Journal of the American Statistical Association

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uasa20

Query-Augmented Active Metric Learning

Yujia Deng, Yubai Yuan, Haoda Fu & Annie Qu

To cite this article: Yujia Deng, Yubai Yuan, Haoda Fu & Annie Qu (2023) Query-Augmented Active Metric Learning, Journal of the American Statistical Association, 118:543, 1862-1875, DOI: 10.1080/01621459.2021.2019045

To link to this article: https://doi.org/10.1080/01621459.2021.2019045

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2023, VOL. 118, NO. 543, 1862–1875: Theory and Methods
https://doi.org/10.1080/01621459.2021.2019045

Query-Augmented Active Metric Learning

Yujia Deng*[a], Yubai Yuan[b], Haoda Fu[c], and Annie Qu[d]

[a]Department of Statistics, University of Illinois, Urbana–Champaign, IL; [b]Department of Statistics, University of California, Irvine, CA; [c]Eli Lilly and Company, Biometrics and Advanced Analytics, Indianapolis, IN; [d]Department of Statistics, University of California, Irvine, CA

ABSTRACT

In this article, we propose an active metric learning method for clustering with pairwise constraints. The proposed method actively queries the label of informative instance pairs, while estimating underlying metrics by incorporating unlabeled instance pairs, which leads to a more accurate and efficient clustering process. In particular, we augment the queried constraints by generating more pairwise labels to provide additional information in learning a metric to enhance clustering performance. Furthermore, we increase the robustness of metric learning by updating the learned metric sequentially and penalizing the irrelevant features adaptively. In addition, we propose a novel active query strategy that evaluates the information gain of instance pairs more accurately by incorporating the neighborhood structure, which improves clustering efficiency without extra labeling cost. In theory, we provide a tighter error bound of the proposed metric learning method using augmented queries compared with methods using existing constraints only. Furthermore, we also investigate the improvement using the active query strategy instead of random selection. Numerical studies on simulation settings and real datasets indicate that the proposed method is especially advantageous when the signal-to-noise ratio between significant features and irrelevant features is low.

ARTICLE HISTORY
Received December 2020
Accepted December 2021

KEYWORDS
Active learning; Metric learning; Selective penalty; Semi-supervised clustering

1. Introduction

In recent years, active learning has become a popular subfield of machine learning due to the fact that the performance of any supervised learning system fundamentally relies on labeled instances which are difficult or expensive to obtain in many applications. For example, the rise of electronic medical records introduces huge amounts of medical data which could be overwhelming and infeasible to examine for the entire populations. In practice, it is desirable to train diagnostic criteria through computing algorithms and preprocess a subset of data to reduce workloads for doctors. In addition, although the original medical record data are high-dimensional, it is important to capture essential medical information through low-dimension summary features. To achieve these goals, instead of formulating the problem as traditional supervised learning which requires training a model on a large dataset labeled by experts, we propose to adjust machine-generated criteria actively through feedback from experts to reduce costs and accelerate the training process.

The idea of incorporating the experts' domain knowledge or the user's feedback has been pursued in the previous clustering methods, e.g., Wagstaff et al. (2001), Basu, Banerjee, and Mooney (2004), Basu et al. (2006), Davidson, Wagstaff, and Basu (2006), Lu (2007), and Liu et al. (2017). Specifically, a user can specify that two instances must either belong to the same cluster or two different clusters. Then, the clustering procedure selects the optimal label assignment by penalizing the assignments that violate these pairwise constraints. Alternatively, instead of directly clustering the instances in the original feature space, metric learning approaches, for example, Xing et al. (2003), Niu et al. (2011), Yang, Jin, and Sukthankar (2012), and Hoi, Liu, and Chang (2010) sought a specific distance metric trained from the constraints. The essential goal of distance metric learning is to identify an appropriate distance metric that encourages "similar" objects to be close together while separating "dissimilar" objects, which improves the performance of the subsequent clustering process.

However, the aforementioned metric learning process could be inefficient and unstable since randomly chosen constraints may provide little information about the group structure in the latent space, especially when the sample size is small. Several active learning solutions have been proposed to solve these problems. For example, Basu, Banerjee, and Mooney (2004) proposed a *explore-consolidate* framework which seeks the skeleton points that are dissimilar to each other first and then use similar pairs to refine the boundary of the clusters. This method is then generalized by Mallapragada, Jin, and Jain (2008) in selecting the most informative pairs in the consolidate phase. Grira, Crucianu, and Boujemaa (2005) proposed active fuzzy constrained clustering, which sequentially queries and collects the labels for instances under boundary of clusters. Alternatively, Huang and Mitchell (2006), Xiong, Azimi, and Fern (2013), and Biswas

CONTACT Annie Qu aqu2@uci.edu Department of Statistics, University of California Irvine, Irvine, CA 92697.
*These authors contributed equally to this work.
Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JASA.

and Jacobs (2014) proposed different models to quantify the uncertainty of the unlabeled pairs. Other methods include Mai et al. (2013) using neighborhood information in density-based clustering, Greene and Cunningham (2007) building an ensemble framework for unlabeled pair selection, and Van Craenendonck, Dumancic, and Blockeel (2018a) using the propagation of similarity relations.

However, approaches exploring active clustering methods which incorporate metric learning simultaneously are still limited. To learn the latent metric from pairwise constraints, Yang, Jin, and Sukthankar (2012) proposed an active Bayesian metric learning that defines a Mahalanobis distance between instances and actively queries using entropy based criteria; Xiong, Azimi, and Fern (2013) proposed an instance-level uncertainty-based active query strategy combining metric pairwise constrained Kmeans (MPCKmeans Bilenko, Basu, and Mooney 2004). The main drawback of their methods is that they do not utilize the unlabeled instance pairs in learning the metric. Although the number of pairwise constraints provided by the user is limited, the relationships between the unlabeled instances can still be inferred based on the clustering structure, which could supply additional information and therefore improve the learning efficiency. Another limitation on the existing active clustering methods is that they do not utilize a dimension reduction strategy for raw data during human–machine interaction. However, identifying and selecting significant features which are consistent with a user's clustering principles are very important for enhancing the similarity within a cluster, and to achieve a more robust and consistent clustering outcome. In addition, dimension reduction also leads to more interpretable clustering criteria from experts. Furthermore, existing models are typically retrained each time the new constraints are added, while the history of training results is ignored. This results in a loss of information which could be utilized to improve clustering performance.

In this article, we propose a new active clustering method with query augmentation and metric aggregation. The novelty of the proposed method is that we incorporate both pairwise constraints from the user's feedback, and the implicit constraints inferred based on the clustering structure to learn the metric. We also integrate the unlabeled instance pairs into the metric learning process through augmented constraints weighted by uncertainty measurement, which leads to more efficient recovery of the underlying feature space. Another novelty is that we pursue dimension reduction by penalizing the irrelevant features adaptively, based on the history of metric learning results in the sequential querying process. Thus, we obtain more precise and robust clustering results consistent with the user's feedback. In addition, we propose a new instance query strategy based on the expected entropy change. Compared with existing active learning methods, we can incorporate the neighborhood structure and transitivity of the constraints through uncertainty measurement, which provides a more accurate evaluation of the potential effect from the queried constraints on the cluster structure. Theoretical and numerical results confirm that the proposed method improves clustering accuracy without adding labeling cost.

The article is organized as follows. Section 2 introduces notations and background for metric learning and active semi-supervised clustering. Section 3 presents a new active metric learning framework and the metric aggregation method. Section 4 introduces algorithms to implement the proposed method. Section 5 establishes the theoretical properties. Section 6 provides the simulation results of the proposed active learning. Section 7 illustrates the application of the proposed method for three real datasets. The last section provides concluding remarks and discussion.

2. Background and Formulation

Given n data points in a p-dimensional feature space, that is, $\boldsymbol{x}_i \in \mathbb{R}^p, i = 1, ..., n$, we assume each $\boldsymbol{x}_i$ is sampled from one of the K clusters, and denote the cluster membership vector as $\boldsymbol{l} = (\ell_1, \ldots, \ell_n)$, where $\ell_i \in \{1, \ldots, K\}$. To ensure the identifiability of the cluster labels, for any two cluster membership vectors $\boldsymbol{l}^{(1)}$ and $\boldsymbol{l}^{(2)}$, we define $\boldsymbol{l}^{(1)} = \boldsymbol{l}^{(2)}$ if there exists a permutation map Γ of $\{1, \ldots, K\}$ such that $\ell_i^{(1)} = \Gamma(\ell_i^{(2)}),\ i = 1, ..., n$. Let the sample space of $\boldsymbol{l}$ be Ω, and then the cardinality $|\Omega|$ equals the total number of ways to partition a set of n objects into K non-empty subsets up to label-switching. We also denote the similarity matrix as $Y \in \mathbb{R}^{n\times n}$, where $y_{ij} = 1$ if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are in the same cluster, and 0 otherwise. Since there is a one-to-one map between Y and $\boldsymbol{\ell}$ determined by $y_{ij} = \mathbb{1}(\ell_i = \ell_j)$, the goal of clustering can be defined as the estimation of either Y or $\boldsymbol{l}$. In *unsupervised* clustering, no elements of Y are known beforehand, while in *semi-supervised* clustering, a part of the elements of Y are queried from users as pairwise constraints. These pairwise constraints are referred to as "similar" and "dissimilar" pairs whose index sets are denoted as $\mathcal{S} = \{(i,j)|y_{ij} = 1, \text{observed}\}$ and $\mathcal{D} = \{(i,j)|y_{ij} = 0, \text{observed}\}$, respectively, while the unlabeled set is denoted as $\mathcal{U} = \left\{(i,j) \mid (i,j) \notin \mathcal{S} \cup \mathcal{D}\right\}$. The pairwise constraints have the following transitivity property:

Property 1 (Transitivity). For different indexes i, j, k, if $(i,j) \in \mathcal{S}$ and $(i,k) \in \mathcal{S}$, then $(j,k) \in \mathcal{S}$. If $(i,j) \in \mathcal{S}$ and $(i,k) \in \mathcal{D}$, then $(j,k) \in \mathcal{D}$.

The transitivity property allows us to generate more constraints within one query, which is essential in improving the efficiency of a query strategy.

Formally, the semi-supervised clustering method aims to maximize the posterior distribution over the cluster memberships $\boldsymbol{l}$, where the queried pairwise labels provide additional information through the prior distribution. Equivalently, we maximize the posterior distribution $\rho(\boldsymbol{l}|\boldsymbol{x}) \propto f(\boldsymbol{x}|\boldsymbol{l})\pi(\boldsymbol{l})$, where $f(\boldsymbol{x}|\boldsymbol{l})$ is the likelihood function, and $\pi(\boldsymbol{l})$ is the prior distribution.

We start with a mixture-Gaussian model such that

$$f(\boldsymbol{x}|\boldsymbol{l}) \propto \exp\left(-\frac{1}{2}\sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{v}_{\ell_i}\|_A^2\right),$$

where $\boldsymbol{v}_k$ is the centroid of the kth cluster, $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_A^2 = (\boldsymbol{x}_i - \boldsymbol{x}_j)^\top A (\boldsymbol{x}_i - \boldsymbol{x}_j)$, and A is the precision matrix. In addition, a low-rank matrix A indicates that the likelihood of the cluster structure can be captured in a linear subspace $\mathbb{R}^r \subset \mathbb{R}^p$, $r \leq p$. By correctly estimating A, we are able to identify the relevant feature space and improve the clustering

performance. In other works, e.g., Xing et al. (2003) and Xiong, Azimi, and Fern (2013), A is also called the *metric matrix* which measures the distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in the linear subspace $\mathbb{R}^r$. Since only limited pairwise constraints are known, one common principle in learning A is through the distance $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_A$, which is small if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ belong to the same cluster, and large if they are from different clusters. Following this principle, one particular metric learning method (Xing et al. 2003) of A is via

$$\min_A \sum_{(i,j)\in\mathcal{S}} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_A^2, \quad \text{s.t.} \sum_{(i,j)\in\mathcal{D}} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_A \geq 1, \quad \text{and} \quad A \succeq 0, \tag{1}$$

where $A \succeq 0$ denotes that A is positive semi-definite. The above training process (1) minimizes the distances between similar pairs while separating dissimilar pairs to avoid trivial solutions with all zeros.

In addition, the prior distribution $\pi(\boldsymbol{l})$ penalizes the probability of cluster label assignments that violate the queried pairwise constraints. Following Basu, Banerjee, and Mooney (2004), we let

$$\pi(\boldsymbol{l}) \propto \exp\left(-\sum_{i,j} V_{ij}(\boldsymbol{l}, \mathcal{S}, \mathcal{D})\right), \tag{2}$$

where

$$V_{ij}(\boldsymbol{l}, \mathcal{S}, \mathcal{D}) = \begin{cases} \mathbb{1}(\ell_i \neq \ell_j) & (i,j) \in \mathcal{S}, \\ \mathbb{1}(\ell_i = \ell_j) & (i,j) \in \mathcal{D}, \\ 0 & \text{otherwise.} \end{cases}$$

A key question in semi-supervised clustering is how to obtain the most informative prior $\pi(\boldsymbol{l})$ given a limited number of queries. To this end, we adopt an *active learning* scheme to update the query criterion sequentially (Xiong, Azimi, and Fern 2013; Basu, Banerjee, and Mooney 2004; Van Craenendonck, Dumancic, and Blockeel 2018a).

3. Methodology

In this section, we introduce an efficient metric learning method with augmented pairwise constraints, and design an active query strategy based on a new uncertainty criterion.

3.1. Metric Learning With Augmented Pairwise Constraints

We start with improving metric learning by augmenting limited numbers of pairwise constraints. One common problem of (1) and existing metric learning methods (Wagstaff et al. 2001; Grira, Crucianu, and Boujemaa 2005; Lu 2007) is that only the violations on queried pairwise constraints are penalized. However, these queried constraints also provide additional prior information on other unlabeled neighborhood pairwise relations implicitly through the underlying cluster structure. To solve this problem, we generalize the queried pairwise constraints $\mathcal{S} \cup \mathcal{D}$ to all y_{ij}'s by inferring the labels of unlabeled instance pairs, and train the metric matrix A with both the queried pairwise constraints and the inferred pairwise constraints.

Specifically, we solve for a fuzzy membership matrix $H \in \mathbb{R}^{n\times K}$ by

$$\begin{aligned} \widehat{H} = \operatorname{argmin}_H & \sum_{(i,j)\in\mathcal{S}\cup\mathcal{D}} (y_{ij} - \boldsymbol{h}_i^T \boldsymbol{h}_j)^2 \\ & + \lambda \sum_{i=1}^{n}\sum_{k=1}^{K} \min(|h_{ik}|, |h_{ik} - 1|), \\ \text{s.t.} \quad & h_{ij} \geq 0, \quad \sum_{k=1}^{K} h_{ik} = 1, \text{ for all } i, \end{aligned} \tag{3}$$

where $\boldsymbol{h}_i^T$ is the ith row of H. In contrast to the discrete labels ℓ_i, h_{ik} is continuous on $[0,1]$ and represents the probability that the ith sample belongs to the kth cluster. The penalty term $\min(|h_{ik}|, |h_{ik} - 1|)$ is a multi-directional separation penalty (MDSP) (Tang, Xue, and Qu 2020), which penalizes h_{ij} to either 0 or 1 depending on the magnitude of h_{ij}. The purpose of adding the MDSP penalty is to prevent strong signals from being pulled toward zero in the process of shrinking weak signals for sparsity pursuit, and thus to reduce the uncertainty on the cluster membership of each instance. In addition, we only infer $\boldsymbol{h}_i$ if at least one element of $\{y_{i\cdot}\}$'s is observed; otherwise we let all the elements of $\boldsymbol{h}_i$ be $1/K$. Note that the augmenting process (3) uses only the queried constraint information without involving the distance between data points since the distance metric is inaccurate during training, which may lead to biased membership inference.

Next, we use $\widehat{H}$ to introduce additional pairwise constraints through the concordance $\widehat{\boldsymbol{h}}_i^\top \widehat{\boldsymbol{h}}_j$. The idea is that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ tend to be similar if $\widehat{\boldsymbol{h}}_i^\top \widehat{\boldsymbol{h}}_j$ is close to 1, and dissimilar if $\widehat{\boldsymbol{h}}_i^\top \widehat{\boldsymbol{h}}_j$ is close to 0. Considering the completely random case when $\widehat{\boldsymbol{h}}_i^\top = \widehat{\boldsymbol{h}}_j^\top = (1/K, ..., 1/K)$ and $\widehat{\boldsymbol{h}}_i^\top \widehat{\boldsymbol{h}}_j = 1/K$, we choose $1/K$ as a threshold for the effective concordance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, and define the augmented constraints as $\widetilde{\mathcal{S}} = \{(i,j) | \widehat{\boldsymbol{h}}_i^\top \widehat{\boldsymbol{h}}_j > 1/K\}$ and $\widetilde{\mathcal{D}} = \{(i,j) | \widehat{\boldsymbol{h}}_i^\top \widehat{\boldsymbol{h}}_j < 1/K\}$. Then we train the metric matrix through

$$\begin{aligned} \min_A \quad \text{Loss}(A) \triangleq & \frac{1}{|\mathcal{S}|} \sum_{(i,j)\in\mathcal{S}} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_A^2 \\ & + \frac{1}{|\widetilde{\mathcal{S}}|} \sum_{(i,j)\in\widetilde{\mathcal{S}}} w_{ij} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_A^2, \\ \text{s.t.} \quad & \frac{1}{|\mathcal{D}|} \sum_{(i,j)\in\mathcal{D}} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_A \\ & + \frac{1}{|\widetilde{\mathcal{D}}|} \sum_{(i,j)\in\widetilde{\mathcal{D}}} w_{ij} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_A \geq 1, \quad A \succeq 0, \end{aligned} \tag{4}$$

where $|\cdot|$ denotes the set cardinality, and $w_{ij} = \frac{K}{K-1} \max\{\widehat{\boldsymbol{h}}_i^\top \widehat{\boldsymbol{h}}_j - \frac{1}{K}, 0\} - K \min\{\widehat{\boldsymbol{h}}_i^\top \widehat{\boldsymbol{h}}_j - \frac{1}{K}, 0\}$. Compared with (1), we involve the augmented similar constraints $\widetilde{\mathcal{S}}$ and the dissimilar constraints $\widetilde{\mathcal{D}}$ in both Loss(A) and the constraint in (4), and we normalize each term by set cardinality to avoid inconsistent scales caused

by imbalanced numbers of similar and dissimilar pairs. In addition, we use $w_{ij} \in [0, 1]$ to quantify the certainty of the inference by imposing less weight on the augmented constraints that are similar to random guess, while imposing a large weight on the constraints queried from users and the inferred constraints if their concordance equals 0 or 1. In this way, we are able to fully utilize the information from the total of $n(n-1)/2$ pairs to learn the metric matrix, instead of $|\mathcal{S}| + |\mathcal{D}|$ as in the conventional metric learning methods.

3.2. Active Query With Minimum Expected Entropy

In this subsection, we introduce a new active query strategy. Instead of randomly selecting a single pair, we propose to select the most important instance whose neighborhood membership affects the expected posterior distribution of the cluster label assignment $\boldsymbol{l}$ significantly. This procedure increases the query efficiency by generating more pairwise constraints within a single query due to the transitivity property.

Formally, we define a neighborhood as a subset of instances which belong to the same cluster based on the queried constraints. We start with one neighborhood which contains a single instance, and sequentially identify the memberships of the instances outside the existing neighborhoods by querying their similarity with the instances within the neighborhoods. Therefore, any pairs within the same neighborhood are similar while any pairs across different neighborhoods are dissimilar. We define the *budget* as the maximum number of queries, which is denoted by B. Meanwhile, we define each *step* as the process to determine the neighborhood membership of a new instance x^* outside the neighborhoods, denoted as $t = 1, ..., T$. Accordingly, we denote the number of queries consumed at the tth step as b^t, then $1 \le b^t \le K - 1$. Note that b^t increases when a query occurs, and the entire active clustering stops only when the budget is exhausted, that is, $\sum_t^T b^t = B$. We also denote the mth neighborhood at the tth step as N_m^t, then for any $\boldsymbol{x}_i, \boldsymbol{x}_j \in N_m^t$, we have $(i, j) \in \mathcal{S}^t$, and for any $\boldsymbol{x}_i \in N_m^t, \boldsymbol{x}_j \in N_{m'}^t, m \neq m'$, we have $(i, j) \in \mathcal{D}^t$, where $\mathcal{S}^t$ and $\mathcal{D}^t$ denote the set of similar pairs and dissimilar pairs at the tth step, respectively.

Moreover, we denote the union of the neighborhoods at the tth step as $\mathcal{N}^t = N_1 \cup \cdots \cup N_{L^t}$, where $L^t \le K$ is the total number of neighborhoods; then for the next step, we select $\boldsymbol{x}_i \notin \mathcal{N}^t$ to determine its neighborhood membership by querying its relationship with $\boldsymbol{x}_j \in N_j, j = 1, \ldots, N_{L^t}$ sequentially until a similar pair is found. If $\boldsymbol{x}_i$ does not belong to any of the neighborhoods, then we formulate a new neighborhood as $\{\boldsymbol{x}_i\}$ and update $\mathcal{N}$ with $\mathcal{N}^t \cup \{\boldsymbol{x}_i\}$ and L with $L^t + 1$. Note this query procedure costs at most K queries, but can generate $|\mathcal{N}^t|$ pairwise constraints due to transitivity. In addition, since $|\mathcal{N}^t|$ increases as t grows, we are able to acquire more constraints with the same cost as the query procedure continues.

Next, we introduce an uncertainty measurement to query an instance outside $\mathcal{N}^t$. We denote the underlying posterior as $\rho_*(\boldsymbol{l}|\boldsymbol{x}) \propto f(\boldsymbol{x}|\boldsymbol{l})\pi_*(\boldsymbol{l})$, where $\pi_*(\boldsymbol{l})$ denotes the prior with the labels of all data pairs; and the posterior of the tth step as $\rho^t(\boldsymbol{l}|\boldsymbol{x}) \propto f(\boldsymbol{x}|\boldsymbol{l})\pi^t(\boldsymbol{l})$, where $\pi^t(\boldsymbol{l})$ involves $\mathcal{S}^t$ and $\mathcal{D}^t$ only. Under this framework, the discrepancy between $\rho^{t+1}(\boldsymbol{l}|\boldsymbol{x})$ and $\rho_*(\boldsymbol{l}|\boldsymbol{x})$ relies on $\pi^{t+1}(\boldsymbol{l})$, which is determined by the query strategy.

We propose to select $\boldsymbol{x}_i \notin \mathcal{N}^t$ whose neighborhood membership is expected to make the posterior distribution closest to the underlying truth via minimizing the Kullback–Leibler divergence (KL-divergence):

$$\begin{aligned} \boldsymbol{x}_i^* &= \underset{\boldsymbol{x}_i \notin \mathcal{N}^t}{\operatorname{argmin}} \sum_{\boldsymbol{l} \in \Omega} \rho_*(\boldsymbol{l} \mid \boldsymbol{x}) \log \left(\frac{\rho_*(\boldsymbol{l} \mid \boldsymbol{x})}{\rho_i^+(\boldsymbol{l} \mid \boldsymbol{x})} \right) \\ &= \underset{\boldsymbol{x}_i \notin \mathcal{N}^t}{\operatorname{argmin}} - \sum_{\boldsymbol{l} \in \Omega} \rho_*(\boldsymbol{l} \mid \boldsymbol{x}) \log \rho_i^+(\boldsymbol{l} \mid \boldsymbol{x}), \end{aligned} \tag{5}$$

where $\rho_i^+(\boldsymbol{l}|\boldsymbol{x})$ denotes the posterior distribution after determining the neighborhood membership of $\boldsymbol{x}_i$. However, since both ρ_* and the true membership of $\boldsymbol{x}_i$ are unobserved, we cannot solve (5) directly. Instead, we consider the following approximation:

$$\begin{aligned} \boldsymbol{x}_i^* = \operatorname{argmin}_{\boldsymbol{x}_i \notin \mathcal{N}^t} & - \sum_{m=1}^{L^t} P^t(\ell_i = m) \\ & \times \sum_{\boldsymbol{l} \in \Omega} \rho_{im}^+(\boldsymbol{l}|\boldsymbol{x}) \log \rho_{im}^+(\boldsymbol{l}|\boldsymbol{x}), \end{aligned} \tag{6}$$

where $\rho_{im}^+(\boldsymbol{l}|\boldsymbol{x})$ denotes the posterior distribution assuming $\boldsymbol{x}_i \in N_m$. The active query strategy (6) can be interpreted as a minimization of the expected entropy of the posterior distribution when new constraints are added, which is equivalent to selecting the instance whose neighborhood membership is the most uncertain based on the information at the tth step. The neighborhood structure is shown to be effective in the normalized point-based uncertainty (NPU) (Xiong, Azimi, and Fern 2013). However, the NPU considers the uncertainty decrease only based on the queried instance, while the proposed method (6) measures the uncertainty decrease over the entire dataset. Therefore, the proposed criterion estimates the information gain from the new query holistically and globally, and thus is able to select more informative instances.

3.3. Metric Aggregation Through Adaptive Penalty

Most of the existing active clustering methods do not incorporate the history of training results in the final model. However, the metric learned during the previous steps may provide additional information for identifying the significant features related to the user-specified clustering principles, and therefore could be utilized to improve the efficiency of learning low-dimensional feature space.

We propose to aggregate the metric matrices learned in each step to extract the underlying significant features by imposing an adaptive penalty on (4), and capture a clustering-oriented subspace. Note that imposing a penalty on all features simultaneously makes a limited impact on the clustering result, since clustering is invariant to the scale of the elements in the metric matrix. Instead, we impose a selective penalty on a subset of features to increase the relative weights of the significant features over the irrelevant ones.

We denote the minimizer of (4) at the tth step as A^t. To determine which features are important for clustering, we aggregate the training results of the previous $T - 1$ steps by imposing a penalty adaptively based on the eigenvalues of A^t. In general,

the features with smaller eigenvalues on average are less relevant in clustering and thus should have smaller weights. We let $\boldsymbol{r}^t = (r_1, \ldots, r_p)$, $t = 1, ..., T-1$ be the rank statistics of p eigenvalues from A^t in the ascending order, and $\bar{\boldsymbol{r}} = \frac{1}{T-1}\sum_{t=1}^{T-1} \boldsymbol{r}^t$ be the average rank. To shrink the weights on irrelevant features, we penalize the top q features with the smallest entries in $\bar{\boldsymbol{r}}$, where q is the number of penalized features. We denote the index set of the penalized features at the Tth step as $\mathcal{G}^T$, $|\mathcal{G}^T| = q$. Then for the Tth step, we train the metric matrix by adding a selective penalty on A through

$$\begin{aligned} \widehat{A} = \operatorname{argmin}_A \quad & \text{Loss}(A) + \gamma \sum_{k\in\mathcal{G}^T} \sigma_k(A), \\ s.t. \quad & \frac{1}{|\mathcal{D}^T|} \sum_{(i,j)\in\mathcal{D}^T} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_A \\ & + \frac{1}{|\widetilde{\mathcal{D}}^T|} \sum_{i,j} w_{ij}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_A \geq 1, \quad A \succeq 0, \end{aligned} \tag{7}$$

where γ is a tuning parameters and $\sigma_k(A)$ denotes the kth eigenvalue of A. The reason we use the rank statistic in determining $\mathcal{G}$ instead of using the eigenvalue directly is that the rank statistic is robust to outliers from the distribution of eigenvalues due to randomness, which lowers the risk of incorrectly penalizing significant features. In particular, when the metric matrix is diagonal, the proposed selective penalizing procedure is equivalent to adding an L_1 penalty to a subset of the diagonal entries of A. Different from the commonly used nuclear norm penalty which penalizes all eigenvalues of A, the selective penalty in (7) only penalizes the eigenvalues in $\mathcal{G}^T$ to prevent the signal features from being penalized; as the Lasso-type penalty tends to shrink all nonzero values as well, and that may lead to biased estimation of metric A.

After acquiring $\widehat{A}$ through (7), we solve for the cluster membership by performing pairwise constrained Kmeans (PCKmeans) (Basu, Banerjee, and Mooney 2004) on the learned linear subspace via

$$\begin{aligned} \widehat{\boldsymbol{l}} = \operatorname{argmin}_{\boldsymbol{l}} & \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{v}_{\ell_i}\|_{\widehat{A}}^2 + \sum_{(i,j)\in\mathcal{S}^T} \mathbb{1}(\ell_i \neq \ell_j) \\ & + \sum_{(i,j)\in\mathcal{D}^T} \mathbb{1}(\ell_i = \ell_j), \end{aligned} \tag{8}$$

where $\boldsymbol{v}_k = \sum_{i=1}^n \boldsymbol{x}_i \mathbb{1}(\ell_i = k)/\sum_{i=1}^n \mathbb{1}(\ell_i = k)$ is the centroid of the kth cluster. Here, we shrink the search space of the membership $\boldsymbol{l}$ by penalizing the cases where the label assignments violate the queried pairwise constraints. Different from the original PCKmeans, we compute the distances between the samples and cluster centers with the learned metric $\widehat{A}$ so that irrelevant features are excluded.

4. Algorithm and Implementation

In this section, we introduce an algorithm to solve the query augmentation problem in (3), the metric learning with selective penalty in (7), and the active query strategy in (6).

We adopt the alternating direction method of multipliers (ADMM) method to solve (3), and decompose the optimization problem (3) into several subproblems that can be solved more easily. The details of ADMM are provided in the supplementary material.

Next, we implement the active query strategy (6) with the neighborhood structure. Notice that computation of the exact expected entropy in (6) requires the enumeration of all possible membership assignments over Ω, which is computationally infeasible. Alternatively, we propose to approximate the expected entropy by taking the summation of the expected entropy for each unlabeled pairs, which contains at most $|\mathcal{U}| \leq n(n-1)/2$ terms. Furthermore, instead of considering the posterior distribution of each pair directly, we estimate the posterior distribution based on the neighborhood membership of each data point to simplify computation.

Mathematically, we let $R^t \in \mathbb{R}^{n\times L^t}$ be the neighborhood membership matrix, where $r_{im}^t = P(\boldsymbol{x}_i \in N_m^t)$. We also denote the probability that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ belong to the same neighborhoods p_{ij}^t. Then we approximate the entropy in (6) by

$$Q(R^t) = -\sum_{(i,j)\in\mathcal{U}} \left\{ p_{ij}^t \log p_{ij}^t + (1-p_{ij}^t)\log(1-p_{ij}^t) \right\},$$

where R^t is implicitly included in each p_{ij}^t and is omitted in the expression for notation simplicity. The expected entropy by identifying the neighborhood membership of $\boldsymbol{x}_i$ is then

$$u^t(\boldsymbol{x}_i) = \sum_{m=1}^{L^t} r_{im}^t Q(\widetilde{R}_{-i,m}^{t+1}), \tag{9}$$

where $\widetilde{R}_{-i,m}^{t+1} \in \mathbb{R}^{n\times L^{t+1}}$ is defined elementwise by

$$\widetilde{r}_{ij}^{t+1} = \begin{cases} r_{kj}^t, & \text{if } k \neq i, \\ 0, & \text{if } k = i \text{ and } j \neq m, \\ 1, & \text{if } k = i \text{ and } j = m. \end{cases}$$

That is, $\widetilde{R}_{-i,m}^{t+1}$ denotes the neighborhood membership matrix assuming that $\boldsymbol{x}_i$ belongs to the mth neighborhood, and $u^t(\boldsymbol{x}_i)$ estimates the expected entropy after obtaining the neighborhood membership of $\boldsymbol{x}_i$. We then select $\boldsymbol{x}^* = \operatorname{argmin}_{\boldsymbol{x}_i\notin\mathcal{N}^t} u^t(\boldsymbol{x}_i)$.

We estimate r_{im}^t and p_{ij}^t by a random forest trained on the clustering result with the learned metric matrix at the tth step. Specifically, p_{ij}^t is estimated by the number of times when $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are assigned to the same leaf of the tree divided by the total number of trees in the random forest, and $r_{im}^t = \sum_{\boldsymbol{x}_j\in N_m^t} p_{ij}^t/|N_m^t|$. The random forest is computationally efficient without assuming the explicit form of $f(\boldsymbol{x}|\boldsymbol{l})$, making the model more flexible for general cluster structures. The random forest has been applied successfully in previous studies (Shi and Horvath 2006; Xiong, Azimi, and Fern 2013) for unsupervised clustering tasks, especially for quantifying the uncertainty of memberships.

Meanwhile, for the selective penalty optimization (7), we first consider the diagonal case. Denote the diagonal entries of A as $\boldsymbol{a}$, then the selective penalty is equivalent to penalizing a subgroup

of entries in $\boldsymbol{a}$ directly. We consider the equivalent form of (7) such that the constraint is linear regarding $\boldsymbol{a}$

$$\begin{aligned}\widehat{\boldsymbol{a}}^T = \arg\max_{\boldsymbol{a}} & \frac{1}{|\mathcal{D}^T|}\sum_{(i,j)\in\mathcal{D}^T}\sqrt{\sum_{m=1}^{p} a_m(x_{im}-x_{jm})^2} \\ & + \frac{1}{|\widetilde{\mathcal{D}}^T|}\sum_{i,j} w_{ij}\sqrt{\sum_{m=1}^{p} a_m(x_{im}-x_{jm})^2} \\ s.t. \quad & \frac{1}{|\mathcal{S}^T|}\sum_{(i,j)\in S^T}\sum_{m=1}^{p} a_m(x_{im}-x_{jm})^2 \\ & + \frac{1}{|\widetilde{\mathcal{S}}^T|}\sum_{i,j}\sum_{m=1}^{p} w_{ij}a_m(x_{im}-x_{jm})^2 \\ & + \gamma\sum_{p\in\mathcal{G}^T} a_p \le 1, \quad a_p \ge 0. \end{aligned} \tag{10}$$

The above function can be maximized using the projected gradient descent method.

For the nondiagonal case, we separate the optimization procedure into two steps. We first seek the best subspace in which data can be clustered more efficiently, and then aggregate the feature weights within the subspace. Mathematically, denote the spectral decomposition of the trained metric at the tth step as $A^t = P^t\Lambda^t(P^t)^\top$, where Λ^t is diagonal, and P^t is orthogonal. We select the metric matrix with the largest eigengap from $\{A^t\}_{t=1}^T$, and denote it as $A^* = P^*\Lambda^*(P^*)^\top$, where the eigengap is defined as the maximum ratio of two adjacent eigenvalues. Next, we project each A^t onto P^* via an orthogonal matrix V^t such that $P^t = P^*V^t$, and thus $A^t = P^*V^t\Lambda^tV^t(P^*)^\top$. Intuitively, this step decomposes the features of $P^t\boldsymbol{x}$ to features in $P^*\boldsymbol{x}$. We let $W^t = V^t\Lambda^t(V^t)^\top$, and denote the diagonal matrix with the same diagonal entries of W^t as D^t. Then, we train the diagonal metric $\widehat{D}$ from $\{D^t\}_{t=1}^T$ via the aggregation procedure for the diagonal case through (10). Finally, we compute $\widehat{A} = P^*\widehat{D}(P^*)^\top$ as our final learned metric matrix.

We remark that although W^t is not necessarily diagonal, D^t is still able to capture the main signals of W^t. To see this, notice that the features of $P^*\boldsymbol{x}$ which are irrelevant to clustering would have small diagonal entries in W^t, and their interaction terms with other features, that is, the off-diagonal entries of W^t, should be close to zero as well. More importantly, the diagonality of $\{D^t\}$ ensures that the order of weights learned at different steps are consistent to features in $P^*\boldsymbol{x}$ so that they can be aggregated in the end. The complete algorithm is summarized in Algorithm 1.

The total computational complexity of Algorithm 1 is $\mathcal{O}\{T(n_\tau n\log n + n^2K + \frac{1}{\epsilon^2}nK^3 + \frac{1}{\epsilon}p^3) + npKt_{pck}\}$, where n_τ is the number of trees in the random forest used during active query; ϵ is the predetermined error bound for constraint augmentation and metric learning; and t_{pck} is the iteration number of PCKmeans. Specifically, for a single query step, the active query selection costs $\mathcal{O}\left(n_\tau n\log n + n^2K\right)$, where $n_\tau n\log n$ is the complexity of training a random forest Breiman (2003) and n^2K is the complexity to compute the information criterion $u^t(\boldsymbol{x}_i)$. In addition, the query augmentation costs $\mathcal{O}\left(\frac{1}{\epsilon^2}nK^3\right)$, and the metric learning procedure requires $\mathcal{O}(\frac{1}{\epsilon}p^2)$ and $\mathcal{O}(\frac{1}{\epsilon}p^3)$ for the diagonal and the non-diagonal A, respectively. Finally, the PCKmeans costs $npKt_{pck}$. Empirically, training a random forest with 50 trees takes 0.05 sec and one loop in the simulation study with $p = 35, n = 300$ and $K = 5$ costs 8 sec on an Intel 4-Core i7-8650U CPU at 1.90GHz.

Algorithm 1 Query-augmented active clustering with metric aggregation

Input: Data $\{\boldsymbol{x}_i\}$, budget B, number of clusters K.
Output: Cluster label $\boldsymbol{l}$.
Initialization: Single neighborhood $\mathcal{N} = \{N_1\}, N_1 = \{\boldsymbol{x}_1\}$, where $\boldsymbol{x}_1$ is randomly selected. Let $\mathcal{G}^0 = \mathcal{S} = \mathcal{D} = \varnothing$, $A^0 = \boldsymbol{I}_p$ and $T = t = 1$.

1. (*Active metric learning*) While $\sum_{t=1}^T b^t \le B$, repeat:
   (a) (*Active query*) Train a random forest to estimate the neighborhood membership matrix R^t. Select the most informative instance $\boldsymbol{x}^*$ to minimize (9). Sort $\{N_i^t\}'s$ in descending order of $p(\boldsymbol{x}^* \in N_i^t)$. Let $b^t = 0$, and $T = t$.
   (b) Query $\boldsymbol{x}^*$ against an instance $x_i \in N_i^t$, update $\mathcal{S}$ or $\mathcal{D}$ according to the feedback, $b^t \leftarrow b^t + 1$.
   (c) Repeat step (b) for the rest of the neighborhoods until a similar link between $\boldsymbol{x}^*$ and $\boldsymbol{x}_i$ is provided by the user or $\sum_{t=1}^T b^t = B$. Let $N_i^t \leftarrow N_i^t \cup \{\boldsymbol{x}^*\}$. If no similar link is found, treat $\boldsymbol{x}^*$ as a new neighborhood. Let $N^* \leftarrow \{\boldsymbol{x}^*\}$ and $\mathcal{N}^t \leftarrow \mathcal{N}^t \cup \{N^*\}$.
   (d) (*Metric update*) Augment the pairwise constraints by solving (3) using the ADMM algorithm. Train metric A^t with the augmented queries by (4). Let $t \leftarrow t + 1$.
2. (*Metric aggregation*) Compute $\mathcal{G}^T$ based on $\{A^t\}_{t=1}^T$, solve for $\widehat{A}$ with selective penalty (7).
3. (*Semi-supervised clustering*) Cluster the instances with PCK-means (8) based on the learned metric $\widehat{A}$ and the acquired pairwise constraints $\mathcal{S}$ and $\mathcal{D}$.

In the following, we provide a brief discussion on the selection of tuning parameters in Algorithm 1, that is, λ in constraint augmentation (3), and γ and q in the selective penalty (7). In practice, we select λ by a 5-fold cross-validation based on the labeled pairwise constraints with a grid search on $[0, 1]$ after each loop of step (1) in Algorithm 1. On the other hand, we select γ by maximizing the Calinski-Harabasz index (Caliński and Harabasz 1974), which evaluates the clustering results by the ratio of the between-cluster variance and the within-cluster variance obtained from the PCKmeans. Different from λ, we tune γ only at Step 2 of Algorithm 1. Finally, q can be selected based on the unpenalized metric learning from Step 1 in Algorithm 1. Specifically, q is selected by the elbow point corresponding to the average eigenvalues of the metric matrices $\{A^t\}_{t=1}^T$, which can be trained without penalty. More details of the parameter tuning for our numerical studies can be found in the supplementary materials.

5. Theory

In this section, we introduce theoretical results for the proposed active clustering method. We first show the advantage of

incorporating the augmented pairwise constraints in the metric learning step, and next demonstrate the improvement of the active query strategy over the passive learning approach by comparing the proposed method with random selection as well as a simple two-step query strategy.

We formulate the metric learning into the semi-supervised learning framework which consists of $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ as pairs of data, and y_{ij}'s as labels. Our goal is to learn a binary classifier parameterized by the metric matrix A trained by the pairwise constraints as labeled data. The number of queries required to achieve a certain prediction accuracy without considering the unlabeled data can be derived by the VC dimension (Devroye, Györfi, and Lugosi 2013). The VC dimension of a function space $\mathcal{C}$ is the maximum number of arbitrarily labeled points that can be classified correctly by the functions in $\mathcal{C}$. However, utilizing the underlying clustering data structure provides additional implicit constraints and reduces the searching space of the target classifier, which requires fewer queries and therefore accelerates the training process. This is achieved by imposing a penalty on the incompatibility of unlabeled pairs with the metric through the augmented labels $\widetilde{\mathcal{S}}$ and $\widetilde{\mathcal{D}}$ in (4). The proposed method is able to minimize both the classification error and the incompatibility simultaneously.

Specifically, the loss function in (1) can be written as

$$\frac{1}{|\mathcal{S}|+|\mathcal{D}|}\sum_{i,j\in\mathcal{S}\cup\mathcal{D}} \mathbb{1}\{y_{ij}=0\}(1-\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_A) + \mathbb{1}\{y_{ij}=1\}(\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_A^2-1),$$

and is a surrogate function to

$$\widehat{e}(h_A)=\frac{1}{|\mathcal{S}|+|\mathcal{D}|}\sum_{i,j\in\mathcal{S}\cup\mathcal{D}}(2y_{ij}-1)h_A(\boldsymbol{x}_i,\boldsymbol{x}_j),$$

where $h_A(\boldsymbol{x}_i,\boldsymbol{x}_j) = \text{sign}(\{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_A^2-1\})$. We denote the joint distribution of $(\boldsymbol{x}_i,\boldsymbol{x}_j)$ as F, then $\widehat{e}(h_A)$ is the empirical estimator of the error rate for the labeled data $e(h_A) = P_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\sim F}(h_A(\boldsymbol{x}_i,\boldsymbol{x}_j) \neq 2y_{ij}-1)$. In addition, we define the *incompatibility* between h_A and F as

$$e_u(h_A) = E_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\sim F}\,\chi(h_A,\boldsymbol{x}_i,\boldsymbol{x}_j), \tag{11}$$

where $\chi(h_A,\boldsymbol{x}_i,\boldsymbol{x}_j) = P(\ell_i=\ell_j)\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_A^2 - P(\ell_i\neq\ell_j)\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_A$. Intuitively, e_u measures the average proximity among data pairs weighted by the probability of being from the same cluster under the metric A, and e_u is small if the metric captures the important features. The empirical estimator of e_u is

$$\widehat{e}_u(h_A)=\frac{2}{n(n-1)}\sum_{i,j} w_{ij}\mathbb{1}\{\widehat{\boldsymbol{h}}_i^\top\widehat{\boldsymbol{h}}_j-1/K>0\}\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_A^2 - w_{ij}\mathbb{1}\{\widehat{\boldsymbol{h}}_i^\top\widehat{\boldsymbol{h}}_j-1/K<0\}\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_A,$$

where w_{ij} and $\widehat{\boldsymbol{h}}_i$ are defined as in (3). Then the proposed augmented metric learning (4) is equivalent to minimizing both $\widehat{e}(h_A)$ and $\widehat{e}_u(h_A)$ at the same time.

Furthermore, we denote the function space of h_A as $\mathcal{C} = \{h_A : A \in \mathbb{R}^{p\times p} \text{ and } A \text{ is semidefinite}\}$. To quantify the complexity of $\mathcal{C}$ regarding the binary classification task for data sampled from F, we can draw a sample of size n independently from F and classify it with the functions in $\mathcal{C}$. The expected number of label assignments that can be correctly classified is denoted as $S_F^{\mathcal{C}}(n)$. Note that $S_F^{\mathcal{C}}(n)$ is a distribution-dependent complexity measure of $\mathcal{C}$. In general, a larger $S_F^{\mathcal{C}}(n)$ implies a larger function space $\mathcal{C}$. In addition, we let $\mathcal{C}_\chi(\tau) = \{h_A\in\mathcal{C} : e_u(h_A)\le\tau\}$ be the function space whose incompatibility with F as defined in (11) is bounded by τ, where τ is a positive constant.

In the following theorem, we show the classification accuracy achieved by the proposed query-augmented metric learning method given the increasing number of pairwise constraints.

Theorem 1. Given any $\epsilon, s, \tau > 0$, and the number of pairwise constraints $n_l = |\mathcal{S}| + |\mathcal{D}|$, assume that

$$\frac{n(n-1)}{2} - n_l = \mathcal{O}\left(\frac{p+1}{\epsilon^2}\log\frac{1}{\epsilon}+\frac{1}{\epsilon^2}\log\frac{2}{\delta}\right),$$

where p is the dimension of A and

$$\delta = 8S_F^{\mathcal{C}_\chi(\tau+2\epsilon)}(2n_l)\exp\left(-\frac{1}{2}\epsilon n_l\right).$$

Then for all $h_A\in\mathcal{C}$ with $\widehat{e}(h_A)\le s$ and $\widehat{e}_u(h_A)\le\tau+\epsilon$, we have $P(e(h_A)\le s+\epsilon)\ge 1-\delta$.

Since $\mathcal{C}_\chi(\tau+2\epsilon)$ is a subset of $\mathcal{C}$, we have

$$S_F^{\mathcal{C}_\chi(\tau+2\epsilon)}(2n_l)\le S_F^{\mathcal{C}_\chi}(2n_l)\le\left(\frac{2en_l}{p+1}\right)^{p+1}, \tag{12}$$

if $2n_l > p+1$. Therefore, δ converges to 0 as n_l increases to infinity, indicating that the classification accuracy using the learned metric converges to the optimal accuracy of $h_A\in\mathcal{C}$ with a probability approaching 1. The second inequality in (12) is derived from the relation between the growth function and the VC dimension (Vapnik 2013), where the growth function is the supremum of $S_F^{\mathcal{C}}(n)$ among all F's and the VC dimension equals $p+1$ in our case. In addition, note that by using the labeled data only, the convergence rate is

$$\delta_0 = 4S_F^{\mathcal{C}}(2n_l)\exp\left(-\frac{1}{2}\epsilon n_l\right).$$

Thus, we have $\delta < \delta_0$ if $S_F^{\mathcal{C}_\chi(\tau+2\epsilon)}(2n_l) < \frac{1}{2}S_F^{\mathcal{C}}(2n_l)$, which can be satisfied if the label of at least one pair from $\mathcal{U}$ can be augmented correctly. With this additional condition, we are able to achieve a faster convergence rate by incorporating the unlabeled pairs with the augmented information.

We remark that Theorem 1 can be generalized when $\widehat{\boldsymbol{h}}$ is a local minimum of (3) instead of the global minimum. In fact, $\widehat{\boldsymbol{h}}_i$ is only involved in the empirical incompatibility $\widehat{e}_u(h_A)$, which needs to satisfy $\widehat{e}_u(h_A)\le\tau+\epsilon$. We only require $\widehat{\boldsymbol{h}}_i$ to be close enough to the global optimizer of (3) such that the difference between $\widehat{e}_u(h_A)$ and its global minimum is of order $\mathcal{O}(\tau)$. On the other hand, the optimization accuracy affects the search space size of the metric matrix A through $S_F^{\mathcal{C}_\chi(\tau+2\epsilon)}(2n_l)$, which is smaller when τ decreases. Under the condition that $\tau=\mathcal{O}(\epsilon)$ as the number of labeled pairs increases, that is, $n_l\to\infty$, the tail probability bound of mislabeling $P(e(h_A)>\epsilon)$ would still be of the same order as the one when we obtain the global minimum. In practice, several numerical approaches can be implemented to find possible global minima or good local minima of (3),

such as the branch-and-bound technique (Land and Doig 2010; Clausen 1999) or using multiple random start points.

Next, we show an improvement on using the proposed active query strategy compared with random selection. In the following, we denote the posteriors of the membership assignment from the random query and active query after acquiring the membership of one extra instance at the tth step as ρ^t_{random} and ρ^t_{active}, respectively, and denote the underlying distribution ρ_* as in Section 3.2. We denote the size of the clusters as $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$, and the size of the neighborhoods at the tth step as $\boldsymbol{\beta}^t = (\beta^t_1, \ldots, \beta^t_{L^t})$. Note that $\boldsymbol{\alpha}$ does not change with t and $\sum_i^K \alpha_i = n$. Without loss of generality, we assume $L^t = K$ in the following; that is, there is one and only one neighborhood in each cluster, which can be achieved when T is sufficiently large. We also denote the minimum of the distance between underlying cluster centers as $\underline{r}$; that is, $\underline{r} = \min_{i,j} \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_A$. We formulate the constrained clustering process from a Bayesian perspective, in that the constraints are added as a prior in the form of (2).

Furthermore, we assume that $\boldsymbol{x}_i$ follows a multivariate generalized Gaussian distribution (Dytso et al. 2018) given the cluster membership ℓ_i; that is,

$$P(\boldsymbol{x}_i|\ell_i = k) \propto \exp\left\{-\left(\frac{1}{\sigma}\|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|_A\right)^d\right\}, \tag{13}$$

where d is a positive constant, $\boldsymbol{\mu}_k$ is the center of the kth cluster and $\sigma^2 = \mathcal{O}\{\text{var}(X_i^2|\ell_i = k)\}$.

The generalized Gaussian distribution assumption specifies a unimodal cluster structure, which is common in the analyses of probabilistic clustering models (Lu and Leen 2007; Marlin et al. 2012; Rodrigues and Engel 2014). Moreover, the generalized Gaussian distribution includes a broad category of distributions, for example, the Laplace distribution for $d = 1$ and the Gaussian distribution for $d = 2$, as well as distributions with different tail behaviors, such as, the super-Gaussian for $d < 2$ and the sub-Gaussian for $d > 2$. In addition, asymmetric clusters are also applicable if they can be transformed to a symmetric distribution.

Theorem 2. Given the neighborhoods at the tth step $\mathcal{N}^t$ and $n > |\mathcal{N}^t|$, under the generalized Gaussian distribution assumption (13), we have

$$KL(\rho_*||\rho^t_{\text{active}}) \le KL(\rho_*||\rho^t_{\text{random}}),$$

with probability at least $1 - \epsilon$, where

$$\epsilon = \mathcal{O}_p\left[n \log K \exp\left\{-n\left(\frac{\underline{r}}{\sigma}\right)^d\right\}\left\{\xi(\boldsymbol{\beta}^t) - \xi(\boldsymbol{\alpha})\right\}\right], \tag{14}$$

with

$$\xi(\boldsymbol{x}) = \left\{\sum_{\substack{i,j \\ i\neq j}}^{K} x_i \exp\left(-x_i - x_j\right)\right\}^n.$$

Theorem 2 implies that the discrepancy between the underlying true distribution and the updated posterior distribution is smaller based on the active query strategy than random selection with a probability close to 1. The tail probability ϵ in (14) depends on the sample size n, neighborhood size $\boldsymbol{\beta}^t$ and the underlying cluster structure. In particular, ϵ converges to 0 as n increases if $\sum_{\substack{i,j \\ i\neq j}}^{K} \beta^t_i \exp\left(-\beta^t_i - \beta^t_j\right) < 1$, which can be satisfied when the sample size in each neighborhood is large enough such that $\exp(2\min_i \beta^t_i)/(\min_i \beta^t_i) \ge \frac{K(K-1)}{2}$. Furthermore, ϵ decreases exponentially with n in a convergence rate bounded by $\min_i \beta^t_i$ and $\max_i \beta^t_i$, since $K^{2n}(\max_i \beta^t_i)^n \exp(-2n\max_i \beta^t_i) \le \xi(\boldsymbol{\beta}^t) \le K^{2n}(\min_i \beta^t_i)^n \exp(-2n\min_i \beta^t_i)$. When n is fixed, ϵ can still converge to 0 if we add pairwise constraints and decrease the value of $\xi(\boldsymbol{\beta}^t) - \xi(\boldsymbol{\alpha})$. Notice that $\beta_i < \alpha_i$ for any $i = 1, \ldots, K$, therefore, we have $\xi(\boldsymbol{\beta}^t) - \xi(\boldsymbol{\alpha}) \ge 0$, and $\xi(\boldsymbol{\beta}^t) - \xi(\boldsymbol{\alpha}) = 0$ when all pairs are queried. In addition, ϵ decreases if the clusters are more separable (i.e., $(\underline{r}/\sigma)^d$ is larger). Here $(\underline{r}/\sigma)^d$ can be interpreted as the signal-to-noise ratio in the clustering task, where $\underline{r}$ measures the closeness among clusters, while σ and d measure the density of data points around the cluster center.

To better illustrate the magnitude of the tail probability, we consider the balanced cluster and balanced neighborhood case

Corollary 1. Assume the conditions for Theorem 2 hold and $\beta_1 = \cdots = \beta_K = \beta, \alpha_1 = \cdots = \alpha_K = \alpha$, then

$$KL(\rho_*||\rho^t_{\text{active}}) \le KL(\rho_*||\rho^t_{\text{random}}),$$

with probability at least $1 - \epsilon$, where

$$\epsilon = \mathcal{O}_p\left\{nK^{2n}\log K \exp\left[-n\left(\frac{\underline{r}}{\sigma}\right)^d\right]\left[\beta^n \exp(-2\beta n) - \alpha^n \exp(-2\alpha n)\right]\right\}. \tag{15}$$

The probability (15) shows that ϵ converges to 0 exponentially as n grows if $\exp(2\beta)/\beta > K^2$, which indicates that more pairwise constraints are needed to ensure convergence when there are more clusters. In addition, the convergence of ϵ is faster when β is larger. That is, as the number of queried constraints increases, we are more confident that the proposed active query strategy selects more important pairs than a passive strategy would.

Theorem 2 compares the proposed method with the random query in one single step. In the following theorem, we compare the proposed method with a non-random selection strategy after T queries.

We define a nonrandom selection strategy as follows. In Step 1, we perform the K-means clustering. In Step 2, we select the top T most uncertain pairs to query based on the clustering result in Step 1. That is, we select the T pairs with similarity probability closest to 0.5. The above query method does not involve a sequential query procedure and can be completed in two steps. Therefore, we refer to this simple nonrandom selection method as the *two-step* strategy, and denote the posterior distribution of the cluster label at the tth step using the two-step strategy as $\rho^t_{ts}(\boldsymbol{\ell}|\boldsymbol{x})$, accordingly. The comparison between the two-step strategy and the proposed strategy is established in Theorem 3.

Theorem 3. Under the generalized Gaussian distribution assumption in (13), for $K^2 < T < Kn$, we have

$$E\left\{KL(\rho_*||\rho^T_{\text{active}}) - KL(\rho_*||\rho^T_{ts})\right\} \le C_1\sqrt{T} - C_2T(1 - e^{-\frac{2T}{K^2}}), \tag{16}$$

where $C_1 = K\sigma/\sqrt{\underline{r}}$, $C_2 = e^{-(\underline{r}/\sigma)^d}(K-1)/K$, and $\underline{r} = \min_{i,j}\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_A$.

Here, we require that $T < Kn$ to ensure that not all the pairs have been labeled in the proposed method. Otherwise, $\rho^t_{\text{active}} = \rho_*$ and $E\left\{KL(\rho_*||\rho^T_{\text{active}}) - KL(\rho_*||\rho^T_{ts})\right\} = -E\left\{KL(\rho_*||\rho^T_{ts})\right\} < 0$.

Theorem 3 compares the Kullback–Leibler divergence of the two methods after T queries, cumulatively. The result of Theorem 3 implies that, with a sufficiently large budget $T > K^2$, we have $E\{KL(\rho_*||\rho^T_{\text{active}})\} < E\{KL(\rho_*||\rho^T_{ts})\}$; that is, the posterior distribution of the label $\boldsymbol{l}$ using the proposed active query strategy is expected to be closer to ρ_* than the two-step method. Moreover, the upper bound in (16) indicates that the discrepancy between the proposed method and the two-step method increases as T grows. Meanwhile, this discrepancy also depends on the variance σ^2, the cluster number K and the minimum center distance $\underline{r}$. The result in (16) demonstrates that the gain of the proposed method is more significant with a larger signal-to-noise ratio $\underline{r}/\sigma$ and a smaller K.

We remark that the result in (16) can also be interpreted as the discrepancy of the prior information gain from the labeled data pairs between the two methods: the cumulative information of the two-step grows in $\mathcal{O}(\sqrt{T})$, while the cumulative information of the proposed method grows in $\mathcal{O}(T)$. This is because the two-step method evaluates the informativeness of pairs based on the initial clustering result only, which might not be accurate or effective for the subsequent query steps. Consequently, the information gain of each selected pair may decrease quickly as querying proceeds. In contrast, the proposed method updates the information criterion at each step, and selects the pairs which benefit the posterior distribution at the current step. This sequential updating strategy mitigates the loss in information gain of queried pairs in the subsequent steps, which also facilitates the accumulation of prior information to increase in a linear order. Moreover, the proposed method incorporates more labeled pairs than the two-step method by utilizing the neighborhood structure and the transitivity property, which further enhances cumulative information gain.

The proofs of the theoretical results in this section are provided in the supplementary material.

6. Simulations

In this section, we illustrate the advantages of the proposed metric learning method and the active query strategy through simulation settings.

6.1. The Advantages of Incorporating Augmented Constraints

We first demonstrate the advantage of incorporating the augmented constraints (4). The data points are generated as $\boldsymbol{x}^\top = \left\{(\boldsymbol{x}^{(1)})^\top, (\boldsymbol{x}^{(2)})^\top\right\}$, where $\boldsymbol{x}^{(1)} \in \mathbb{R}^{p_1}$ includes the significant features for determining the cluster memberships, and $\boldsymbol{x}^{(2)} \in \mathbb{R}^{p_2}$ include irrelevant features.

Specifically, $\boldsymbol{x}^{(i)}$ is sampled from a Gaussian mixture model, that is, $\boldsymbol{x}^{(i)}|z^{(i)} \sim \mathcal{N}(\boldsymbol{\mu}^{(i)}_{z^{(i)}}, \boldsymbol{I}_{p_i})$, for $i = 1, 2$, where $z^{(i)}$ is uniformly sampled from $\{1, \ldots, p_i\}$, and $\boldsymbol{\mu}^{(i)}_{z^{(i)}} = c\left(\mathbb{1}\left\{z^{(i)} = 1\right\}, \mathbb{1}\left\{z^{(i)} = 2\right\}, \ldots, \mathbb{1}\left\{z^{(i)} = p_i\right\}\right)$; that is, all elements are zero except that the $z^{(i)}$th element equals c. Here, $c > 0$ denotes the distance between the center of the clusters and the origin. A larger c implies a more separable cluster structure.

We let the cluster label $\ell = z^{(1)}$ and the number of clusters $K = p_1$, so the cluster memberships are fully determined by the first p_1 features. An illustration of the simulation data with $K = p_1 = p_2 = 3$ is shown in Figure 1.

In this experiment, we select $p_1 = 6$, $p_2 = 3$, $c = 5, n = 60$ and $K = 6$. We train the metric matrix A with randomly selected pairwise constraints and compare the clustering performance with or without augmented constraints $\widetilde{\mathcal{S}}$ and $\widetilde{\mathcal{D}}$ from (4), which is evaluated by the Adjusted Random Index (ARI) (Rand 1971). A higher ARI indicates a clustering result more consistent with the true cluster memberships. Figure 2 shows the boxplots of ARI with different numbers of pairwise constraints, which demonstrates that incorporating the augmented constraints consistently improves the clustering performance under varying numbers of queried constraints. The advantage is more obvious when the number of constraints is large, since more entries in the similarity matrix Y are labeled and the augmented constraints are more accurate. In addition, the trend that ARI increases as the number of constraints grows is more stable with the augmented constraints compared with its counterpart, indicating that the proposed method also increases robustness in clustering. One possible reason is that the proposed method uses all $n(n-1)/2$ pairs during metric training instead of selected constraints only, which alleviates randomness and avoids overfitting labeled pairs.

6.2. Active Clustering With Low Signal-to-Noise Ratio

Another novelty of the proposed method is performing feature selection in the process of active clustering. In this simulation, we demonstrate this advantage in a low signal-to-noise ratio setting where the number of irrelevant features is much larger than the true features. We adopt the data generating procedure in Section 6.1, except letting $P(\mu^{(2)}_j = c\mathbb{1}\{j = z^{(2)}_i\}) = P(\mu^{(2)}_j = -c\mathbb{1}\{j = z^{(2)}_i\}) = \frac{1}{2}$ to make clustering more difficult in that the irrelevant features are well-separated. Therefore, clustering without identifying the true features is likely to underperform in this case.

We compare the proposed method with other popular active semi-supervised clustering methods under the setting of $p_1 = 5$, $p_2 = 30$, $c = 3$, $K = 5$ and $p_1 = 10$, $p_2 = 30$, $c = 3$, $K = 10$, respectively. In each case, we generate $n = 300$ samples which are evenly distributed sampled from K clusters. In addition, we let the penalty parameter $\lambda = 0.5$ and the number of penalized features $q = p_2/2$.

The competing methods include constrained Kmeans (COP-Kmeans) (Wagstaff et al. 2001), pairwise constrained Kmeans

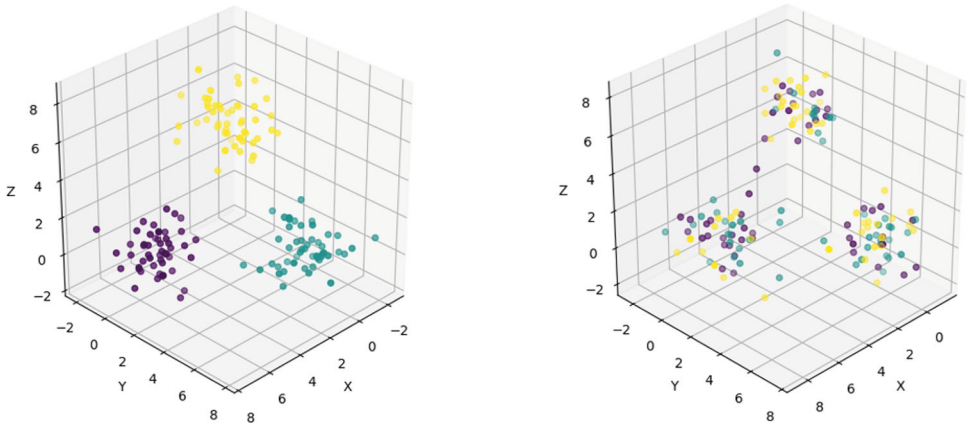


Figure 1. Illustration of the simulated data with $K = p_1 = p_2 = 3$, showing the first three dimensions (*left*) and the last three dimensions (*right*). The cluster membership is determined by the first three dimensions, illustrated by different colors.

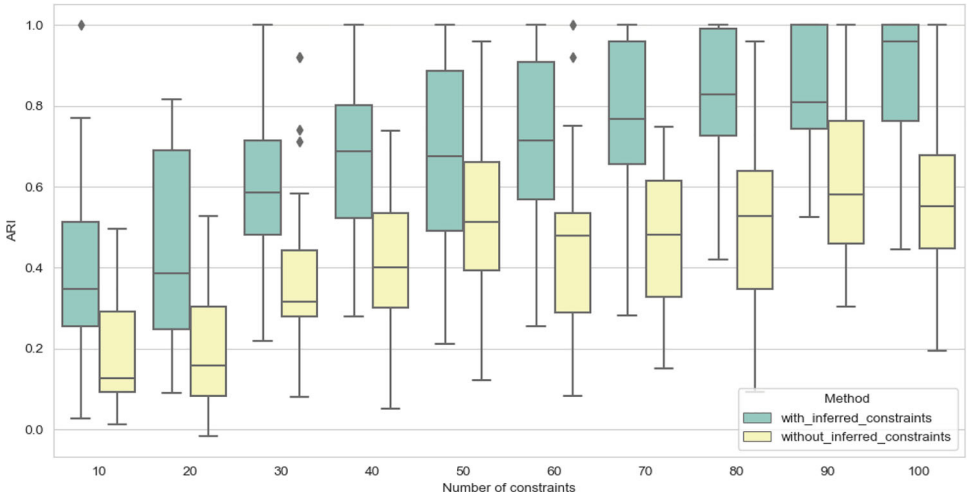


Figure 2. The ARI comparisons of the simulation setting with $p_1 = 6, p_2 = 3$ and $c = 5$ using random queries with (*green*), and without (*yellow*) augmented constraints, based on 30 replications for each number of constraints.

(PCKmeans) (Basu, Banerjee, and Mooney 2004), metric pairwise constrained Kmeans (MPCKmeans) (Bilenko, Basu, and Mooney 2004), Constraint-based Repeated Aggregation (COBRA) (Van Craenendonck, Dumancic, and Blockeel 2018a) , and the Constraint-based Repeated Aggregation and Splitting (COBRAS) (Van Craenendonck et al. 2018b). Among these methods, COP-Kmeans, PCKmeans, and MPCKmeans are centroid-based clustering algorithms, and COP-Kmeans and PCKmeans do not involve metric learning. The aforementioned methods were originally designed for one-time-selected pairwise constraints. To make a fair comparison, we combined these three methods with the NPU active query strategy implemented by Švehla (2018). On the other hand, COBRA is a model-free hierarchical clustering method, by first preclustering the instances into several local neighborhoods called *super-instances*, and then further combining these super-instances based on pairwise constraints. The budget of query is controlled indirectly through the initial number of super-instances. Finally, COBRAS extends COBRA by controlling the number of queries directly, and is not biased toward the ellipsoidal clusters. We implement the COBRA and COBRAS with packages provided by Craenendonck (2017) and Craenendonck (2020), respectively.

To illustrate the improvement in our clustering performance from each section separately, we break down the proposed method into two parts: the Augmented Query Metric Learning Method (AQM), and the Minimum Expected Entropy (MEE) active query strategy. We denote the full implementation of the proposed method as AQM+MEE. In addition, we also provide the results of combining the proposed metric learning method with the competing active strategy NPU denoted as AQM+NPU.

Tables 1 and 2 present the average ARI and standard deviation based on 30 replications under different numbers of queries, implying that the combination of AQM+MEE method achieves the best clustering result when the signal-noise ratio is low, regardless of the number of queries. In particular, the proposed AQM method achieves the largest improvements on ARI by more than 50% compared with the MPCKmeans when both methods adopt the NPU strategy. In addition, the comparison between AQM+NPU and AQM+MEE shows that the proposed active strategy MEE can further enhance clustering efficiency, especially when $p_1 = 5$ and $p_2 = 30$. We also notice that for the methods without metric learning, namely PCKmeans, COPKmeans, COBRA and COBRAS, their performances are similar as they are not designed for extracting features through metric learning. However, although

Table 1. Comparisons on the simulation data with $p_1 = 5, p_2 = 30, c = 3$, and $K = 5$, showing average ARI of clustering with standard deviations.

Number of queries	60	120	180	240	300
PCKmeans + NPU	0.291(0.083)	0.347(0.100)	0.377(0.115)	0.407(0.112)	0.429(0.123)
COPKmeans + NPU	0.276(0.125)	0.369(0.147)	0.453(0.186)	0.552(0.198)	0.644(0.202)
MPCKmeans + NPU	0.275(0.115)	0.355(0.113)	0.434(0.131)	0.444(0.157)	0.529(0.157)
COBRAS	0.168(0.083)	0.250(0.083)	0.300(0.084)	0.348(0.077)	0.417(0.077)
AQM + NPU	0.443(0.128)	0.587(0.112)	0.688(0.090)	0.768(0.132)	0.860(0.151)
AQM + MEE	**0.474(0.124)**	**0.630(0.123)**	**0.725(0.147)**	**0.845(0.110)**	**0.921(0.078)**
Number of super instances	10	50	90	130	170
Number of queries	19.100(2.737)	101.900(14.614)	179.533(20.713)	250.000(24.220)	321.933(28.578)
ARI of COBRA	0.174(0.065)	0.275(0.070)	0.365(0.065)	0.473(0.057)	0.584(0.056)

NOTE: The second half of the table provides the results of COBRA.

Table 2. Comparisons on the simulation data with $p_1 = 10, p_2 = 30, c = 3$ and $K = 10$, showing average ARI of clustering with standard deviations.

Number of queries	60	120	180	240	300
PCKmeans + NPU	0.109(0.021)	0.131(0.034)	0.177(0.033)	0.200(0.049)	0.228(0.055)
COPKmeans + NPU	0.107(0.030)	0.122(0.031)	0.146(0.041)	0.174(0.048)	0.194(0.047)
MPCKmeans + NPU	0.086(0.024)	0.129(0.034)	0.164(0.040)	0.189(0.044)	0.233(0.042)
COBRAS	0.064(0.031)	0.089(0.035)	0.084(0.037)	0.140(0.051)	0.157(0.068)
AQM + NPU	0.119(0.040)	**0.209(0.050)**	**0.267(0.062)**	0.307(0.083)	0.350(0.082)
AQM + MEE	**0.131(0.049)**	0.203(0.055)	0.263(0.063)	**0.327(0.076)**	**0.350(0.090)**
Number of super instances	10	30	50	70	90
Number of queries	28.067(4.626)	116.433(13.728)	187.700(19.338)	259.633(20.180)	317.900(24.347)
ARI of COBRA	0.078(0.023)	0.107(0.029)	0.146(0.028)	0.188(0.024)	0.238(0.032)

NOTE: The second half of the table provides the results of COBRA.

Table 3. ARI comparisons with cluster center on sphere with $K = 5, p_1 = 100, p_2 = 400, r = 5$, and $n = 250$, showing the average ARI of clustering with standard deviations.

Number of queries	60	120	180	240	300
PCKmeans + NPU	**0.391(0.051)**	0.456(0.071)	0.575(0.074)	0.687(0.074)	0.819(0.103)
COPKmeans + NPU	0.326(0.027)	0.374(0.041)	0.490(0.059)	0.562(0.067)	0.673(0.074)
MPCKmeans + NPU	0.325(0.094)	0.435(0.077)	0.564(0.060)	0.713(0.056)	0.824(0.049)
COBRAS	0.313(0.042)	0.349(0.035)	0.367(0.044)	0.438(0.037)	0.501(0.053)
AQM + MEE	0.384(0.045)	**0.489(0.099)**	**0.595(0.097)**	**0.773(0.037)**	**0.906(0.083)**
Number of super instances	30	70	90	110	150
Number of queries	55.400(5.389)	141.300(5.675)	180.200(12.983)	222.500(12.902)	320.000(9.571)
ARI of COBRA	0.248(0.017)	0.351(0.023)	0.396(0.016)	0.454(0.015)	0.597(0.020)

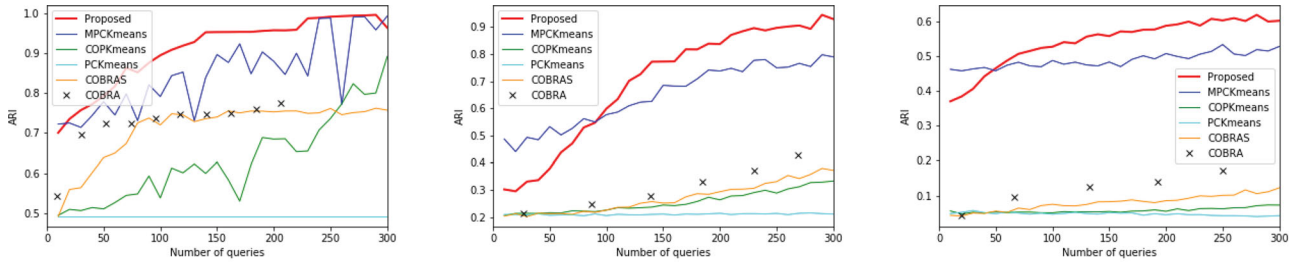
MPCKmeans involves metric learning, its accuracy is still relatively low since it does not exclude the irrelevant features. In contrast, the proposed methods improves the clustering accuracy significantly in both simulation settings, indicating the effectiveness of both AQM and MEE for clustering tasks actively.

In addition, we compare the proposed method and the competing methods under a high-dimensional setting. Specifically, we sample the relevant features from a K-mode Gaussian mixture model, where the K centers are located on a $p_1 - 1$ sphere with a radius r, such that the kth center is located at $r(\cos(\phi_1^i), \sin(\phi_1^i)\cos(\phi_2^i), ..., \sin(\phi_1^i)\sin(\phi_{p-2}^i)\cos(\phi_{p-1}^i), \sin(\phi_1^i)\sin(\phi_{p-2}^i)\sin(\phi_{p-1}^i))$, where $\phi_1^i = \cdots \phi_{p-2}^i = i\pi/K$ and $\phi_{p-1}^i = 2i\pi/K$. We let $K = 5$, $p_1 = 100$, $p_2 = 400$, $r = 5$, and generate 250 samples in total. Table 3 provides the adjusted random index (ARI) of AQM+MEE and the competing methods with different numbers of queries. The new simulation results show that the proposed method still achieves the highest clustering accuracy, under a higher dimension of p. This simulation also affirms that the proposed method is applicable under high-dimensional cases, and can adapt to different clustering center distributions.

7. Real Data

We apply the proposed method on three real datasets with high dimensional features. The first dataset is the breast cancer diagnostic data from the UCI machine learning repository (Dua and Graff 2017). The dataset contains 569 samples with 30 features extracted from the diagnostic images of a breast mass, which are labeled as either malignant or benign. The second dataset is the MEU-Mobile dataset which records 71 keystroke features of phone users, including finger area, pressure and hold time, etc. We use a subset of the keystroke data from 9 users. Each user repeats typing the same password 51 times, so there are 459 samples in total. The third dataset is the urban land cover dataset which contains 675 multi-scale remote sensing images. For each resolution scale, 21 features are measured, including area, brightness, asymmetry, etc. These features are repeatedly constructed for the same image under 7 different resolutions, resulting in 147 features in total. Based on the extracted features, the images of the third dataset are categorized into 9 urban land cover classes including trees, grass, soil, concrete, asphalt, buildings, cars, pools and shadows.

The goal of our study is to cluster the datasets with sequentially queried pairwise constraints while identifying important



(a) Breast cancer (b) MEU-Mobile (c) Urban land cover

Figure 3. Performance comparison on three real datasets, showing average ARI against number of constraints based on 30 replications. (The three competing methods MPCKmeans, COPKmeans and PCKmeans are combined with NPU strategy.)

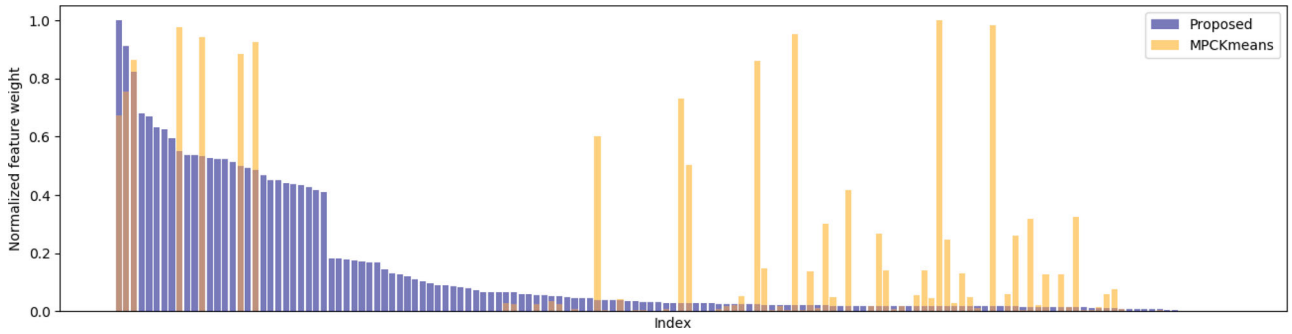


Figure 4. The estimated weights of 147 features from the urban land cover dataset, showing the significant features identified by the proposed method compared with MPCKmeans.

features. In our implementation, we cluster the three datasets into 2, 9, and 9 categories, respectively. The category labels in the raw datasets are used only in determining the similarity of the queried instance pairs, and in verifying the accuracy of the clustering outcomes through ARI.

Figure 3 provides the ARI comparison between the proposed method (AQM+MEE) and the competing methods. The proposed method has the overall best performance in all three datasets in terms of the average ARI with varying number of constraints. The most competitive method is the MPCKmeans, as it also applies metric learning to extract the important features, while the other competing approaches, that is, PCKmeans, COPKmeans and COBRA do not. For the breast cancer data, although the proposed method reaches a similar accuracy as the MPCKmeans when the number of pairwise constraints is large, the proposed method has a higher ARI when the number of constraints is relatively small due to the higher efficiency in utilizing constraints. In particular, the proposed method achieves an ARI of 0.928 with 80 queries while the MPCKmeans achieves only 0.732 with the same number of queries. For the MEU-Mobile data and urban land cover data, the proposed method improves the average ARI by 20% and 17%, respectively, compared with the MPCKmeans given 300 queries. In addition, the ARI of MPCKmeans and COPKmeans fluctuate significantly in clustering breast cancer data as the number of queries grows. In contrast, the proposed method leads to a more stable clustering with consistently increasing ARI when the query process continues. One possible reason is that the proposed method removes the irrelevant features through selecting penalty functions. Another possible reason is that the proposed MEE strategy selects the unlabeled pairs which consistently contribute to clustering by evaluating the information gain of the selected queries more accurately compared with the NPU method.

Furthermore, we investigate the interpretability of the features selected by the proposed method via the example of the urban land cover data. We compare the significance of each feature by plotting the diagonal entries of the estimated metric matrix A in a decreasing order as shown in Figure 4. The barplot of Figure 4 implies that the proposed method divides the feature into 3 groups. The first 3 largest coefficients correspond to the 19th, 40th, and 61st features in the original data. These three features are associated with the Normalized Difference Vegetation Index (NDVI) on three different resolution scales, respectively, which implies that NDVI is a crucial factor identified by the proposed method in determining the image category. The second group consists of 25 features, and the third group consists of the rest of the features which are less important in image clustering. In contrast, the MPCKmeans tends to assign high importance to the irrelevant features and the selected features are not consistent across different resolution scales.

To verify the importance of the selected features, we perform the Kmeans with the entire 147 features, the first 3 features and the 28 features in the first two groups, respectively, without imposing any pairwise constraints. With all 147 features, the Kmeans returns an ARI of 0.03. In contrast, with the first three features only, the ARI increases significantly to 0.29. With the selected 28 features, the ARI further increases to 0.34. The above results imply that the first three features extracted by the proposed method play an essential role in determining the category of the sampled remote sensing image. Although involving less important features from the second group can improve the performance, the improvement is quite negligible. On the other hand, including more features brings more irrelevant informa-

tion in measuring the similarity between two images, leading to a noisy metric space with more difficulties in clustering. The real data analyses confirm that the proposed method is able to identify the low-dimensional feature space which is highly correlated to clustering analyses.

8. Discussion

In this article, we propose an active clustering method with metric learning. The article has three main contributions: First, we augment the queried instance-level similarity by generalizing the pairwise constraints using the cluster structure that is typically ignored in the existing metric learning methods. Second, we improve the robustness of the metric learning process by selectively penalizing the potentially irrelevant features based on history training results. Third, we propose a new active query strategy based on the expected entropy change, which makes a more accurate evaluation of the information gain from a query. We also investigate the theoretical properties of the proposed approach, especially on the advantage between the active query strategy over random selection from the perspective of the posterior distribution of the cluster membership, which has not been studied in the existing literature to the best of our knowledge. Finally, we demonstrate the efficacy of the proposed method through simulation studies and real data applications in breast cancer diagnosis, keystroke recognition and multi-scale remote sensing images.

The proposed framework can be extended to online training in that both constraint augmentation and metric aggregation can be adapted into a progressive method without retraining at each step when new constraints are added, which can improve computation efficiency. In addition, the proposed method can be generalized to fit the non-ellipsoidal clusters through nonlinear feature transformation, such as the kernel method (Anand et al. 2014; Abin and Beigy 2015), or adopting other constrained clustering methods instead of PCKmeans, for example, spectral clustering methods (Wang and Davidson 2010; Huang, Chuang, and Chen 2012) (See supplementary material for details). On the other hand, the theoretical properties indicate that metric learning and active query can be interpreted as optimizing the likelihood function and the prior function sequentially regarding cluster membership distribution, respectively. Therefore, one future research direction is to develop a unified framework by quantifying the randomness of metric learning in the active query process. Another potential research direction is to extend the current method under the setting of the model-free constraints generating process using deep learning tools such as generative adversarial networks (GAN) (Goodfellow et al. 2020).

Supplementary Materials

The online supplement contains the ADMM algorithm to solve (3), all technical proofs and additional numerical results. The code is available at *https://github.com/dyj9999/query_augmented_active_metric_learning*.

Acknowledgments

The authors are grateful to reviewers, the associate editor and the editor for their insightful comments and suggestions which have improved the article significantly.

References

Abin, A. A., and Beigy, H. (2015), "Active Constrained Fuzzy Clustering: A Multiple Kernels Learning Approach," *Pattern Recognition*, 48, 953–967. [1874]

Anand, S., Mittal, S., Tuzel, O., and Meer, P. (2014), "Semi-Supervised Kernel Mean Shift Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36, 1201–1215. [1874]

Basu, S., Banerjee, A., and Mooney, R. J. (2004), "Active Semi-Supervision for Pairwise Constrained Clustering," in *Proceedings of the 2004 SIAM International Conference on Data Mining*, pp. 333–344. Society for Industrial and Applied Mathematics. [1862,1864,1866,1871]

Basu, S., Bilenko, M., Banerjee, A., and Mooney, R. J. (2006), "Probabilistic Semi-Supervised Clustering With Constraints," in *Semi-Supervised Learning*, eds. O. Chapelle, B. Scholkopf, and A. Zien, pp. 73–102. Cambridge: The MIT Press. [1862]

Bilenko, M., Basu, S., and Mooney, R. J. (2004), "Integrating Constraints and Metric Learning in Semi-Supervised Clustering," in *Twenty-first International Conference on Machine Learning - ICML'04*, p. 11, Banff, Alberta: ACM Press. [1863,1871]

Biswas, A., and Jacobs, D. (2014), "Active Image Clustering With Pairwise Constraints From Humans," *International Journal of Computer Vision*, 108, 133–147. [1863]

Breiman, L. (2003), "Rf/Tools: A Class of Two-Eyed Algorithms," in *SIAM Workshop*, pp. 1–56. [1867]

Caliński, T., and Harabasz, J. (1974), "A Dendrite Method for Cluster Analysis," *Communications in Statistics-Theory and Methods*, 3, 1–27. [1867]

Clausen, J. (1999), "Branch and Bound Algorithms-Principles and Examples," Department of Computer Science, University of Copenhagen, pp. 1–30. [1869]

Craenendonck, T. V. (2017), "Active Semi-supervised Clustering Algorithms for Scikit-learn," *https://bitbucket.org/toon_vc/cobra/src/master/*. [1871]

——— (2020), "Semi-Supervised Clustering With Cobras," available at: *https://github.com/ML-KULeuven/cobras*. [1871]

Davidson, I., Wagstaff, K. L., and Basu, S. (2006), "Measuring Constraint-Set Utility for Partitional Clustering Algorithms," in *European Conference on Principles of Data Mining and Knowledge Discovery*, eds. Fürnkranz, J., Scheffer, T., and Spiliopoulou, M., pp. 115–126. Berlin: Springer. [1862]

Devroye, L., Györfi, L., and Lugosi, G. (2013), *A Probabilistic Theory of Pattern Recognition* (Volume 31), Springer Science & Business Media, New York: Springer. [1868]

Dua, D., and Graff, C. (2017), "UCI Machine Learning Repository." [1872]

Dytso, A., Bustin, R., Poor, H. V., and Shamai, S. (2018), "Analytical Properties of Generalized Gaussian Distributions," *Journal of Statistical Distributions and Applications*, 5, 6. [1869]

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2020), "Generative Adversarial Networks," *Communications of the ACM*, 63, 139–144. [1874]

Greene, D., and Cunningham, P. (2007), "Constraint Selection by Committee: An Ensemble Approach to Identifying Informative Constraints for Semi-Supervised Clustering," in *Machine Learning: ECML 2007*, eds. J. N. Kok, J. Koronacki, R. L. d. Mantaras, S. Matwin, D. Mladenič, and A. Skowron, Vol. 4701, pp. 140–151. Berlin: Springer. [1863]

Grira, N., Crucianu, M., and Boujemaa, N. (2005), "Active Semi-Supervised Fuzzy Clustering for Image Database Categorization," in *Proceedings of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval - MIR'05*, p. 9, Singapore: Hilton, ACM Press. [1862,1864]

Hoi, S. C., Liu, W., and Chang, S.-F. (2010), "Semi-Supervised Distance Metric Learning for Collaborative Image Retrieval and Clustering," *ACM Transactions on Multimedia Computing, Communications, and Applications*, 6, 1–26. [1862]

Huang, H.-C., Chuang, Y.-Y., and Chen, C.-S. (2012), "Affinity Aggregation for Spectral Clustering," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 773–780. doi 10.1109/CVPR.2012.6247748. [1874]

Huang, Y., and Mitchell, T. M. (2006), "Text Clustering With Extended User Feedback," in *Proceedings of the 29th Annual International ACM*

SIGIR Conference on Research and Development in Information Retrieval - SIGIR '06, p. 413, Seattle, WA: ACM Press. [1862]

Land, A. H., and Doig, A. G. (2010), "An Automatic Method for Solving Discrete Programming Problems," in *50 Years of Integer Programming 1958–2008*, Heidelberg: Springer, pp. 105–132. [1869]

Liu, W., Ye, M., Wei, J., and Hu, X. (2017), "Fast Constrained Spectral Clustering and Cluster Ensemble With Random Projection," *Computational Intelligence and Neuroscience*, 2017, 1–14. [1862]

Lu, Z. (2007), "Semi-Supervised Clustering With Pairwise Constraints: A Discriminative Approach," in *Artificial Intelligence and Statistics* (Vol. 2), M. Meila and X. Shen, San Juan, Puerto Rico: PMLR, pp. 299–306. [1862,1864]

Lu, Z., and Leen, T. K. (2007), "Penalized Probabilistic Clustering," *Neural Computation*, 19, 1528–1567. [1869]

Mai, S. T., He, X., Hubig, N., Plant, C., and Böhm, C. (2013), "Active Density-Based Clustering," in *2013 IEEE 13th International Conference on Data Mining*, pp. 508–517. [1863]

Mallapragada, P. K., Jin, R., and Jain, A. K. (2008), "Active Query Selection for Semi-Supervised Clustering," in *2008 19th International Conference on Pattern Recognition*, pp. 1–4. [1862]

Marlin, B. M., Kale, D. C., Khemani, R. G., and Wetzel, R. C. (2012), "Unsupervised Pattern Discovery in Electronic Health Care Data Using Probabilistic Clustering Models," in *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pp. 389–398. [1869]

Niu, G., Dai, B., Yamada, M., and Sugiyama, M. (2011), "SERAPH: Semi-Supervised Metric Learning Paradigm With Hyper Sparsity," arXiv:1105.0167 [cs, stat]. arXiv: 1105.0167. [1862]

Rand, W. M. (1971), "Objective Criteria for the Evaluation of Clustering Methods," *Journal of the American Statistical Association*, 66, 846–850. [1870]

Rodrigues, T. F., and Engel, P. M. (2014), "Probabilistic Clustering and Classification for Textual Data: An Online and Incremental Approach," in *2014 Brazilian Conference on Intelligent Systems*, pp. 288–293. [1869]

Shi, T. and Horvath, S. (2006). "Unsupervised Learning with Random Forest Predictors," *Journal of Computational and Graphical Statistics*, 15, 118–138. [1866]

Tang, X., Xue, F., and Qu, A. (2020), "Individualized Multidirectional Variable Selection," *Journal of the American Statistical Association*, pp. 1–17, DOI: 10.1080/01621459.2019.1705308 [1864]

Van Craenendonck, T., Dumancic, S., and Blockeel, H. (2018a), "COBRA: A Fast and Simple Method for Active Clustering With Pairwise Constraints," arXiv: 1801.09955. [1863,1864,1871]

Van Craenendonck, T., Dumančić, S., Van Wolputte, E., and Blockeel, H. (2018b), "COBRAS: Interactive Clustering With Pairwise Queries," in *International Symposium on Intelligent Data Analysis*, pp. 353–366. Switzerland: Springer. [1871]

Vapnik, V. (2013), *The Nature of Statistical Learning Theory*, Springer Science & Business Media, New York, NY. [1868]

Švehla, J. (2018), "Active Semi-Supervised Clustering Algorithms for Scikit-Learn." Available at: *https://github.com/datamole-ai/active-semi-supervised-clustering*. [1871]

Wagstaff, K., Cardie, C., Rogers, S., and Schrödl, S. (2001), "Constrained k-Means Clustering With Background Knowledge," in *ICML* (Vol. 1), pp. 577–584. [1862,1864,1870]

Wang, X., and Davidson, I. (2010), "Active Spectral Clustering," in *2010 IEEE International Conference on Data Mining*, pp. 561–568. [1874]

Xing, E. P., Jordan, M. I., Russell, S. J., and Ng, A. Y. (2003), "Distance Metric Learning With Application to Clustering With Side-Information," in *Advances in Neural Information Processing Systems*, pp. 521–528. [1862,1864]

Xiong, S., Azimi, J., and Fern, X. Z. (2013). "Active Learning of Constraints for Semi-supervised Clustering," *IEEE Transactions on Knowledge and Data Engineering*, 26, 43–54. [1862,1863,1864,1865,1866]

Yang, L., Jin, R., and Sukthankar, R. (2012), "Bayesian Active Distance Metric Learning." arXiv:1206.5283. [1862,1863]