.1109/3477.623232

[11] P. Qian, Y. Jiang, S. Wang, K.-H. Su, J. Wang, L. Hu, and R. F. Muzic. 2017. Affinity and Penalty Jointly Constrained Spectral Clustering With All-Compatibility, Flexibility, and Robustness. *IEEE Transactions on Neural Networks and Learning Systems* 28, 5 (2017), 1123–1138. https://doi.org/10.1109/TNNLS.2015.2511179

[12] N. Shental, A. bar Hillel, T. Hertz, and D. Weinshall. 2003. Computing Gaussian Mixture Models with EM Using Equivalence Constraints.. In *Proc. Advances in Neural Information Processing Systems (NIPS)*. Vancouver.

[13] M. Smieja and B. C. Geiger. 2017. Semi-supervised cross-entropy clustering with information bottleneck constraint. *Information Sciences* 421 (Dec 2017), 254–271. https://doi.org/10.1016/j.ins.2017.07.016

[14] N. Tishby and N. Slonim. 2000. Data Clustering by Markovian Relaxation and the Information Bottleneck Method. In *Proc. Advances in Neural Information Processing Systems (NIPS)*. MIT Press, Cambridge, MA, USA, 619–625.

[15] M. Vidyasagar. 2010. Reduced-order modeling of Markov and hidden Markov processes via aggregation. In *Proc. IEEE Conf. on Decision and Control (CDC)*. IEEE, Atlanta, GA, 1810–1815. https://doi.org/10.1109/CDC.2010.5717206

[16] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. 2001. Constrained k-means clustering with background knowledge. In *Icml*, Vol. 1. 577–584.

[17] Y. Xu, S.M. Salapaka, and C. Beck. 2014. Aggregation of Graph Models and Markov Chains by Deterministic Annealing. *IEEE Trans. Automat. Control* 59 (10 2014). https://doi.org/10.1109/TAC.2014.2319473

[18] M. Śmieja, O. Myronov, and J. Tabor. 2018. Semi-supervised discriminative clustering with graph regularization. *Knowledge-Based Systems* 151 (2018), 24–36. https://doi.org/10.1016/j.knosys.2018.03.019