HIERARCHICAL DATA VISUALIZATION VIA PCA TREES

Miguel Á. Carreira-Perpiñán and **Kuat Gazizov**
Dept. of Computer Science & Engineering, UC Merced

1 Introduction

We propose a new model for dimensionality reduction, the PCA tree, which works like a regular autoencoder, having explicit projection and reconstruction mappings. The projection is effected by a sparse oblique tree, having hard, hyperplane splits using few features and linear leaves. The reconstruction mapping is a set of local linear mappings.

Work partially supported by NSF award IIS–2007147.

3 Optimization and time complexity

Our objective function is the regular reconstruction error of an autoencoder $\mathbf{T}$: $\mathbb{R}^D \rightarrow \mathbb{R}^D$ with an ℓ_1 regularization term of hyperparameter $\lambda \geq 0$ on a training set $\{\mathbf{x}_n\}_{n=1}^N \subset \mathbb{R}^D$:

$$E(\mathbf{\Theta}) = \sum_{n=1}^{N} \|\mathbf{x}_n - \mathbf{T}(\mathbf{x}_n; \mathbf{\Theta})\|_2^2 + \lambda \sum_{i \in \mathcal{D}} \|\mathbf{w}_i\|_1 \quad \text{s.t.} \quad \mathbf{U}_j^T \mathbf{U}_j = \mathbf{I},\ \forall j \in \mathcal{L}.$$

We use a variation of the Tree Alternating Optimization (TAO) approach to train the model. It repeatedly updates the nodes in turn: at a leaf it solves a PCA, and at a decision node it solves an ℓ_1-regularized, logistic regression problem. The training complexity is at most linear in N and quadratic in D, just like in regular PCA. As a useful summary, *the rough cost is $\Theta(ND^2)$ for shallow trees and $\mathcal{O}(ND)$ for deep trees*—which is asymptotically faster than PCA!

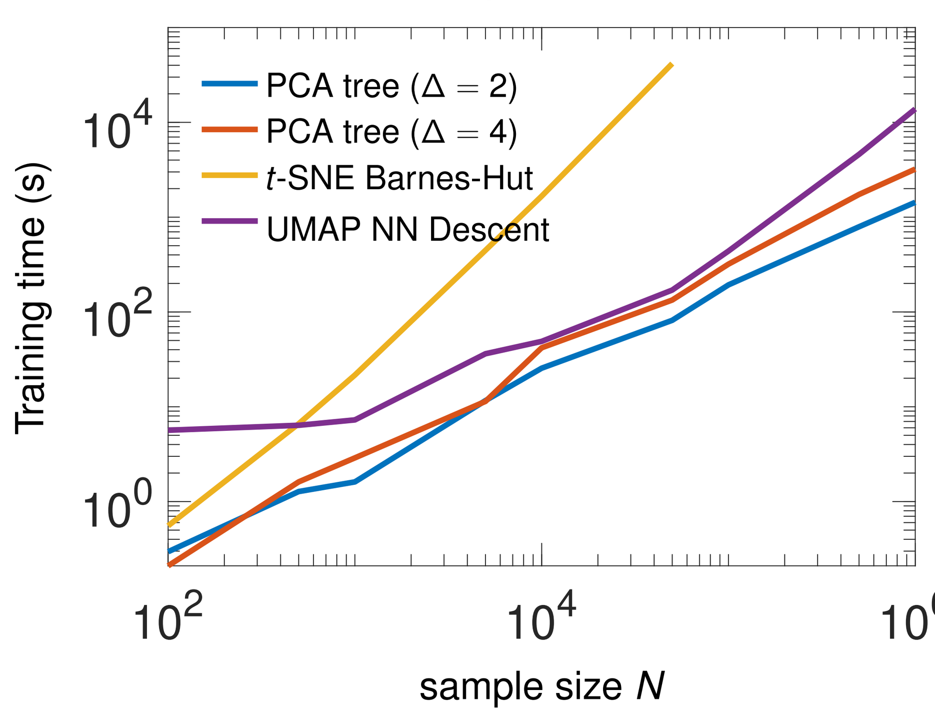
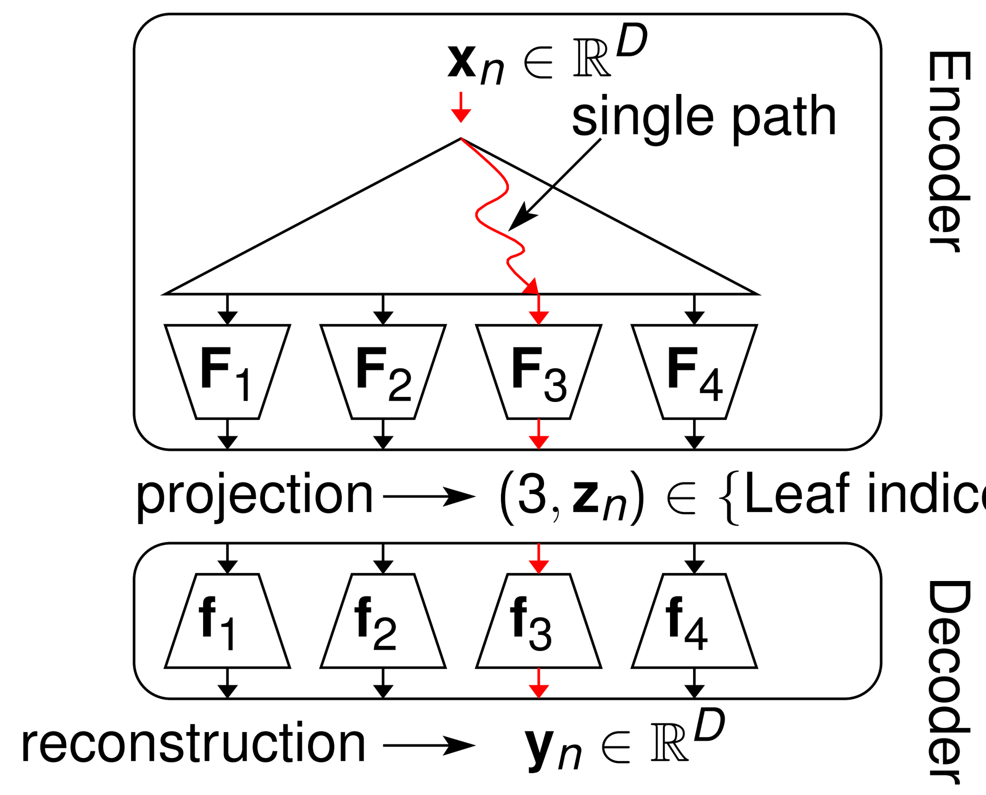


Figure: Training time per iteration on Infinite MNIST for PCA trees with different N, Δ, and for t-SNE and UMAP.

2 Tree autoencoder

The encoder is given by a tree mapping $\mathbf{T}^e(\mathbf{x}; \mathbf{\Theta})$: $\mathbb{R}^D \rightarrow \mathcal{L} \times \mathbb{R}^L$ where the predictor for leaf j has the form of a linear mapping $\mathbf{F}_j(\mathbf{x}; \mathbf{U}_j, \boldsymbol{\mu}_j) = \mathbf{U}_j^T(\mathbf{x} - \boldsymbol{\mu}_j)$, where $\mathbf{U}_j \in \mathbb{R}^{D \times L}$ is an orthogonal matrix and $\boldsymbol{\mu}_j \in \mathbb{R}^D$. The encoder parameters are $\mathbf{\Theta} = \{\mathbf{w}_i, w_{i0}\}_{i \in \mathcal{D}} \cup \{\mathbf{U}_j, \boldsymbol{\mu}_j\}_{j \in \mathcal{L}}$. Thus, the encoder maps an input instance $\mathbf{x} \in \mathbb{R}^D$ to a leaf index $j \in \mathcal{L}$ and an L-dimensional real vector $\mathbf{z} = \mathbf{U}_j^T(\mathbf{x} - \boldsymbol{\mu}_j)$, which at an optimum will be the PCA projection in that leaf. This means that the PCA tree does not have a common latent space of dimension L where all instances are projected. Instead, it has one separate L-dimensional PCA space per leaf.

The decoder maps a leaf index j and L-dimensional vector $\mathbf{z}$ (in $\mathcal{L} \times \mathbb{R}^L$) to a vector in $\mathbb{R}^D$. It consists of a set of linear mappings of the form $\mathbf{f}_j(\mathbf{z}; \mathbf{U}_j, \boldsymbol{\mu}_j) = \mathbf{U}_j\mathbf{z} + \boldsymbol{\mu}_j$ for $j \in \mathcal{L}$.



4 Advantages

The PCA tree provides significant, complementary advantages over previous methods:

1. It optimizes the reconstruction error, which has a clear meaning
2. It does not require a neighborhood graph (and perplexity parameter, etc.), which is tricky to estimate so it captures manifold structure, and computationally very costly
3. It is highly interpretable and extracts a wealth of information from complex datasets.
4. It defines nonlinear out-of-sample mappings
5. PCA map shows clusters, these are real—in contrast with t-SNE's tendency to create false clusters
6. The loss function is really a self-supervised regression problem. This makes it possible to use cross-validation to determine the hyperparameters

5 Fashion MNIST visualizations

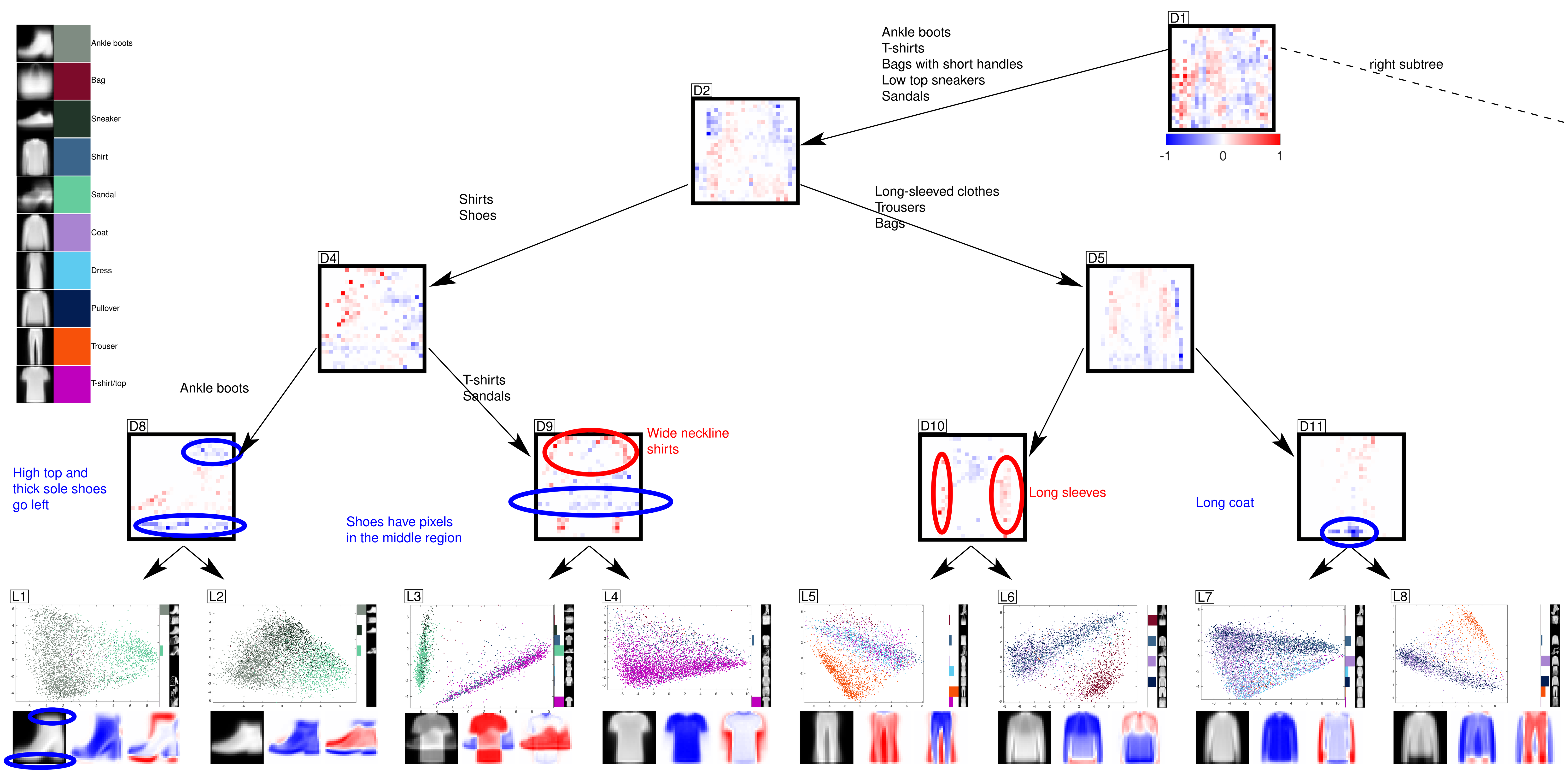


Figure: PCA tree trained on Fashion MNIST. The decision nodes' weight vectors (and the leaves' PCs $\mathbf{U}_j$) are shown as 28×28 images, with negative/zero/positive values colored blue/white/red. Each leaf shows a 2D PCA scatterplot of its RS, and below it the mean $\boldsymbol{\mu}_j$ (grayscale) image and the 2 PCs $\mathbf{U}_j$ (in color). To the right of the scatterplot, a bar chart displays class proportions and class means. The legend (top left) shows each class's grayscale mean and color (for the scatterplots).