Statistical Topological Data Analysis

Bhuwan Dhingra
Language Technology Institute
Carnegie Mellon University
Pittsburgh, PA 15213
bdhingra@andrew.cmu.edu

Chaitanya Ahuja
Language Technology Institute
Carnegie Mellon University
Pittsburgh, PA 15213
cahuja@andrew.cmu.edu

1 Topological Data Analysis

Topological Data Analysis (TDA) refers to data analysis methods which study properties such as shape, topology and connectedness of the data. In this project we plan to study some of the key techniques used in TDA along with their established theoretical properties from a statistical perspective. TDA provides the data analyst with a set of tools to visualize and analyze a data sample. Below we give an overview of three such techniques summarized from [15], and then we take a deep dive into the density clustering and present some results from [6] which discusses how to construct confidence sets for density tree estimates. Assume that we observe a data sample $X_1, X_2, \dots X_N \sim P$, where $X_i \in \mathbb{R}^d$.

Density clustering refers to methods which use an estimate of the true density (such as the Kernel Density Estimator (KDE) $\hat{p}_h(x)$) to find clusters where the data concentrates. This can be done by first constructing the estimated upper level sets $\hat{L}_t = \{x : p(x) > t\}$, and then finding its connected components (which are the clusters). Varying t gives a whole set of clusters at every level, which can be conveniently visualized as a tree known as the *density tree*. Statistical inference on these trees is studied in [6], and we give a detailed overview in the following section.

Manifold learning starts with the assumption that the density P is supported (mostly) on a subset S of dimension r, where $r < d$. The goal then, is to either find an estimate of this subset $\hat{S}$, or embed the observed data in an r-dimensional subspace while preserving its topological properties. In this project we focus on the first problem, an estimator for which can be constructed as $\hat{S} = \cup_{i=1}^{N} B(X_i, \epsilon)$, i.e., the union of ϵ-radius balls around each observed sample. Detailed analysis of this estimator is given in [10] for the no noise case, and in [11] in the presence of noise. While manifold learning assumes that the density is supported on S, an alternative is to find *density ridges* of small dimension where most of the density concentrates. These can be estimated from $\hat{p}_h(x)$ using the SCMS algorithm described in [12]. Statistical properties of the estimates of density ridges are studied in [4].

Persistent homology studies the topological properties of the data sample at varying scales. Specifically, let $L_\epsilon = \{x : d_S(x) < \epsilon\}$ denote the lower level set of the distance function for a set S given by $d_S(x) = \inf_{y \in S} \|x - y\|$. Persistent homology studies the connected components and holes of L_ϵ as ϵ is varied. The *birth* and *death* times (in terms of ϵ) for each hole and connected component can be plotted to give what is known as the *persistence diagram* D. An estimator $\hat{L}_\epsilon$ for the lower level is given by the same union of balls $\cup_{i=1}^{N} B(X_i, \epsilon)$ used for estimating S in manifold learning, and this can be used in turn to derive an estimate $\hat{D}$ for the persistence diagram. However, this diagram can be unstable in the presence of outliers, and instead upper level sets of the KDE may be used to record the birth and death times of the components. Another alternative is to replace the distance function with another metric – *distance to a measure* (DTM). These approaches are studied in detail in [3].

Our motivation in studying TDA for the course project stems from the desire to connect these methods to recent work on representing words and sentences in vector spaces in Natural Language Processing (NLP). Visualizing language based data involves converting a discrete space to a continuous one. Skip-gram based Word Embeddings[8, 9] have been shown to perform well empirically, but the theoretical

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

grounding is not well understood as of now. As part of the project we aim to use density clustering to visualize these embeddings to verify the distance metric in a practical setting. Methods like t-SNE[7] and Elastic Embeddings[1] demonstrate interesting insights for dimensionality reduction, however our goal is to obtain multi-resolution cluster visualization of the data in the embedding space of the vectors themselves.

Dimensionality reduction to visualization could potentially remove important information, hence topological methods like density based cluster trees could a viable alternative. It is possible to objectively decide if it makes sense to visualize a given density estimate [6] along with finding confidence sets for the density function. Confidence sets are key for pruning the cluster tree, hence giving clearer visualizations.

1.1 Statistical Inference for Density Trees

In this section we discuss statistical inference over density trees, also known as cluster trees since they provide a visualization of a range of clusterings of the estimated density[1]. In particular we follow Kim et al [6] and describe the construction of confidence sets for the trees, followed by some pruning strategies to remove statistically insignificant features of the trees. We start with a formal definition of a tree and then define a distance function over the members of a tree which endows them with a metric topology. To formally define edges within the tree (leaves or internal branches of the tree), we define the notion of an equivalence class of the tree. Then each equivalence class corresponds to an edge on the tree. We also define the l_∞ metric which measures the distance between two trees.

Next we outline an important lemma which shows that the cluster tree constructed from the biased Kernel Density Estimate (KDE) with a small but fixed bandwidth, and the cluster tree of the true distribution have the same metric topology above. Hence, it is not necessary to let the bandwidth tend to 0, which leads to a much better rate for cluster tree estimation compared to density estimation. This makes intuitive sense, since for the cluster tree we are only interested in its topological properties which are in some sense easier to estimate than the full distribution. Then we outline the bootstrap method for constructing the confidence set for the true cluster tree using the l_∞ metric.

The confidence set obtained using the bootstrap consists of infinitely complex trees, since small perturbations of the estimated tree result in extra leaves and branches without leaving the confidence set. One of the main contributions of [6] is to come up with *pruning* rules which can remove statistically insignificant features of the estimated tree. In the last section, we summarize a partial ordering for a collection of trees which captures the notion of *simplicity* of the trees, and present the formal definition of the pruning rules in terms of the definitions of the tree edge. Lastly, we outline a result which shows that the tree obtained after pruning is indeed simpler than the original tree, comes from a valid density function, and that it lies in the bootstrap confidence interval.

2 Definitions and Setup

Assume that we have an iid sample from the true density $X_1, X_2, \ldots, X_N \sim p$, where $X_i \in \mathcal{X} \subset \mathbb{R}^d$. The cluster tree is defined in terms of a function $f : \mathcal{X} \to [0, \infty)$, which is usually a density function.

Definition 1. *Given $f : \mathcal{X} \to [0, \infty)$, the cluster tree of f is a function $T_f : \mathbb{R} \to 2^{\mathcal{X}}$, where $2^{\mathcal{X}}$ denotes all subsets of $\mathcal{X}$, such that $T_f(\lambda) = \text{connected}(\{x \in \mathcal{X} : f(x) \geq \lambda\})$, where $\text{connected}(S)$ denotes the connected components of set S. The collection of all the connected components at all levels λ is denoted by $\{T_f\} = \cup_\lambda T_f(\lambda)$.*

We will sometimes use $C \in T_f$ to denote that $C \in \{T_f\}$. Figure 1 shows an example. Note that the cluster tree is, in fact, a tree, i.e for $A, B \in \{T_f\}$ either $A \subset B$ or $B \subset A$ or $A \cap B = \phi$. The cluster tree for the true density will be denoted by T_p, and the cluster tree of the KDE will be denoted by $T_{\hat{p}_h}$, where,

$$\hat{p}_h(x) = \frac{1}{nh^d} \sum_{i=1}^{N} K(\frac{\|x - X_i\|}{h}) \tag{1}$$

[1] We will use these terms interchangeably in the rest of the paper.

Next we define a topology on the collection of sets $\{T_f\}$ which will be used later to justify the use of a fixed bandwidth for the KDE when constructing the confidence set. Kim et al [6] define the following distance function between sets in $\{T_f\}$:

Definition 2. *For any two sets $C_1, C_2 \in \{T_f\}$, the tree distance function $d_{T_f} : \{T_f\} \times \{T_f\} \to [0, \infty)$ is defined as:*

$$d_{T_f}(C_1, C_2) = \lambda_1 + \lambda_2 - 2m_f(C_1, C_2), \tag{2}$$

where $\lambda_1 = \sup\{\lambda : C_1 \in T_f(\lambda)\}$ and similarly λ_2 for C_2, and $m_f(C_1, C_2) = \sup\{\lambda \in \mathbb{R} : \exists C \in T_f(\lambda) \ s.t. \ C_1, C_2 \subset C\}$ is called the merge height *of the two sets.*

We can also define the merge height $m_f(x, y)$ between two points $x, y \in \mathcal{X}$ analogously as $m_f(x, y) = \sup\{\lambda \in \mathbb{R} : \exists C \in T_f(\lambda) \ s.t. \ x, y \in C\}$.

Lemma 1. *Let T_f be a cluster tree and let d_{T_f} be the tree distance function defined above. Then d_{T_f} is a metric on $\{T_f\}$.*

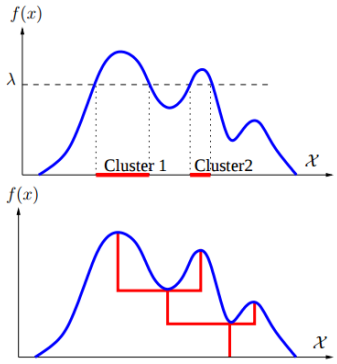

Figure 1: **Top:** A density) along with its level set clusters $T_f(\lambda)$ at level λ. **Bottom:** A pictorial representation of the entire cluster tree $\{T_f\}$ at for all levels $\lambda > 0$. Figure borrowed from [2].

Proof. Non-negativity ($d_{T_f}(C_1, C_2) \geq 0$) and symmetry ($d_{T_f}(C_1, C_2) = d_{T_f}(C_2, C_1)$) follow trivially from the definition above.

For identity of indiscernibles ($C_1 = C_2 \Leftrightarrow d_{T_f}(C_1, C_2) = 0$), note that if $C_1 = C_2$ then $\lambda_1 = \lambda_2 = m_f(C_1, C_2)$, and hence $d_{T_f}(C_1, C_2) = 0$. If $d_{T_f}(C_1, C_2) = 0$ then, since $\lambda_1, \lambda_2 \geq m_f(C_1, C_2)$, it must be that $\lambda_1 = \lambda_2 = m_f(C_1, C_2)$. Hence, $\exists C \in T_f(\lambda_1)$ s.t. $C_1 \subset C$ and $C_2 \subset C$ (from definition of m_f). But since $C_1, C_2, C \in T_f(\lambda_1)$, hence $C_1 \cap C \neq \phi$ and $C_1 = C$. Similarly, $C_2 = C$ which leads to the conclusion that $d_{T_f}(C_1, C_2) \Rightarrow C_1 = C_2$.

To prove sub-additivity ($d_{T_f}(C_1, C_2) + d_{T_f}(C_2, C_3) \leq d_{T_f}(C_1, C_3)$), note that $\max\{m_f(C_1, C_2), m_f(C_2, C_3)\} \leq \lambda_2$ since both $m_f(C_1, C_2)$ and $m_f(C_2, C_3)$ are $\leq \lambda_2$. Also note that $\min\{m_f(C_1, C_2), m_f(C_2, C_3)\} \leq m_f(C_1, C_3)$ since there exist sets $C_{12}, C_{23} \in T_f(\min\{m_f(C_1, C_2), m_f(C_2, C_3)\})$ s.t. $C_1, C_3 \in C_{12} = C_{23}$. Hence, it follows that,

$$\begin{aligned}
&d_{T_f}(C_1, C_2) + d_{T_f}(C_2, C_3) \\
&= \lambda_1 + \lambda_2 - 2m_f(C_1, C_2) + \lambda_2 + \lambda_3 - 2m_f(C_2, C_3) \\
&= \lambda_1 + \lambda_3 - 2(\min\{m_f(C_1, C_2), m_f(C_2, C_3)\} + \max\{m_f(C_1, C_2), m_f(C_2, C_3)\} - \lambda_2) \\
&\geq \lambda_1 + \lambda_3 - 2m_f(C_1, C_2) \\
&= d_{T_f}(C_1, C_3)
\end{aligned}$$

Hence, we have shown that d_{T_f} is non-negative, symmetric, sub-additive and follows the identity of indiscernibles. Hence it is a metric on $\{T_f\}$. □

The collection of sets $\{T_f\}$ along with the metric d_{T_f} form a metric topology, whose ϵ-balls are defined by $B(C, \epsilon) = \{C' \in \{T_f\} : d_{T_f}(C, C') < \epsilon\}$. We say that two trees are homeomorphic or $T_f \cong T_g$ if the sets of collected components $\{T_f\}$ and $\{T_g\}$ are homeomorphic, i.e. there exists a bijective continuous function between the two which has a continuous inverse.

Next we define edges in the cluster tree, i.e. the red vertical segments in Figure 1 (bottom). First note that, intuitively, an edge can be defined as the set of clusters in $\{T_f\}$ which have the same inclusion relationship with respect to all other clusters in the tree. To formalize this, for $A, B \in \{T_f\}$ define an interval as $[A, B] = \{C \in \{T_f\} : A \subset C \subset B\}$. Also, define an equivalence relationship $\sim$ and write $A \sim B$ if and only if for all $C \in \{T_f\}$ s.t. $C \notin [A, B] \cup [B, A]$, $C \subset A$ iff $C \subset B$ and $A \subset C$ iff $B \subset C$. This relation is reflexive, symmetric and transitive. Edges in $\{T_f\}$ are then given by the equivalence classes in $\{T_f\}$ formed by this relation. We denote them by $E(T_f) = \{T_f\}/\sim$.

3 Confidence Sets

3.1 Tree Metrics

To quantify the closeness of trees, tree metrics need to be defined which are suitable for statistical inference. Let $p, q : \mathcal{X} \to [0, \infty)$ be non-negative functions which represent probability density functions. T_p and T_q correspondingly represent density trees.

A common metric is to find the maximum distance between the function values over all points in the domain space. It is called l_∞ metric and is defined as

$$d_\infty(T_p, T_q) = \sup_{x \in \chi} |p(x) - q(x)| = \|p - q\|_\infty. \tag{3}$$

Another metric is the Merge Distortion Metric [5], which is defined in a way that the distance between the clusters maps to the distance between the merge heights of the given two cluster trees.

$$d_M(T_p, T_q) = \sup_{x,y \in \chi} |m_p(x, y) - m_q(x, y)|, \tag{4}$$

where $m_p(x, y)$ is the merge height of $x.y$ for the distribution $p(x)$.

Even though, Merge Distortion Metric can handle the perturbations in l_∞, d_∞ and d_M turn out to be equivalent if the distributions are continuous (see Lemma 2 below). Hence, we can use either of them to construct confidence sets.

There is another metric known as Modified Merge Distortion Metric (d_{MM}) which is not used in this paper, because it is not point-wise Hadamard differentiable and hence does not guarantee stability to perturbations in the input distribution.

Lemma 2. *When p and q are continuous, then* $d_\infty(T_p, T_q) = d_M(T_p, T_q)$.

The outline of the proof is as follows.

Proof. To show that $d_\infty(T_p, T_q) = d_M(T_p, T_q)$, it is shown that $d_\infty(T_p, T_q) \geq d_M(T_p, T_q)$ and $d_\infty(T_p, T_q) \leq d_M(T_p, T_q)$.

The proof of $d_\infty(T_p, T_q) \geq d_M(T_p, T_q)$ involves finding the lowerbound of l_∞ metric. To make the process more straightforward, choose a set $C_0 \in T_p(m_p(x, y) - \epsilon)$ where $\epsilon > 0$. C_0 represents all the connected sets just below some merge point. For all $z \in C_0$ and some $x, y \in C_0$, we have

$$\begin{aligned} q(z) &> p(z) - d_\infty(T_p, T_q), \\ &\geq m_p(x, y) - \epsilon - d_\infty(T_p, T_q), \end{aligned}$$

hence, $C_0 \subset q^{-1}(m_p(x, y) - \epsilon - d_\infty(T_p, T_q), \infty)$ and C_0 is connected, which shows that x, y are in the same connected component of $q^{-1}(m_p(x, y) - \epsilon - d_\infty(T_p, T_q), \infty)$

$$m_q(x, y) \leq m_p(x, y) - \epsilon - d_\infty(T_p, T_q),$$

with similar arguments we get

$$m_p(x, y) \leq m_q(x, y) - \epsilon - d_\infty(T_p, T_q),$$

which implies

$$\begin{aligned} |m_p(x, y) - m_q(x, y)| &\leq d_\infty(T_p, T_q), \\ sup_{x,y \in \chi} |m_p(x, y) - m_q(x, y)| &\leq d_\infty(T_p, T_q), \\ d_M(T_p, T_q) &\leq d_\infty(T_p, T_q). \end{aligned}$$

For the second part of the proof, choose x such that $|p(x) - q(x)| > d_\infty(T_p, T_q) - \frac{\epsilon}{2}$. Also, there exists a finite ball of size $\delta > 0$ such that $B(x, \delta) \subset p^{-1}\left(p(x) - \frac{\epsilon}{2}, \infty\right) \cap q^{-1}\left(q(x) - \frac{\epsilon}{2}, \infty\right)$. As

$B(x,\infty)$ is connected, we have $p(x) - \frac{\epsilon}{2} \leq m_p(x,y) \leq p(x)$ and $q(x) - \frac{\epsilon}{2} \leq m_q(x,y) \leq q(x)$. This implies that

$$\begin{aligned} |m_p(x,y) - m_q(x,y)| &\geq |p(x) - q(x)| - \frac{\epsilon}{2}, \\ &> d_\infty(T_p, T_q) - \epsilon, \\ sup_{x,y\in\chi}|m_p(x,y) - m_q(x,y)| &\geq d_\infty(T_p, T_q), \\ d_M(T_p, T_q) &\geq d_\infty(T_p, T_q), \end{aligned}$$

as this is true for any $\epsilon > 0$. Hence, $d_M(T_p, T_q) = d_\infty(T_p, T_q)$. □

3.2 Confidence Sets via Bootstrap

A valid confidence interval is defined as $C_\alpha = \{T : d_\infty(T, T_{\widehat{p_h}}) \leq t_\alpha\}$ for T_{p_h}. Consider a bootstrap sample $\{\tilde{X}_1^1, \ldots, \tilde{X}_n^1\}, \ldots, \{\tilde{X}_1^B, \ldots, \tilde{X}_n^B\}$. This sample is used to estimate cluster trees $\{\tilde{T}_{p_h}^1, \ldots, \tilde{T}_{p_h}^B\}$ using kernel density estimation. These estimates are used to construct a cumulative distribution function $\hat{F}$, which is finally used to estimate t_α.

$$\begin{aligned} \hat{F}(s) &= \frac{1}{B}\sum_{i=1}^{B} \mathbb{I}(d_\infty(\tilde{T}_{p_h}^i, T_{\widehat{p_h}}) < s), \\ \hat{t_\alpha} &= \hat{F}^{-1}(1-\alpha), \end{aligned}$$

where $\mathbb{I}$ is the indicator function.

Theorem 1. *Under regularity conditions on the kernel, an asymptotically valid confidence interval is*

$$\mathbb{P}\left(T_{p_h} \in \hat{C}_\alpha\right) = 1 - \alpha + O\left(\frac{\log^7 n}{nh^d}\right)^{\frac{1}{6}}, \tag{5}$$

where $\hat{C}_\alpha = \{T : d_\infty(T, T_{\widehat{p_h}}) \leq \hat{t}_\alpha\}$

In Theorem 1 we can see that the choice of h can change the rate. Also the confidence interval has been constructed for the cluster estimate (T_{p_h}) instead of the true distribution (T_{p_0}). Instead of make h(bandwidth) go to zero, we set h to a small positive value and use Lemma 3 to show that T_{p_0} and T_{p_h} have the same topology. In addition to this, the rate becomes dimension independent to $O\left(\frac{\log^7 n}{n}\right)^{\frac{1}{6}}$.

Lemma 3. *If the true unknown density p_0, has non-degenerate critical points, then $\exists$ a constant $h_0 > 0$, such that $\forall 0 < h \leq h_0$, the 2 cluster trees, T_{p_0} and T_{p_h} have the same topology.*

Proof. Assume that p is a Morse Function supported on a compact set S with finitely many and distinct critical values. By properties of the Morse function, $\exists$ a constant C_0 such that for a smooth function $q : S \to \mathbb{R}$ and $\|q - p\|_\infty, \|\nabla q - \nabla p\|_\infty, \|\nabla^2 q - \nabla^2 p\|_\infty < C_0$. This implies that q is a Morse function.

As described in [3], there exist 2 different diffeomorphisms $\phi : S \to S$ and $h : \mathbb{R} \to \mathbb{R}$ such that $q = h \circ p \circ \phi$, which implies **Case 1:** $q \circ \phi^{-1} = h \circ p$ and **Case 2:** $h^{-1} \circ q = p \circ \phi$

Case 1: For any connected component $C \in T_p(\lambda)$, $\phi^{-1}(C)$ is a connected component of $T_q(h(\lambda))$. Hence we can define a mapping $\Phi : \{T_p\} \to \{T_q\}$ where $\Phi = \phi^{-1}$ for all C in the support of Φ. As ϕ is diffeomorphic, Φ is also diffeomorphic, hence $C_1 \subset C_2$ iff $\Phi(C_1) \subset \phi(C_2)$. Using Definition 3, we can say that $T_p \preceq T_q$

Case 2: The proof is similar to Case 1. For any connected component $C \in T_q(\lambda)$, $\phi(C)$ is a connected component of $T_p(h^{-1}(\lambda))$. Hence we can define a mapping $\Phi : \{T_q\} \to \{T_p\}$ where $\Phi = \phi$ for all C in the support of Φ. As ϕ is diffeomorphic, Φ is also diffeomorphic, hence $C_1 \subset C_2$ iff $\Phi(C_1) \subset \phi(C_2)$. Using Definition 3, we can say that $T_q \preceq T_p$.

Combining Case 1 and Case 2 and using Lemma 4, we can say that T_p and T_q have the same topology (as defined as Section 2). Also non-parametric theory [14] gives us that $\exists$ a constant $C_1 > 0$ such

that $\|p_h - p\|_2, \max < C_1 h^2$ when $h < 1$. Using the definition of morse function it can be said that if $0 \leq h \leq \sqrt{\frac{C_0}{C_1}}$, T_h and T have the same Topology. Replacing T with T_p and T_h with T_{p_h}, the given lemma is proved. □

4 Tree Pruning

The confidence set constructed above is asymptotically valid, however it contains infinitely complex trees. We define the notion of tree "complexity" formally below, but intuitively we can see that given a tree $T \in \hat{C}_\alpha$, infinitesimal perturbations of this tree which lead to extra edges will also belong in $\hat{C}_\alpha$. Hence, in this section we review the pruning rules presented in [6] to remove statistically insignificant features from the estimated tree such that the resulting tree is a simple representative from the confidence set.

4.1 Notions of Tree Simplicity

We first need to define a notion of tree simplicity, for which we define the following partial ordering:

Definition 3. *For any $f, g : \mathcal{X} \to [0, \infty)$ and their trees T_f, T_g we say $T_f \preceq T_g$ if $\exists$ a map $\Phi : \{T_f\} \to \{T_g\}$ which preserves set inclusion relationships, i.e. for any $C_1, C_2 \in \{T_f\}$ we have $C_1 \subset C_2$ iff $\Phi(C_1) \subset \Phi(C_2)$.*

A partial order must satisfy the following properties – (i) reflexivity, (ii) transitivity, (iii) antisymmetry. The first two are trivial to check in the above case, for the third one we outline the following lemma:

Lemma 4. *Let $f, g : \mathcal{X} \to [0, \infty)$ be continuous functions, and let T_f, T_g be their finite cluster trees, i.e. their edge sets $E(T_f)$ and $E(T_g)$ are finite. Then $T_f \preceq T_g$ and $T_g \preceq T_f$ is true if and only if there exists a homeomorphism $\Phi : \{T_f\} \to \{T_g\}$ which preserves the root, i.e. $\Phi(\mathcal{X}) = \mathcal{X}$.*

Note that a homeomorphism between the two trees implies that they are topologically equivalent. We give an outline of the proof below, for details see [6].

Proof Sketch. For the *only if* direction, from the definition of the partial order and since $T_f \preceq T_g$, we have a map Φ which maps sets in $\{T_f\}$ to sets in $\{T_g\}$. This can be easily extended to another map $\bar{\Phi} : E(T_f) \to E(T_g)$ over the edge sets of the two trees which can be shown to be injective. Hence, $|E(T_f)| \leq |E(T_g)|$. Similarly, starting from $T_g \preceq T_f$ we get that $|E(T_g)| \leq |E(T_f)|$. Hence, $|E(T_f)| = |E(T_g)|$ and since both of these are finite, the map $\bar{\Phi}$ is a bijection. It can be shown that this map $\bar{\Phi}$ sends adjacent edges in $E(T_f)$ to adjacent edges in $E(T_g)$ and the root to root. From this, and the fact that f, g are continuous we can extend $\bar{\Phi}$ to a homeomorphism.

For the *if* direction, we note that for any $C \in \{T_f\}$, the interval $[C, \mathcal{X}]$ is mapped to another interval $\Phi([C, \mathcal{X}])$ which is uniquely determined by its boundary points, i.e. $\Phi([C, \mathcal{X}]) = [\Phi(C), \Phi(\mathcal{X})]$. From this we can show that if $C_1, C_2 \in \{T_f\}$ and $C_1 \subset C_2$, then $\Phi(C_1) \subset \Phi(C_2)$. Hence, $T_f \preceq T_g$. Since $\Phi^{-1} : \Phi(\{T_f\}) \to \{T_g\}$ is also a homeomorphism from exactly the same argument we can get that $T_g \preceq T_f$. □

We can verify that the above partial order matches our intuitive notions of tree complexity. We state the following properties of this order without proofs, which can be found in [6]:

1. If $T_f \preceq T_g$, then $|E(T_f)| \leq |E(T_g)|$. Hence a tree obtained by removing edges from another tree is necessarily simpler. This justifies the pruning rules presented below.
2. If T_g can be obtained by adding edges to T_f, then $T_f \preceq T_g$ holds.
3. The existence of a topology preserving embedding from $\{T_f\}$ to $\{T_g\}$ implies that $T_f \preceq T_g$.

4.2 Pruning Rules

Pruning rules are methods for removing statistically insignificant edges of the estimated tree $T_{\hat{p}_h}$ such that the resulting tree $\tilde{T}$ satisfies $\tilde{T} \preceq T_{\hat{p}_h}$ and $\tilde{T} \in C_\alpha$. Two such schemes are presented in [6] which can be informally described as follows (recall that $\hat{t}_\alpha$ is the d_∞ threshold selected via bootstrap):

1. **Pruning only leaves**: Remove all leaves of the tree with length less than $2\hat{t}_\alpha$.
2. **Pruning leaves and internal branches:** Remove all leaves and internal branches of the tree with *cumulative length* less than $2\hat{t}_\alpha$.

The following definition gives a formal description of these rules in terms of a function `life` which maps edges of the tree to a value representing its significance. First we define a partial order on the edge set $E(T_f)$ as follows: for $[C_1], [C_2] \in E(T_f)$ we say that $[C_1] \leq [C_2]$ if and only if for all $A \in [C_1]$ and $B \in [C_2]$ we have $A \subset B$.

Definition 4. *Suppose the function* life : $E(T_f) \to [0, +\infty]$ *satisfies that* $[C_1] \leq [C_2]$ *implies that* life$([C_1]) \leq$ life$([C_2])$. *Then the pruned tree* $Pruned_{\text{life},\hat{t}_\alpha}(T_{\hat{p}_h}) : \mathbb{R} \to 2^{\mathcal{X}}$ *is defined as,*

$$Pruned_{\text{life},\hat{t}_\alpha}(T_{\hat{p}_h})(\lambda) = \left\{ C \in T_{\hat{p}_h}(\lambda - \hat{t}_\alpha) : \text{life}([C]) > \hat{t}_\alpha \right\}.$$

We need to define the function life which corresponds to the two rules presented above. First, define the *level* and *cumulative level* of any edge $[C] \in E(T_f)$ as,

$$\begin{aligned} level([C]) &= \{\lambda : \exists A \in [C] \cap T_{\hat{p}_h}(\lambda)\}, \\ cumlevel([C]) &= \{\lambda : \exists A \in T_{\hat{p}_h}(\lambda), B \in [C] \quad s.t. \quad A \subset B\}. \end{aligned}$$

Intuitively, $level$ extends from the bottom to the top of the edge, and $cumlevel$ extends from the bottom of the edge to the top of the tree branch on which the edge lies. Then the following life function corresponds to the first pruning rule,

$$\text{life}^{leaf}([C]) = \begin{cases} \sup\{level([C])\} - \inf\{level([C])\} & \text{if} \sup\{level([C])\} = \sup\{cumlevel([C])\} \\ +\infty & o.w. \end{cases}.$$

The following life function corresponds to the second pruning rule,

$$\text{life}^{top}([C]) = \sup\{cumlevel([C])\} - \inf\{cumlevel([C])\}.$$

Note that life$^{top}([C]) \leq$ life$^{leaf}([C])$. The following lemma states that for any life function which is lower bounded by life^{top}, the pruned tree is a valid tree in the confidence set $\hat{C}_\alpha$ and is simpler than the original tree from which it is constructed.

Lemma 5. *Suppose that the function* life : $E(T_f) \to [0, +\infty]$ *satisfies:* $\forall C \in E(T_f)$, life$^{top}([C]) \leq$ life$([C])$, *then:*

(i) $Pruned_{\text{life},\hat{t}_\alpha}(T_{\hat{p}_h}) \preceq T_{\hat{p}_h}$.

(ii) There exists a function $\tilde{p}$ *s.t.* $T_{\tilde{p}} = Pruned_{\text{life},\hat{t}_\alpha}(T_{\hat{p}_h})$.

(iii) $\tilde{p}$ *in (ii) satisfies* $\tilde{p} \in \hat{C}_\alpha$.

Proof. (i) This follows since $Pruned_{\text{life},\hat{t}_\alpha}(T_{\hat{p}_h})$ is obtained by removing edges from $T_{\hat{p}_h}$.

(ii) $Pruned_{\text{life},\hat{t}_\alpha}(T_{\hat{p}_h})$ is generated from the following $\tilde{p}$:

$$\tilde{p}(x) = \sup\left\{\lambda : \exists C \in T_{\hat{p}_h}(\lambda) \text{ s.t. } x \in C \text{ and } \text{life}([C]) > 2\hat{t}_\alpha\right\} + \hat{t}_\alpha \tag{6}$$

(iii) The outline of the proof is as follows. First note that we can rewrite the KDE estimate as,

$$\hat{p}(x) = \sup\{\lambda : \exists C \in T_{\hat{p}_h} \text{ s.t. } x \in C\}. \tag{7}$$

Hence, $\tilde{p}(x) \leq \hat{p}(x) + \hat{t}_\alpha$. Next, define $e_x = \{e : x \in e, \text{life}(e) \leq 2\hat{t}_\alpha\}$ and observe that:

$$\{\lambda : \exists C \in T_{\hat{p}_h}(\lambda) \text{ s.t. } x \in C, \text{life}([C]) \leq 2\hat{t}_\alpha\} \subset cumlevel(e_x).$$

But the LHS above is itself a superset of the difference of sets whose supremes are $\tilde{p}(x)$ (6) and $\hat{p}(x)$ (7). Hence,

$$\begin{aligned} \hat{p}(x) + \hat{t}_\alpha - \tilde{p}(x) &\leq \sup\{cumlevel(e_x)\} - \inf\{cumlevel(e_x)\} \\ &= \text{life}^{top}(e_x) \\ &\leq \text{life}(e_x) \leq 2\hat{t}_\alpha. \end{aligned}$$

Hence, $\hat{p}(x) - \hat{t}_\alpha \leq \tilde{p}_x$. Combined with the upper bound on $\tilde{p}(x)$ above this implies that $\tilde{p}(x)$ lies in the confidence set. □

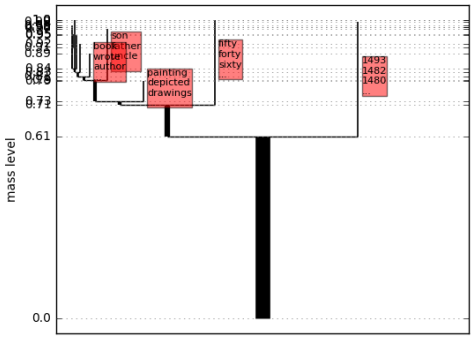

Figure 2: Leonardo Da Vinci

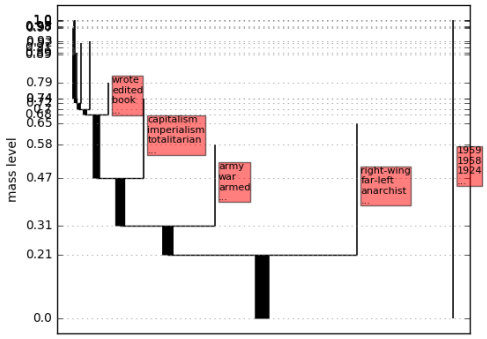

Figure 3: Noam Chomsky

Figure 4: Density Trees of words from 2 Wikipedia pages with GloVe word embeddings used as data for constructing the probability distribution. Figure 2 and Figure 3 have been created from documents of 'Leonardo da Vincia' and 'Noam Chomsky'

5 Simulations – Visualizing Word Embeddings

We perform simulations on the GloVe Word Embedding dataset [13] which was created using text data found in the real world. This dataset can be summarized as a mapping function $\phi : A \to \mathbb{R}^d$, where A is a set of 400 thousand words from the english language and d is the dimension of the vector space where the words are mapped to. We take $d = 50$ for our simulations. This dataset is used to construct a probability distribution p_h which is used to construct cluster trees. For the sake of clarity in visualization we choose a smaller set of words $W_0 = A \cap W_{\text{page}}$ where W_{page} is a set of words extracted from a wikipedia page. We chose 'Noam Chomsky' and 'Leonardo da Vinci' as their lives were multi-faceted and we hoped to find words from multiple domains in these documents.

The density trees are shown in Figure 4. It is interesting to note that words with similar meaning (or context in natural language) end up being a part of the same cluster leaf. This is probably the reason why words denoting years ended up together while 'far-left, anarchist, right-wing' which are political affiliations were another cluster.

6 Conclusion

In this project we explore Topological data analysis which studies the shape, topological and connectedness properties of the data. Specifically, we study Density based clustering. We start of by defining tree topology. These definitions were crucial to the unambiguous construction of density trees, which we later use for the construction of confidence sets for these trees via bootstrap. Finally, these confidence intervals were used to remove statistically insignificant features of the trees resulting in a 'simpler' tree. It was also shown that the pruned tree is generated from a valid density function and it lies in the constructed confidence set.

References

[1] Miguel A Carreira-Perpinán. The elastic embedding algorithm for dimensionality reduction. In *ICML*, volume 10, pages 167–174, 2010.

[2] Kamalika Chaudhuri, Sanjoy Dasgupta, Samory Kpotufe, and Ulrike von Luxburg. Consistent procedures for cluster tree estimation and pruning. *IEEE Transactions on Information Theory*, 60(12):7900–7912, 2014.

[3] Frédéric Chazal, Brittany T Fasy, Fabrizio Lecci, Bertrand Michel, Alessandro Rinaldo, and Larry Wasserman. Robust topological inference: Distance to a measure and kernel distance. *arXiv preprint arXiv:1412.7197*, 2014.

[4] Yen-Chi Chen, Christopher R Genovese, Larry Wasserman, et al. Asymptotic theory for density ridges. *The Annals of Statistics*, 43(5):1896–1928, 2015.

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

[6] Jisu Kim, Yen-Chi Chen, Sivaraman Balakrishnan, Alessandro Rinaldo, and Larry Wasserman. Statistical inference for cluster trees. *arXiv preprint arXiv:1605.06416*, 2016.

[7] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[8] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[9] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[10] Partha Niyogi, Stephen Smale, and Shmuel Weinberger. Finding the homology of submanifolds with high confidence from random samples. *Discrete & Computational Geometry*, 39(1-3):419–441, 2008.

[11] Partha Niyogi, Stephen Smale, and Shmuel Weinberger. A topological view of unsupervised learning from noisy data. *SIAM Journal on Computing*, 40(3):646–663, 2011.

[12] Umut Ozertem and Deniz Erdogmus. Locally defined principal curves and surfaces. *Journal of Machine Learning Research*, 12(Apr):1249–1286, 2011.

[13] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[14] Larry Wasserman. All of nonparametric statistics. In *Springer Science & Business Media*, 2016.

[15] Larry Wasserman. Topological data analysis. *arXiv preprint arXiv:1609.08227*, 2016.