Principle Component Trees and Their Persistent Homology

Ben Kizaric\textsuperscript{1,3}, Daniel Pimentel-Alarcón \textsuperscript{2,3}

\textsuperscript{1}Department of Electrical Engineering, University of Wisconsin-Madison
\textsuperscript{2}Department of Biostatistics, University of Wisconsin-Madison
\textsuperscript{3}Wisconsin Institute For Discovery
benkizaric@gmail.com, pimentelalar@wisc.edu

Abstract

Low dimensional models like PCA are often used to simplify complex datasets by learning a single approximating subspace. This paradigm has expanded to union of subspaces models, like those learned by subspace clustering. In this paper, we present Principal Component Trees (PCTs), a graph structure that generalizes these ideas to identify mixtures of components that together describe the subspace structure of high-dimensional datasets. Each node in a PCT corresponds to a principal component of the data, and the edges between nodes indicate the components that must be mixed to produce a subspace that approximates a portion of the data. In order to construct PCTs, we propose two angle-distribution hypothesis tests to detect subspace clusters in the data. To analyze, compare, and select the best PCT model, we define two persistent homology measures that describe their shape. We show our construction yields two key properties of PCTs, namely ancestral orthogonality and non-decreasing singular values. Our main theoretical results show that learning PCTs reduces to PCA under multivariate normality, and that PCTs are efficient parameterizations of intersecting union of subspaces. Finally, we use PCTs to analyze neural network latent space, word embeddings, and reference image datasets.

1 Introduction

Learning a low-dimensional structure that approximates a high-dimensional dataset is a fundamental problem in science and engineering — e.g., learning an approximating subspace using \textit{principal component analysis} (PCA) (Pearson 1901; Hotelling 1933; Jolliffe 2002). This paper introduces a new approximating structure, which we call \textit{principal component trees} (PCT), of which PCA and generalized PCA (also known as subspace clustering (Elhamifar and Vidal 2013)) are special cases.

PCT builds on the union of subspaces model, which assumes that each data sample lies near one of several low-dimensional subspaces. Subspace clustering is then the task of learning the structure of these subspaces and has shown effectiveness in modeling many kinds of data (Vidal, Tron, and Hartley 2008; Mahmood and Pimentel-Alarcón 2022; Li, Zhao, and Zhu 2023). Baked into the majority of subspace clustering algorithms, such as the effective Sparse Subspace Clustering (SSC) method (Elhamifar and Vidal 2013), is the assumption that the subspaces containing the data are disjoint. That is, all points near the span of one subspace are not near the span of any other subspace.

It is this disjoint assumption that motivates Principal Component Trees, which can represent a single subspace, multiple disjoint subspaces, and multiple intersecting subspaces. In essence, a PCT is a graph summarizing the hierarchical structure of the principal components in the data. Each node of a PCT corresponds to a single principal component, defined by a principal vector and a singular value. Each edge of a PCT links principal vectors that, together, define a set of subspaces that contain the data. Each of these subspaces is spanned by a \textit{branch} of the PCT, which is the set of nodes that lie on the path between each leaf of the PCT and its root. Branches that share nodes correspond to intersecting subspaces (see Figure 1).

Our algorithm to learn a Principal Component Tree analyzes the structure of the data dynamically, recursively adding child nodes, depending on the structure of the data. Each node will have more than one child only if we detect disjoint subspace clusters. We make this determination with a non parametric hypothesis test which compares the distribution of pairwise angles of the data points to the known distribution of uniform angles. If no disjoint structure is detected at any node, the PCT will result in a degenerate tree.
In this case, PCT reduces to PCA. Similarly, the union of subspaces model used by classical subspace clustering is the special case of PCT where a disjoint structure is detected at the root and nowhere else. In general, data following a multivariate normal distribution will yield tall trees, and data exhibiting subspace clustered structure will yield wide trees.

To quantitatively measure tree structure, we turn to tools in persistent homology that measure properties of graphs as they are filtered into sub-graphs. Specifically, we perform a height filtration that measures how many nodes are below a given height, and a width filtration, which measures the width of the tree as we prune the least significant nodes. These measures yield bounded, monotonic curves that summarize the subspace structure of the PCT, and can be used to perform hypothesis testing between PCTs. Our experimental results show the effectiveness of both summaries on synthetic and real-world data, including multilingual word embeddings and the latent space of neural networks.

Our main theoretical contribution shows that any distribution that can be parameterized using PCA or generalized PCA can be equivalently represented with a PCT with as many or fewer parameters. Our second theoretical contribution relates to our non parametric hypothesis test. We are able to show that with high probability (w.h.p) our test will accurately detect clusters of correlated data, which we treat as a proxy for subspace structure. As a corollary of this result, we show that w.h.p our algorithm will result in a degenerate tree (i.e., will produce the same result as PCA) whenever data is best represented by a single subspace.

Similar to PCA, our PCT enjoys several favorable properties. Firstly, the principal vector of each node is orthogonal to all of its ancestors and descendants, making projection of data onto each node of the tree efficient and decomposable. We also show that the singular value of a node is strictly less than or equal to that of its parent and ancestors. In standard PCA, the components corresponding to the largest singular values are the subset of components enabling optimal projection approximation of the datapoints. We prove that the monotonically decreasing singular values gives PCT a similar property, that the optimal approximating subtree of a PCT is the subtree with nodes with largest singular values. The decreasing singular values are also essential to enable the aforementioned width filtration topological measure.

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 (Vidal, Tron, and Hartley 2008) and Hyperspectral Imaging (Li, Zhao, and Zhu 2023), others are more interested in using the learned subspace structure for downstream tasks. For example, subspace clustering has shown to be effective in matrix completion problems (Fan and Udell 2019), or as a feature extraction method (Hu et al. 2023; Kizaric and Pimentel-Alarcón 2022). Methods for performing subspace clustering utilize sparse self-expressiveness (Elhamifar and Vidal 2013), expectation maximization (Lipor et al. 2021), deep learning (Zhu et al. 2019), and fusion over the Grassmannian (Mahmood and Pimentel-Alarcón 2022).

An important distinction in subspace clustering techniques is the difference between axis-parallel and linear methods. In axis-parallel methods such as CLIQUE (Agrawal et al. 1998) and (Yuan et al. 2013) the subspaces are assumed be be spanned by a discrete subset of the variables / features of the dataset. In linear methods, the subspaces are allowed to span any linear combination of the variables, in a continuous space known formally as the Grassmanian. As illustrated in figure 1, the nodes of Principal Component Trees represent an arbitrary direction in the data, and as such is a generalization of linear subspace clustering. Although it is comparatively well-studied, the discreet sample space in axis-parallel regimes enables the use of an entirely different class of algorithms, so we focus our comparisons only on linear methods.

Of particular interest to this paper are hierarchical subspace clustering techniques. These approaches come in top-down (Rafiezadeh Shahi et al. 2020; Wu et al. 2015) and bottom-up (Menon, Muthukrishnan, and Kalyani 2020) flavors. (Rafiezadeh Shahi et al. 2020) and (Wu et al. 2015) are similar to Principal Component Trees in that they are top-down methods which recursively partition the space into lower dimensional subspaces, and the termination of node splitting is based on a reconstruction error criteria.

While the construction of PCTs is fundamentally a hierarchical subspace clustering algorithm, it is the only one whose nodes represent a single basis, and that can generalize to PCA, disjoint subspaces, and intersecting subspaces. Similarly named similarity search methods like (McCartin-Lim, McGregor, and Wang 2012) and (Wichert 2009) exist, but these methods don’t use subspace clustering.

Angle Distributions Established results on the distribution of pairwise angles in high dimensional space are important to PCT’s adaptive splitting construction. These results were first established in (Cai, Fan, and Jiang 2013) and used to estimate intrinsic dimensionality in (Thordsen and Schubert 2015). (Menon, Muthukrishnan, and Kalyani 2020) uses pairwise angle distributions in subspace clustering, but for merging subspaces from the bottom-up, unlike our top-down approach. Hypothesis tests of angle uniformity are central to directional statistics, and are explored in (García-Portugués, Navarro-Esteban, and Cuesta-Albertos 2023).

Dictionary Learning and Sparse Coding Primarily a compression and denoising technique, Dictionary Learning and Sparse Coding seek to find a dictionary $V$ and sparse codes $U$ in order to minimize $\|X - UV\|_F + \alpha \|U\|_1$. Implementations utilize autoencoders (Ng et al. 2011), alternating optimization (Tosic and Frossard 2011), and other techniques. Structured Sparse Coding places additional constraints on $U$ and $V$ in order to form advantageous structures. For example, (Jenatton et al. 2010) force the non-zero coefficients to form a pre-defined tree hierarchy, and its optimization procedure promotes orthogonality in the branches of dictionary atoms in the tree. The atoms in (Szlam, Gregor, and LeCun 2012) are placed into overlapping groups which are learned from data, but don’t form a tree.

Topological Data Analysis & Persistent Homology Be-
cause of its ability to discover robust patterns in arbitrary distributions like networks and point clouds, topological data analysis (TDA) has emerged as a promising approach in medical imaging and other areas (Chazal and Michel 2021). Persistent Homology is a sub-field of TDA that focuses on measuring properties of graphs as they are filtered down (Pun, Lee, and Xia 2022). While persistent homology is most commonly used to construct Birth-Death decomposition charts, Rip’s Filtration (Bauer 2021) and Graph Filtration (Lee et al. 2012) are robust approaches that enable theoretically sound statistical inference. The key property of Graph Filtration is that it tracks the first two Betti numbers of directed edges.

Of particular relevance to our analysis of PCTs are persistent homology based characterizations of tree structure. (Chung et al. 2015) build minimum spanning trees from structural covariance networks then apply graph filtration to create curves used to compare network structure. Persistent Homology has also been applied to tree-structured data. Both (Li et al. 2017) and (Kanari et al. 2016) filter the binary tree structure of neurons and brain arteries based on height / distance to the neuron’s cell body. Persistent structures are then described by a birth-death composition chart. Our analysis of PCTs borrows this height filtration approach, but instead creates monotonic curves as in (Chung et al. 2015).

3 Principal Component Tree

Consider a data matrix $X$ whose columns $\{x\}$ represent samples in $\mathbb{R}^D$. A principal component tree $T$ to model data $X$ is defined by a set of nodes $N_i$, $i \in \{1, \ldots, |T|\}$, and a set of directed edges $\mathcal{E}_i$, indicating that node $N_i$ is the parent of node $N_j$. Given node $N_j$, we denote its (unique) parent as $\mathcal{P}_i$, its set of ancestors as $\mathcal{A}_i$, with $N_i \in \mathcal{A}_i$, its set of children as $\mathcal{C}_i$, and its set of descendants as $\mathcal{D}_i$. We use $\mathcal{S}_i$ to denote the set of siblings of $N_i$, defined as the set of nodes with the same parent as $N_i$.

A node $N_i$ represents two objects: (a) its associated principal component, given by a unit-length vector $v_i \in \mathbb{R}^D$, and (b) its singular value $\sigma_i \geq 0$. Much like PCA, each node’s $v_i$ represents a direction of variation in the data, and its $\sigma_i$ is proportional to the amount of variation along this direction. However, unlike PCA, these quantities describe variation directions of specific subsets of the data, not just the data as a whole. These subsets of data and their particular variation directions are determined by the nodes and their lineage. To see this, define the ancestral basis associated to node $N_i$ as

$$V_i := \{v_i \mid N_i \in \mathcal{A}_i\}.$$ 

This basis spans a subspace that approximates a subset of the samples in $X$. Specifically, we assign each node of the tree a subset of the data $X_i$, defined recursively as:

$$X_i := \left\{ x \in X \mid \arg \max_{j \in \mathcal{S}_i} |\langle v_j, x \rangle| = i \right\}.$$

Intuitively, the subset of columns assigned to node $N_i$ is the subset of columns that were assigned to its parent, and that are closer to the principal component of $N_i$ than to that of any of its siblings. See Figure 2 for a one sample example.

Notice that under these definitions, the bases and data assignments of ancestors and descendants are nested, i.e., for any $N_j \in \mathcal{A}_i$,

$$\text{span}(V_j) \subseteq \text{span}(V_i) \text{ and } X_j \supseteq X_i$$

That is, the basis for an ancestor is a subspace of the basis for all its descendants, and the data assigned to an ancestor is a superset of the data assigned to its descendants.

Approximating Data

We will now show how Principal Component Trees can be used to approximate data in a similar fashion to PCA. That is, to create an approximation $\tilde{X}$ of $X$. Whereas PCA approximations can be computed by a single matrix operation, PCT approximations are a union of approximations given individual nodes of their respective assigned data. For a single node, its node approximation and node residual are:

$$\tilde{X}_i = V_i V_i^T X_i \text{ and } R_i := X_i - \tilde{X}_i$$

respectively. The node approximation is simply a node’s assigned data projected onto its ancestral basis. Ordinary least squares projection would require that we also compute $(V_i V_i^T)^{-1}$, but our tree construction actually gives an orthonormal basis for all $V_i$ (Property 6.1). The node residuals are used extensively in our tree construction; the descendants of $N_i$ are learned to best approximate $R_i$. Intuitively, the number of children of $N_i$ is the number of disjoint subspace clusters detectable in $R_i$ and the principal components of those children are the first singular vectors of those disjoint subspaces.

To compute the approximation of the entire dataset, we make use of the fact that for leaves of the tree (nodes with no children), data assignments are disjoint. Furthermore, the subspaces defined by leaf nodes are not nested. More exactly, for two leaf nodes $N_l$ and $N_g$:

$$\text{span}(V_l) \not\subseteq \text{span}(V_g) \text{ and } X_l \cap X_g = \emptyset$$

Now that we have single node approximations and have shown that data is disjointly partitioned between leaf nodes, we can compute the full approximation of some data $X$ as:

$$\tilde{X} := \bigcup_{i=1}^{|N_L|} \tilde{X}_l$$

Where $N_L$ is the set of all leaves of the tree.
4 Subspace Structure Test

Testing the Presence of Subspace Clusters

Key to our construction of the Principal Component Tree is our adaptive splitting algorithm, which builds the tree top to bottom, expanding node $N_i$ with either (a) a single child at a time, if its data residuals $R_i$ show no sign of subspace structure, or (b) with multiple children, if we detect sufficient subspace structure in $R_i$.

To make this determination we use the Cramér-von Mises test to compare the distribution of the angles of the residuals $R_i$ with the known distribution under uniformity on the hypersphere. We motivate the use of this test as follows. If data is subspace clustered, there will be directions in the space that are crowded with points and directions that are comparatively empty. This non-uniform use of the space will cause the distribution of angles between points of subspace clustered data to differ from the known distribution for uniform points. We treat the degree of angle non-uniformity as a proxy for the degree of subspace clusteredness.

This measure of subspace clusteredness is summarized in a single $p$-value $p$ where a small $p$ indicates subspace clustering. To compute this $p$-value, we require three things. 1) The empirical cumulative distribution function (CDF) $\hat{P}_D(C)$ which describes the observed distribution of angles of our $D$-dimensional residuals. 2) The CDF $P_D(C)$ which defines the expected distribution of angles under the null hypothesis of no subspace structure i.e. uniformity. 3) The Cramér-von Mises test statistic $Q$ which quantifies the difference between $\hat{P}_D(C)$ and $P_D(C)$, and therefore the degree of subspace clustering.

For notational clarity, we henceforth discuss the test as applied to a generic data matrix $X$ whose columns $\{x_1, x_2, ..., x_N\}$ represent samples in $\mathbb{R}^D$.

To perform this test, we first measure the angle distribution of our data. More precisely, we measure the distribution of absolute cosine similarities between pairs of points of $X$.

Taking all pairs, we compute the $\frac{N(N-1)}{2}$ angles of $X$,

$$\angle X := \left\{ \frac{|\langle x_k, x_j \rangle|}{||x_k|| \cdot ||x_j||} \bigg| k < j \right\} \angle x_i \in [0, 1].$$

We use absolute cosine similarity because subspaces pass through the origin. All points on a line through the origin to should be deemed similar, regardless of whether they are on the same side of the origin. Then we describe the distribution of angles via the empirical distribution function,

$$\hat{P}_D(C) := \frac{1}{N(N-1)} \sum_{i=1}^{N(N-1)} 1\{\angle x_i \leq C\}$$

In order to determine if our data $X$ exhibits subspace clusteredness, we need to compare the observed distribution of angles to an expected null distribution. In this test, our null hypothesis is that our data has angular uniformity; that the unit length normalized data is sampled uniformly from the hypersphere embedded in $\mathbb{R}^D$. For example, the isotropic gaussian distribution has angular uniformity.

Thankfully, the expected distribution of pairwise angles on the hypersphere has already been derived by (Cai, Fan, and Jiang 2013) for standard angles $\theta \in [0, 2\pi]$ and (Thorsen and Schubert 2022) for absolute cosine similarity $C \in [0, 1]$. The latter is a simple transformation of the univariate beta distribution, whose CDF is:

$$P_D(C) = 2 \text{BetaCDF} \left( \frac{1+C}{2}; \alpha = \frac{\beta-1}{2}, \beta = \frac{\beta-1}{2} \right) - 1.$$

With the observed and expected/null distributions in hand, we can finally compare them via the Cramér-von Mises test. This test quantifies the distance between two distributions into the test statistic

$$Q = N(N-1) \int_0^1 \left( P_D(C) - \hat{P}(C_D) \right)^2 dC.$$}

The known domain of $C$ allows us to bound the otherwise improper integral of the CvM statistic. Computation of the final $p$-value is given in (Csörgő and Faraway 1996), equation I.8.

**Practical Considerations.** For simplicity and interpretability, we leave three important implementation details about this and the following test to the technical appendix. 1) In order to reasonably compare angle distributions to those of hyperspheres, we first whiten $X$ before taking the angle distribution. 2) This explicit whitening makes the expected angle distribution $P_D(C)$ dependent on $N$, so we used one construct a simulation to estimate $P_D(C)$ for $N < 100D$. 3) Whenever comparing the subspace clusteredness of two data matrices, we first project both to a common dimension.

Testing an Intersection of Subspace Clusters

In the previous subsection, we state that we will split a node $N_i$ in two if we detect subspace clusters in its residuals $R_i$. This is a simplification; if we only do this, we may split nodes prematurely and leave our PCT construction unable to find intersections of subspaces. For example, take $X_B$ and $X_C$ in Figure 2. Both $X_B$ and $X_C$ have a very small $p$-value, small enough to correctly conclude there is subspace
PCT. Put simply, we build a tree by selecting the
most interpretable topological measures that describe the shape of
clustering in the data. However, if we split \( X_B \) right away, we
would miss the fact that the two subspaces have a prominent 1D intersection (labeled \( v \)).

In order to build PCTs with the capacity to represent intersecting subspaces, we now present a test that extends on the previous subsection. This test evaluates whether or not potential subspace clusters plausibly intersect along a given axis / basis \( v \). We propose that if some data \( X \) projected onto the null space of \( v \) appears more subspace clustered than the original data, then \( v \) is a plausible intersection of subspace clusters present in \( X \). More exactly, our test \( \textsc{intersect-test}(X, v) \) returns \texttt{true} if

\[
\textsc{cluster-test}(X - vv^T X) < \textsc{cluster-test}(X)
\]

Back to the example of Figure 3, notice that \( X_C \) is actually the projection of \( X_B \) onto the null space of \( v \), and that \( X_C \) has an even smaller p-value than \( X_B \). In this case, we would conclude that \( v \) is a plausible subspace intersection of \( X_B \).
PCTs. Inspired by (Chung et al. 2015), both of these measures take the form of bounded, monotonic curves measuring the size or shape of a tree as nodes are pruned away.

As a guiding principal, the more subspace clustered X is, the more nodes will be split into multiple children, and the tree will be wider; it will have more leaves. Conversely, the tree will be taller the less subspace clustered X is. In the case where X follows a multivariate normal distribution, the tree will be degenerate and maximally tall. Our two topological measures quantify this intuition by defining curves that describe the height and width of the tree. Assume that the empty root node is left out of calculations.

**Height Curves.** The first topological measure of a PCT is the tree’s height curve. Put simply, the height curve characterizes the distribution of node heights, similar to a CDF. For a tree T, it is defined as:

\[
H_T(n) = \sum_{i=1}^{\lvert T \rvert} \mathbb{I}\{h_i \leq n\}.
\]

Intuitively, the height curve will be higher for more subspace clustered trees. More subspace clusters lead to more nodes with multiple children which leads to more nodes at lower heights. The height curve is lower bounded by \(H_T(n) = n\). This occurs only for degenerate / “PCA” PCTs, where there is exactly one node at each height. The height curve is upper bounded by \(H_T(n) = \lvert T \rvert\). Only trees where all \(\lvert T \rvert\) nodes are children of the root will have this curve. In this case, the tree models a union of \(\lvert T \rvert\) one dimensional disjoint subspace clusters.

**Width Curves.** We also propose width curves as a topological descriptor of PCT shape. For this curve, we measure tree width (number of leaves) of the subtree of T consisting of only the most significant nodes, according to their singular values. Formally, this subtree consisting of the n most significant nodes is

\[
T_{[n]} = \{N_i \mid \sigma_i \geq \sigma(n)\}, \{E_{ij} \mid \sigma_i, \sigma_j \geq \sigma(n)\}
\]

where \(\sigma(n)\) is the n-th largest singular value of nodes in T. In property 6.3 we prove that \(T_{[n]}\) is the best approximating subtree of T. With this subtree defined, a PCT’s width curve is then

\[
W_T(n) = \sum_{i=1}^{\lvert T \rvert} \mathbb{I}\{i \in T_{[n]} \land C_i \cap T_{[n]} = \emptyset\}.
\]

That is, the number of nodes who are both in \(T_{[n]}\) and leaves of \(T_{[n]}\). It may be easier to imagine recording the width of a tree as the least significant nodes are removed.

Similar to height curves, the width curves will be higher for more subspace clustered trees. Also like height curves, width curves are lower bounded by \(W_T(n) = 1\) and upper bounded \(W_T(n) = \lvert T \rvert\) for degenerate trees and maximally wide trees, respectively.

**Area Under Curves** We have just shown that \(H_T(n)\) and \(W_T(n)\) are monotonic and maximized under a large degree of subspace clusteredness. Therefore, the area under each curve is an effective one number summary of the subspace-clusteredness of the dataset used to construct a tree. The area under height and width curves are

\[
H_T = \sum_{n=1}^{\lvert T \rvert} H_T(n) \quad \text{and} \quad W_T = \sum_{n=1}^{\lvert T \rvert} W_T(n).
\]

In section 7 we perform hypothesis testing to show differences in subspace clusteredness between various datasets.

### 6 Main Theoretical Results

In this section we present two theoretical results that demonstrate the representational capacity of Principal Component Trees and the ability of their construction to perform PCA under correct conditions. Detailed proofs in the Appendix.

**Theorem 6.1** Any union of subspaces, or distribution lying on a union of subspaces can be equivalently represented by a PCT using the same or fewer parameters.

**Proof Sketch.** We prove this result for a union of two subspaces. Any mixture of 2 d-dimensional subspaces will require 2d vector parameters. But if these subspaces have a c-dimensional intersection, we can a) find that intersection analytically, and b) create a PCT representing that same mixture with only 2d - c parameters.
Theorem 6.2 When presented data following a multivariate normal distribution, PCT construction will reduce to PCA, yielding a degenerate tree with probability at least $(1 - \alpha_{test})^D$.

**Proof Sketch.** If the data is truly multivariate normal, then all projection residuals will also be multivariate normal. Therefore, when we test for subspace clustering in $R_i$ for any node, we will erroneously split a node with probability $\alpha_{test}$. Repeating this test $D$ times, we will never split a node with probability $(1 - \alpha_{test})^D$.

We also prove that our PCT learning algorithm leads to two simple but useful properties of PCTs. Firstly, nodes are orthogonal to their ancestors, making calculation of data approximations fast and decomposable. Secondly, node singular values decrease with height. Together these two properties are used in property 6.3, which provides a simple and efficient way to select the best approximating subtree of an existing tree by selecting its most significant nodes. This is like selecting the best principal components of PCA.

**Property 6.1** The singular vector of any node is orthogonal to those of its ancestors. Formally,
\[ v_i \perp v_j \forall j \in A_i \]
as a consequence, all ancestral bases are orthonormal.

**Proof Sketch.** This property arises from the tree construction. As a singular vector of $R_i$, each $v_i$ is somewhere in the null space of $v_{P_i}$, its parent’s basis, so $v_i \perp v_{P_i}$. By induction, it is orthogonal to all its ancestors.

**Property 6.2** A node’s singular value is less than or equal to its ancestors. Formally,
\[ \sigma_i \leq \sigma_j \forall j \in A_i \]

**Proof Sketch.** We show this by contradiction. Consider a node $N_i$ and its parent $P_i$. Let $R_{P_i}$ be the residual used to create the parent $P_i$. Both $\sigma_{P_i}$ and $\sigma_i$ are derived from $R_{P_i}$, but $\sigma_i$ is its singular value. If $\sigma_i > \sigma_{P_i}$, it would violate the Eckart-Young theorem. Therefore, $\sigma_i \leq \sigma_{P_i}$. By induction, $\sigma_i \leq \sigma_j, j \in A_i$.

**Property 6.3** The best size $n$ approximating subtree of $T$ is given by $T_{[n]}$ (equation 1), i.e. selecting the $n$ nodes with largest singular values. More exactly,
\[
\arg\min_{T' \subset T} ||X - \hat{X}_{T'}||_F := T_{[n]}
\]

**Proof Sketch.** We prove this in three parts. 1) We show that property 6.1 means that the approximation error is inversely proportional to the sum of singular values in the tree. 2) We can therefore minimize the approximation error by selecting the nodes with largest singular values. 3) Property 6.2 implies that $T_{[n]}$ will share the root of $T$, and will have no gaps, that is $N_i \in T_{[n]} \implies N_j \in T_{[n]} \forall N_j \in A_i$.

7 Experiments

With our persistent homology measures of tree shape defined, we now analyze the structure of some real world high-dimensional datasets by comparing the structure of Principal Component Trees learned from those datasets. Our goal is not to discriminate between samples, but to conclude something about the clusteredness of the datasets themselves.

In order to do this, we: 1) take the datasets we wish to compare, and pre-process them by normalizing, but not completely whitening their singular values. This way, differences in PCT structure are purely the result of subspace structure, not anisotropic covariance. 2) Bootstrap sample each dataset into a number of equally sized sub-datasets. Differently sized bootstrap samples between datasets would impact the hypothesis test used to construct the trees. 3) Construct a PCT and calculate $H_T$ and $W_T$ for each bootstrap sample. 4) Compare $H_T$ and $W_T$ for each dataset using a wilcoxon rank-sum test. We also plot these metrics on scatter plots.

**MNIST Digits vs Fashion.** For our first experiment, we compare the widely known MNIST Digits and Fashion datasets (Deng 2012; Xiao, Rasul, and Vollgraf 2017). On paper, they are very similar: 60,000 32x32 images with 10 balanced classes each.

When we compare the subspace clusteredness of the digits and fashion datasets, the results are clear: MNIST Fashion is much more subspace clustered. The results in Figure 4 tell the story. Both the average height and width curve in 4b is higher for the fashion dataset, and this is reflected in areas under each curve. Anecdotally, the sample tree (4c) for MNIST digits shows a single dominant “branch” of nodes whereas the fashion tree has two dominant branches.

**Neural Network Latent Space & Non-Linear Dimensionality Reduction** “What do neural networks learn?” is a long running question in Machine Learning, and there is no shortage of answers in the literature. In this experiment, we answer this question by analyzing the latent space of different neural network architectures via Principal Component Trees. Specifically, we compare the latent space of a Multi Layer Perceptron classifier, simple auto encoder (AE), and variational auto encoder (VAE) which have been trained on the MNIST Digits dataset. We also include the digits after...
To ensure our word embeddings cover similar concepts, we use the TED 2020 dataset (Cettolo, Girardi, and Federico 2012), where each language’s embedding were trained on the same TED talks translated into each language. Amongst the Neural Network latent spaces, the latent space of the simple autoencoder (labeled AE & VAE), the results are consistent with the objective function and effectiveness in enhancing high-dimensional clusters. Amongst the Neural Network latent spaces, the latent space of the classification model is much more subspace clustered than the original data or the other architectures. This is ultimately unsurprising. The end goal of this model is to transform the original data into one-hot vectors which are maximally clustered under perfect accuracy. We posit that this objective leads to more subspace clustering in the intermediate layers of the model.

Looking at the standard and variational autoencoder (labeled AE & VAE), the results are consistent with the objective of these models. The subspace clusteredness of the original data is preserved in the latent space of the simple autoencoder, likely because it simply tried to accurately reproduce the training data. For the variational autoencoder, the latent space is less subspace clustered than the original. This is expected as the loss function of VAEs encourages the latent space to take the form of an isotropic gaussian function. For this comparison, it appears that $W_T$ is a more sensitive measure than $H_T$.

**Multilingual word embeddings.** For our final experiment, we compare the subspace clusteredness of multilingual word embeddings in three languages. We use the embeddings presented in (Ferreira, Martins, and Almeida 2016), where each language’s embedding were trained on the TED 2020 dataset (Cettolo, Girardi, and Federico 2012), where the same TED talks are translated into each language. To ensure our word embeddings cover similar concepts, we examine the 1000 most common words from each language. Under these circumstances, we would expect the embeddings to be structurally similar across languages, but that’s not what we see in Figure 5. Each of the three kinds of embeddings have different degrees of subspace clusteredness, with Italian embeddings being the least and Russian embeddings being the most clustered.

**8 Conclusion**

In conclusion, this paper introduced Principal Component Trees (PCTs) as a framework for identifying mixtures of components that collectively represent the subspace structure within high-dimensional datasets. The proposed PCTs extend conventional low-dimensional models such as PCA and union of subspaces by incorporating a graph structure. Each node in the PCT corresponds to a principal component, with edges representing the necessary combinations of components required to form a subspace that approximates specific portions of the data.

To construct PCTs, two angle-distribution hypothesis tests are introduced to detect intersecting subspace clusters within the data. Additionally, for the analysis, comparison, and selection of optimal PCT models, we use two novel and interpretable persistent homology measures to examine their shape. The constructed PCTs exhibit two interesting properties: ancestral orthogonality and non-decreasing singular values, which in turn facilitate the computation of persistent homology measures.

Theoretical results demonstrate that learning PCTs reduces to PCA under multivariate normality, and that Principal Component Trees serve as efficient parameterizations for intersecting unions of subspaces. Finally, we showcase the practical applicability of PCTs by analyzing neural network latent spaces, word embeddings, and reference image datasets. This research advances our understanding of subspace modeling and offers a powerful tool for dissecting complex high-dimensional datasets in various domains.
References


Li, Q.; Zhao, X.; and Zhu, H. 2023. Semi-supervised Sparse Subspace Clustering Based on Re-weighting. Engineering Letters, 31(1).


