Principle Component Trees and their Persistent Homology

Ben Kizaric, MS
Department of Electrical and Computer Engineering
University of Wisconsin-Madison
Madison, Wi 53706
bakizaric@wisc.edu

Abstract

Multivariate normality of data has served as a bedrock assumption of many statistical methods for the better part of a century. We often use an ordered sequence of orthonormal basis vectors known as "Principle Components" to model and characterize such distributions. A natural extension of this framework for high-dimensional data is the union of subspaces perspective, which views data as being generated by several disjoint low-dimensional gaussians and thereby several sequences of principle components. Unfortunately, real-world distributions rarely fit the assumptions of either model exactly. In this paper, we propose "Principle Component Trees" (PCTs), a structure made to marry these two perspectives and facilitate comparison and visualization of the subspace structure of high-dimensional distributions. Each node in a PCT corresponds to a principle component vector of some subspace of the distribution, and is split into one or two children based on the results of a "subspace structure" hypothesis test which uses the distribution of pairwise angles of points assigned to that node. Through the use of interpretable persistent homology measures on the structure of this tree, we show it's ability to discriminate between and analyse complex distributions.

1 Introduction

The ability to compare and characterize complex distributions is at the core of statistics and by extension machine learning. Defined by a mean $\boldsymbol{\mu}$ and covariance matrix Σ, the multivariate normal distribution is perhaps the most used tool for this task, and our foundational tool for understanding multivariate normal distributions is Principle Component Analysis. By analyzing the eigenvectors / principle components of this covariance matrix, we can determine the primary directions in which the distribution varies, enabling visualization and summarization of high dimensional patterns. By analyzing the eigenvalues of the covariance matrix, we can determine whether the data is distributed evenly among the whole space, or if it lies mostly in a lower dimensional subspace.

Of course PCA is not restricted to only normal distributions, and exists a whole pantheon of modeling tools and parameterizations for multivariate data. One such parameterization is the union of subspaces model, where we assume data is best modeled not on a single full-rank gaussian, but on several low-rank gaussians, whose principle components define disjoint subspaces. Subspace clustering is then the task of learning the structure of these subspaces and has shown effectiveness in modeling many kinds of data [28, 22, 17].

Baked into the majority of subspace clustering algorithms, such as the effective Sparse Subspace Clustering [10] is the assumption that subspaces learned are disjoint, that is all points on the span of one subspace are not on the span of any other subspace (with the exception of the origin). However real-world data is rarely so disjoint. For example, imagine two groups with 3 attributes: the first

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

attribute has approximately the same variation in both groups, but there is no variation of the 2_{nd} attribute for group A and no variation of the 3_{rd} attribute for group B. In this axis-aligned example, both groups can be modeled as 2D subspaces of the 3D space, but yet they both share the 1D subspace corresponding to attribute 1. See figure 1 for a visualization.

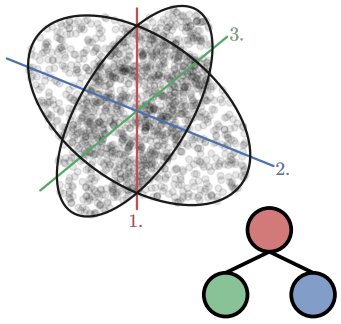

Figure 1: Two 2D subspace clusters with a shared 1D subspace and the corresponding PCT.

In this paper, we show that the relaxation of this disjoint subspace assumption is key to bridging the gap between multivariate gaussians and subspace clustering. We propose and algorithm to build a "Principle Component Tree", where each node is a principle component of some subset of the data, and the tree is built recursively, top-down, by expanding each node one at a time. Key to the expressive power of the tree are 3 properties. 1: **Path Orthogonality** Each node's basis vector is orthogonal to the basis vectors of it's ancestor and descendent nodes. 2: **Residual Sub-trees** When expanding a node into a sub-tree, we use only the residuals after projecting that node's assigned subset of the data onto the orthonormal basis spanned by it and it's ancestors. 3 **Adaptive Splitting** : We only split the node into two children if we detect sufficient subspace clustered structure in it's assigned data points, as determined by a novel pairwise angle distribution test. Otherwise, the node is given only one additional child, corresponding to the next most dominant principle component of data assigned to that node.

With this construction, if no subspace clustered structure is detected the tree becomes a "linked list", in which each node has no children, and it's basis vector is the i^{th} principle component of the original data, where i is the depth of the node. In general, data more closely matching some multivariate gaussian will yield "Tall Trees", and data exhibiting subspace clustered structure will yield "Wide Trees". To quantitatively measure this intuition, we turn to persistent homology, a branch of topological data analysis that concerns itself with measuring properties of graphs as they are filtered into sub-graphs. Specifically, we employ a simple tree filtration that measures how many nodes are less than a given height, yielding a bounded, monotonically increasing curve for each PCT, which serves a summary of subspace structure, and can be used to perform hypothesis testing between distributions.

Our experimental results show the effectiveness of this tree construction on synthetic and real-world distributions, including on the structural covariance of brain grey matter, and the latent space of neural networks.

2 Related Work

Subspace Clustering As mentioned, the task of subspace clustering is to partition a set of data such that each partition lies on the span of a low dimensional subspace. While some works are principally interested in the clustering / partitioning of points, such as in Motion Segmentation [28] and Hyperspectral Imaging [17], others are more interested in the using the subspace structure for downstream tasks. For example, subspace clustering has shown to be effective in matrix completion problems [11, 22], or as a feature extraction method [13, 15]. Outside of axis-parallel methods like [1, 20, 5], which assume the subspaces run parallel to some subset of the given features, methods for performing subspace clustering utilize sparse self-expressiveness [10], expectation maximization [19], deep learning [32], and fusion over the space of all possible subspaces [22].

Of particular interest to this paper are hierarchical subspace clustering techniques, usually inspired by methods in traditional clustering which enable discovery of an adaptive number of clusters. These approaches come in top-down [26, 30] and bottom-up [24, 21] flavors. [26] and [30] follow a construction similar to Principle Component trees in that they are top-down methods which use existing flat methods of subspace clustering to recursively partition space into lower dimensional subspaces, and the termination of node splitting is based on a reconstruction error criteria. The existence of similarly named methods like [23] and [29] should be acknowledged, but note that these methods make no use of subspace clustering; each *level* of their trees is a single basis, and these methods are used primarily to accelerate vector similarity search. While the construction of PCTs is

fundamentally a hierarchical subspace clustering algorithm, it stands alone as the only one where each node is a single basis vector, with Path Orthogonality, Residual Sub-tree Construction, and Adaptive Splitting.

Central to PCT's adaptive splitting construction are established results on the distribution of pairwise angles in high dimensional space. First established in [3] and used to estimate intrinsic dimensionality in [27], we leverage the distribution of angles to test for the presence of subspace structure in our data. [24] is the only other work to use pairwise angle distributions in subspace clustering, but they are used here to determine which subspaces to merge in an agglomerative hierarchical clustering approach, unlike our top-down approach. The use of parametric and non-parametric tests of angle uniformity is central to directional statistics, and is explored in depth in [12].

Topological Data Analysis & Persistent Homology Because of it's ability to discover highly robust patterns in arbitrary distributions like networks and point clouds, topological data analysis has emerged as a promising approach to many problems, and has seen much success in medical imaging and other areas [4]. Persistent Homology is a sub-field of TDA that focuses on measuring properties of graphs and other structures as they are filtered down [25]. The most common use of persistent homology is in the construction of Birth-Death decomposition charts, which summarize how sub-structures are created and destroyed in some filtration [9]. While this decomposition can be a very powerful tool, Rips filtration [2] and it's more interpretable cousin Graph Filtration [16] have emerged as robust approaches that enable theoretically sound statistical inference. The key property of graph filtration that enables this is that it tracks the first two Betti numbers of the graph, which are monotonically increasing or decreasing over edge filtration. Because of this, it has proven useful in the analysis of brain networks.

Of particular relevance to our analysis of PCTs are persistent homology based characterizations of tree structure. [7] first build minimum spanning trees from structural covariance networks then applies graph filtration to create curves that can be used to compare network structure. Persistent Homology has also been applied to explicitly tree-structured data, especially in neuroscience. [18, 14] both filter the binary tree structure of neurons and brain arteries based height, where "height" is defined as the distance to the neuron's cell body. Persistent structures are then recorded onto a birth-death decomposition chart and embedded into a vector space. Our analysis of PCTs borrows this "height filtration" approach, but instead creates monotonic curves as in [7].

3 Methodology

At a high-level, we build a principle component tree from a set N, D-dimensional data points $X_{N\times D}$ using a top-down, recursive subspace clustering algorithm. Each node is assigned some subset of X, and we use the residuals obtained from projecting that subset of X onto the basis formed by that node's ancestors to test for pairwise angle uniformity in some subspace of the residuals. The test is a non-parametric KS test comparing the observed angle distribution of the residuals with what would be expected from an isotropic gaussian. As such, we whiten the residuals to ensure that differences from the expected distribution occur because of subspace clustering, not because of anisotropic covariance. If sufficient subspace clustered structure is detected, we split the node into two children. If not, we expand it with only one child. We form the tree up to a pre-defined number of nodes by greedily splitting the leaf with the largest projection residual sum of squares. After constructing the tree, we use a persistent homology filtration to capture the distribution of the heights of each node, yielding a monotonic curve that can then be used to compare trees.

3.1 Building the subspace tree

The PCT construction algorithm uses a few hyper-parameters. Firstly, we greedily build the tree up to a maximum number of nodes $|T|_{max}$, after which we end construction, and index nodes $\mathcal{N}_i$ $i = 1...|T|$, where $\mathcal{N}_1$ is the root node of the tree. In the test of subspace clustering, we test for the angle uniformity of a nodes residuals only in a W-Dimensional subspace, where $W \geq 2$. We also specify a significance level $\alpha_{test} \leq 0.05$ which specifies how confident the test must be that there is subspace clustered / angle non-uniformity structure in the data. Finally, when we split a node in two, we use an arbitrary subspace clustering algorithm to partion the data. We denote this $\mathcal{SC}(X) \rightarrow (X_A, X_B)$.

We define the following properties of the node $\mathcal{N}_i$ as follows. T_i is the sub-tree of T rooted at $\mathcal{N}_i$. h_i is the number of ancestors / height of the node. If a node has two children, they are labelled $\mathcal{N}_{i,left}$ and $\mathcal{N}_{i,right}$; if only one child, $\mathcal{N}_{i,cent}$. The sequence $\mathcal{P}_i = \mathcal{N}_1 ... \overline{\mathcal{N}_i}, \mathcal{N}_i$ is the unique path from the root of the tree to N_i, and has h_i elements. $\overline{\mathcal{N}_i}$ is the parent of $\mathcal{N}_i$. Each node is assigned a subset of $n_i \leq N$ samples from X labeled as X_i. $X_i \subseteq \overline{X_i}$. We denote each node's basis vector as $\mathbf{v}_i \in \mathbb{R}^D, ||\mathbf{v}_i|| = 1$. Equation 2 defines $\mathbf{V}_i$: the orthonormal "ancestral" basis of node $\mathcal{N}_i$, $\mathbf{E}_i$: the projection residuals of X_i onto $\mathbf{V}_i$, and R_i: the sum of squares of the residuals. The primary algorithm for tree construction is given in Algorithm 3.1.

$$\text{Let } \overline{\mathbf{E}}_{\bar{i}} \text{ be the subset of } \overline{\mathbf{E}}_i \text{ assigned to } \mathcal{N}_i. \text{ We define } \mathbf{V}_i \text{ as the first right singular value of } \overline{\mathbf{E}}_{\bar{i}}. \quad (1)$$

$$\mathbf{V}_i^{D \times h_i} = [\mathbf{v}_j | \mathcal{N}_j \in \mathcal{P}_i] \qquad \mathbf{E}_i^{n_i \times D} = X_i - X_i \mathbf{V}_i \mathbf{V}_i^T = \overline{\mathbf{E}}_{\bar{i}} - \overline{\mathbf{E}}_{\bar{i}} \mathbf{v}_i \mathbf{v}_i^T \qquad R_i = ||\mathbf{E}_i||_F^2 \quad (2)$$

Algorithm 1 Tree Construction

procedure BUILD_TREE
  ROOT $\leftarrow$ NODE(X, <NO PARENT>); LEAVES $\leftarrow$ [ ]
  while ROOT has less than $|T|_{max}$ descendants **do**
    $\mathcal{N}_{next} = \text{argmax}_{R_i}$ LEAVES ▷ Select the leaf with largest residual sum of squares.
    LEAVES $\leftarrow$ LEAVES $- \mathcal{N}_{next}$
    if $\mathcal{N}_{next}$ can be expanded **then** ▷ rank($\mathbf{E}_{next}$) ≥ 2. and $\mathbf{E}_{next}$ has 3+ rows.
      LEAVES $\leftarrow$ LEAVES + EXPAND_NODE($\mathcal{N}_{next}$)
    end if
  end while
  RETURN ROOT
end procedure
procedure EXPAND_NODE($\mathcal{N}_i$)
  if SUBSPACE_CLUSTER_TEST($\mathbf{E}_i$) **then**
    $(X_{i,left}, X_{i,right}) \leftarrow \mathcal{SC}(\mathcal{W}(\mathbf{E}_i))$ ▷ subspace cluster the whitened projection residuals.
    RETURN (NODE($X_{i,left}, \mathcal{N}_i$), NODE($X_{i,right}, \mathcal{N}_i$))
  end if
  RETURN (NODE($X_i, \mathcal{N}_i$))
end procedure

Test of Subspace Structure / Angle Non-Uniformity Key to our construction of the Principle Component Tree is our adaptive splitting algorithm, which will only split a node *in two* if we detect sufficient subspace clustered structure in it's assigned residuals $\mathbf{E}_i$. To make this determination, we use the distribution of the absolute value of angles/cosine distances between points of $\mathbf{E}_i$: $\mathbf{E}_i$, $P(|\frac{\mathbf{e}_i}{||\mathbf{e}_i||} \cdot \frac{\mathbf{e}_j}{||\mathbf{e}_j||}|)$. To support this, we make use of results from [3] and [27] which derive the distribution of angles between points drawn uniformly from the hypersphere in $\mathbb{R}^d$. They are given for angles $\theta \in [0, 2\pi]$ and absolute cosine distances $|cos(\theta)| \in [0, 1]$ in equation 3, which we abbreviate $\text{p}(C_d)$. These papers also note that as dimension increases, points become more perpendicular.

$$\text{p}(\theta) = \frac{\Gamma(\frac{d}{2})}{\Gamma(\frac{1}{2})\Gamma(\frac{d-1}{2})} sin(\theta)^{d-2} \qquad \text{p}(C_d) = 2Beta(\chi; \frac{d-1}{2}, \frac{d-1}{2}) - 1, \text{where} \chi = \frac{1 + C_d}{2} \quad (3)$$

We posit that if $\mathbf{E}_i, \text{rank}(\mathbf{E}_i) = 2d$ falls on a union of two rank-d subspaces $\mathcal{S}_A$ and $\mathcal{S}_B$ corresponding to a partitioning into $\mathbf{E}_{i,A} \sim N(\mathbf{0}, \mathbf{I}_d)$ and $\mathbf{E}_{i,B} \sim N(\mathbf{0}, \mathbf{I}_d)$, then the distribution of angles of $\mathbf{E}_i$, denoted $\angle\mathbf{E}_i$ will be significantly different than what would be expected if $\mathbf{E}_i \sim N(\mathbf{0}, \mathbf{I}_{2d})$. Specifically, angles within $\mathbf{E}_{i,A}$ and $\mathbf{E}_{i,A}$ would follow $\text{p}(C_d)$, whereas angles *between* $\mathbf{E}_{i,A}$ and $\mathbf{E}_{i,A}$ would be more perpendicular. If $\mathcal{S}_A$ was exactly perpendicular to $\mathcal{S}_B$, $\text{p}(\mathbf{E}_{i,A} \angle \mathbf{E}_{i,B}) = 0$.

Of course, the angle distributions shown in equation 3 are for angles of points on hyper**spheres**, and the distribution of angles will be different for anisotropic gaussians (see figure 3.1a). To isolate differences in $\angle\mathbf{E}_i$ and $\text{p}(C_d)$ that occur because of subspace clustering behavior and not an

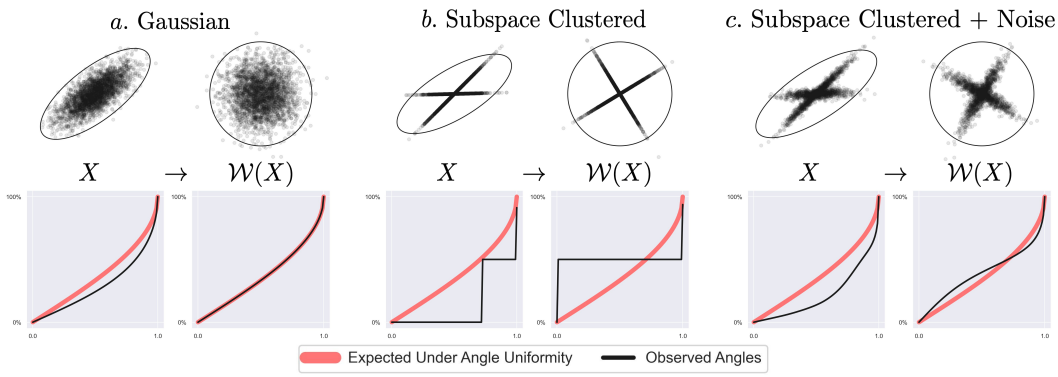

Figure 2: Distribution of pairwise angles (absolute value of dosine distance between points) with whitened and un-whitened data distributions. a) Multivariate Gaussian. After whitening, the data becomes "circular", and the angle distribution matches expected near-perfectly. b) Subspace Clustered. After whitening, angles are 50/50 perpendicular or parallel.

isotropic covariance, we whiten $\mathbf{E}_i$ such that it's diagonal matrix of singular values will be exactly I_d. *Caveat*: whitening forces singular values to be all equal, which essentially makes the distribution of angles in explicitly whitened data dependent on n, not just d. For example, if $n = d$, all points become perpendicular to each other. To address this, for $n < 10d$, we used monte-carlo simulation ($N = 100,000,000$) to estimate the angle distribution.

To actually test for angle non-uniformity, we perform a one-sample Kolmogorov-Smirov test to compare the empirical CDF of a whitened angle distribution of points $\hat{\mathrm{P}}(C_d)$ vs the expected CDF $\mathrm{P}(C_d)$. This test gives us a test statistic $D_i = sup_c|\hat{\mathrm{P}}(c) - \mathrm{P}(c)|$, the largest absolute difference in CDFs, and a p-value p where a sufficiently small p suggests subspace structure. Although it's not the most powerful non-parametric test to compare distributions, we use the KS test because it's null distribution is well-understood.

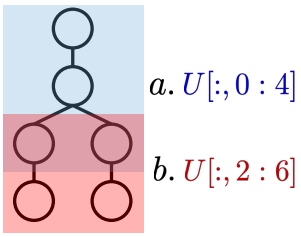

Figure 3: Data lies on 2 4D subspaces that share a two-dimensional subspace.

With these tools in hand, the full algorithm for determining whether a node should be split into one or two children is given in Algorithm 3.1. At a high-level, we only split a node into two if there is evidence for subspace clustering in subspace spanned by the first W singular vectors of $\mathbf{E_i}$ **and** the inclusion of information carried by the first singular vector contributes to the evidence for subspace clustering. Consider the non-disjoint subspace clustered structure shown in figure 3. With a window size of $W = 4$, we first test the angle uniformity of the data projected onto the first 4 singular vectors of $\mathbf{E}_i$ (figure 3a). This space is indeed subspace clustered, which our test would indicate. But if we immediately split the data after the first node, our tree structure wouldn't represent that the subspaces share low-dimensional structure. If we look "deeper" into the data by projecting onto the 2_{nd} through 6_{th} singular vectors (figure 3b), the shared low-dimensional structure disappears, and our test of angle uniformity will likely indicate there is *even more* evidence for subspace clustering deeper in the data.

3.2 Persistent Homology Analysis

Now that we have described and algorithm to construct the principle component tree, we introduce a simple persistent homology approach to analyze and compare the structure of PCTs. Motivated by proposition 4.2, we define a cumulative height function $H_T(h)$ which describes the distribution of node heights of a PCT T. $H_T(h)$ is trivially monotonically increasing. The larger the value of H_T at some h, the more the tree has splits at nodes with a height less than h, and therefore the more subspace clustered the data is deemed to be. For simple, interpretable comparison between trees, we also define AUC_T, the "area under tree" for T. The larger AUC_T, the more the data is deemed to be subspace clustered. Proposition 4.5 shows that $H_T(h)$ and therefore AUC_T is bounded above and

Algorithm 2 Adaptive Node Splitting

procedure SUBSPACE_CLUSTER_TEST($\mathbf{E}_i$)
  $U, s, V^T \leftarrow SVD(\mathbf{E}_i)$; ▷ Implicitly Whitens U.
  $\alpha_1 \leftarrow$ KS-TEST($\angle U[:, :W], \mathrm{p}(C_W)$)
  if $\alpha_1 > \alpha_{test}$ **then** ▷ No angle non-uniformity detected in window.
    RETURN FALSE
  end if
  $\alpha_2 \leftarrow$ KS-TEST($\angle U[:, 1:W+1], \mathrm{p}(C_W)$)
  if $\alpha_1 > \alpha_2$ **then** ▷ More angle non-uniformity structure "deeper" in subspace.
    RETURN FALSE
  end if
  RETURN TRUE
end procedure

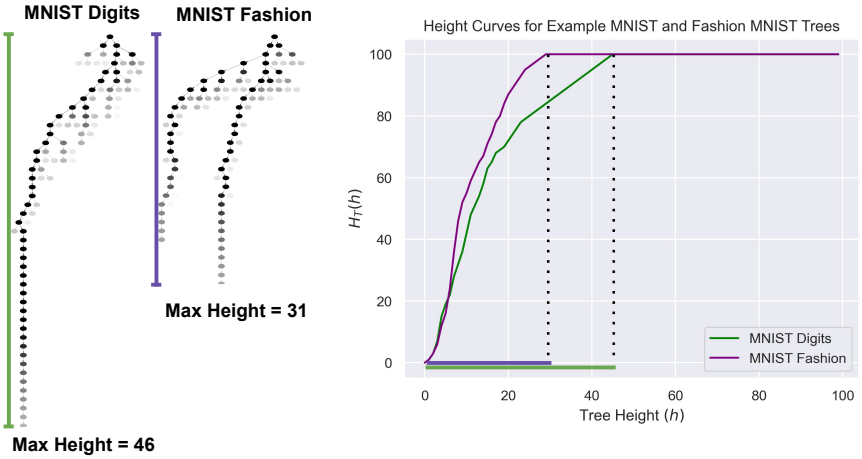

Figure 4: Example PCTs and height curves from the MNIST Digits and Fashion-MNIST datasets. The shade of each node it proportional to it's residual sum of squares.

below. To test for equal subspace structure in two distributions, we use the approach from [7], and perform jackknife-like sampling from the two distributions, and then run a difference of medians hypothesis test on the AUC_T from trees built from each sample.

$$H_T(h) = \Sigma_{\mathcal{N}_i \in T} \mathbb{1}\{h_i \leq h\}. \qquad H_T(|T|) = |T|, H_T(0) = 0 \qquad AUC_T = \Sigma_{h=0}^{h=|T|} H_T(h) \tag{4}$$

4 Theoretical Results

Proposition 4.1. *The basis vectors of all nodes in the subtree rooted by $\mathcal{N}_i$ are orthogonal to $\mathbf{v}_i$.*

Justification. All basis vectors $\mathbf{v}_k | \mathcal{N}_k \in T_i$ are either singular vectors of $\mathbf{E}_i$ or singular vectors of some subspace of $\mathbf{E}_i$. Therefore all $\mathbf{v}_k \in \mathrm{span}(\mathbf{E}_i)$. Because $\mathbf{E}_i^{n_i \times D} = \overline{\mathbf{E}}_{\bar{i}} - \overline{\mathbf{E}}_{\bar{i}} \mathbf{v}_i \mathbf{v}_i^T$, $\mathbf{v}_i$ forms a basis for the null space of $\mathbf{E}_i$, and $\mathbf{v}_i \perp \mathbf{v}_k \forall k$.

Corollary 4.0.1. *For all i, $\mathbf{V}_i$ is orthonormal. That is, all nodes within a path from the root to some node have orthonormal basis vectors.*

Justification. By our construction of V_i, the j_{th} column is the basis vector of a node in the sub-trees of nodes corresponding to the $\leq j_{th}$ columns. By proposition 4.1, all columns are orthogonal to each other, and by virtue of being some singular vector, all columns are unit length.

Proposition 4.2. *If all nodes in the tree have exactly one child, then the basis vector for the node at the l_{th} level of the tree is the l_{th} singular vector of X.*

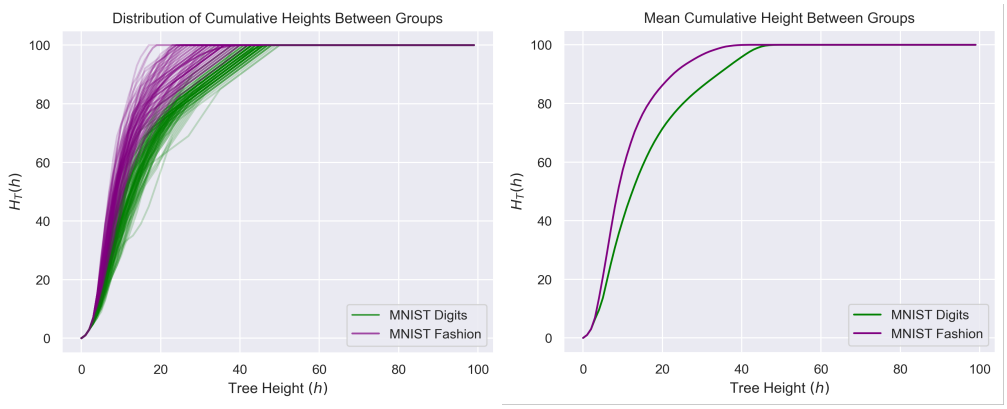

Figure 5: Tree height curves for trees build on the MNIST and MNIST-Fashion datasets.

Justification. Using the alternative node indexing based on node height, $\mathbf{E}_0 = X$, and $\mathbf{v}_1$ is the first right singular vector of X. $\mathbf{v}_2 = X - X\mathbf{v}_1\mathbf{v}_1^T$, where by definition of the SVD, $X\mathbf{v}_1 = \sigma_1\mathbf{u}_1$, so $\mathbf{v}_2 = X - \sigma_1\mathbf{u}_1\mathbf{v}_1^T$, therefore $\mathbf{v}_2$ is the 2_{nd} right singular value of X. More generally, $\mathbf{v}_l$ is the l_{th} right singular vector of X.

Proposition 4.3. *If the edge weight of the edge between a node $\mathcal{N}_i$ and it's parent $\overline{\mathcal{N}_i}$ is $\overline{R_i} - R_i$, the edge weights along any path $\mathcal{P}_i$ from root to node $\mathcal{N}_i$ are monotonically decreasing.*

Justification. By the Eckart-Young theorem, the edge weight $\overline{R_i} - R_i$ is equal to the first singular value of $\overline{\mathbf{E_i}}$. If the edge weight from parent to child $\overline{R_i} - R_i$ was greater than grandparent to parent $\overline{\overline{R_i}} - \overline{R_i}$, this would imply that more variation of a subset of $\overline{\mathbf{E}_i}$ is captured by $\mathbf{v}_i$ than $\overline{\mathbf{v}_i}$, which is impossible because $\overline{\mathbf{v}_i}$ is the first singular vector of $\overline{\overline{\mathbf{E}_i}}$.

Proposition 4.4. *If $X \in \mathbf{R}^D \sim N(0, \Sigma)$, then the PCT tree will have D nodes and no splits with at least probability $1 - (\alpha_{test})^D$.*

Justification. A linear transformation of any multivariate normal is still a multivariate normal, including the projection of X onto some of it's singular vectors in SUBSPACE_CLUSTER_TEST(). Therefore when we test for subspace clustering in the projection of that data, we would erroneously split a node with probability α_{test}. We repeat this test D times.

Proposition 4.5. *$H_T(h)$ is lower and upper bounded. $h \geq H_T(h) \leq 2^h - 1$ $H_T(h)$: is maximized when T forms a complete binary tree, which has $2^h - 1$ nodes at height $\leq h$, and minimised when T has no splits, so it has h nodes at height $\leq h$.*

5 Experimental Results

We now present experimental results demonstrating the ability to use the structure of PCTs to effectively discriminate between distributions. Prior to constructing a PCT for two distributions, we perform two normalization steps to ensure fair comparability. First, we use PCA / the truncated SVD to reduce the dimensionality of the the two distributions to the dimensionality of the lower-dimensional distribution. We do this so that the maximum height of the PCTs constructed on either distribution is equal, and our $H_T()$ curves reflect actual subspace structure differences, not just that one distribution is higher-dimensionality than the other. Secondly, we "color" the distributions to have equal singular values, specifically the i_{th} singular value becomes $D - i$. PCT construction should be invariant to differences in relative singular values as long as their order doesn't change, but we normalize this just to be sure.

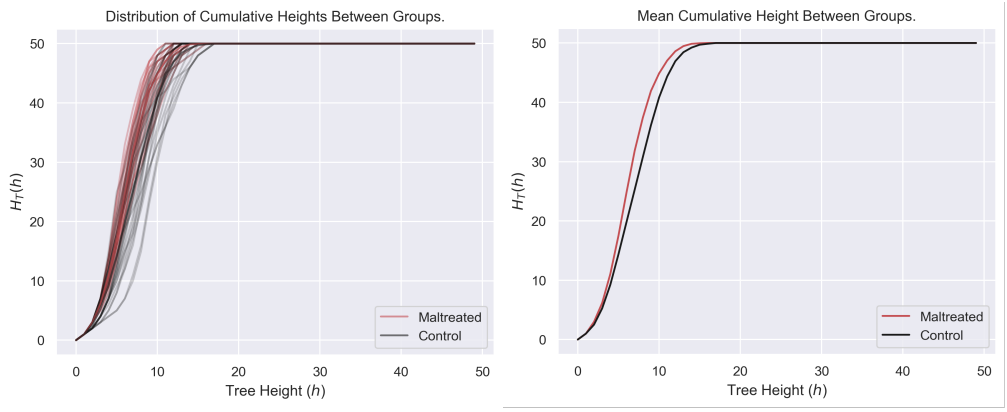

Figure 7: Tree height curves for trees built on Maltreated and Typical child brain jacobians.

5.1 MNIST & Fashion-MNIST

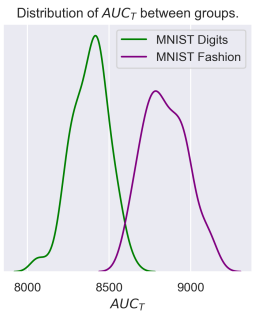

Figure 6: Distribution of AUC_T for MNIST and MNIST-Fashion.

For our first test, we compare the subspace structure of the well-known MNIST dataset [8] and it's more recently made sister, Fashion-MNIST [31]. Both datasets contain 70,000 28x28 black and white images belonging to 10 balanced classes. MNIST is of handwritten digits and Fashion-MNIST is of articles of clothing and accessories, and was designed to be a harder drop-in replacement for the original MNIST. To perform a hypothesis test of difference in subspace clustering, we perform a bootstrap-like rank-sum test of difference of median AUC_T. Specifically, we draw 50 random samples of size 10,000 from each distribution, then within each of those 50 samples, we construct a PCT with 100 nodes, it's corresponding height curve $H_T(h)$, and AUC_T. We also normalize both groups to $D = 64$ for comparability with results in section 5.3. After collecting 50 AUC_T for each group, we perform a two-sample Wilcoxon rank-sum test for differences in median. An example of a tree constructed from both groups is shown in figure 3.2. The distribution and mean of curves from the 50 samples are shown in figure 6, and the distribution of AUC_T is shown in figure 6. In the end, we obtain a **P-Value** of 3.3×10^{-31}, and thus conclude there is a significant difference in subspace cluster structure between the distribution of MNIST and Fashion-MNIST points.

5.2 Maltreated Brain Structure.

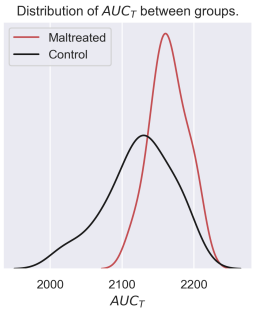

Figure 8: Distribution of AUC_T for brain kinds.

For our second test, we turn to a smaller neuroscience dataset. First introduced and analyzed in [6], this dataset consists of brain measurements on a group of 23 maltreated and developmentally typical children of comparable age and demographics. For each child, the dataset provides a pre-processed collection of 18,881 measurements of jacobian curvature on the child's brain. These are essentially measurements of the degree of curvature at 18,881 locations at the brain's surface. It also notes two smaller subsets of these measurements (548 and 1856) which are well-aligned and representative of the larger structure. With such a small sample size, we are unable to compare the subspace structure of the two groups in the space of their measurements relative to individuals, so we instead compare the space of individuals relative to their measurements. This is also the approach taken by [7], which creates a 548x548 correlation matrix for each group, and then compares the structure of this matrix. In essence, we are testing for difference in subspace structure in how the two group's measurements are internally similar.

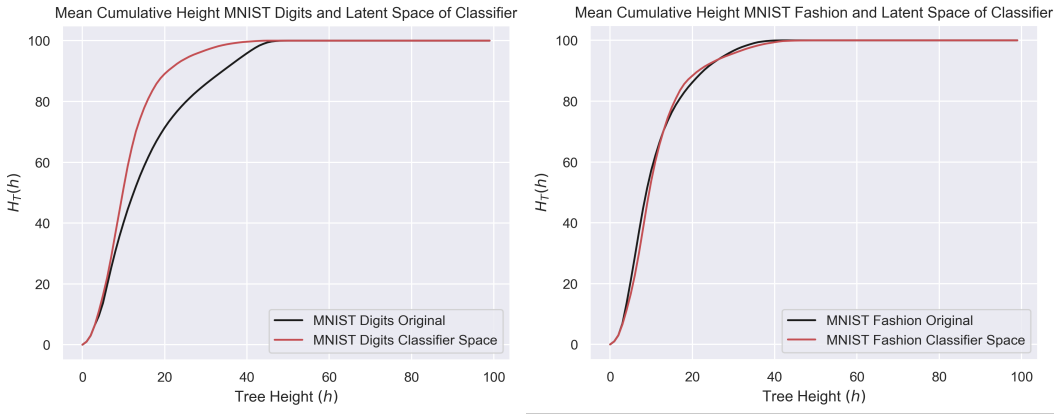

Figure 9: Tree height curves original for the MNIST and MNIST-Fashion datasets, both the original distribution, and the latent space of a classifier neural network trained on the dataset.

Specifically, for the maltreated group, we use leave-one-out jackknife sampling to create 23 matrices of size $18,881 \times 22$, where each matrix is lacking one participant's measurements. For the control group, we do the same, creating 31 matrices of $18,881 \times 30$. We then normalize all of these matrices as described above. Following this re-sampling, we construct trees, height curves (figure 5.2), and AUC_T (figure 8) as in section 5.1 then performing a rank sum test of medians. In the end, we obtain a **P-Value** of 0.00049, and thus conclude there is a significant difference in subspace cluster structure among the structural covariance in maltreated children's brains and neurotypical children's brains.

5.3 Neural Network Latent Space

In the final experiment, we wish to see how the subspace clustered structure of the latent space of neural networks compares to that of the data the network was trained on. To accomplish this, we train networks to perform classification for the MNIST and Fashion-MNIST datasets, then pass the datasets through the first few layers of the dataset before and build a PCT on their intermediate representation. For both datasets, we use a simple multi-layer-perceptron architecture with two hidden layers of size 128 and 64. To train the network, we use the ADAM optimizer and run for 50 epochs, afterwards we pass the data through the first 2 hidden layers, obtaining 64D embeddings. To test for differences in subspace structure, we follow the same setup as section 5.1, generating 50 random samples of 10,000 points, training a new neural network and thus obtaining different embedding for each of the 50 trials for both datasets. We compare the subspace clustered structure of the original points and their classification-trained embeddings using the same hypothesis testing framework.

For the original MNIST dataset, we obtain a very small p-value (2.2×10^{-21}), indicating subspace cluster difference. In some sense, this is expected. For a 10-class classification neural network to perform well, output of the network will be very nearly one-hot 10D vectors. This output is "subspace clustered" in that each class's one-hot vector should be perpendicular to other classes. Therefore, we would expect the output of intermediate layers to approach this structure, and thus become more subspace-clustered. While this assumption seems to hold true for the original MNIST dataset, it doesn't for MNIST fashion, where we detect no difference in the subspace-clustered structure of the original dataset and the classifier latent space. (**P-Value** $= 0.988$). We offer two explanations for this. Firstly, the Fashion-MNIST dataset is already comparatively subspace structured, and $H_T(h)$ is upper-bounded, so it's possible that there isn't any room for the latent space to become more subspace-clustered structured. Furthermore, the classifier for MNIST fashion performs much worse than original MNIST (89% vs 97%), so the lack of induced subspace clustered structure may just be a reflection of this lower performance.

6 Conclusion & Future Work

In summarize, the assumption of multivariate normality has been fundamental to many statistical methods, and principal component analysis (PCA) has been commonly used to model and characterize

such distributions. However, high-dimensional real-world data rarely fit the assumptions of either PCA or the union of subspaces perspective. In this paper, we propose a novel structure, Principle Component Trees (PCTs), which bridges the gap between the two perspectives and facilitates comparison and visualization of the subspace structure of high-dimensional distributions. PCTs correspond to principle component vectors of subspaces and are split into children based on a subspace structure hypothesis test. By using interpretable persistent homology measures, our experimental results demonstrate the effectiveness of PCTs in discriminating and analyzing complex distributions, including brain grey matter and neural network latent spaces.

In the future, we wish to:

- Improve the efficiency and stability of the principle component tree construction. As it stands, the subspace clustering method operates on an NxN matrix, which is computationally prohibitive for large datasets, so we only perform subspace clustering on a random subset, then assign all points to the nearest cluster. This randomness introduces instability into each tree. Also, for easier implementation, we run the SVD multiple times even for the exact same $\mathbf{E}_i$, which can be avoided for an easy speed-up.
- Apply PCTs to new kinds of datasets, in particular applying it directly to correlation matrices, or perhaps graph laplacians.
- Leverage some of the unique properties of PCTs to define more interesting distances between trees, or perhaps to embed them in some vector space. In particular, the path orthogonality and monotonically decreasing edge weights may prove useful.
- Derive theory for the angle-distribution based test of subspace clustering, including deriving a distribution of angles for explicitly whitened gaussians to we don't have to use a monte-carlo based distribution for small sample size, another source of randomness in tree construction. **OR**, use an information-theoretic subspace clustering test instead of an angle based one.

References

[1] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data. *Data Mining and Knowledge Discovery*, 11:5–33, 2005.

[2] U. Bauer. Ripser: efficient computation of vietoris–rips persistence barcodes. *Journal of Applied and Computational Topology*, 5(3):391–423, 2021.

[3] T. T. Cai, J. Fan, and T. Jiang. Distributions of angles in random packing on spheres. *Journal of Machine Learning Research*, 14:1837, 2013.

[4] F. Chazal and B. Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *Frontiers in artificial intelligence*, 4:667963, 2021.

[5] C.-H. Cheng, A. W. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 84–93, 1999.

[6] M. K. Chung, J. L. Hanson, H. Lee, N. Adluru, A. L. Alexander, R. J. Davidson, and S. D. Pollak. Persistent homological sparse network approach to detecting white matter abnormality in maltreated children: Mri and dti multimodal study. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013: 16th International Conference, Nagoya, Japan, September 22-26, 2013, Proceedings, Part I 16*, pages 300–307. Springer, 2013.

[7] M. K. Chung, J. L. Hanson, J. Ye, R. J. Davidson, and S. D. Pollak. Persistent homology in sparse regression and its application to brain morphometry. *IEEE transactions on medical imaging*, 34(9):1928–1939, 2015.

[8] L. Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

[9] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. In *Proceedings 41st annual symposium on foundations of computer science*, pages 454–463. IEEE, 2000.

[10] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2765–2781, 2013.

[11] J. Fan and M. Udell. Online high rank matrix completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8690–8698, 2019.

[12] E. García-Portugués, P. Navarro-Esteban, and J. A. Cuesta-Albertos. On a projection-based class of uniformity tests on the hypersphere. *Bernoulli*, 29(1):181–204, 2023.

[13] P. Hu, X. Li, N. Lu, K. Dong, X. Bai, T. Liang, and J. Li. Prediction of new-onset diabetes after pancreatectomy with subspace clustering based multi-view feature selection. *IEEE Journal of Biomedical and Health Informatics*, 2023.

[14] L. Kanari, P. Dłotko, M. Scolamiero, R. Levi, J. Shillcock, K. Hess, and H. Markram. Quantifying topological invariants of neuronal morphologies. *arXiv preprint arXiv:1603.08432*, 2016.

[15] B. A. Kizaric and D. L. Pimentel-Alarcón. Classifying incomplete data with a mixture of subspace experts. In *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1–8. IEEE, 2022.

[16] H. Lee, H. Kang, M. K. Chung, B.-N. Kim, and D. S. Lee. Weighted functional brain network modeling via network filtration. In *NIPS Workshop on Algebraic Topology and Machine Learning*, volume 3. Citeseer, 2012.

[17] Q. Li, X. Zhao, and H. Zhu. Semi-supervised sparse subspace clustering based on re-weighting. *Engineering Letters*, 31(1), 2023.

[18] Y. Li, D. Wang, G. A. Ascoli, P. Mitra, and Y. Wang. Metrics for comparing neuronal tree shapes based on persistent homology. *PloS one*, 12(8):e0182184, 2017.

[19] J. Lipor, D. Hong, Y. S. Tan, and L. Balzano. Subspace clustering using ensembles of k-subspaces. *Information and Inference: A Journal of the IMA*, 10(1):73–107, 2021.

[20] G. Liu, K. Sim, J. Li, and L. Wong. Efficient mining of distance-based subspace clusters. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 2(5-6):427–444, 2009.

[21] Y. Ma, H. Derksen, W. Hong, and J. Wright. Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE transactions on pattern analysis and machine intelligence*, 29(9):1546–1562, 2007.

[22] U. Mahmood and D. Pimentel-Alarcón. Fusion subspace clustering for incomplete data. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022.

[23] M. McCartin-Lim, A. McGregor, and R. Wang. Approximate principal direction trees. *arXiv preprint arXiv:1206.4668*, 2012.

[24] V. Menon, G. Muthukrishnan, and S. Kalyani. Subspace clustering without knowing the number of clusters: A parameter free approach. *IEEE Transactions on Signal Processing*, 68:5047–5062, 2020.

[25] C. S. Pun, S. X. Lee, and K. Xia. Persistent-homology-based machine learning: a survey and a comparative study. *Artificial Intelligence Review*, 55(7):5169–5213, 2022.

[26] K. Rafiezadeh Shahi, M. Khodadadzadeh, L. Tusa, P. Ghamisi, R. Tolosana-Delgado, and R. Gloaguen. Hierarchical sparse subspace clustering (hessc): An automatic approach for hyperspectral image analysis. *Remote Sensing*, 12(15):2421, 2020.

[27] E. Thordsen and E. Schubert. Abid: Angle based intrinsic dimensionality—theory and analysis. *Information Systems*, 108:101989, 2022.

[28] R. Vidal, R. Tron, and R. Hartley. Multiframe motion segmentation with missing data using powerfactorization and gpca. *International Journal of Computer Vision*, 79:85–105, 2008.

[29] A. Wichert. Subspace tree. In *2009 Seventh International Workshop on Content-Based Multimedia Indexing*, pages 38–43. IEEE, 2009.

[30] T. Wu, P. Gurram, R. M. Rao, and W. U. Bajwa. Hierarchical union-of-subspaces model for human activity summarization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1–9, 2015.

[31] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[32] P. Zhu, B. Hui, C. Zhang, D. Du, L. Wen, and Q. Hu. Multi-view deep subspace clustering networks. *arXiv preprint arXiv:1908.01978*, 2019.