Composing Tree Graphical Models with Persistent Homology Features for Clustering Mixed-Type Data

Xiuyan Ni [1] **Novi Quadrianto** [2 3] **Yusu Wang** [4] **Chao Chen** [1]

Abstract

Clustering data with both continuous and discrete attributes is a challenging task. Existing methods often lack a principled probabilistic formulation. In this paper, we propose a clustering method based on a tree-structured graphical model to describe the generation process of mixed-type data. Our tree-structured model factorizes into a product of pairwise interactions, and thus localizes the interaction between feature variables of different types. To provide a robust clustering method based on the tree-model, we adopt a topographical view and compute peaks of the density function and their attractive basins for clustering. Furthermore, we leverage the theory from topology data analysis to adaptively merge trivial peaks into large ones in order to achieve meaningful clusterings. Our method outperforms state-of-the-art methods on mixed-type data.

1. Introduction

Clustering is one of the most widely used techniques in data analysis (Xu & Wunsch, 2005; Jain, 2010). Despite a rich literature on pure continuous data or pure categorical data, the clustering problem remains challenging for *mixed-type data*, i.e., data with both types of attributes (Everitt et al., 2001). Mixed-type data are ubiquitous in real world domains, e.g., social science, biomedicine and finance, where categorical attributes often describe demographic information or questionnaire responses, and continuous attributes often correspond to quantitative measurements. However, only a very limited number of clustering methods have been proposed for such data (Everitt et al., 2001; Huang, 1998). The major challenge is the lack of a good geometric intuition of data on the mixed-type domain; such intuition is the important basis of many successful geometric clustering methods, e.g., k-means (MacQueen et al., 1967), Ward's method (Ward Jr, 1963), DBSCAN (Ester et al., 1996), to name a few. In practice, what usually being done is to convert mixed-type data to either pure continuous or pure categorical domain, and subsequently use existing geometric clustering methods. A metric for directly dealing with mixed-type data is also available, based on Gower's coefficient (1971). The uptake of geometric clustering methods is mostly driven by their lightweight computational requirements. However, these methods lack a well justified underlying probabilistic model, are sensitive to the choice of underlying metric, and do not give a principled answer to the fundamental question of required number of clusters for the data at hand.

In this paper, we propose a probabilistic clustering method for mixed-type data, which admits at least four attractive properties. First, our probabilistic method goes beyond the widely-adopted class conditional independence assumption of feature variables, e.g., as in the latent class model (McCutcheon, 1987). Second, our method is based on the global topographical features, i.e., peaks and mountains, of the density function, rather than the distances between data points. The argument for topographical features is to sidestep a premature specification of the metric space in which our mixed-type data will achieve the best grouping. Third, our method is able to utilize a persistent homology theory to automatically determine the number of clusters in the data. Fourth, the proposed method can be easily parallelized to achieve a competitive running time with respect to many lightweight geometric clustering methods.

From the modeling perspective, we compose tree graphical models with topographical features to achieve a probabilistic mixed-type clustering model. Graphical models provide a way of factorizing a joint probability distribution into a product of local interactions. These local interactions capture dependency among feature variables. While a Bayesian network or a Markov random field can be built with a set of nodes representing each feature variable. The graph structure and parameter estimation can be computationally expensive. By constraining the graph to be a tree, the structure and parameter can be learned efficiently. Other than computational benefits, tree-structured

[1]City University of New York (CUNY), New York, USA [2]University of Sussex, Falmer, United Kingdom [3]National Research University Higher School of Economics, Moscow, Russia [4]Ohio State University, Columbus, USA. Correspondence to: Chao Chen <chao.chen.cchen@gmail.com>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

graphical models also provide a modeling elegance; with a tree structure, we have a factorization that explicitly corresponds to empirical univariate and bivariate marginal distributions. For the bivariate distributions, we can then adapt the product kernel density estimation (Scott, 2015) to capture interaction between continuous-continuous variables, between categorical-categorical variables, and between categorical-continuous variables.

Having modeled the data generation process via a tree graphical model, we are left with finding a robust approach for assigning each data point to its cluster. To achieve this, we adopt a topological perspective, namely, we view a probability distribution as a terrain function, called the *density landscape*, and capture its topographical features as the basis for defining clusters. The *topographical features* include modes (peaks) and their attractive basins. For high-dimension and sparse data, it is natural to have many modes. To avoid over-segmentation of the data and generation of many clusters with only few members, we employ a persistent homology theory (Edelsbrunner & Harer, 2010) to measure the saliency of all modes and merge the trivial ones. Our principled method for clustering mixed-type data respects the underlying topographical features of the density landscape and achieves competitive performance on real data.

1.1. Related Work

Clustering has been extensively studied in machine learning and data mining. Many comprehensive surveys have been produced detailing the landscape of clustering problems and models. Here we will review related work in the context of geometric versus probabilistic clustering methods for mixed-type data and clustering methods that rely on topographical features such as modes and their attractive basins.

Geometric clustering methods A straightforward approach for mixed-type data clustering is to map them into either pure continuous or pure categorical domains before applying a standard clustering method. A metric based on Gower's coefficient (Gower, 1971) has been proposed for mixed-type data, which rescales the difference in all dimensions, continuous or categorical, and take the average. One can apply any distance based method using these metrics. However, all these methods are heuristic; there is no good justification for the underlying geometric intuition of these methods on such a counter-intuitive metric space, despite some successful stories in practice. For example, K-Prototypes algorithm (Huang, 1998) uses a weighted sum of the Euclidean distance and Hamming distance and adopts the K-Means method (Faber, 1994), which iteratively finds the mean of each cluster and re-associates data to different clusters. When the data is pure categorical, the method is called K-Modes (Huang, 1997). Chiu *et al*. (2001) proposed a hierarchical clustering method, in which distance between clusters are measured using their log-likelihood, which treat continuous and categorical domain separately.

Probabilistic clustering methods Graphical models have been applied to clustering before. Zhang (2004) proposed a latent tree model, i.e., a Bayesian tree whose leaf nodes correspond to all observed dimensions and internal nodes are latent variables determining different clusters. Such tree structure can be learned using efficient algorithms (Chen et al., 2012; Liu et al., 2015). However, this method is only restricted to categorical data. Lee & Hastie (2015) proposed a loopy graphical model to model mixed-type data. Their model reduces to a discrete Markov random field when all attributes are categorical, and a Gaussian graphical model when all attributes are continuous. Parameters are learned using pseudo-likelihood estimation (Besag, 1975) and edges are selected using group sparsity penalties (Yuan & Lin, 2006; Huang & Zhang, 2010). However, an efficient inference model is missing in order to apply such model to clustering.

Clustering by mode-seeking The density landscape has been exploited before to extract global properties of the data and to achieve better clustering quality. Mode-seeking methods, i.e., associating data to modes representing clusters, have been proposed before in continuous domain (Cheng, 1995; Comaniciu & Meer, 2002b). But such methods rely on a kernel density estimation, which suffers from the curse of dimensionality and thus do not scale to high dimensions (Wasserman, 2013, chap. 20). Chen & Quadrianto (2016) proposed a mode-seeking method for categorical data clustering. However, their method tends to produce trivial modes/clusters and thus over-segments the data, mainly due to the lack a principled way to merge modes into clusters of proper size.

Persistent homology for merging clusters In recent years, novel approaches have been proposed to merge modes/clusters based on the topographical landscape of the density function. Chazal et al. (2013) used topological persistence to guide the merging of data into clusters. Their method, although theoretically sound, relies on a k-nearest neighbor graph of the data and a given density function, e.g., a kernel density estimation (Silverman, 1986) or a distance from measure (Chazal et al., 2011). This method assumes that the data is a high quality sample of the domain and the k-nearest neighbor graph faithfully captures the topographical characteristics of the distribution. However, this condition is often too strong to assume in practice, where most datasets are relatively sparse. In this paper, we propose to start with mode-seeking, and leverage these modes and the gradient paths as a more accurate account of the density landscape. Our idea proves to be a better solution and a good complement to the theoretical tool. We also refer to other topological and geometrical studies into the

global structures of hierarchical clustering (Eldridge et al., 2015; Carlsson & Mémoli, 2010).

2. Background

A probabilistic graphical model (Koller & Friedman, 2009) consists of a set of inter-dependent random variables $X = (X_1, \ldots, X_D)$, a potential function f, and a graph $G = (\mathcal{V}, \mathcal{E})$. Each element in the node set $\mathcal{V}$ represents one random variable from X. The edges represents the dependence relations between pairs of variables. There are two different kinds of variables in our setting: continuous ones and discrete ones variables. For simplification, we assume each discrete variable takes discrete values $X_i \in \mathcal{L} = \{1, \ldots, L\}$. In this paper, we use discrete and categorical interchangeably and focus on non-ordinal discrete variables, although ordinal discrete variables are of interest in practice as well. In our setting, only Hamming distance can be used for discrete variables.

A value assignment to all random variables $x = (x_1, \ldots, x_D)$ is called a configuration. A *potential function* $f : x \to \mathbb{R}$ assigns to each configuration a real value, which is inversely proportional to the logarithm of the probability distribution, $p(x) = \exp(-f(x) - A)$, where A is the log-partition function. In this paper, we focus on tree structured graphical models, represented by $T = (\mathcal{V}, \mathcal{E})$. For a tree model, the probability and potential of a configuration can be factorized into a product (Bach & Jordan, 2003):

$$p(x) = \prod_{(i,j)\in\mathcal{E}} \frac{p(x_i, x_j)}{p(x_i)p(x_j)} \prod_{k\in\mathcal{V}} p(x_k), \tag{2.1}$$

where $p(x_i, x_j)$ is the bivariate marginal density of the variable X_i and X_j, and $p(x_k)$ is the univariate marginal density of the variable X_k.

When the true distribution can be represented by a tree, we can use the algorithm by Chow & Liu (1968) to reconstruct the tree model. First, we compute the *mutual information* between all pairs of variables:

$$MI_{ij} = \int_{x_i,x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)} dx_i dx_j,$$

using empirical univariate and bivariate marginals. The integral is replaced by sum when X_i and X_j have discrete values. Next, we compute the maximum spanning tree of a complete graph with D nodes, using the mutual information as edge weights. The computed tree is the desired tree model with the optimal KL-divergence from the true tree distribution (Liu et al., 2011). More details of the selection of the models for univariate and bivariate densities will be given in Section 3.

3. Method

Our method first estimates the underlying probabilistic density function from given data. We choose tree-models as they strike a elegant balance between computational efficiency and flexibility of the model. Next, we propose to cluster data based on the density landscape: associating data with modes/peaks of the density, and merge them based on advanced persistent homology theory. First, we formalize the definition of modes in the mixed-type domain. Then we present algorithms for modes-seeking (Section 3.2) and for modes-merging (Section 3.3).

We first formalize what a mode is in a D-dimensional mixed-type data domain. Our definition is not restricted to the underlying model. Denote by $\mathcal{I}_d$ and $\mathcal{I}_c$ the index sets of discrete- and continuous-valued random variables. Denote by $\mathrm{dist}_H(x, x')$ the Hamming distance between x and x' within the discrete dimensions, and $\mathrm{dist}_{L2}(x, x')$ the L_2 distance within the continuous dimensions. We call a discrete neighborhood of x with radius $\delta > 0$ as all elements with no more than δ Hamming distance and zero Euclidean distance from x, formally,

$$\mathcal{N}_\delta^d(x) = \{x' \mid \mathrm{dist}_d(x, x') \le \delta \wedge \mathrm{dist}_c(x, x') = 0\}.$$

Similarly, we define a continuous neighborhood of x with radius $\epsilon > 0$ as

$$\mathcal{N}_\epsilon^c(x) = \{x' \mid \mathrm{dist}_d(x, x') = 0 \wedge \mathrm{dist}_c(x, x') \le \epsilon\}.$$

Given a probability density function, $p(X)$, a mode is a local maximum in both the continuous neighborhood and discrete neighborhood, formally:

Definition 1 (Modes). *A point $x \in \mathcal{X}$ is a mode if and only if there exists positive numbers $\epsilon > 0$ and $\delta > 0$ such that (1) $p(x) \ge p(x')$ for any $x' \in \mathcal{N}_\epsilon^c(x)$; and (2) $p(x) \ge p(x')$ for any $x' \in \mathcal{N}_\delta^d(x)$.*

It suffices to use the smallest positive integer for the discrete neighborhood, $\delta = 1$. In this paper, we focus on a tree-structured graphical model. Next, we describe our tree model in details within the mixed-type setting.

3.1. Instantiating the Tree Model

We formalize the univariate and bivariate marginal densities $p(x_i)$ and $p(x_i, x_j)$ in the tree model (Eq. (2.1)). We assume a set of N data $\{y^1, y^2, \cdots, y^N\}$ is given. For discrete dimensions, we use Multinoulli distribution with Dirichlet prior $\alpha = 1, \forall i, j \in \mathcal{I}_d$:

$$p(x_i) = \frac{N_{x_i} + 1}{N + L}, \text{ with } N_{x_i} = \sum_{n=1}^{N} [\![y_i^n = x_i]\!],$$

$$p(x_i, x_j) = \frac{N_{x_i,x_j} + 1}{N + L^2},$$

$$\text{with } N_{x_i,x_j} = \sum_{n=1}^{N} [\![y_i^n = x_i \wedge y_j^n = x_j]\!].$$

For continuous variables, we use one-dimensional kernel density estimation for univariate density, and product kernel (Scott, 2015) for univariate and bivariate marginal density. Formally, $\forall i, j \in \mathcal{I}_c$,

$$p(x_i) = \frac{1}{N}\sum_{n=1}^{N} \mathrm{K}_{h_{1i}}(y_i^n - x_i), \text{ and}$$
$$p(x_i, x_j) = \frac{1}{N}\sum_{n=1}^{N} \left\{\mathrm{K}_{h_{2i}}(y_i^n - x_i)\,\mathrm{K}_{h_{2j}}(y_j^n - x_j)\right\}, \tag{3.1}$$

We use a one-dimensional Gaussian kernel, denoted as $\mathrm{K}_h(z) = \frac{1}{\sqrt{2\pi}h}\exp\left(-\frac{z^2}{2h^2}\right)$. Following standard non-parametric statistics literature (Fan & Gijbels, 1996; Tsybakov, 2009), the kernel bandwidths for univariate and bivariate density are chosen as

$$h_{ti} = 1.06 \cdot \min\left\{\sigma_i^*, \frac{q^*_{i,0.75} - q^*_{i,0.25}}{1.34}\right\} \cdot N^{-\frac{1}{2\beta+t}}, t = 1, 2,$$

where σ_i^*, $q^*_{i,0.75}$ and $q^*_{i,0.25}$ are the standard deviation, the 75% and 25% sample quantiles of X_i, respectively. The variable β is the order of the kernel (Fan & Gijbels, 1996) and is set to 2 by default.

The choice of a product kernel is justified by two reasons. First, a product kernel reduces to the product of one-dimensional kernels, which are more reliable that a direct 2D kernel density estimation. Second, the product kernel proves to be convenient to be adopt to bivariate densities for variables with mixed-type as follows. For a mixed-type pair of variables, (X_i, X_j), $i \in \mathcal{I}_c, j \in \mathcal{I}_d$, we take the limit of h_{2i} to zero in the product kernel formula (Equation (3.1)). The first kernel becomes the Dirac-delta function, leading to the following bivariate marginal

$$p(x_i, x_j) = \frac{1}{N}\sum_{n=1}^{N} \left\{[\![y_j^n = x_j]\!]\,\mathrm{K}_{h_{2i}}(y_i^n - x_i)\right\}.$$

Building the tree model. Using these empirical univariate and bivariate marginal densities, we estimate all pairwise mutual information, and then compute the tree $(\mathcal{V}, \mathcal{E})$ using the Chow-Liu algorithm. Plugging the univariate and bivariate marginal densities into Eq. (2.1), we have the complete density distribution (the tree model). Next, we present our algorithm for finding the modes over the density landscape of the computed model.

3.2. Mode-Seeking Algorithm

Our algorithm assigns each data to a mode via a gradient ascent procedure. For a mixed-domain, a gradient is not well defined. Following the definition of modes (Def. 1), we formulate a gradient step as an optimization within either the continuous neighborhood $\mathcal{N}_\epsilon^c(x)$ or the discrete neighborhood $\mathcal{N}_\delta^d(x)$, with $\delta = 1$. The two procedures have to be taken alternatively in order to continue increasing the probability until a mode is reached.

Our algorithm starts at each data, s, iteratively walks to a nearby point with bigger probability until convergence. The final position is the mode of interest and will be associated with the data, s. For ease of computation, we use the potential function $f(x)$ instead of the probability density function:

$$f(x) = -\sum_{(i,j)\in\mathcal{E}} \log p(x_i, x_j) - \sum_{i\in\mathcal{V}} (1 - d_i)\log p(x_i), \tag{3.2}$$

in which d_i is the degree of node i in the tree. It is easy to verify that $p(x) \propto -f(x)$. Therefore, modes of $p(x)$ are the local minima of $f(x)$, following the same definition in Def. 1. We follow the aforementioned iterative procedure, except at each step, we find a nearby point with smaller potential.

At each step of the algorithm, we first update all discrete variables until no better elements exist within the discrete neighborhood $\mathcal{N}_\delta^d(x)$ with $\delta = 1$. Next, we update all continuous variables using gradient descent, until the gradient of f at continuous dimensions $\nabla_c f$ becomes zero. Our main algorithm is summarized in Alg. 1.

Algorithm 1 Mode-Seeking Algorithm

Input: Data $\mathcal{D} = \{s_i \mid i = 1, \cdots, N\}$; a potential function f.
Output: A set of modes, $\mathcal{M}$; mode indices associated to each data $\{c_i \mid i = 1, \cdots, N\}$
$\mathcal{M} \leftarrow \emptyset$
for $i = 1$ **to** N **do**
  $x \leftarrow s_i$
  repeat
    repeat
      $x \leftarrow \mathrm{argmin}_{z\in\mathcal{N}_1^d(x)} f(z)$
    until x converges
    repeat
      $x \leftarrow x - \eta\nabla_c f$
    until x converges
  until x converges
  if $x \notin \mathcal{M}$ **then**
    $\mathcal{M} \leftarrow \mathcal{M} \cup \{x\}$
  end if
  $c_i \leftarrow$ the index of x in $\mathcal{M}$
end for

Here η is the stepsize. The best neighbor within Hamming distance one, $\mathrm{argmin}_{z\in\mathcal{N}_1^d(x)} f(z)$, can be computed using dynamic programming. This can be achieved by directly adapting the algorithm by (Chen & Quadrianto, 2016).

It remains to compute the gradient of f in the contin-

uous domain, $\nabla_c f$. For each continuous variable $i \in \mathcal{I}_c$, relevant terms in the energy function (Eq. (3.2)) can be divided into three groups, the univariate term, the bivariate terms with a continuous neighbor, $j \in \mathcal{I}_c$, and the bivariate terms with a discrete neighbor, $j \in \mathcal{I}_d$. Treating them differently, the partial derivative:

$$\frac{\partial f(x)}{\partial x_i} = -(1-d_i)\frac{\sum_{n=1}^{N} \mathrm{K}_{h_{1i}}(y_i^n - x_i)\frac{y_i^n - x_i}{h_{1i}^2}}{\sum_{n=1}^{N} \mathrm{K}_{h_{1i}}(y_i^n - x_i)}$$
$$- \sum_{j \in \mathcal{I}_c:(i,j)\in\mathcal{E}} \frac{\sum_{n=1}^{N} \mathrm{K}_{h_{2i}}(y_i^n - x_i)\,\mathrm{K}_{h_{2j}}(y_j^n - x_j)\frac{y_i^n - x_i}{h_{2i}^2}}{\sum_{n=1}^{N} \mathrm{K}_{h_{2i}}(y_i^n - x_i)\,\mathrm{K}_{h_{2j}}(y_j^n - x_j)}$$
$$- \sum_{k \in \mathcal{I}_d:(i,j)\in\mathcal{E}} \frac{\sum_{n=1}^{N} \mathrm{K}_{h_{2i}}(y_i^n - x_i)[\![y_k^n = x_k]\!]\frac{y_i^n - x_i}{h_{2i}^2}}{\sum_{n=1}^{N} \mathrm{K}_{h_{2i}}(y_i^n - x_i)} \tag{3.3}$$

Algorithm 2 Merging Data Using Topological Persistence

Input: $\widehat{\mathcal{G}} = (\widehat{\mathcal{V}}, \widehat{\mathcal{E}})$, density function $p : \widehat{\mathcal{V}} \to \mathbb{R}^+$, persistence threshold τ
Output: Clusters $\mathcal{C}$
$\mathcal{C} \leftarrow \emptyset$
Sort elements in $\widehat{\mathcal{V}}$ according to the density function values, so that $p(v_i) \geq p(v_{i+1}), \forall v_i, v_{i+1} \in \widehat{\mathcal{V}}$.
for $i = 1$ **to** $|\widehat{\mathcal{V}}|$ **do**
  $nbd \leftarrow \{v_j \mid (v_i, v_j) \in \widehat{\mathcal{E}} \wedge j < i\}$
  // neighbors of v_i with smaller indices (bigger p)
  if $nbd = \emptyset$ **then**
    create a new cluster $c = \{v_i\}$
    $\mathrm{birth}(c) \leftarrow p(v_i)$
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{c\}$
  else
    $\mathcal{C}_{nbd} \leftarrow$ all clusters containing nodes in nbd
    $c_{max} \leftarrow \mathrm{argmax}_{c \in \mathcal{C}_{nbd}} \mathrm{birth}(c)$
    for all $c \in \mathcal{C}_{nbd}$ **and** $c \neq c_{max}$ **do**
      $\mathrm{persistence}(c) \leftarrow \mathrm{birth}(c) - p(v_i)$
      if $\mathrm{persistence}(c) < \tau$ **then**
        // merge c into c_{max}
        $c_{max} \leftarrow c_{max} \cup c$
        $\mathcal{C} \leftarrow \mathcal{C} \backslash \{c\}$
      end if
    end for
    // assign v_i to c_{max}
    $c_{max} \leftarrow c_{max} \cup \{v_i\}$
  end if
end for

3.3. Merging Clusters Using Topological Persistence

The modes computed in Alg. 1 provide a clustering of the data. However, in practice, the data is often relatively sparse. In such cases, the method tends to produce a large number of modes, and thus over-segments the data into small clusters. There are ways to merge these small clusters (Ward Jr, 1963; Day & Edelsbrunner, 1984). But they rely on a distance metric to measure similarities between clusters. Instead, we propose a principled approach that is only based on the density landscape, i.e., the topographical features such as peaks, ridges, valleys. Our method is built on the theory of *persistent homology*. We focus on zero-dimensional topological structures in this paper, although the theory is much more general.

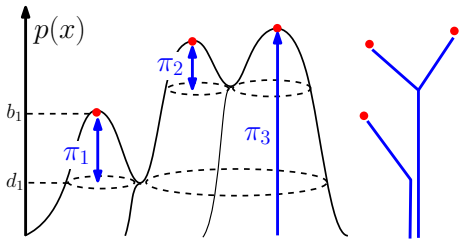

Figure 1. Left: the density landscape with three modes (red). The z axis is the density function. The persistence (length of the blue vertical bars, π_1, π_2, π_3) measure their saliency. The function values of the ends of these bars are the birth and death times (e.g., b_1 and d_1 for the left mode). The global maxima has infinite persistence, $\pi_3 = +\infty$. Right: an illustration of the merging hierarchy of clusters, when $\pi_1 > \tau > \pi_2$. The second mode is merged into the third one when the first mode remains separated. The first mode remains separated as its persistence is bigger than the threshold. Two clusters remain after merging. Although in this example, the domain is pure continous, we believe that the intuition carries to the mixed domain.

Persistence of modes. We estimate the saliency of a peak (mode) using its "relative height", namely, the difference between its height and the level at which its basin of attraction meets the one of another higher mode. Formally, we filter the domain using a function value threshold t from $+\infty$ to $-\infty$. As t decreases, we monitor the topological changes of the progressively growing *superlevel set*, $\mathcal{X}^t = \{x \in \mathcal{X} \mid p(x) \geq t\}$, that is, the domain whose probability density value is no smaller than t. Each mode attributes to the birth of a new connected component in the superlevel set and the component is killed when it meets another component created by a higher mode. The density value of the creating mode and the density value of the point at which the two components meet (called a *saddle*) are called the *birth* and *death times*, and their difference, called the *persistence*, measures the saliency of this mode. See Figure 1 for an illustration.

The merging of connected components as we decrease the threshold t provides a natural way to merge modes; when two connected components meet, we merge them if one of them has $\leq \tau$ persistence (Figure 1). This gives us a principled way to merge modes. Based on the convergence of tree-model estimation (Liu et al., 2011) and the stability

of persistent homology (Cohen-Steiner et al., 2007), this method is guaranteed to be robust to noise and L_∞ perturbation of the density function.

Sample-based persistence computation. Given a dense uniform sampling of the whole domain $\mathcal{X}$, we can trust these samples will describe the density landscape faithfully. In practice, however, a uniform sampling will have exponential size to the dimension. Chazal et al. (2013) used the k-nearest neighbor graph of the input data, $\mathcal{D}$, assuming they are good samples from the density function. However, in practice, the data is often relatively sparse and cannot represent the landscape well enough to produce a high quality mode-merging hierarchy. In fact, it is very likely that the modes are not included in the data and thus the birth time (as well as the persistence) will be under-estimated. See Figure 2(left) for an illustration.

In this paper, we propose to compute persistence based on all points we encountered during the mode-seeking procedure. In Algorithm 1, we collect the point x computed after each iteration (after line 12). The gradient step also provides a natural edge connecting these points. This tree structured graph give us a high-quality description of the attractive basin of each mode. This provides us a well-suited underlying graph describing the density landscape. See Figure 2(right). Finally, to ensure the graph is fully connected, and the space between modes are well described, we add edges (green edges) connecting points from neighboring attractive basins, as well as the lowest point along these edges (green markers). Note that this is the only time when the distance metric plays a role in our model. We use a sum of the Hamming distance and Euclidean distance.

Algorithm. Given a graph $\widehat{\mathcal{G}} = (\widehat{\mathcal{V}}, \widehat{\mathcal{E}})$, in which each node is assigned a probability density, we compute the persistence-based merge tree as follows. Sort all nodes in decreasing order of their density function values. Add them into the superlevel set one-by-one. To add a node v_i, we check whether it is adjacent to any nodes that have been included. If not, v_i, which must be a mode itself, creates a new connected component with the birth time $p(v_i)$. If v_i is connected to multiple existing connected components, we keep the one with the earliest birth time, c_{max}, and merge some others into c_{max}. In particular, for each other adjacent connected component, we check whether its life length so far is less than τ. The ones with $\leq \tau$ life length will be merged into c_{max}. We add v_i into the connected component c_{max} See Figure 3 for an illustration. See Alg. 2 for the pseudocode.

4. Experiments

We compare our methods with existing clustering methods on several real world mixed-type datasets from UCI repository (Lichman, 2013): Contraceptive Method Choice dataset (CMC), Credit Approval dataset (CRX), German Credit Approval (German), and Statlog Heart Disease dataset (Heart). See the table below for more details. All datasets have 60% to 70% of the features being discrete.

Table 1. Datasets

Data	# of samples	Dimension	# of clusters
CMC	1473	9	3
Heart	297	13	5
CRX	653	15	2
German	1000	20	2

Our method can be straightforwardly parallelized. We run the mode-seeking for all data points (the for-loop in Alg. 1) in parallel. On average, the mode-seeking of a single data takes 6 gradient ascent steps and 5.87 seconds. On a cluster with 48 cores, our program finishes within 3 minutes for any of the datasets. If running in a sequential manner, the time will be linear to the dataset size. After all data are processed, we collect all relevant points and run a persistence-based merging sequentially. This step takes less than 20 seconds for any of the datasets. The persistence-based merging depends on a threshold τ. It is hard to select a universal one due to the large variation among datasets. Instead, we choose the τ for each dataset so that the desired the nubmer of clusters remain after merging. This is a fair comparison; all clustering methods we compare with use an oracle number of clusters. We empirically set the parameter δ to one. Using a bigger δ hurts the performance as it would try to 'smooth' the landscape in the categorical domain.

All methods can be grouped into five different groups, based on the underlying domain and the approach. The first group assumes a continuous domain and an Euclidean metric. We project the mixed-type data into the continuous domain and directly apply such methods, including k-means (Faber, 1994), Affinity Propagation (Frey & Dueck, 2007), Mean Shift (Cheng, 1995; Comaniciu & Meer, 2002a), Spectral Clustering (Kamvar et al., 2003), Ward's algorithm (Ward Jr, 1963), Agglomerative clustering (Day & Edelsbrunner, 1984) and DBSCAN (Ester et al., 1996).

The second group are methods designed for pure categorical domain, e.g., K-Modes (Huang, 1997), ROCK (Guha et al., 1999), mixture of multinoulli (latent class analysis) (McCutcheon, 1987). We convert mixed-type data into categorical data by thresholding continuous values at the median. We also include Affinity Propagation, Spectral Clustering and DBSCAN in this group; these methods can be applied to any distance metrics. We compute pairwise Hamming distance between data as the input of these three methods.

For the third group, we use these three methods, but using a distance matrix based on Gower's coefficient (Gower, 1971), which was designed specifically for mixed-domain. The fourth group uses a simply sum of the Euclidean distance (restricted to continuous dimensions) and Hamming distance (restricted to categorical dimensions). A good rep-

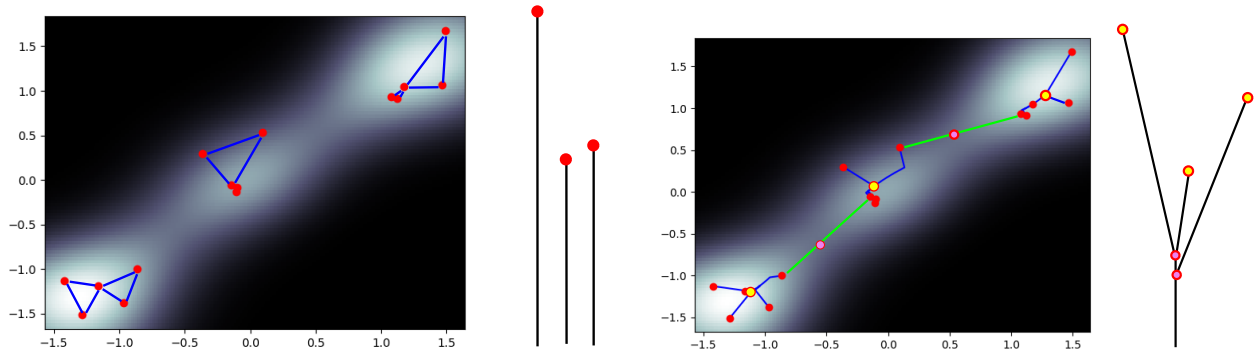

Figure 2. Comparison of the topological methods: (Chazal et al., 2013) v.s. ours. The data is sampled from a mixture of three 2D Gaussian functions. Left: k-nearest neighbor graph with $k = 2$ on the original data (red markers), plotted on the density function. Next to it is the merging tree of clusters. Since the three components are disconnected, we cannot accurately estimate a proper merging time between them. We also consider the middle and upper-right mode equally salient, as we underestimates the birth time of the upper-right mode using the probability of one of its data point. Right: our method using the gradient steps as edges (blue) and explicit edges connecting adjacent attractive basins (green), as well as more points collected during the procedure. Capturing the actual modes gives us an accurate estimation of the birth time of each basin, furthermore, the lowest point along the green edge (purple marker) gives us a good estimation of the saddle points and thus the death time of each connected component. In the illustration, we use $\tau = \infty$ so all three modes merged into a single cluster. The goal is to show that our method better captures the merging tree compared with (Chazal et al., 2013). The domain is $\mathbb{R}^2$. But we believe that the intuition carries to the high-dimensional mixed domain.

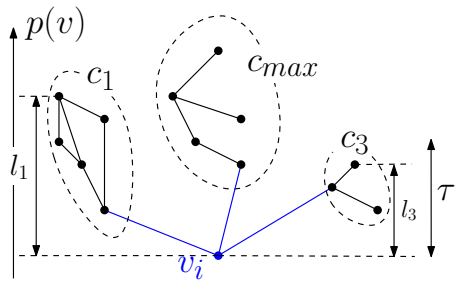

Figure 3. Adding a new node v_i into the superlevel set. Component c_3 is merged into c_{max} as its life length ($l_3 = \text{birth}(c_3) - p(v_i)$) is smaller than τ. The first component c_1 remains separated after v_i as its life length l_1 is bigger than τ.

resentative in such group is K-Prototypes (Huang, 1998). We again applied the three methods (Affinity, Spectral and DBSCAN) on this new metric.

In the last group, we compare our method and a few other topological methods. We compare to the method using only modes for clustering. This is essentially an adaptation of (Chen & Quadrianto, 2016) to the mixed-type domain. We also compare to (Chazal et al., 2013) by computing the persistence on the k-nearest neighbor graph, using our tree-model as the underlying density estimation. Finally, we also show the result of our method.

The results are listed in Table 2. We use the Adjusted Mutual Information (AMI) (Vinh et al., 2010) and Adjusted Rand Score (ARS) (Hubert & Arabie, 1985) to evaluate all methods. For all methods requiring random initializations, we run each one for 10 times and take the average performance. When necessary, we provide a true number of clusters as an oracle. The cells with N/A correspond to the cases when the program crashes. It is most likely because the Gower's coefficient and Hamming distance does not give us a well-conditioned distance matrix for the spectral clustering method.

Discussion. Our method outperforms most methods from all other four groups, using different types of metrics. We also observe that a few methods based on pure categorical domain are quite competitive. Similarly, K-prototype, a popular tool for mixed-type data, has good performance on some data. Outperforming other topological methods (modes only and persistence only) demonstrate the significance of our contribution.

Our current experiments assume the correct number of clusters is given. It is possible to prove that with sufficient samples and the correct threshold τ, the persistence-based clustering can find the correct number of cluster and the right clustering for most data points in a sense similar to the elegant result in (Chazal et al., 2013). A closely related theoretical result is in (Eldridge et al., 2015), which shows that the hierarchical clustering tree constructed by a similar merging procedure is consistent for points sampled from a nice density distribution over $\mathbb{R}^D$.

Table 2. Results on Real Datasets

	Adjusted Random Score (ARS)				Adjusted Mutual Information (AMI)			
	CMC	Heart	CRX	German	CMC	Heart	CRX	German
Continuous Domain (Euclidean Metric)								
KMeans	0.017	0.142	0.005	0.016	0.029	0.164	0.020	0.003
Affinity	0.001	0.000	0.014	0.003	0.013	0.000	0.051	0.015
MeanShift	-0.004	0.012	0.004	-0.007	0.006	0.001	0.001	0.000
Spectral	-0.002	0.123	-0.006	0.003	0.004	0.158	0.032	-0.001
Ward	0.019	0.030	0.001	-0.026	0.031	0.166	0.002	0.008
Agglomerative	-0.013	0.026	0.001	0.006	0.012	0.013	0.000	0.000
DBSCAN	-0.014	0.075	0.001	-0.038	0.016	0.023	0.044	0.011
Categorical Domain (Hamming Metric)								
Kmodes	0.020	0.108	0.230	0.020	0.017	0.125	0.178	**0.017**
ROCK	0.000	0.000	0.001	0.000	0.005	0.002	0.011	0.000
Mixture (LCA)	0.010	0.127	0.006	**0.068**	0.025	0.109	0.022	0.016
Affinity	0.003	0.024	0.022	0.002	0.018	0.077	0.086	**0.017**
Spectral	N/A	-0.003	-0.001	0.000	N/A	-0.010	0.000	0.000
DBSCAN	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Mixed Domain (Gower's Coefficient)								
Affinity	0.001	0.042	0.034	0.002	0.013	0.091	0.096	0.016
Spectral	N/A	-0.022	-0.001	0.000	N/A	0.004	0.006	0.000
DBSCAN	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Summed Distance Metric (Euclidean + Hamming)								
Affinity	0.006	0.000	0.030	0.000	0.013	0.091	0.096	0.016
Spectral	N/A	0.002	0.002	0.003	N/A	0.003	0.004	0.001
DBSCAN	0.000	0.000	-0.010	0.000	0.000	0.000	0.019	0.000
K Prototype	**0.026**	0.066	0.003	0.052	0.030	0.040	0.006	0.010
Topological Methods								
Modes Only	0.000	0.000	0.000	0.000	0.008	0.000	0.000	0.000
Persistence Only	-0.001	0.083	-0.004	-0.005	0.000	0.028	0.004	0.001
Ours (Modes + Pers.)	**0.026**	**0.285**	**0.318**	0.040	**0.038**	**0.183**	**0.230**	0.006

5. Conclusions

In this paper, we propose a probabilistic clustering method for mixed-type data. We design a tree-structured graphical model for the mixed-type domain. We also develop methods based on a topographical view of the density landscape. We design algorithms to capture modes of the density landscape and merge trivial modes based on the theory of persistent homology.

Acknowledgments. XN and CC have been partly funded by the grant PSC-CUNY 69844-00 47. NQ has been partly funded by the Russian Academic Excellence Project '5-100'. YW has been partly supported by the grant NSF DMS-1547357. The authors gratefully acknowledge use of the services and facilities of CUNY Queens Colleges Center for Computational Infrastructure for the Sciences (CCIS).

References

Bach, Francis R and Jordan, Michael I. Beyond independent components: trees and clusters. *The Journal of Machine Learning Research*, 4:1205–1233, 2003.

Besag, Julian. Statistical analysis of non-lattice data. *The statistician*, pp. 179–195, 1975.

Carlsson, Gunnar and Mémoli, Facundo. Characterization, stability and convergence of hierarchical clustering methods. *Journal of Machine Learning Research*, 11: 1425–1470, 2010.

Chazal, Frédéric, Cohen-Steiner, David, and Mérigot, Quentin. Geometric inference for measures based on distance functions. *Foundations of Computational Mathematics*, 11(6):733–751, 2011.

Chazal, Frédéric, Guibas, Leonidas J, Oudot, Steve Y, and Skraba, Primoz. Persistence-based clustering in Rieman-

nian manifolds. *Journal of the ACM (JACM)*, 60(6):41, 2013.

Chen, Chao and Quadrianto, Novi. Clustering high dimensional categorical data via topographical features. In *International Conference on Machine Learning (ICML)*, 2016.

Chen, Tao, Zhang, Nevin L, Liu, Tengfei, Poon, Kin Man, and Wang, Yi. Model-based multidimensional clustering of categorical data. *Artificial Intelligence*, 176(1):2246–2269, 2012.

Cheng, Yizong. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

Chiu, Tom, Fang, DongPing, Chen, John, Wang, Yao, and Jeris, Christopher. A robust and scalable clustering algorithm for mixed type attributes in large database environment. In *Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 263–268. ACM, 2001.

Chow, C and Liu, C. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

Cohen-Steiner, David, Edelsbrunner, Herbert, and Harer, John. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.

Comaniciu, D. and Meer, P. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on (PAMI)*, 24(5): 603–619, 2002a.

Comaniciu, Dorin and Meer, Peter. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5): 603–619, 2002b.

Day, William HE and Edelsbrunner, Herbert. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of classification*, 1(1):7–24, 1984.

Edelsbrunner, Herbert and Harer, John. *Computational Topology: an Introduction*. AMS, 2010.

Eldridge, Justin, Belkin, Mikhail, and Wang, Yusu. Beyond hartigan consistency: Merge distortion metric for hierarchical clustering. In *COLT*, pp. 588–606, 2015.

Ester, Martin, Kriegel, Hans-Peter, Sander, Jörg, Xu, Xiaowei, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pp. 226–231, 1996.

Everitt, Brian S., Landau, Sabine, Leese, Morven, and Stahl, Daniel. *Cluster Analysis (5th Edition)*. Wiley, 2001.

Faber, Vance. Clustering and the continuous k-means algorithm. *Los Alamos Science*, 22(138144.21), 1994.

Fan, Jianqing and Gijbels, Irene. *Local polynomial modelling and its applications: monographs on statistics and applied probability 66*, volume 66. CRC Press, 1996.

Frey, Brendan J and Dueck, Delbert. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

Gower, John C. A general coefficient of similarity and some of its properties. *Biometrics*, pp. 857–871, 1971.

Guha, Sudipto, Rastogi, Rajeev, and Shim, Kyuseok. ROCK: A robust clustering algorithm for categorical attributes. In *International Conference on Data Engineering (ICDE)*, pp. 512–521, 1999.

Huang, Junzhou and Zhang, Tong. The benefit of group sparsity. *The Annals of Statistics*, 38(4):1978–2004, 2010.

Huang, Zhexue. A fast clustering algorithm to cluster very large categorical data sets in data mining. In *DMKD*, pp. 0, 1997.

Huang, Zhexue. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery*, 2(3):283–304, 1998.

Hubert, Lawrence and Arabie, Phipps. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

Jain, Anil K. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

Kamvar, Sepandar D, Klein, Dan, and Manning, Christopher D. Spectral learning. In *Proceedings of the 18th international joint conference on Artificial intelligence*, pp. 561–566. Morgan Kaufmann Publishers Inc., 2003.

Koller, Daphne and Friedman, Nir. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Lee, Jason D and Hastie, Trevor J. Learning the structure of mixed graphical models. *Journal of Computational and Graphical Statistics*, 24(1):230–253, 2015.

Lichman, M. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

Liu, Han, Xu, Min, Gu, Haijie, Gupta, Anupam, Lafferty, John, and Wasserman, Larry. Forest density estimation. *Journal of Machine Learning Research*, 12:907–951, 2011.

Liu, Teng-Fei, Zhang, Nevin L, Chen, Peixian, Liu, April Hua, Poon, Leonard KM, and Wang, Yi. Greedy learning of latent tree models for multidimensional clustering. *Machine learning*, 98(1-2):301–330, 2015.

MacQueen, James et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pp. 281–297. Oakland, CA, USA., 1967.

McCutcheon, Allan L. *Latent class analysis*. Number 64. Sage, 1987.

Scott, David W. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

Silverman, Bernard W. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

Tsybakov, Alexandre B. Introduction to nonparametric estimation. revised and extended from the 2004 french original. translated by vladimir zaiats, 2009.

Vinh, Nguyen Xuan, Epps, Julien, and Bailey, James. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11 (Oct):2837–2854, 2010.

Ward Jr, Joe H. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

Wasserman, Larry. *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013.

Xu, Rui and Wunsch, Donald. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3): 645–678, 2005.

Yuan, Ming and Lin, Yi. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

Zhang, Nevin L. Hierarchical latent class models for cluster analysis. *The Journal of Machine Learning Research*, 5: 697–723, 2004.