MC-GTA: Metric-Constrained Model-Based Clustering using Goodness-of-fit Tests with Autocorrelations

Zhangyu Wang [1] **Gengchen Mai** [2 3] **Krzysztof Janowicz** [4 1] **Ni Lao** [5]

Abstract

A wide range of (multivariate) temporal (1D) and spatial (2D) data analysis tasks, such as grouping vehicle sensor trajectories, can be formulated as clustering with given metric constraints. Existing metric-constrained clustering algorithms overlook the rich correlation between feature similarity and metric distance, i.e., metric autocorrelation. The model-based variations of these clustering algorithms (e.g. TICC and STICC) achieve SOTA performance, yet suffer from computational instability and complexity by using a metric-constrained Expectation-Maximization procedure. In order to address these two problems, we propose a novel clustering algorithm, MC-GTA (**M**odel-based **C**lustering via **G**oodness-of-fit **T**ests with **A**utocorrelations). Its objective is only composed of pairwise weighted sums of feature similarity terms (square Wasserstein-2 distance) and metric autocorrelation terms (a novel multivariate generalization of classic semivariogram). We show that MC-GTA is effectively minimizing the total hinge loss for intra-cluster observation pairs not passing goodness-of-fit tests, i.e., statistically not originating from the same distribution. Experiments on 1D/2D synthetic and real-world datasets demonstrate that MC-GTA successfully incorporates metric autocorrelation. It outperforms strong baselines by large margins (up to 14.3% in ARI and 32.1% in NMI) with faster and stabler optimization (>10x speedup).

[1]Department of Geography, University of California Santa Barbara, CA, USA [2]Department of Geography, University of Georgia, GA, USA [3]SEAI Lab, Department of Geography and the Environment, University of Texas at Austin, TX, USA [4]Faculty of Geosciences, Geography and Astronomy, University of Vienna, Vienna, Austria [5]Google, Mountain View, CA, USA. Correspondence to: Zhangyu Wang <zhangyuwang@ucsb.edu>, Gengchen Mai <gengchen.mai@austin.utexas.edu>, Krzysztof Janowicz <krzysztof.janowicz@univie.ac.at>, Ni Lao <noon99@gmail.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

1. Introduction

Clustering is one of the most fundamental problems in unsupervised learning, which deals with the data partitioning when ground-truth labels are unknown (Xu & Tian, 2015). Most existing clustering algorithms only consider the similarity among observations in the feature (attribute) space. However, in real-world applications, additional *metric constraints* (e.g., temporal continuity and geospatial proximity) often matter, especially in temporal and spatial data mining (Birant & Kut, 2007a; Hu et al., 2015; Mai et al., 2018; Belhadi et al., 2020). In other words, observations have both features and positions in a metric space. Hence, observations that are put in the same cluster should be similar in terms of features **and** their positions in the metric space (e.g., timestamps or geographic locations) should also satisfy some constraints. Metric constraints can be generalized to even higher dimensions as long as a meaningful distance measure is defined. This kind of problems is commonly known as *metric-constrained clustering* (Veldt et al., 2019). It is **not** a trivial task to design a larger composite space from these two spaces because they may have completely different metrics. For example, concatenating word embedding with geo-coordinates makes learning good similarity functions difficult because the former uses cosine distance while the latter uses Euclidean/geodesic distance, whose values can not be directly compared.

The current state-of-the-art metric-constrained clustering algorithms, namely TICC (Hallac et al., 2017) (for temporal clustering) and STICC (Kang et al., 2022) (for spatial clustering), consider both spaces by combining model-based clustering (Gormley et al., 2022) with a soft metric penalty $\arg\min_{\Theta,\mathcal{C}} \sum_{k=1}^{K} \Big[\| \lambda \circ \theta_{C_k} \|_1 + \sum_{X_i \in C_k} \big(-ll(X_i, \theta_{C_k}) + \beta \mathbb{1}\{\tilde{X}_i \notin C_k\} \big) \Big]$. Here K is the total number of clusters. X_i is one observation we need to assign to a cluster, and $\tilde{X}_i$ is the nearest neighbor of X_i in the metric space. θ_{C_k} are the estimated model parameters for cluster k, C_k is the set of observations of cluster k, $-ll(X_i, \theta_{C_k})$ is the negative log-likelihood of observation X_i belonging to cluster C_k given model parameter θ_{C_k}, and $\beta \mathbb{1}\{\tilde{X}_i \notin C_k\}$ is the soft metric penalty. λ is an L-1 normalization hyperparameter to prevent overfitting. The advantage of this

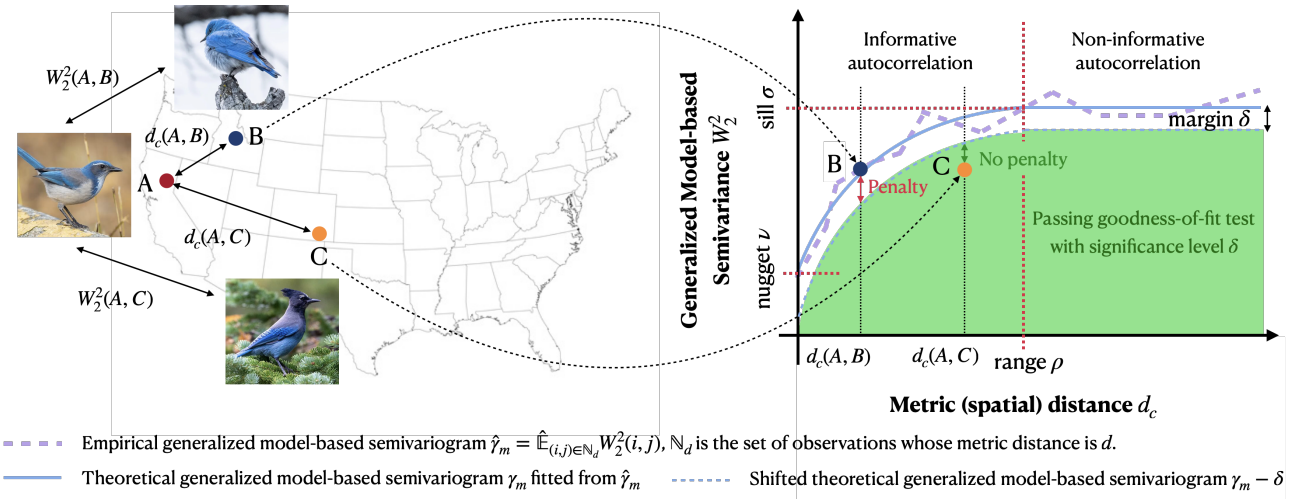


Figure 1. Motivation of MC-GTA using iNaturalist-2018 dataset as an example. We wish to cluster wild animal photos based on both their image similarity and spatial adjacency. For any pair of observations, we obtain their metric distance d_c and generalized model-based semivariance W_2^2 (square Wasserstein-2 distance), which quantifies feature similarity via underlying models. In the presence of metric autocorrelation, the expected generalized model-based semivariance is in theory an increasing function of d_c within range ρ and levels off beyond ρ, namely *theoretical generalized model-based semivariogram* γ_m. We fit γ_m from the empirical generalized model-based semivariogram $\hat{\gamma}_m$. MC-GTA penalizes observation pairs whose W_2^2 is close to or exceeding γ_m via a hinge loss with margin δ. An observation pair having no hinge loss penalty equals passing a goodness-of-fit test with significance level δ.

strategy is three-fold. Firstly, similarity computed based on underlying models is by nature more robust to noise and outliers than that based on raw feature vectors (Wang et al., 2016). Secondly, estimated underlying models provide better interpretability (Hallac et al., 2017). Finally, the magnitude of penalty can be tuned to adjust the emphasis on metric constraints, preferable to methods that enforce metric constraints as hard rules, such as ST-DBSCAN (Birant & Kut, 2007a), MDST-DBSCAN (Choi & Hong, 2021), and semi-supervised algorithms that discretize the metric constraints into graphs (Wagstaff et al., 2001; Basu et al., 2004; Lu, 2007; Bibi et al., 2019; Boecking et al., 2022). However, these approaches also have major drawbacks.

The most critical weakness of all existing clustering algorithms is that they ignore the effects of metric autocorrelation, e.g., temporal/spatial autocorrelation (Goodchild, 1987; Anselin, 1988; Fortin et al., 2002; Gubner, 2006), when applying the metric constraints. Metric autocorrelation effectively asserts that within a cluster, feature vectors observed at metrically distant positions naturally have higher empirical variance than metrically adjacent ones. Figure 1 shows how considering metric autocorrelation may even reverse the clustering results: suppose observation B and observation C are equally similar to observation A featurewise, but C is farther away from A than B. Without metric autocorrelation, B should be preferred to be clustered with A, since it has smaller metric distance; but if we do consider metric autocorrelation, C should be preferred instead, because autocorrelation implies that C would have been more similar to A if it were in B's metric position. Using log-likelihood as a clustering objective makes it hard to integrate metric constrains in a generative process. Although log-likelihood alone as loss function fits well with model selection theories such as Akaike Information Criterion (AIC), the sum of log-likelihood and weighted distance penalties lacks statistical meaning. Moreover, the presence of the penalty term breaks the convergence guarantee of EM iterations. Empirically, TICC/STICC is highly non-convex and difficult to optimize (Hallac et al., 2017; Kang et al., 2022). Other issues include high computational complexity, sensitivity to initial conditions, and expensive hyperparameter tuning. Please refer to Section 2 and Appendix A.7 for detailed discussions. These problems, however, can be avoided by getting rid of log-likelihood and EM iterations. One strategy is to perform clustering only according to pairwise similarity measurements, similar to DBSCAN.

In this paper, inspired by the analysis above, we propose a novel model-based clustering method named MC-GTA (Model-based Clustering via Goodness-of-fit Tests with Autocorrelations) that expilicitly accounts for the metric autocorrelation by designing a Wasserstein-2 distance-based multivariate generalization of the spatial semivariogram, which is widely used in geostatistics (Isaaks & Srivastava, 1989). MC-GTA first fits an underlying model (Gaussian Markov Random Field) for each observation using its neighbors. Then we compute the generalized model-based semivariance (i.e., square Wasserstein-2 distance) and metric distance for all observation pairs. Next, we fit the theoretical generalized model-based semivariogram and form our clustering objective as a total hinge loss based on the

difference between the empirical semivariance and the theoretical semivariogram. This injects metric constraints into clustering. Finally, we develop an algorithm that minimizes the loss. We prove that our objective can be theoretically interpreted in terms of goodness-of-fit tests. We use extensive experiments to demonstrate that MC-GTA can mitigate the computational complexity and convergence instability problems of TICC/STICC, achieving significantly better clustering quality.

To summarize, the **major contributions** of this paper are:

- We propose a Wasserstein-2 distance-based generalization of semivariogram that explicitly accounts for multivariate metric autocorrelation.
- We propose a novel model-based clustering objective based on goodness-of-fit tests. It simultaneously enables incorporating metric autocorrelation information and improves computational stability/efficiency. We believe that clustering based on statistic tests is a promising direction for future research developments.
- We compare our method with existing works comprehensively on various 1D and 2D synthetic and real-world datasets. We demonstrate that our method outperforms the baselines in clustering quality, computational stability, and computational efficiency.

2. Related Works

Spatial and Temporal Clustering. Clustering temporal subsequences and spatial subregions is a well-studied subfield of clustering (Appendix A.1). Some works treat temporal/spatial information as indices, such as dynamic time warping (Begum et al., 2015; Keogh, 2002; Keogh & Pazzani, 2000; Rakthanmanon et al., 2012), time point clustering (Gionis & Mannila, 2003; Zolhavarieh et al., 2014) and geo-tagged images (Liu et al., 2018), and some works cluster the spatio-temporal trajectories directly (Belhadi et al., 2020; Kisilevich et al., 2010). We are mostly interested in the first case, i.e., clustering temporally/geospatially referenced observations. However, these methods generally perform clustering based on feature similarity, which can be problematic or even unreliable (Keogh et al., 2003), because it only considers the structure of features, ignoring that the observations are also distributed over time and space.

To address this problem, two main strategies are explored in previous works. The first strategy is to enforce metric constraints as hard rules. For example, in ST-DBSCAN (Birant & Kut, 2007a) and MDST-DBSCAN (Choi & Hong, 2021), only temporally dense observations are considered candidates for core observations. The second strategy is to add a soft metric penalty to the clustering optimization objective. TICC (Hallac et al., 2017) is the first work to introduce Markov Random Fields to model temporal dependency structures of subsequences together with a soft temporal penalty. STICC (Kang et al., 2022), following this work, modified the algorithm to suit 2-dimensional spatial subregion clustering. They are both model-based clustering algorithms, like ARMA (Xiong & Yeung, 2004), GMM (Fraley & Raftery, 2006) and HMMs (Smyth, 1996). TICC/STICC achieves state-of-the-art performance in temporal/spatial clustering tasks.

3. Problem Formulation

3.1. Metric-Constrained Clustering

Given a dataset $\mathcal{D}$ of N observations $\{X_i\}_{i=1}^N$ (e.g., points of interest in an urban area, sensor measurements at different time points, etc), we need to assign each X_i to a set C_k, i.e. cluster k. The set of all clusters $\mathcal{C} = \{C_k\}_{k=1}^K$ is a *cluster assignment* or a *clustering*. K is called the number of clusters, either predefined or inferred from data.

Each observation $X_i = (\mathbf{f}_i, \mathbf{p}_i)$ is a tuple of two vectors: $\mathbf{f}_i$ is a d_F-dimensional feature vector (e.g., attributes of a POI) in a feature space F, while $\mathbf{p}_i$ is a d_M-dimensional position vector (e.g., geo-coordinates of this POI) in a metric space (M, d_c) (e.g., Earth surface with geodesic distance), where d_c is a predefined metric. With a dissimilarity measurement $d_f(\cdot,\cdot)$ in the feature space, e.g., cosine distance, a classic clustering problem without metric constraints $\hat{\mathcal{C}}_K = \arg\min_{\mathcal{C}} \mathcal{L}(\mathcal{C})$ is to minimize the loss:

$$\mathcal{L}(\mathcal{C}) = \sum_{\{C_k \in \mathcal{C}\}} \sum_{\{i,j \in C_k\}} d_f(\mathbf{f}_i, \mathbf{f}_j) - \alpha \sum_{\{C_k, C_l \in \mathcal{C}, k \neq l\}} \sum_{\{i \in C_k, j \in C_l\}} d_f(\mathbf{f}_i, \mathbf{f}_j) \tag{1}$$

where the first term is the intra-cluster cohesion objective and the second is the inter-cluster separation objective. α is a hyperparameter balancing cohesion and separation. Many applications emphasize more on intra-cluster cohesion. Following TICC/STICC (Hallac et al., 2017; Kang et al., 2022), we set $\alpha = 0$ in this study.

A *metric constraint* is an additional loss $\mathcal{L}^{\text{mc}}$ that assigns penalty based on metric distance and feature similarity. A *metric-constrained clustering* problem is to find an optimal cluster assignment that minimizes a multi-objective $\hat{\mathcal{C}}_K^{\text{mc}} = \arg\min_{\mathcal{C}} \left[\mathcal{L}(\mathcal{C}) + \beta \mathcal{L}^{\text{mc}}(\mathcal{C})\right]$ where β is a hyperparameter that determines how soft the constraints are, and

$$\mathcal{L}^{\text{mc}}(\mathcal{C}) = \sum_{\{C_k \in \mathcal{C}\}} \sum_{\{i,j \in C_k\}} r(d_f(\mathbf{f}_i, \mathbf{f}_j), d_c(\mathbf{p}_i, \mathbf{p}_j)). \tag{2}$$

r is a function of the metric distance and the feature dissimilarity, called the *metric penalty function*, designed to properly enforce the metric constraints. For example, in ST-DBSCAN (Birant & Kut, 2007a), temporal continuity

is the metric constraint. Conceptually it corresponds to $r(d_f(\mathbf{f}_i, \mathbf{f}_j), d_c(\mathbf{p}_i, \mathbf{p}_j)) = \mathbb{1}\{d_c(\mathbf{p}_i, \mathbf{p}_j) > \epsilon_t\}$ with ϵ_t being the preset radius of temporal neighborhood, and $\beta = \infty$. It effectively means that cluster assignments with temporal discontinuity are hard eliminated.

3.2. Metric-Constrained Model-Based Clustering

Metric-Constrained Model-based (MCM) clustering is a special case of metric-constrained clustering, which views the feature vector $\mathbf{f}_i$ of observation X_i as a random sample drawn from a parametric distribution $\mathcal{M}(\theta_i)$. We say $\mathcal{M}(\theta_i)$ is the underlying model of X_i and θ_i is a specification of the parameters. The family of distribution (e.g. Gaussian) and exact parameterization (e.g. mean and covariance matrix) of $\mathcal{M}(\theta_i)$ is chosen a priori based on domain knowledge and computational considerations. In MCM clustering, the feature dissimilarity measure $d_f(\mathbf{f}_i, \mathbf{f}_j)$ is replaced with $d_m(i,j) = d_m(\mathbf{f}_i, \mathbf{f}_j, \theta_i, \theta_j; \mathcal{M})$, named as *model-based dissimilarity*, e.g. negative log-likelihood.

In summary, MCM clustering can be formulated as minimizing the MCM loss[1]:

$$\mathcal{L}^{\text{mcm}}(\mathcal{C}) = \sum_{\{C_k \in \mathcal{C}\}} \sum_{\{i,j \in C_k\}} [d_m(i,j) + \beta r(i,j)] \quad (3)$$

As TICC authors (Hallac et al., 2017) argued, distance-based metrics have been shown to yield unreliable results in certain situations. While model-based approaches prevents overfitting and allows us to discover types of patterns that other approaches are unable to find. See Section 6.3.1 for a detailed analysis of the tasks in this study.

4. Preliminaries

In this section, we introduce a few useful statistical tools for our proposed clustering algorithm.

4.1. Classic Univariate Semivariogram

In Section 1, we argued for the importance of metric autocorrelation in metric-constrained clustering. In order to incorporate it into the clustering process, we need to appropriately quantify it. While an abundance of statistics for autocorrelation tests are developed in classic temporal and spatial analysis, such as Durbin–Watson statistic (Durbin & Watson, 1950; 1951) and Moran's I (Moran, 1950), the semivariogram (Matheron, 1963) fits our end best. This is because the theoretical semivariogram, denoted as $\gamma(\mathbf{p}_i, \mathbf{p}_j)$, is a function describing the degree of spatial dependence of a spatial random field or stochastic process, which is literally the fundamental assumption of model-based clustering.

Given a dataset (which is a sample generated from the spatial stochastic process) of N univariate observed variables $\{z_1, \cdots z_N\}$ together with their spatial positions $\{\mathbf{p}_1, \cdots \mathbf{p}_N\}$, there are N^2 pairs of variables (z_i, z_j) and their corresponding pairs of spatial positions $(\mathbf{p}_i, \mathbf{p}_j)$. The empirical semivariogram is defined as

$$\hat{\gamma}(h \pm \epsilon) := \frac{1}{2|N(h \pm \epsilon)|} \sum_{\{(\mathbf{p}_i, \mathbf{p}_j) \in N(h \pm \epsilon)\}} |z_i - z_j|^2 \quad (4)$$

where $N(h \pm \epsilon) := \{(\mathbf{p}_i, \mathbf{p}_j) | h - \epsilon \leq d_c(\mathbf{p}_i, \mathbf{p}_j) \leq h + \epsilon\}$, a set of spatial positions, and $|N(h \pm \epsilon)|$ is the size of the set. This is essentially the half empirical variance of all pairs whose spatial distance falls into the same distance bin centered at h of width 2ϵ.

In a semivariogram, the x and y axes indicate the spatial distance and semivariance $\hat{\gamma}(h \pm \epsilon)$, respectively. In the beginning the semivariance rises as distance increases, which indicates spatial autocorrelation. Then it levels off, which indicates that now semivariance no longer provides useful information. The range ρ is the distance beyond which spatial autocorrelation levels off. The sill σ is the semivariance when spatial autocorrelation levels off. The nugget ν is the semivariance when distance is almost zero, which is considered an intrinsic variance of the stochastic process.

Semivariogram can also be applied to metric spaces other than 2D or 3D geospatial space, such as temporal space, spatio-temporal space, and even multi-dimensional, non-Euclidean spaces (Nguyen et al., 2014). However, they are designed to quantify the autocorrelation between univariate observations and can not be applied to multivariate cases as shown in Figure 1. This is because the concepts of range, sill and nugget are defined as turning/intercepting points of the function. If γ is multivariate, the three core concepts are not well-defined. One of our novel contributions is to generalize the definition of semivariogram to multivariate observations. Figure 1 illustrates this generalization.

4.2. Wasserstein-2 Distance and Gaussian Markov Random Fields

To generalize semivariograms to multivariate cases, our strategy is to replace the univariate distance in Equation 4 with a model dissimilarity measurement. There are various statistical metrics or quasi-metrics that can be used, such as divergence (such as KL-divergence), total variation, discrepancy, and Wasserstein-2 distance (Gibbs & Su, 2002). We wish to choose one that is compatible with the classic semivariogram's definition. Specifically, we need to show that having a small semivariance in terms of model dissimilarity guarantees having a small semivariance in terms of feature difference. The weakest possible condition that satisfies this

[1] For the rest of the paper, we abbreviate the notations by omitting the arguments and only keeping the indices. For example, $d_m(\mathbf{f}_i, \mathbf{f}_j, \theta_i, \theta_j; \mathcal{M})$ is written in short as $d_m(i,j)$, $d_c(\mathbf{p}_i, \mathbf{p}_j)$ as $d_c(i,j)$, $r(d_m(i,j), d_c(i,j))$ as $r(i,j)$, respectively.

requirement is weak convergence, also known as convergence in distribution. Intuitively it says if a model weakly converges to another model, the observations generated from them will become statistically indistinguishable, consequently having indistinguishable semivariance. Therefore, we need to find a statistical metric W_2 that metricizes weak convergence, i.e., $(W_2(i,j) \to 0) \Rightarrow (\mathcal{F}_i \xrightarrow{\text{D}} \mathcal{F}_j)$. Here $\mathcal{F}_i, \mathcal{F}_j$ are cumulative distribution functions parametrized by θ_i, θ_j respectively, and $\xrightarrow{\text{D}}$ denotes convergence in distribution. Among all such metrizations, Lévy-Prokhorov metric and Wasserstein's distance (Earth Mover's Distance) are the two most important cases. By (Gibbs & Su, 2002), Lévy-Prokhorov metric is the tightest bound of the distance between two distributions, and the Wasserstein's distance is only looser up to a constant factor. Please refer to Appendix A.4 for the definition of both distance metrics.

Whereas the Lévy-Prokhorov Metric is in general not computable, the square Wasserstein-2 distance between two Gaussian Markov Random Fields (GMRFs) has a beautiful closed-form:

$$W_2^2(\theta_i, \theta_j) = d_2^2(\mu_i, \mu_j) + Tr\big(\Sigma_i + \Sigma_j - 2A\big) \quad (5)$$

Here μ_i, μ_j are mean vectors, Σ_i, Σ_j are covariance matrices, and $\theta_i = (\mu_i, \Sigma_i)$. $A = (\Sigma_i^{1/2}\Sigma_j\Sigma_i^{1/2})^{1/2}$ and $Tr(\cdot)$ is the trace of a matrix. d_2^2 is the square L2 norm. For the simplicity of notations, we use $W_2^2(i,j)$ in abbrevation of $W_2^2(\theta_i, \theta_j)$ throughout this paper.

The analysis above demonstrates that the combination of Wasserstein-2 distance and GMRFs is essentially the only choice we have that both satisfies our requirement of weak convergence and comes with practical computability. Please refer to Appendix A.2 for more background about GMRF. In Section 5 we show that our clustering objective based on square Wasserstein-2 distance has clear statistical meaning.

5. Method

5.1. Generalized Model-based Semivariogram

We propose a novel multivariate generalization of the classic semivariogram, called *generalized model-based semivariogram*, to appropriately quantify the multivariate metric autocorrelation. Unlike existing work such as Abzalov (2016), which modifies the definitions of range, sill, and nugget analogously to simultaneous confidence intervals, we derive a natural generalization by replacing the variance $|z_i - z_j|^2$ in Equation 4 with $d_m := W_2^2$. As in model-based clustering, every observed feature vector $\mathbf{f}_i$ has an underlying model $\mathcal{M}(\theta_i)$. Though the difference between the feature vectors is multivariate, the difference between the underlying models is univariate. We define the empirical generalized model-based semivariogram $\hat{\gamma}_m$ as

$$\hat{\gamma}_m(h \pm \epsilon) := \frac{1}{2|N(h \pm \epsilon)|} \sum_{(\mathbf{p}_i, \mathbf{p}_j) \in N(h \pm \epsilon)} W_2^2(i,j) \quad (6)$$

Following that, we can fit a theoretical generalized model-based semivariogram γ_m on $\hat{\gamma}_m$ by using well-established, classic univariate semivariogram fitting methods (Müller, 1999). ρ is the range of the fitted theoretical semivariogram.[2]

The soundness of this generalized definition is theoretically supported by goodness-of-fit tests (Section 5.2). It is also verified on real-world datasets. Figure 2 is the empirical generalized model-based semivariogram computed on a large geo-tagged image dataset iNaturalist-2018 (Cui et al., 2018). We use the top 16 PCA components of the pretrained image embedding as the feature vector, and the distance is the great circle distance between the geo-tags. For each image, we use its 15/20/30-nearest neighbors to estimate a GMRF as the underlying model. Then we compute the generalized semivariogram using Equation 6. We can see the empirical semivariogram conforms very well with the theory.

5.2. Clustering Objective as Goodness-of-Fit Tests

The conventional likelihood-based EM iterations of model-based clustering algorithms malfunction when metric constraints are involved. To avoid this situation, we propose to formulate the loss function in terms of only pairwise computations between observations. Our solution mainly relies on goodness-of-fit tests (i.e., whether two samples come from a statistically identical distribution) as the pairwise computation. We punish the pairs that have large goodness-of-fit test statistics beyond a significance threshold with a hinge loss. This threshold is based on the average goodness-of-fit test statistic dependent on metric distance. Specifically, the generic $\mathcal{L}^{\text{mcm}}$ loss (Equation 3) can be realized with goodness-of-fit tests as follows:

$$\begin{aligned}\mathcal{L}^{\text{MC-GTA}}(\mathcal{C}) = \sum_{\{C_k \in \mathcal{C}\}} \sum_{\{i,j \in C_k\}} \Big[\\ \lfloor W_2^2(i,j) - (\widehat{\mathbb{E}}_{i',j' \in \mathbb{N}} W_2^2(i',j') - \delta^0) \rfloor_+ \\ + \beta \lfloor W_2^2(i,j) - (\widehat{\mathbb{E}}_{i',j' \in \mathbb{N}_{i,j}} W_2^2(i',j') - \delta) \rfloor_+ \Big]\end{aligned} \quad (7)$$

Here $\mathbb{N}$ is the set of all observation pairs, and $\mathbb{N}_{i,j} = N(d_c(i,j) \pm \epsilon)$ are the pairs in (i,j)'s distance bin as defined in the generalized model-based semivariogram (Section 5.1). $\lfloor x \rfloor_+ = \max(0, x)$ is a rectifier (hinge) function. $W_2^2(i,j)$ is a goodness-of-fit statistic (Panaretos & Zemel, 2019).

This proposed loss has the following properties:

[2] For simplicity, in the rest of the paper, when mentioning semivariogram/semivariance, we always refer to the generalized model-based definitions unless otherwise specified.

- $W_2^2(i,j)$ is a goodness-of-fit test statistic. Thus, $W_2^2(i,j)$ being smaller than certain threshold implies observations i and j can pass the goodness-of-fit hypothesis test under certain significance level.

- $\widehat{\mathbb{E}}_{i',j'\in\mathbb{N}} W_2^2(i,j) - \delta^0$ and $\widehat{\mathbb{E}}_{i',j'\in\mathbb{N}_{i,j}} W_2^2(i,j) - \delta$ can be seen as two thresholds based on the average test statistics plus a desired significance level. In the case of Wasserstein-2 distance (Panaretos & Zemel, 2019), the test statistics follows a normal distribution with the square of Wasserstein-2 distance of the true underlying models as the mean. We do not have access to the true means, so we use empirical means as estimations. While $\widehat{\mathbb{E}}_{i',j'\in\mathbb{N}} W_2^2(i,j) - \delta^0$ is independent to metric autocorrelation, $\widehat{\mathbb{E}}_{i',j'\in\mathbb{N}_{i,j}} W_2^2(i,j) - \delta$ is dependent. We call them *non-metric threshold* and *metric threshold*, respectively. By definition, $\widehat{\mathbb{E}}_{i',j'\in\mathbb{N}_{i,j}} W_2^2(i,j)$ is exactly double of the empirical generalized model-based semivariogram defined in Section 5.1.

- The rectifier (hinge) function avoids assigning negative values to observation pairs that pass the test, which increases computational stability. The idea is similar to the idea of margin and hinge loss in SVMs. Our ablation study (Appendix 6.3) shows that this choice increases computational stability and clustering accuracy.

Since both $\widehat{\mathbb{E}}_{i',j'\in\mathbb{N}} W_2^2(i,j)$ and δ^0 are constants, for computational efficiency we can simplify Equation 7 by defining $\delta^0 = \widehat{\mathbb{E}}_{i',j'\in\mathbb{N}} W_2^2(i,j)$ and $r(i,j) = \lfloor W_2^2(i,j) - (\widehat{\mathbb{E}}_{i',j'\in N_{i,j}} W_2^2(i,j) - \delta)\rfloor_+$, and rewrite our loss as

$$\mathcal{L}^{\text{MC-GTA}}(\mathcal{C}) = \sum_{\{C_k\in\mathcal{C}\}} \sum_{\{i,j\in C_k\}} [W_2^2(i,j) + \beta r(i,j)] \quad (8)$$

which is exactly Equation 3. This demonstrates that by choosing appropriate d_m and $r(i,j)$, we can theoretically formulate clustering as minimizing the penalty for the intra-cluster pairs that do not pass goodness-of-fit tests. This is the central formula that our algorithm is based on.

5.3. Model-based Clustering via Goodness-of-fit Tests with Autocorrelations (MC-GTA)

The MC-GTA algorithm is naturally derived from incorporating the properties of metric autocorrelation with goodness-of-fit tests. As we have discussed in Section 5.2, the penalty function can be defined as

$$r(i,j) = \lfloor W_2^2(i,j) - [\gamma_m(d_c(i,j)) - \delta]\rfloor_+ \quad (9)$$

Notice we replace the empirical generalized model-based semivariance $\hat{\gamma}_m = \widehat{\mathbb{E}}_{i',j'\in N_{i,j}} W_2^2(i,j)$ with the fitted theoretical model-based semivariance γ_m, because the fitted semivariogram is smooth. In addition, we need to further condition the penalty function on the fitted range ρ (Equation 10). The reason is that when $d_c > \rho$, the second term in Eq 8 is much smaller than the first term and can be ignored to spare computation. Figure 6 shows an empirical analysis on the iNaturalist-2018 dataset. The average contribution of the penalty function to the total loss beyond ρ quickly drops below 15% and remains flat, which is non-informative and can be omitted in practice. We empirically verified in Appendix 6.3.5 that using the conditional form of penalty function is both beneficial for sparing computation and improving clustering performance. Thus, the final form of the penalty function is defined as

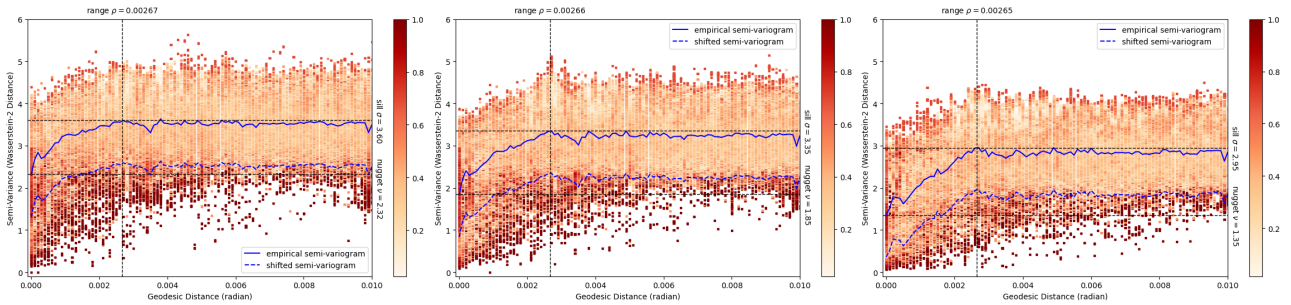
$$r_\rho(i,j) = \left\{ \begin{array}{ll} r(i,j), & d_c(i,j) \le \rho \\ 0, & d_c(i,j) > \rho \end{array} \right\} \quad (10)$$

Intuitively, r_ρ penalizes observations whose semivariance lies above the theoretical semivariogram. It is **local** (only effective within the range ρ), **monotonically decreasing**, and **continuous** (because the semivariogram is monotonically increasing and continuous within the range).

δ is a critical hyperparameter called *margin*. The motivation of using this hyperparameter comes from both theoretical analysis and empirical observations. Theoretically, the semivariogram is an average measurement of intra-cluster model dissimilarity. Inter-cluster pairs may also happen to have smaller than average semivariance. To reduce the chance of wrongly identifying inter-cluster pairs as intra-cluster, only observations whose semivariance is significantly small (at least δ below the theoretical semivariogram) are exempt from penalty. This is equivalent to shifting the semivariogram downwards by δ. Empirically, by plotting the percentage of ground-truth intra-cluster pairs in each bin, Figure 2 shows there is a clear boundary between the dark region (with higher percentage of intra-cluster pairs) and the light region (with lower percentage of intra-cluster pairs). We notice that by vertically shifting down for an appropriate distance δ, the semivariogram partially overlaps with the boundary. Then penalizing the observation pairs above the shifted semivariogram becomes practically equivalent to penalizing the observation pairs for falling into the region of low intra-cluster probability. Since we do not know the ground-truth value of δ, it is a hyperparameter that needs to be tuned. Algorithm 1 shows the core MC-GTA algorithm which is implemented based on Equation 6, 8 and 10.

6. Experiments

We perform extensive experiments on two synthetic and seven real-world datasets which cover both temporal and spatial clustering tasks. We compare MC-GTA with a wide range of baselines. The detailed experiment setup, baseline algorithms and evaluation metrics can be found in Appendix A.5. We conduct hyperparameter tuning on the number of neighbors n, the weight β, and the margin δ. Results show



(a) semivariogram with $n = 15$ (b) semivariogram with $n = 20$ (c) semivariogram with $n = 30$

Figure 2. Empirical generalized model-based semivariogram under different hyperparameter settings. n is the number of nearest neighbors used for fitting the GMRF models for each observation. The color represents the percentage of observation pairs that belong to the same ground-truth cluster in each 0.0001(geodesic) $\times$ 0.01(Wasserstein-2) bin.

Algorithm 1 MC-GTA Algorithm

Input : A dataset $\mathcal{D}$ of N observations $\{X_i = (\mathbf{f}_i \in F, \mathbf{p}_i \in M)\}_{i=1}^N$. The distance function d_m. The metric penalty function r. The model fitting algorithm $\mathbf{GL}$. The density-based clustering algorithm $\mathbf{DB}$. The number of neighbors n used for model fitting. The metric-constraint strength β. The margin hyperparameter δ.

Output : A clustering $\mathcal{C} = \{C_k\}_{k=1}^K$

for each observation $X_i \in \mathcal{D}$
  find n nearest observations $\mathbb{N}_i$ in the metric space
  fit the model parameters $\theta_i \leftarrow \mathbf{GL}(\mathbb{N}_i)$ (Sec 5.1)
for each pair of observations, compute their
  model dissimilarity $d_m(i,j) \leftarrow W_2^2(\theta_i, \theta_j)$ (Eq 5)
  metric distance $d_c(i,j) \leftarrow d_c(\mathbf{p}_i, \mathbf{p}_j)$
compute empirical generalized semivariogram $\hat{\gamma}_m$ (Eq 6)
fit theoretical generalized semivariogram γ_m from $\hat{\gamma}_m$
compute range ρ from γ_m (Sec 4.1)
compute loss matrix $M^w_{i,j} \leftarrow d_m(i,j) + \beta r_\rho(i,j)$ (Eq 10)
run density-based clustering algorithm $\mathcal{C} \leftarrow \mathbf{DB}(M^w)$
return $\mathcal{C}$

that the search spaces of all hyperparameters have good convexity. Moreover, unlike TICC and STICC, we do not need to re-compute the covariance metrics during hyperparameter tuning for MC-GTA. That makes hyperparameter tuning for MC-GTA much faster (see more details in Appendix A.8).

6.1. Main Result

From Table 1 we can see in general, model-based algorithms handle spatio-temporally distributed data better than feature-based clustering algorithms. Our method (MC-GTA-w) outperforms the strong baselines (TICC and STICC) in all tasks. Besides performance improvement, MC-GTA is also more flexible and generally applicable. MC-GTA performs consistently well throughout different constraint dimensions, dataset sizes and cluster numbers, whereas TICC and STICC can only handle either 1D or 2D metric constraints and do not converge stably (NC in Table 1), especially when the feature dimension and the dataset size are large. Furthermore, comparing TICC, STICC and MC-GTA-w with their non-constrained version (i.e., TICC ($\beta = 0$), STICC ($\beta = 0$) and MC-GTA-wo), we can see that metric constraints do improve the clustering quality.

One important observation is that SKATER (Assunção et al., 2006) performs even better than MC-GTA on the Climate dataset, but works extremely bad on the iNaturalist-2018/POI/Land-use datasets. There are two take-aways: (1) The ground-truth clusters of Climate dataset are contiguous regions with disjoint boundaries, whereas those of the latter three datasets overlap with each other (Figure 4). SKATER splits the metric space into a Voronoi diagram, so it fits the former dataset well but fails the latter. It is not as generally applicable as MC-GTA. (2) The performance of MC-GTA is worse on the Climate dataset because the observations are spatially sparse, e.g., the maximum distance between 30-nearest neighbors may be as large as 1 radian. This brings a dilemma: in order to have enough samples to estimate the underlying models, we must risk including metrically distant observations, which by the metric autocorrelation, have high variance. This is a limitation of our algorithm. See Appendix A.6 for more discussions of the result.

6.2. Stability, Robustness and Efficiency

MC-GTA is robust and computationally stable in two ways. Firstly, it is a sequential algorithm. TICC/STICC, on the other hand, uses EM iterations which may accumulate error. For example, we observe that if the initial cluster assignment is too imbalanced, TICC/STICC will self-enhancingly increase this imbalance until it fails to converge. Secondly, MC-GTA has only three hyperparameters to tune: the number of neighbors n; the penalty weight β and the shift δ. β and n are common hyperparameters that all model-based clustering algorithms share. Thus, only the shift δ is unique

Table 1. Performance on 1-D (temporal) and 2-D (spatial) tasks. d denotes the feature dimension, c denotes the ground-truth cluster number, and N denotes the size of each dataset. MC stands for metric constraint. **Bold** numbers and underlined numbers indicate the best and second best performances. (S)TICC means applying TICC to temporal datasets and STICC to spatial datasets. $\beta = 0$ means there is no temporal/spatial penalty term applied. - means the method is not suitable for this dataset. NC means the algorithm does not converge. MC-GTA-wo/MC-GTA-w represents MC-GTA loss without/with metric constraints respectively.

		Synthetic Datasets				Real-world Datasets													
						Temporal						Spatial							
		Temporal d=5, c=5 N=1,000		Spatial d=5, c=5 N=10,000		Pavement d=10, c=3 N=1,055		Vehicle d=7, c=5 N=16,641		Gesture d=3, c=8 N=704,970		Climate d=5, c=14 N=4,741		iNat2018 d=16, c=6 N=24,343		POI d=7, c=10 N=23,019		Landuse d=7, c=5 N=8,964	
Model Type	Model	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI
No-Constraint Model-Free	k-Means	1.03	1.69	1.26	1.66	8.02	6.59	8.94	21.54	2.78	5.23	5.47	22.14	6.91	14.71	18.37	43.44	2.39	4.21
	DBSCAN	2.44	2.50	3.69	5.38	15.25	18.75	33.67	41.83	1.18	2.07	3.61	17.89	34.91	34.69	15.03	39.29	11.91	7.19
	HDBSCAN	0.90	0.61	1.00	1.39	7.10	11.66	37.51	41.64	-	-	11.52	28.01	7.65	17.92	20.78	62.55	1.00	7.64
	DTW	2.52	2.13	-	-	17.13	17.55	8.11	23.35	-	-	-	-	-	-	-	-	-	-
Constrained Model-Free	PCK-Means	5.12	5.68	2.30	2.89	7.42	5.13	4.80	14.17	NC	NC	18.50	34.67	25.51	28.96	0.12	0.18	0.11	0.26
	MDST-DBSCAN	-	-	1.12	5.73	-	-	-	-	-	-	11.32	27.89	8.43	18.13	1.33	0.97	1.29	1.01
	SKATER	-	-	23.87	32.29	-	-	-	-	-	-	**23.44**	**44.10**	0.51	0.35	1.52	0.91	1.03	0.74
No-Constraint Model-Based	GMM	7.82	9.54	9.26	10.35	28.05	28.74	57.87	58.78	2.44	4.15	19.06	34.97	21.72	35.91	16.38	42.96	2.86	4.61
	(S)TICC-β=0	80.11	83.95	91.28	89.28	58.54	58.83	40.12	45.86	3.26	6.56	13.30	30.53	NC	NC	13.29	27.08	7.22	12.60
	MC-GTA-wo	86.38	84.56	87.34	84.74	76.10	74.36	63.31	58.60	8.12	33.60	16.63	36.73	21.90	36.47	30.45	66.23	12.91	28.72
Constrained Model-Based	(S)TICC	84.88	86.13	91.84	89.85	62.27	61.89	50.53	53.68	12.20	23.20	17.62	37.29	NC	NC	NC	NC	11.04	15.35
	MC-GTA-w	**90.50**	**87.96**	**94.49**	**91.98**	**77.64**	**77.22**	**65.04**	**59.36**	**26.51**	**55.34**	20.08	40.91	**42.70**	**40.49**	**39.81**	**68.27**	**36.54**	**42.97**

to our method. Furthermore, by comparing Figure 2a, 2b and 2c, we find that the key factors of the semivariogram (range, sill and nugget) remain relatively stable. This finding is critical because our method is heavily based on the reliable construction of the semivariogram.

MC-GTA is more efficient than TICC/STICC by removing EM iterations. The underlying model of each observation is only estimated once throughout the entire algorithm of MC-GTA , whereas TICC/STICC must re-estimate models in every iteration. Empirically, the execution speed of our method is 5 to 15 times faster than TICC/STICC. Moreover, the estimated underlying models can be archived and reused, making hyperparameter tuning much easier than TICC/STICC. Please see Appendix A.7 for a detailed comparison of theoretical and empirical runtime complexity between MC-GTA and TICC/STICC.

6.3. Ablation Studies

We conduct a series of ablation studies on the Pavement dataset to investigate the necessity and effectiveness of the components we adopt in our algorithm.

6.3.1. Wasserstein-2 Distance vs other feature similarity measures

To demonstrate how the choice of different feature similarity measures matters in clustering, we replace the Wasserstein-2 distance in Equation 6 with various model-free/model-based measures and report the experiment results on the Pavement dataset in Table 2. We can see that the model-based Wasserstein-2 distance significantly outperforms both the model-free and the model-based alternatives.

Table 2. Comparing different feature similarity measures

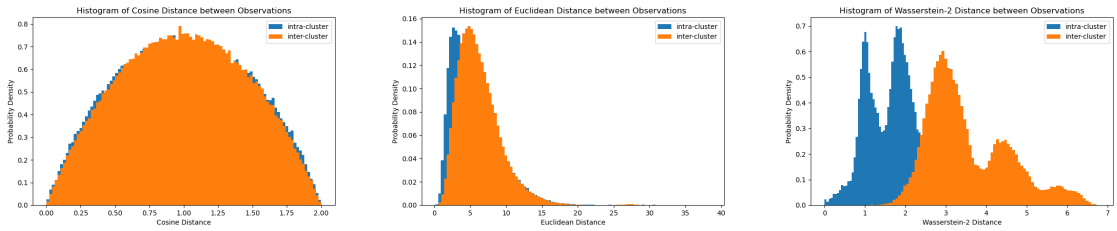
Wasserstein-2		Euclidean		Cosine		Total Var.		KL-D		JS-D	
ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI
77.64	77.22	23.11	22.43	0.34	1.36	3.37	3.61	56.73	66.10	15.55	18.84

Here we present a simple experiment to show how a model-based approach outperforms model-free distance when the underlying model is properly selected. Figure 3 depicts the histograms of pairwise Wasserstein-2 distance, Euclidean distance, and cosine distance between two observations in the Pavement dataset. The blue represents intra-cluster pairs and the orange represents inter-cluster pairs. We can see the model-based Wasserstein-2 distance itself makes distinguishing intra-cluster and inter-cluster pairs a lot easier than using Euclidean or cosine distance. It clearly demonstrates that raw feature similarity measures can not capture the complex patterns. Besides, Euclidean distance is known to be inefficient in high dimensions due to sparsity (Aggarwal et al., 2001), and the cosine distance is unable to represent differences in magnitudes.

Then we analyze why the other model-based measures yield inferior results. Total variation and Jensen-Shannon (JS) divergence perform poorly because they are difficult to accurately compute in high-dimensional spaces. KL-divergence underperforms our Wasserstein-2 distance because it is less sensitive to fine-grained model differences.

6.3.2. GLasso vs other covariance estimaters

We use Graphical Lasso as the covariance estimation algorithm, but the effectiveness of MC-GTA does not rely on this specific implementation. As an ablation study, we replace Graphical Lasso with Minimum Covariance Deter-



(a) Cosine distance (b) Euclidean distance (c) Wasserstein-2 distance

Figure 3. The histograms of pairwise distance between intra-cluster and inter-cluster observations in the Pavement dataset. The distributions of intra/inter-cluster Wasserstein-2 distance show more distinctive patterns than those of cosine distance and Euclidean distance.

minant (MIN-COV)[3] and Shrunk Covariance (SHRUNK)[4] and apply MC-GTA on the Pavement dataset. Clustering performance is reported in Table 3. All results are under the best hyperparameters based on grid search. In general, the more robust and more accurate the covariance estimation algorithm is, the better the clustering performance is. Shrunk is the least robust covariance estimation algorithm among the three, thus its performance is obviously lower than GLasso and MinCov. However, different variations of MC-GTA still significantly outperform the strongest baseline, TICC. It indicates the effectiveness of MC-GTA.

Table 3. Comparing different covariance estimation methods

Method	TICC (Baseline)		MC-GTA (GLASSO)		MC-GTA (MIN-COV)		MC-GTA (SHRUNK)	
Performance	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI
	62.27	61.89	77.64	77.22	80.82	73.78	74.70	71.42

6.3.3. DBSCAN VS OTHER DISTANCE-BASED CLUSTERING ALGORITHMS

Similarly, MC-GTA does not rely on any specific implementation of the clustering algorithm. Table 4 again demonstrates that though clustering performances are affected by the choice of distance-based clustering algorithms, MC-GTA still outperforms the baselines by large margins.

Table 4. Comparing different distance-based clustering algorithms

Method	TICC (Baseline)		MC-GTA (DBSCAN)		MC-GTA (HDBSCAN)		MC-GTA (OPTICS)	
Performance	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI
	62.27	61.89	77.64	77.22	72.35	69.61	69.77	68.58

6.3.4. THE HINGE OPERATION IN THE LOSS FUNCTION

In our ablation experiment, removing the rectifier (hinge) operation causes computational instability. Since we apply a distance-based clustering algorithm on top of the weighted distance matrix, all entries are required to be positive. When we remove the max operation, the weighted distance sometimes becomes negative and the clustering algorithm fails.

6.3.5. THE RANGE CONDITION IN THE PENALTY

As an ablation study, we ignore the range condition in Equation 10, i.e., define $r_\rho(i,j)$ simply as

$$r_\rho(i,j) = \lfloor W_2^2(i,j) - [\gamma_m(d_c(i,j)) - \delta] \rfloor_+ \qquad (11)$$

This means we need to compute the penalty term for all possible pairs of data points. We then apply MC-GTA. The comparison of clustering performance using Equation 10 (Conditional) and using Equation 11 (Unconditional) is demonstrated in Table 5. Ignoring the range does not only negatively affects the clustering performance, but also wastes resources, since we can spare the computation of penalty terms of pairs out of range.

Table 5. Comparing conditional and unconditional penalty

Method	MC-GTA (Conditional)		MC-GTA (Unconditional)	
Performance	ARI	NMI	ARI	NMI
	77.64	77.22	76.91	76.25

7. Conclusion and Future Works

In this paper, we propose a novel clustering technique called MC-GTA that injects knowledge of metric autocorrelation into model-based clustering algorithms by computing pairwise Wasserstein-2 distance between estimated model parameterizations for each observation. MC-GTA provides a unified solution to clustering problems with temporal/spatial/higher-dimensional metric constraints and achieves SOTA performance on both synthetic and real-world datasets. Moreover, by minimizing the total hinge loss of pairwise goodness-of-fit tests, MC-GTA is more computationally efficient and stable than the strongest baselines TICC and STICC, which optimize data likelihood through EM procedures during clustering.

For future work, it is worth extending MC-GTA to non-Gaussian, general Markov Random Fields using their corresponding pairwise Wasserstein-2 distances.

[3]https://scikit-learn.org/stable/modules/generated/sklearn.covariance.MinCovDet.html
[4]https://scikit-learn.org/stable/modules/generated/sklearn.covariance.ShrunkCovariance.html

Acknowledgements

This work is mainly funded by the National Science Foundation under Grant No. 2033521 A1 – KnowWhereGraph: Enriching and Linking Cross-Domain Knowledge Graphs using Spatially-Explicit AI Technologies. Gengchen Mai acknowledges support from the Microsoft Research Accelerate Foundation Models Academic Research (AFMR) Initiative. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Software and Data

All the datasets and packages we use are publicly accessible online and properly cited. The implementation of our algorithm and a tutorial is publicized on GitHub via https://github.com/Octopolugal/MC-GTA.git.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. We do not foresee any potential negative societal impacts of the current work.

References

Abzalov, M. *Multivariate Geostatistics*, pp. 287–290. Springer International Publishing, Cham, 2016. ISBN 978-3-319-39264-6. doi: 10.1007/978-3-319-39264-6_20. URL https://doi.org/10.1007/978-3-319-39264-6_20.

Aggarwal, C. C., Hinneburg, A., and Keim, D. A. On the surprising behavior of distance metrics in high dimensional space. In *Database Theory—ICDT 2001: 8th International Conference London, UK, January 4–6, 2001 Proceedings 8*, pp. 420–434. Springer, 2001.

Ahmed, M., Seraj, R., and Islam, S. M. S. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8):1295, 2020.

Anselin, L. *Spatial econometrics: methods and models*, volume 4. Springer Science & Business Media, 1988.

Assunção, R. M., Neves, M. C., Câmara, G., and da Costa Freitas, C. Efficient regionalization techniques for socio-economic geographical units using minimum spanning trees. *International Journal of Geographical Information Science*, 20(7):797–811, 2006.

Basu, S., Banerjee, A., and Mooney, R. J. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the 2004 SIAM international conference on data mining*, pp. 333–344. SIAM, 2004.

Begum, N., Ulanova, L., Wang, J., and Keogh, E. Accelerating dynamic time warping clustering with a novel admissible pruning strategy. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pp. 49–58, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336642. doi: 10.1145/2783258.2783286. URL https://doi.org/10.1145/2783258.2783286.

Belhadi, A., Djenouri, Y., Nørvåg, K., Ramampiaro, H., Masseglia, F., and Lin, J. C.-W. Space–time series clustering: Algorithms, taxonomy, and case study on urban smart cities. *Engineering Applications of Artificial Intelligence*, 95:103857, 2020. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2020.103857. URL https://www.sciencedirect.com/science/article/pii/S0952197620302141.

Bibi, A., Wu, B., and Ghanem, B. Constrained k-means with general pairwise and cardinality constraints. *CoRR*, abs/1907.10410, 2019. URL http://arxiv.org/abs/1907.10410.

Birant, D. and Kut, A. St-dbscan: An algorithm for clustering spatial–temporal data. *Data and Knowledge Engineering*, 60(1):208–221, 2007a. ISSN 0169-023X. doi: https://doi.org/10.1016/j.datak.2006.01.013. URL https://www.sciencedirect.com/science/article/pii/S0169023X06000218. Intelligent Data Mining.

Birant, D. and Kut, A. St-dbscan: An algorithm for clustering spatial–temporal data. *Data & knowledge engineering*, 60(1):208–221, 2007b.

Boecking, B., Jeanselme, V., and Dubrawski, A. Constrained clustering and multiple kernel learning without pairwise constraint relaxation. *Advances in Data Analysis and Classification*, pp. 1–16, 2022.

Cai, L., Janowicz, K., Mai, G., Yan, B., and Zhu, R. Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting. *Transactions in GIS*, 24(3):736–755, 2020.

Choi, C. and Hong, S.-Y. Mdst-dbscan: A density-based clustering method for multidimensional spatiotemporal data. *ISPRS International Journal of Geo-Information*, 10 (6), 2021. ISSN 2220-9964. doi: 10.3390/ijgi10060391. URL https://www.mdpi.com/2220-9964/10/6/391.

Cui, Y., Song, Y., Sun, C., Howard, A., and Belongie, S. Large scale fine-grained categorization and domain-specific transfer learning, 2018.

Durbin, J. and Watson, G. S. Testing for serial correlation in least squares regression: I. *Biometrika*, 37(3/4):409–428, 1950. ISSN 00063444. URL http://www.jstor.org/stable/2332391.

Durbin, J. and Watson, G. S. Testing for serial correlation in least squares regression. ii. *Biometrika*, 38(1/2):159–177, 1951. ISSN 00063444. URL http://www.jstor.org/stable/2332325.

Elmustafa, A., Rozi, E., He, Y., Mai, G., Ermon, S., Burke, M., and Lobell, D. Understanding economic development in rural africa using satellite imagery, building footprints and deep models. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, pp. 1–4, 2022.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pp. 226–231. AAAI Press, 1996.

Fortin, M.-J., Dale, M. R., and Ver Hoef, J. Spatial analysis in ecology. *Encyclopedia of environmetrics*, 4:2051–2058, 2002.

Fraley, C. and Raftery, A. Mclust version 3: An r package for normal mixture modeling and model-based clustering. *University of Washington Tech Report*, 504:51, 09 2006.

Friedman, J., Hastie, T., and Tibshirani, R. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 12 2007. ISSN 1465-4644. doi: 10.1093/biostatistics/kxm045. URL https://doi.org/10.1093/biostatistics/kxm045.

Gao, S., Janowicz, K., and Couclelis, H. Extracting urban functional regions from points of interest and human activities on location-based social networks. *Transactions in GIS*, 21(3):446–467, 2017.

Gelbrich, M. On a formula for the l2 wasserstein metric between measures on euclidean and hilbert spaces. *Mathematische Nachrichten*, 147(1):185–203, 1990. doi: https://doi.org/10.1002/mana.19901470121. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/mana.19901470121.

Gibbs, A. L. and Su, F. E. On choosing and bounding probability metrics, 2002.

Gionis, A. and Mannila, H. Finding recurrent sources in sequences. In *Proceedings of the Seventh Annual International Conference on Research in Computational Molecular Biology*, RECOMB '03, pp. 123–130, New York, NY, USA, 2003. Association for Computing Machinery. ISBN 1581136358. doi: 10.1145/640075.640091. URL https://doi.org/10.1145/640075.640091.

Goodchild, M. F. *Spatial autocorrelation / [by] Michael F. Goodchild.* Concepts and techniques in modern geography ; no. 47. Geo, 1987. ISBN 0860942236.

Gormley, I., Murphy, T., and Raftery, A. Model-based clustering. *Annual Review of Statistics and Its Application*, 10, 10 2022.

Gubner, J. A. *Probability and Random Processes for Electrical and Computer Engineers*. Cambridge University Press, 2006. doi: 10.1017/CBO9780511813610.

Hallac, D., Vare, S., Boyd, S., and Leskovec, J. Toeplitz inverse covariance-based clustering of multivariate time series data. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 215–223, 2017.

Hartman, L. and Hössjer, O. Fast kriging of large data sets with gaussian markov random fields. *Computational Statistics and Data Analysis*, 52(5):2331–2349, 2008.

Hu, Y., Gao, S., Janowicz, K., Yu, B., Li, W., and Prasad, S. Extracting and understanding urban areas of interest using geotagged photos. *Computers, Environment and Urban Systems*, 54:240–254, 2015.

Hubert, L. J. and Arabie, P. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

Isaaks, E. H. and Srivastava, R. M. Applied geostatistics. *(No Title)*, 1989.

Jain, A. K. and Dubes, R. C. *Algorithms for Clustering Data*. Prentice-Hall, Inc., USA, 1988. ISBN 013022278X.

Janowicz, K., McKenzie, G., Hu, Y., Zhu, R., and Gao, S. Using semantic signatures for social sensing in urban environments. In *Mobility patterns, big data and transport analytics*, pp. 31–54. Elsevier, 2019.

Kang, Y., Zhang, F., Peng, W., Gao, S., Rao, J., Duarte, F., and Ratti, C. Understanding house price appreciation using multi-source big geo-data and machine learning. *Land Use Policy*, 111:104919, 2021.

Kang, Y., Wu, K., Gao, S., Ng, I., Rao, J., Ye, S., Zhang, F., and Fei, T. STICC: a multivariate spatial clustering method for repeated geographic pattern discovery with consideration of spatial contiguity. *International Journal of Geographical Information Science*, 36(8):1518–1549, 2022.

Keogh, E. Exact indexing of dynamic time warping. In *Proceedings of the 28th International Conference on Very Large Data Bases*, VLDB '02, pp. 406–417. VLDB Endowment, 2002.

Keogh, E., Lin, J., and Truppel, W. Clustering of time series subsequences is meaningless: implications for previous and future research. In *Third IEEE International Conference on Data Mining*, pp. 115–122, 2003. doi: 10.1109/ICDM.2003.1250910.

Keogh, E. J. and Pazzani, M. J. Scaling up dynamic time warping for datamining applications. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '00, pp. 285–289, New York, NY, USA, 2000. Association for Computing Machinery. ISBN 1581132336. doi: 10.1145/347090.347153. URL https://doi.org/10.1145/347090.347153.

Kisilevich, S., Mansmann, F., Nanni, M., and Rinzivillo, S. *Spatio-temporal clustering*. Springer, 2010.

Law, S., Paige, B., and Russell, C. Take a look around: using street view and satellite images to estimate house prices. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(5):1–19, 2019.

Liu, H., Zhan, Q., Yang, C., and Wang, J. Characterizing the spatio-temporal pattern of land surface temperature through time series clustering: Based on the latent pattern and morphology. *Remote Sensing*, 10(4), 2018. ISSN 2072-4292. doi: 10.3390/rs10040654. URL https://www.mdpi.com/2072-4292/10/4/654.

Liu, K., Gao, S., Qiu, P., Liu, X., Yan, B., and Lu, F. Road2vec: Measuring traffic interactions in urban road system from massive travel routes. *ISPRS International Journal of Geo-Information*, 6(11):321, 2017.

Lu, Z. Semi-supervised clustering with pairwise constraints: A discriminative approach. In *Artificial Intelligence and Statistics*, pp. 299–306. PMLR, 2007.

Mac Aodha, O., Cole, E., and Perona, P. Presence-only geographical priors for fine-grained image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9596–9606, 2019.

Mai, G., Janowicz, K., Hu, Y., and Gao, S. Adcn: An anisotropic density-based clustering algorithm for discovering spatial point patterns with noise. *Transactions in GIS*, 22(1):348–369, 2018.

Mai, G., Lao, N., He, Y., Song, J., and Ermon, S. Csp: Self-supervised contrastive spatial pre-training for geospatial-visual representations. In *the Fortieth International Conference on Machine Learning (ICML 2023)*, 2023a.

Mai, G., Xuan, Y., Zuo, W., He, Y., Song, J., Ermon, S., Janowicz, K., and Lao, N. Sphere2vec: A general-purpose location representation learning over a spherical surface for large-scale geospatial predictions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 202: 439–462, 2023b.

Manvi, R., Khanna, S., Mai, G., Burke, M., Lobell, D. B., and Ermon, S. Geollm: Extracting geospatial knowledge from large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Matheron, G. Principles of geostatistics. *Economic Geology*, 58(8):1246–1266, 12 1963. ISSN 0361-0128. doi: 10.2113/gsecongeo.58.8.1246. URL https://doi.org/10.2113/gsecongeo.58.8.1246.

McInnes, L., Healy, J., and Astels, S. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11): 205, 2017.

McKenzie, G., Janowicz, K., Gao, S., and Gong, L. How where is when? on the regional variability and resolution of geosocial temporal signatures for points of interest. *Computers, Environment and Urban Systems*, 54:336–346, 2015.

Moran, P. A. P. Notes on continuous stochastic phenomena. *Biometrika*, 37(1/2):17–23, 1950. ISSN 00063444. URL http://www.jstor.org/stable/2332142.

Müller, W. G. Least-squares fitting from the variogram cloud. *Statistics & Probability Letters*, 43(1):93–98, 1999. ISSN 0167-7152. doi: https://doi.org/10.1016/S0167-7152(98)00250-8. URL https://www.sciencedirect.com/science/article/pii/S0167715298002508.

Nguyen, H., Osterman, G., Wunch, D., O'Dell, C., Mandrake, L., Wennberg, P., Fisher, B., and Castano, R. A method for colocating satellite x co 2 data to ground-based data and its application to acos-gosat and tccon. *Atmospheric Measurement Techniques*, 7(8):2631–2644, 2014.

Panaretos, V. M. and Zemel, Y. Statistical aspects of wasserstein distances. *Annual review of statistics and its application*, 6:405–431, 2019.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., and Keogh, E. Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD

'12, pp. 262–270, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450314626. doi: 10.1145/2339530.2339576. URL https://doi.org/10.1145/2339530.2339576.

Reynolds, D. A. et al. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009.

Rue, H. and Tjelmeland, H. Fitting gaussian markov random fields to gaussian fields. *Scandinavian journal of Statistics*, 29(1):31–49, 2002.

Smyth, P. Clustering sequences with hidden markov models. In *Proceedings of the 9th International Conference on Neural Information Processing Systems*, NIPS'96, pp. 648–654, Cambridge, MA, USA, 1996. MIT Press.

Veldt, N., Gleich, D. F., Wirth, A., and Saunderson, J. Metric-constrained optimization for graph clustering algorithms. *SIAM Journal on Mathematics of Data Science*, 1(2):333–355, 2019.

Vinh, N., Epps, J., and Bailey, J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11:2837–2854, 10 2010.

Wagstaff, K., Cardie, C., Rogers, S., and Schrödl, S. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pp. 577–584, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558607781.

Wang, C., Komodakis, N., and Paragios, N. Markov random field modeling, inference & learning in computer vision & image understanding: A survey. *Computer Vision and Image Understanding*, 117(11):1610–1627, 2013.

Wang, J.-L., Chiou, J.-M., and Müller, H.-G. Functional data analysis. *Annual Review of Statistics and its application*, 3:257–295, 2016.

Xiong, Y. and Yeung, D.-Y. Time series clustering with arma mixtures. *Pattern Recognition*, 37(8):1675–1689, 2004. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2003.12.018. URL https://www.sciencedirect.com/science/article/pii/S0031320304000585.

Xu, D. and Tian, Y. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2:165–193, 2015.

Yadav, M. and Alam, M. A. Dynamic time warping (dtw) algorithm in speech: a review. *International Journal of Research in Electronics and Computer Engineering*, 6(1): 524–528, 2018.

Yan, B., Janowicz, K., Mai, G., and Gao, S. From itdl to place2vec: Reasoning about place type similarity and relatedness by learning embeddings from augmented spatial contexts. In *Proceedings of the 25th ACM SIGSPATIAL international conference on advances in geographic information systems*, pp. 1–10, 2017.

Yang, D., Zhang, D., Zheng, V. W., and Yu, Z. Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(1):129–142, 2015. doi: 10.1109/TSMC.2014.2327053.

Yeh, C., Meng, C., Wang, S., Driscoll, A., Rozi, E., Liu, P., Lee, J., Burke, M., Lobell, D. B., and Ermon, S. Sustainbench: Benchmarks for monitoring the sustainable development goals with machine learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Zolhavarieh, S., Aghabozorgi, S., and Wah, T. A review of subsequence time series clustering. *The Scientific World Journal*, 07 2014. doi: 10.1155/2014/312521.

A. Appendix

A.1. A General Objective of Clustering

While the complete definition of clustering has not yet come to an agreement, three principles in general apply (Jain & Dubes, 1988): (1) **Intra-Cluster Cohesion**: observations, in the same cluster, must be as similar as possible; (2) **Inter-Cluster Separation**: observations, in different clusters, must be as different as possible; (3) **Interpretability**: measurement for similarity and dissimilarity must be clear and have practical meanings. Following these principles, the two key components of a clustering algorithm are the similarity/dissimilarity measurement and the algorithm that optimizes intra-cluster cohesion and inter-cluster separation. There is no cure-all clustering algorithm. Different data structures require different similarity/dissimilarity measurements (e.g. cosine distance, Euclidean distance, graph distance) and different optimization algorithms (e.g. hierarchical, iterative, estimation-maximization-based), resulting in a variety of clustering algorithms such as partition-based clustering, hierarchical clustering, density-based clustering, model-based clustering, etc.

Conceptually, if we define measurements similarity $s(\cdot,\cdot)$ and dissimilarity $d(\cdot,\cdot)$ between two observations X_i, X_j, we can view a clustering problem as finding an optimal cluster assignment $\hat{C}_K$ that maximizes the objective:

$$\hat{C}_K = \arg\max_C \Big[\sum_{\{C_k \in C\}} \sum_{\{i,j \in C_k\}} s(X_i, X_j) + \beta \sum_{\{C_k, C_l \in C, k \neq l\}} \sum_{\{i \in C_k, j \in C_l\}} d(X_i, X_j) \Big] \tag{12}$$

The first term corresponds to the intra-cluster cohesion principle and the second term corresponds to the inter-cluster separation principle. β is a hyperparameter chosen to control how much we weigh these two terms, since the two objectives may compete. The choice of s and d, in turn, corresponds to the interpretability principle. Usually we simply let $s(\cdot,\cdot) = -d(\cdot,\cdot)$, thus the objective becomes

$$\hat{C}_K = \arg\min_C \Big[\sum_{\{C_k \in C\}} \sum_{\{i,j \in C_k\}} d(X_i, X_j) - \beta \sum_{\{C_k, C_l \in C, k \neq l\}} \sum_{\{i \in C_k, j \in C_l\}} d(X_i, X_j) \Big] \tag{13}$$

A.2. Gaussian Markov Random Field (GMRF)

A Gaussian Markov Random Field (GMRF) is a special case of the general Markov Random Field (MRF) (Wang et al., 2013), which additionally requires the joint and marginal distributions of variables to be Gaussian. Using GMRFs introduces several advantages. The first advantage is high computational efficiency. A (centered) GMRF can be efficiently represented and fitted as a sparse covariance matrix, through Graphical LASSO[5] (Friedman et al., 2007). Secondly, a GMRF can provide interpretable insights into variable correlations. Finally, a GMRF can be used to properly model continuous data in a wide range of situations (Rue & Tjelmeland, 2002; Hartman & Hössjer, 2008). For example, in spatial data mining, many commonly used real-valued features, such as place check-in numbers (McKenzie et al., 2015; Janowicz et al., 2019; Yan et al., 2017), traffic volume (Liu et al., 2017; Cai et al., 2020), customer rating (Gao et al., 2017), sustainability indices (Yeh et al., 2021; Elmustafa et al., 2022; Manvi et al., 2024), and real-estate pricing (Law et al., 2019; Kang et al., 2021), can be treated as normal distributions after standardization. In addition to that, the covariance representation of a GMRF can be easily extended into a Toeplitz matrix that models inter-observation dependency, which is very important in understanding the interactions across time (Hallac et al., 2017) and space (Kang et al., 2022). Due to the above advantages of GMRF, we choose it as the parametrization of the underlying models in our method. Furthermore, the other important component of our method, the Wasserstein-2 distance (Gibbs & Su, 2002), works best with GMRFs. It is mathematically proved that the Wasserstein-2 distance has a closed-form solution on GMRF models, which ensures the efficiency and stability of our method.

A.3. Metricization of Weak Convergence

To read more detailed discussions of the metrization of probability convergence, see Gibbs & Su (2002) for a comprehensive summary. According to the same paper, two important propositions are worth notification: 1) the Lévy-Prokhorov metric is "precisely the minimum distance 'in probability' between random variables distributed according to μ and ν", and 2) the Lévy-Prokhorov metric and the Wasserstein's distance satisfy the following quantitative relation:

$$\pi \leq W_p \leq (\text{diam}(\Omega) + 1)\pi \tag{14}$$

[5] https://scikit-learn.org/stable/modules/generated/ sklearn.covariance.graphical_lasso.html

where $\text{diam}(\Omega) := \sup\{d(x, y) : x, y \in \Omega\}$ is the diameter of the sample space Ω. These two propositions justify that though the Wasserstein's distance is not the tightest bound (i.e., the Prokhorov metric), it converges as fast up to a constant factor, so long as the metric space is bounded.

Since both the Lévy-Prokhorov metric and the Wasserstein's distance has guaranteed convergence, the choice of d_m is mainly upon computational efficiency. Whereas both metrizations have no simple algorithms for computation in the general case, the Wasserstein-2 distance between two multi-variate Gaussian distributions has a neat closed-form formula in terms of mean vectors and covariance matrices. Gelbrich (1990) gives the formula of the squared Wasserstein-2 distance as follows:

$$W_2^2(\theta_1, \theta_2) = d_2^2(\mu_1, \mu_2) + \text{Tr}(\Sigma_1 + \Sigma_2 - 2(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2})^{1/2}) \tag{15}$$

Here μ_1, μ_2 are mean vectors and Σ_1, Σ_2 are covariance matrices. $\theta = (\mu, \Sigma)$. Tr is the trace of a matrix.

By computing the pairwise Wasserstein-2 distance between the estimated models, we obtain a distance matrix. Any density-based clustering algorithms that support pre-computed distance matrix, such as DBSCAN (Ester et al., 1996), can be seamlessly applied without any modification. Since these clustering algorithms are designed to minimize intra-cluster distance and maximize inter-cluster distance, it follows immediately that the intra-cluster observations follow as similar as possible distributions whereas the inter-cluster observations follow as dissimilar as possible distributions, by the fact that the Wasserstein's distance is a metrization of weak convergence.

With this dimension reduction, we can finally transform the original metric-constrained model-based clustering problem in the product space $F \times M$ to a simpler problem in the product space $\mathcal{R} \times M$. Since $\mathcal{R}$ and M are both metric spaces, density-based algorithms that are supported on product metric spaces such as ST-DBSCAN(Birant & Kut, 2007a) can be then applied. However, these algorithms treat the two metric spaces independently without considering the correlation introduced by the metric constraint. In Section 5.3, we discuss how to address this issue.

A.4. Definitions of Distance Metrics

The definitions of Lévy-Prokhorov Metric and Wasserstein's distance are as follow.

Lévy-Prokhorov Metric: given a separable metric space (M, d) together with its Borel sigma algebra $\mathcal{B}(M)$, define the ϵ-neighborhood of $A \subset M$ as $A^\epsilon := \{p \in M : \exists q \in A \text{ s.t. } d(p, q) < \epsilon\}$. Then the Lévy-Prokhorov metric π of two probability measures μ, ν is defined as

$$\pi(\mu, \nu) := \inf\{\epsilon > 0 : \mu(A) \leq \nu(A^\epsilon) + \epsilon \text{ and } \nu(A) \leq \mu(A^\epsilon) + \epsilon, \forall A \in \mathcal{B}(M)\} \tag{16}$$

Wasserstein's Distance: given a Radon metric space (M, d), for $p \in [1, \infty)$, the Wasserstein-p distance W_p between two probability measures μ, ν is defined as

$$W_p := (\inf_{\gamma \in \Gamma(\mu,\nu)} \mathbb{E}_{(x,y)\sim\gamma d(x,y)^p})^{1/p}. \tag{17}$$

Here $\Gamma(\mu, \nu)$ is the set of all possible couplings of μ and ν.

A.5. Experiment Setup

A.5.1. Baseline Models and Evaluation Metrics

Baseline Models. We compare our method to both density-based and model-based clustering algorithms. See Table 1 for details. Among them, TICC(Hallac et al., 2017) can only deal with 1-dimensional constraint and STICC(Kang et al., 2022) can only deal with 2-dimensional constraint. Thus, the former will only be evaluated against 1-dimensional datasets and the latter only against 2-dimensional datasets. All other models that do not incorporate metric constraint information are evaluated on both 1-dimensional and 2-dimensional datasets.

We compare our MC-GTA with a wide range of baseline clustering algorithms. (1) General non-constrained clustering algorithms: kMeans (Ahmed et al., 2020), DBSCAN (Ester et al., 1996), HDBSCAN (McInnes et al., 2017). (2) Must-link/Cannot-link-based constrained clustering algorithms: PCKMeans (Basu et al., 2004)) , which use distance matrix to sample must-links/cannot-links. (3) Temporal/Spatial clustering algorithms: DTW (Yadav & Alam, 2018) (temporal), MDST-DBSCAN (Choi & Hong, 2021) (the multivariate version of ST-DBSCAN (Birant & Kut, 2007b),

spatial-temporal), SKATER (Assunção et al., 2006) (spatial). (4) Model-based clustering algorithms GMM (Reynolds et al., 2009), TICC (Hallac et al., 2017) and STICC (Kang et al., 2022).

Evaluation Metrics of Clustering Quality. For the fairness of comparison, we adopt the most commonly used ground-truth label based metrics, Adjusted Rand Index (ARI) (Hubert & Arabie, 1985) and Normalized Mutual Information (NMI) (Vinh et al., 2010). We use their implementation in `sklearn`(Pedregosa et al., 2011). We do not adopt the Macro-F1 metric that TICC(Hallac et al., 2017) uses because this metric is only well-defined when cluster number is fixed, while our method is density-based, which does not preset a cluster number.

A.5.2. Synthetic Dataset

We generate 1-dimensional and 2-dimensional synthetic datasets following the MC-GTA assumption discussed in Section 5.3. The only hyperparameters we preset are cluster number K, feature dimension D, noise scale α and sample batch size k. All other hyperparameters such as sequence length, cluster size and so forth are completely randomly generated for the sake of fair comparison.

1-Dimensional Synthetic Dataset. We generate the 1-dimensional synthetic dataset following the MC-GTA assumption discussed in Section 5.3:

- Choose hyperparameters: cluster number K, feature dimension D, noise scale α, sample size k.
- Randomly choose a subsequence number N.
- Randomly generate K different $D \times D$ ground-truth covariance matrices $\{\Sigma_1, \Sigma_2 \cdots \Sigma_K\}$. It is required that the pairwise Wasserstein-2 distances should all be greater than 1.0. This is to make sure that observations of different clusters are statistically different.
- Generate a random list of N ground-truth subsequence cluster labels $\{C_1, C_2 \cdots C_N\}, C_i \in \{0..K-1\}$, and a random list of N subsequence lengths $\{L_1, L_2 \cdots L_N\}$.
- For each subsequence label C_i and subsequence length L_i, generate L_i perturbed covariance matrices $\{P_{i,1}, P_{i,2} \cdots P_{i,L_i}\}$ by adding Gaussian noise to Σ_i. Notice, in order to conform with the monotonic assumption, we add noise with noise scale $j\alpha$ as we generate $P_{i,j}$, and the maximum noise scale should be no larger than 10% of the maximum entry in the ground-truth covariance matrix, in order to meet the continuous assumption.
- Sample k D-dimensional feature vectors from each $P_{i,j}$ sequentially and concatenate them all together into a $k\Sigma_{i=1}^{N} L_i$ list, with each entry being a D-dimensional feature vector. The corresponding position list is simply $\{1, 2 \cdots k\Sigma_{i=1}^{N} L_i\}$. Pairing the feature list and the position list makes the dataset.

2-Dimensional Synthetic Dataset. We generate the 2-dimensional synthetic dataset, also following the MC-GTA assumption discussed in Section 5.3:

- Choose hyperparameters: cluster number K, feature dimension D, noise scale α.
- Randomly choose a list of cluster sizes $\{N_1..N_K\}$.
- Randomly generate K points $\{\mathbf{p}_1..\mathbf{p}_K\}$ on the $X-Y$ plane as the metric center of clusters. Randomly generate K 2×2 covariance matrices $\{S_1..S_K\}$. For each $\mathbf{p}_i$, generate N_i points $\{\mathbf{p}_{i,1}..\mathbf{p}_{i,N_i}\}$ from the bivariate Gaussian distribution specified by S_i. For each generated point, its ground-truth cluster label is i.
- Randomly generate K different $D \times D$ ground-truth covariance matrices $\{\Sigma_1, \Sigma_2 \cdots \Sigma_K\}$. It is required that the pairwise Wasserstein-2 distances should all be greater than 1.0. This is to make sure that observations of different clusters are statistically different.
- For each point $\mathbf{p}_{i,j}$, compute its Euclidean distance $d_{i,j}$ to the cluster center $\mathbf{p}_i$. For this point, generate a perturbed covariance matrix $P_{i,j}$ by adding Gaussian noise of scale $d_{i,j}\alpha$ to the ground-truth covariance matrix Σ_i. Sample a D-dimensional feature vector $\mathbf{f}_{i,j}$ from $P_{i,j}$. Similarly the maximum noise scale should be no larger than 10% of the maximum entry in the ground-truth covariance matrix. Then the collection of all $(\mathbf{f}_{i,j}, \mathbf{p}_{i,j})$ makes the dataset.

Choice of hyperparameters: Larger α makes the synthetic data noisier and cluster boundaries fuzzier. Larger k makes model estimation more accurate and stable, thus better clustering results.

A.5.3. Real-world Datasets

(1) Pavement Dataset. This dataset is a sensor-based, originally univariate time series collected by experts. Car sensors collect data while driving on different pavements (cobblestone, dirt and flexible). There are in total 1055 successive, variable-length subsequences of accelerometer readings sampled at 100 Hz. Each subsequence has a label from the aforementioned

three pavement types. We use the first 10 entries of each subsequence as its feature vector, and treat the truncated data as a 1055-long, 10-dimensional multivariate time series. Our task is to put subsequences of the same pavement labels into the same clusters.

The detailed information can be found at https://timeseriesclassification.com/description.php?Dataset=AsphaltPavementType.

(2) Vehicle Dataset. This is a multivariate time series dataset collected by tracking the working status of commercial vehicles (specifically, dumpers) using smart phones and published in the literature. The original paper is here: http://kth.diva-portal.org/smash/record.jsf?pid=diva2

(3) Gesture Dataset. This dataset records hand-movements as multivariate time series. Each movement record is 315-time-step long, and each time-step has a 3-dimensional vector, representing the spatial coordinates of the center of the hand. There are in total 2238 records, each record 315-time-step long, thus the entire length of the dataset is 704,970 time-steps. All the records belong to one of the eight gestures. We randomly shuffle the order of the records, so that it is more challenging. Our task is to cluster time-steps into different gestures. The detailed information about this dataset can be found in https://timeseriesclassification.com/description.php?Dataset=UWaveGestureLibrary.

(4) Climate Dataset. This dataset consists of locations on the earth and their 5 climate attributes (temperature, precipitation, wind, etc.) based on the WorldClim database (https://www.worldclim.org/data/worldclim21.html). The ground-truth labels are the climate types of each location. There are in total 4741 locations, belonging to 14 different climate types. We use the great circle distance as the spatial distance metric for this dataset.

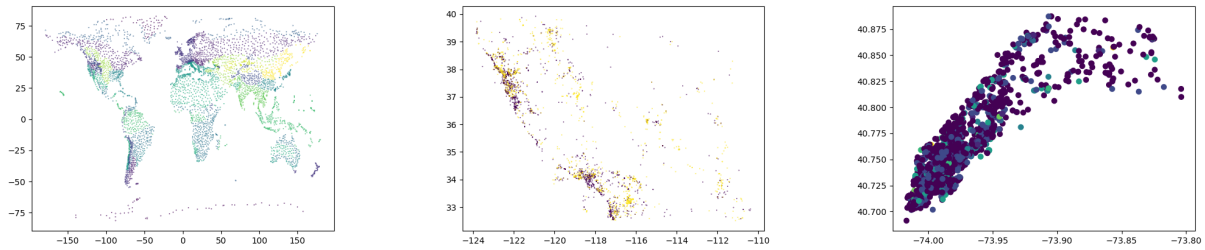
(5) iNaturalist-2018 Dataset. This dataset contains images of species from all over the world together with their geotags (longitude and latitude). The entire dataset is huge and geospatially highly imbalanced (e.g., there are in total 24343 images in the test set, but 10792 out of them are in the contiguous US). We use the ImageNet-pretrained Inception V3 model to embed each image into a 2048-dimensional vector as (Mac Aodha et al., 2019; Mai et al., 2023b;a) did, and reduce it to a 16-dimensional vector using PCA, for the sake of computability of STICC. The ground-truth labels of each image are hierarchical (i.e., from the top kingdom types to the bottom class types), and we use the 6 kingdom types as the cluster labels. Dataset (4) gives an example of spatially-constrained clustering in the multivariate raw feature space, and Dataset (5) extends the boundary to the latent representation space of images.

(6) (7) NYC Foursquare check-in dataset. We use the NYC Check-in data proposed by (Yang et al., 2015). This dataset contains check-in data in New York City by Foursquare, based on social media records. Each record includes VenueId, VenueCateg (POI Type), check-in timestamp (Weekday + Hour) and geospatial coordinate (Longitude + Latitude). We define the feature vector to be the normalized check-in vector, i.e., sum up the Hour attribute grouped by Week, and normalized this 7-dimensional vector. It is a feature vector representing the check-in patterns from Monday to Sunday. For evaluation, we construct 2 sets of ground-truth labels. One is from the **NYC Check-in data itself**: for each observation, we add up the one-hot POI type vectors of its nearest 50 neighbors and normalize it to be the POI embedding of this observation. Then, we cluster over these POI embeddings, and use the clustering labels as the ground-truth. Notice there is no information leak because our algorithm is fitted on check-in data and geo-coordinates only. The other is based on the **Primary Land Use Tax Lot Output (PLUTO) dataset** from NYC Open Data[6]. We extract the land-use records and assign to each observation the nearest land-use record as its ground-truth land-use label. For the sake of data quality, we only use the records of Manhattan and Bronx.

A.6. Further Discussion on Experiment Results

In Dataset (4) and Dataset (5) STICC/MC-GTA without spatial constraints yield lower performance than GMM, because the spatial sampling rate is too low (i.e., there are too few data points within a unit distance). There is a dilemma to STICC/MC-GTA algorithms: in order to obtain an adequate number of samples, we need to increase the sampling radius; however, as the sampling radius gets bigger, the samples become more noisy. In both cases the estimated distributions are inaccurate. Essentially, this problem originates from the balance between data sparsity (when having small neighborhood), and temporal/spatial incontiguity (when having large neighborhood) We address this problem by introducing a global prior. Since GMM can give a fairly good global estimation of the distribution of each cluster, we can use it as the prior distribution and update it in a maximum likelihood/Bayesian way given subsequence/subregion observations. This approach demonstrates a large increase in clustering performance for iNaturalist-2018. Again it demonstrates how important spatial

[6] https://data.cityofnewyork.us/City-Government/Primary-Land-Use-Tax-Lot-Output-PLUTO-/64uk-42ks/data



(a) Climate dataset ground-truth clusters (b) iNaturalist-2018 dataset ground-truth clusters (subset) (c) NYU POI/Land-use dataset

Figure 4. Comparison of the Climate dataset, the iNaturalist-2018 dataset, the NYU POI/Land-use datasets. For visualization we only plot a subset of iNaturalist-2018 (California, two species). It is obvious that 1) the Climate dataset is very sparse and the ground-truth clusters have clear-cut borders, and 2) the iNaturalist/NYU datasets are dense and the ground-truth clusters overlap each other.

metric information is in clustering data points that satisfy local metric constraints. This finding may lead to future works.

A.7. Theoretical Complexity and Empirical Runtime

We denote d as the data dimension, n as the number of data points, and K as the number of clusters. Theoretically, the complexity of MC-GTA is $O(n^2d^2)$. Firstly, we need to estimate covariance matrices for each data point, which is $O(n^2d\min(n,d))$. Since in most cases, $n >> d$, the complexity becomes $O(n^2d^2)$. After estimating the covariances, we compute the pairwise Wasserstein-2 distances, which is again $O(n^2d^2)$, because we need to do matrix multiplication ($O(d^2)$) n^2 times. Finally, we apply a distance-based clustering algorithm like DBSCAN on the Wasserstein-2 distance matrix, which is again $O(n^2)$. Thus the overall time complexity of MC-GTA is $O(n^2d^2)$. This means, theoretically the execution time of TICC/STICC is $C \cdot K$ times of that of MC-GTA.

Next, we show that the time complexity of the SOTA models (TICC and STICC) is $O(C \cdot K \cdot n^2d^2)$, where C is how many iterations it takes to converge, which usually increases as K and n increase.

TICC/STICC needs to 1) compute an initial cluster assignment by kMeans, which is $O(n^2)$; 2) estimate cluster-wise covariance matrices and compute the likelihood of each data point against each cluster, which is $O(K \cdot n^2d^2)$; 3) update cluster assignment, which is reported $O(K \cdot n)$ in the original papers; 4) repeat (1) to (3) C times until convergence. Thus the overall time complexity is $O(C \cdot K \cdot n^2d^2)$.

We also evaluated the empirical time complexity of each clustering algorithm. Please refer to the "RT" column in Table 6. We can see that TICC/STICC is much slower than our MC-GTA. Notice the time TICC/STICC takes highly depends on how many iterations it takes to converge.

Finally, the spatial complexity of both MC-GTA and TICC/STICC is $O(n \cdot d^2)$, since all we need to store is the covariance matrices of each data point.

A.8. Hyperparameter Tuning

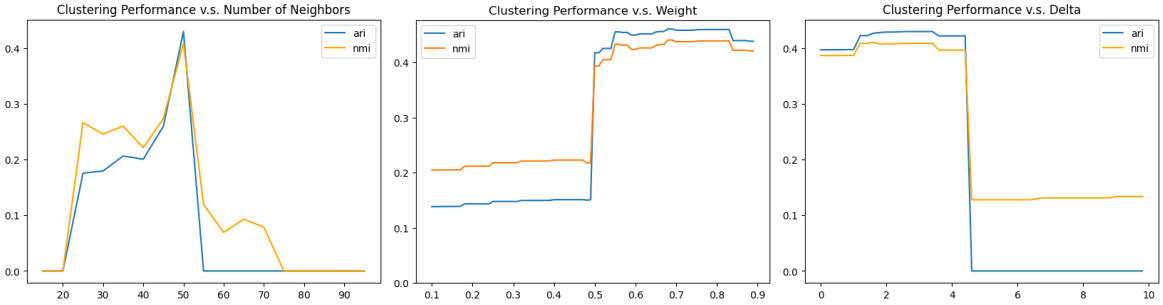
We include an ablation study to investigate the influence of the number of neighbors n, the weight β, and the margin δ using the most complicated iNaturalist 2018 dataset. Figure 5 demonstrates that the search space of single hyperparameters has good convexity. Thus, we can easily and quickly tune the hyperparameters by hierarchical grid search.

For tuning β and δ, we do not need to re-compute the covariance matrices. Instead, we only need to re-run the density-based clustering algorithm, such as DBSCAN. Thus the time complexity of a complete grid search is only $O(A \cdot B \cdot Cn^2)$, where A and B are the grid sizes of n, β and δ.

Unlike MC-GTA, the competing baselines TICC/STICC must re-run the entire algorithm when tuning hyperparameters. That means the complete grid search is $O(A \cdot B \cdot C \cdot K \cdot n^2d^2)$, even if we only tune the most important λ and β hyperparameters.

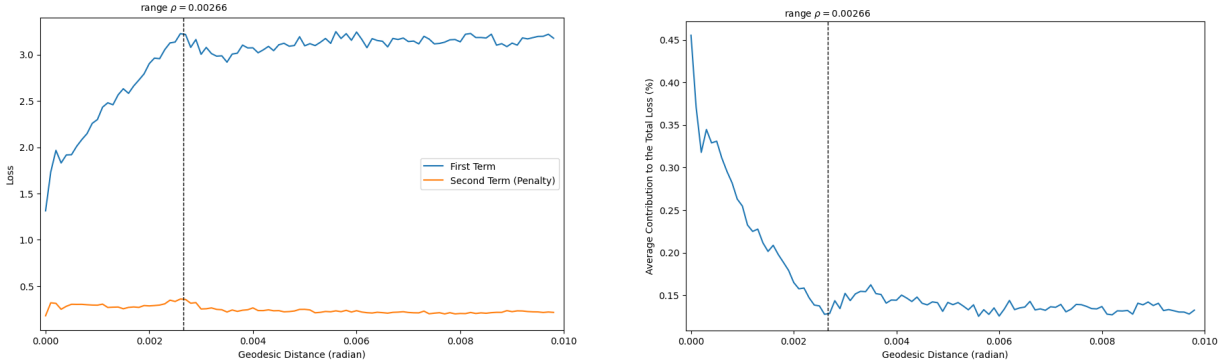
Table 6. Performance and runtime comparison across different model-based clustering algorithms on 1-D (temporal) and 2-D (spatial) real-world datasets. d denotes the feature dimension, c denotes the ground-truth cluster number, and N denotes the size of each dataset. RT denotes the average run-time in seconds. **Bold** numbers and underlined numbers indicate the best and second best performances. TICC applies to 1-D datasets and STICC applies to 2-D datasets. β_0 means there is no temporal/spatial penalty term applied. NC means the algorithm does not converge. MC-GTA-wo/MC-GTA-w represents MC-GTA loss without/with metric information respectively.

Model	Temporal Dataset (1-D)									Spatial Dataset (2-D)					
	Pavement d=10, c=3 N=1,055			Vehicle d=7, c=5 N=16,641			Gesture d=3, c=8 N=704,970			Climate d=5, c=14 N=4,741			iNat2018 d=16, c=6 N=24,343		
	ARI	NMI	RT	ARI	NMI	RT	ARI	NMI	RT	ARI	NMI	RT	ARI	NMI	RT
GMM	28.05	28.74	< 1s	57.87	58.78	3s	2.44	4.15	14s	19.06	34.97	< 1s	21.72	35.91	9s
(S)TICC-β_0	58.54	58.83	383s	40.12	45.86	441s	3.26	6.56	4782s	13.30	30.53	1277s	NC	NC	6881s
(S)TICC	62.27	61.89	508s	50.53	53.68	566s	12.20	23.20	4511s	17.62	37.29	1204s	NC	NC	6325s
MC-GTA-wo	76.10	74.36	14s	63.31	58.60	74s	8.12	33.60	573s	16.63	36.73	746s	21.90	36.47	588s
MC-GTA-w	**77.64**	**77.22**	14s	**65.04**	**59.36**	76s	**26.51**	**55.34**	554s	**20.08**	**40.91**	755s	**42.70**	**40.49**	594s



(a) Tuning the number of neighbors n (b) Tuning the weight β (c) Tuning the margin δ

Figure 5. The performance curve with regard to the grid-searched hyperparameters n, β and δ



(a) The average value of the first and second term in Eq 8 (b) The contribution of the second term in Eq 8 to the total loss.

Figure 6. Analysis of the loss composition in Eq 8. The average contribution of the metric-constraint penalty term to the total loss beyond the range quickly drops down to below 15%, which can be ignored in practice.