Learning Interpretable, Tree-Based Projection Mappings for Nonlinear Embeddings

Arman Zharmagambetov and **Miguel Á. Carreira-Perpiñán**

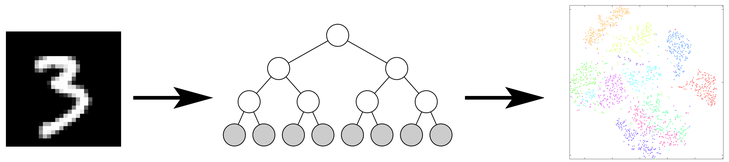
Dept. Computer Science & Engineering, University of California, Merced, USA

1 Abstract

- We consider the problem of learning interpretable out-of-sample mappings for nonlinear embedding methods such as t-SNE.
- Recall that such DR methods do not naturally define an out-of-sample mapping, rather they directly learn a low-dimensional projection for each training point.
- We argue for the use of sparse oblique decision trees because they strike a good tradeoff between accuracy and interpretability which can be controlled via a hyperparameter.
- The resulting optimization problem is difficult because decision trees are not differentiable.
- By using an equivalent formulation of the problem, we give an algorithm that can learn such a tree for any given nonlinear embedding objective.
- We illustrate experimentally how the resulting trees provide insights into the data beyond what a simple 2D visualization of the embedding does.

Work supported by NSF award IIS–1423515 and IIS–2007147

2 Proposed model and motivation



- Our proposed mapping is a sparse oblique decision tree $\mathbf{F}$, which maps a high-dimensional point $\mathbf{x}$ in D dimensions to a low-dimensional point $\mathbf{z}$ in $L \ll D$ dimensions: $\mathbf{F}$: $\mathbf{x} \in \mathbb{R}^D \rightarrow \mathbf{z} \in \mathbb{R}^L$. Each leaf uses a sparse linear mapping. Such a tree is learned by optimizing:

$$\min_{\Theta} \sum_{n=1}^{N} L(\mathbf{y}_n, \mathbf{F}(\mathbf{x}_n; \Theta)) + \lambda \phi(\mathbf{F}). \tag{1}$$

- It can model nonlinear mappings using very few nodes compared to an axis-aligned tree.
- It is especially convenient when clusters exist in the data, which can be captured by the tree hierarchy.
- It can make full use of any and all features of an instance.
- We can control the tree complexity (no. of nodes, features, etc.) via the regularization hyperparameter λ. This offers a convenient way to achieve a range of explanation levels, from detailed and accurate to simple and less accurate.

3 Jointly learning an optimal tree and embedding

A nonlinear embedding method defines an objective function $E(\mathbf{Z})$ over the low-dimensional coordinates $\mathbf{Z}_{L\times N} = (\mathbf{z}_1, \dots, \mathbf{z}_N)$ of the training points $\mathbf{X}_{D\times N} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$. For example, consider the elastic embedding loss (can also be t-SNE or any other loss):

$$E(\mathbf{Z}) = \sum_{n,m=1}^{N} \left(w_{nm} \|\mathbf{z}_n - \mathbf{z}_m\|^2 + \alpha e^{-\|\mathbf{z}_n - \mathbf{z}_m\|^2} \right)$$

Call the resulting $\mathbf{z}$ the free embedding. If we want an out-of-sample mapping $\mathbf{F}$ so we can project new points, then $\mathbf{z} = \mathbf{F}(\mathbf{x})$ by definition and we have a parametric embedding objective function:

$$E(\mathbf{F}) = \sum_{n,m=1}^{N} \left(w_{nm} \|\mathbf{F}(\mathbf{x}_n) - \mathbf{F}(\mathbf{x}_m)\|^2 + \alpha e^{-\|\mathbf{F}(\mathbf{x}_n) - \mathbf{F}(\mathbf{x}_m)\|^2} \right) + \lambda \phi(\mathbf{F}) \tag{2}$$

where $\phi(\mathbf{F})$ is a regularization term on the mapping. Eq. (2) is not easy to optimize since $\mathbf{F}$ is non-differentiable and non-convex mapping. Solution: apply the method of auxiliary coordinates (MAC) [1]. Consider the following equivalent constrained problem with "auxiliary coordinates" $\mathbf{Z}$:

$$\min_{\mathbf{Z},\mathbf{F}} E(\mathbf{Z}) + \lambda \phi(\mathbf{F}) \quad \text{s.t.} \quad \mathbf{Z} = \mathbf{F}(\mathbf{X}) \tag{3}$$

We solve (3) using a penalty method. We describe the quadratic-penalty method for simplicity, but in the experiments we use the augmented Lagrangian. This defines a new, unconstrained objective function:

$$\min_{\mathbf{Z},\mathbf{F}} E(\mathbf{Z}) + \lambda \phi(\mathbf{F}) + \mu \|\mathbf{Z} - \mathbf{F}(\mathbf{X})\|^2. \tag{4}$$

Finally, we optimize (4) by alternating optimization over $\mathbf{Z}$ and $\mathbf{F}$:

- Over $\mathbf{Z}$, eq. (4) is the original embedding objective E but with a quadratic regularization term on $\mathbf{Z}$:

  $$\min_{\mathbf{Z}} E(\mathbf{Z}) + \mu \|\mathbf{Z} - \mathbf{F}(\mathbf{X})\|^2.$$

  This can be easily solved by reusing an algorithm to optimize the original embedding (t-SNE, the elastic embedding or whatever), with a minor modification to handle the additional quadratic term.
- Over $\mathbf{F}$, eq. (4) reduces to a regression fit of a tree (see eq. (1)) which we solve using the Tree Alternating Optimization (TAO) [2]:

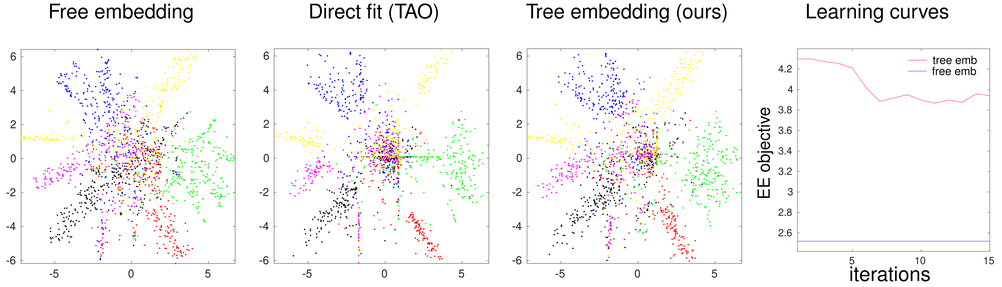
  $$\min_{\mathbf{F}} \|\mathbf{Z} - \mathbf{F}(\mathbf{X})\|^2 + \frac{\lambda}{\mu} \phi(\mathbf{F}).$$

  The ability of the TAO algorithm to take an initial tree and improve over it is essential here to make sure that the step over $\mathbf{F}$ improves over the previous iteration, and to be able to use warm-start to speed up the computation.

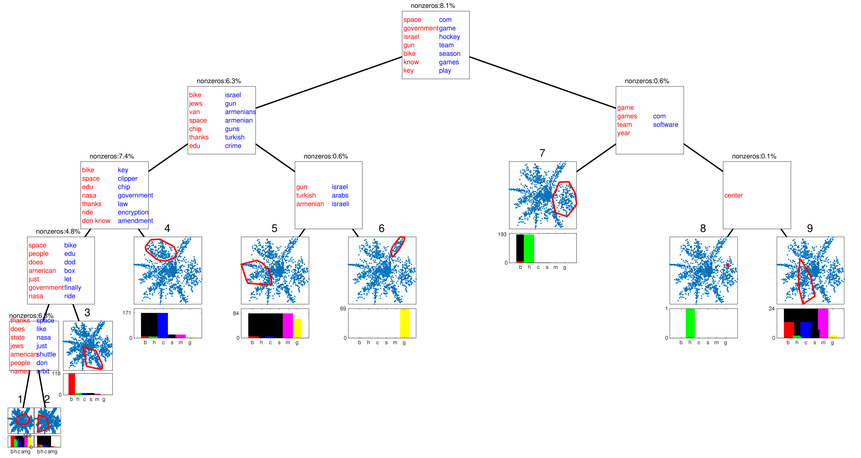
References

1. Carreira-Perpiñán, M. Á. and Wang, W. Distributed optimization of deeply nested systems. AISTATS 2014.
2. Zharmagambetov, A. and Carreira-Perpiñán, M. Á. Smaller, More Accurate Regression Forests Using Tree Alternating Optimization. ICML 2020.

4 Experiments



- Results on 20-newsgroups dataset: 6 classes, tf-idf statistics on unigrams and bigrams as features (1000 features in total).
- We used elastic embedding to produce the free embedding.
- Direct fit trains an oblique tree (using TAO) directly to a free embedding, i.e. it uses free embedding as a label.
- The first iteration ($\mu = 0$) in learning curves (left plot) represents a direct fit. Our proposed approach (tree embedding) improves over this baseline (see iterations).



- Visualization of the tree embedding. For each decision node, we show up to top-7 features (words) which corresponds to the largest non-zero values in the weight vector. Words with the highest positive/negative values are responsible for sending an instance to a certain child.
- For each leaf, we show the region of its responsibility by convex hull of the mappings falling into that leaf and provide histogram counts of classes.
- There is a clear clustering structure in the hierarchy as most of leaves focus on few classes. The hierarchy respects class ontology by merging instances of semantically similar classes under one subtree, e.g. "g" gun and "m" mideast (in leaves #5 and #6), whereas their locations in 2D are not next to each others.
- Leaf #8 has only one point which is clearly an outlier (several topics are discussed in one document).