Optimal Interpretable Clustering Using Oblique Decision Trees

Magzhan Gabidolla   Miguel Á. Carreira-Perpiñán

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

Abstract

Recent years have seen a renewed interest in interpretable machine learning, which seeks insight into how a model achieves a prediction. Here, we focus on the relatively unexplored case of interpretable clustering. In our approach, the cluster assignments of the training instances are constrained to be the output of a decision tree. This has two advantages: 1) it makes it possible to understand globally how an instance is mapped to a cluster, in particular to see which features are used for which cluster; 2) it forces the clusters to respect a hierarchical structure while optimizing the original clustering objective function. Rather than the traditional axis-aligned trees, we use sparse oblique trees, which have far more modelling power, particularly with high-dimensional data, while remaining interpretable. Our approach applies to any clustering method which is defined by optimizing a cost function and we demonstrate it with two k-means variants.

Work supported by NSF award IIS–2007147.

Defining "Interpretable" Clustering

- We aim at explaining how an input instance $\mathbf{x} \in \mathbb{R}^D$ (not necessarily in the training set) is mapped to a particular cluster. We call this the out-of-sample mapping.
- The optimal out-of-sample mapping for k-means is given by assigning the instance $\mathbf{x}$ to its closest centroid. But this mapping is not very helpful in explaining how the input features in $\mathbf{x}$ determine the cluster.
- For other clustering methods (e.g. spectral clustering) a natural out-of-sample mapping is much harder to determine.
- Therefore, we want to determine an out-of-sample mapping that is interpretable, and in a way that is agnostic to how the clustering cost is defined, so it is generally applicable.

Joint optimization algorithm

- We consider clustering algorithms defined by a cost function E, and demand the cluster assignments come from a classification tree $\mathbf{T}(\mathbf{x}, \boldsymbol{\Theta})$. To jointly learn both clustering $\boldsymbol{\Psi}$ and tree $\boldsymbol{\Theta}$ parameters:

$$\min_{\boldsymbol{\Psi}, \boldsymbol{\Theta}} E(\mathbf{T}(\mathbf{X}; \boldsymbol{\Theta}), \boldsymbol{\Psi}) + \lambda\, \phi(\boldsymbol{\Theta}). \tag{1}$$

- We rewrite this as a constrained problem using assignment variables $\mathbf{Z}$:

$$\min_{\mathbf{Z}, \boldsymbol{\Psi}, \boldsymbol{\Theta}} E(\mathbf{Z}, \boldsymbol{\Psi}) + \lambda\, \phi(\boldsymbol{\Theta}) \quad \text{s.t.} \quad \mathbf{Z} = \mathbf{T}(\mathbf{X}; \boldsymbol{\Theta}), \quad \mathbf{Z}^T\mathbf{1} = \mathbf{1}, \quad \mathbf{Z} \in \{0,1\}^{K \times N}. \tag{2}$$

We apply a penalty method to the equality constraints that involve $\mathbf{T}$ and define the problem:

$$\min_{\mathbf{Z}, \boldsymbol{\Psi}, \boldsymbol{\Theta}} E(\mathbf{Z}, \boldsymbol{\Psi}) + \lambda\, \phi(\boldsymbol{\Theta}) + \mu\, P(\mathbf{Z}, \mathbf{T}(\mathbf{X}; \boldsymbol{\Theta})) \quad \text{s.t.} \quad \mathbf{Z}^T\mathbf{1} = \mathbf{1}, \quad \mathbf{Z} \in \{0,1\}^{K \times N}. \tag{3}$$

where $\mu \geq 0$ is a penalty parameter and P is a penalty function satisfying $P(\mathbf{z}, \mathbf{z}) = 0$ and $P(\mathbf{z}, \mathbf{z}') > 0$ if $\mathbf{z} \neq \mathbf{z}'$. If $\mu \to \infty$ then both have the same solutions. We follow a path of solutions starting from small μ, and for each μ, we perform alternating optimization:

- Clustering step (over $\mathbf{Z}, \boldsymbol{\Psi}$ given $\boldsymbol{\Theta}$):

$$\min_{\mathbf{Z}, \boldsymbol{\Psi}} E(\mathbf{Z}, \boldsymbol{\Psi}) + \mu \sum_{n=1}^{N} P(\mathbf{z}_n, \overline{\mathbf{z}}_n) \quad \text{s.t.} \quad \mathbf{Z}^T\mathbf{1} = \mathbf{1}, \quad \mathbf{Z} \in \{0,1\}^{K \times N} \tag{4}$$

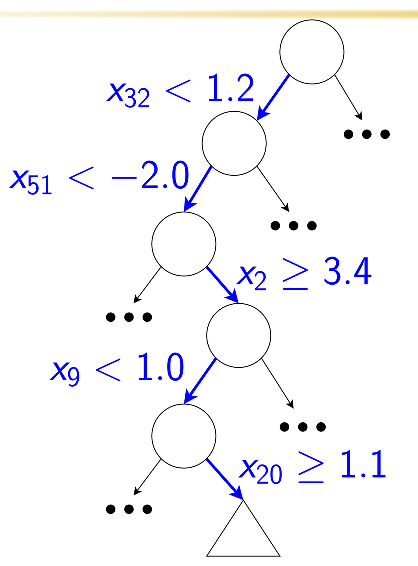
where $\overline{\mathbf{z}}_n = \mathbf{T}(\mathbf{x}_n; \boldsymbol{\Theta})$ is a constant vector for $n = 1, \dots, N$. This is very similar to the unconstrained clustering problem, but with a regularization term that pulls the assignments $\mathbf{Z}$ towards $\overline{\mathbf{Z}}$.

- Tree step (over $\boldsymbol{\Theta}$ given $\mathbf{Z}, \boldsymbol{\Psi}$):

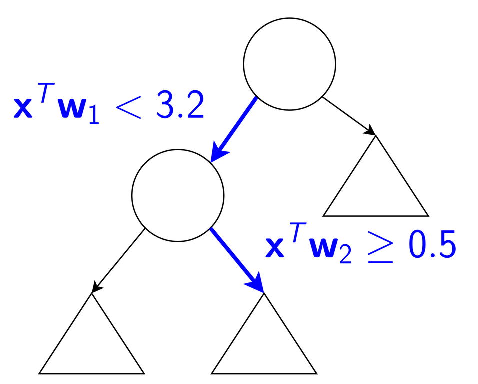
$$\min_{\boldsymbol{\Theta}} \sum_{n=1}^{N} P(\mathbf{z}_n, \mathbf{T}(\mathbf{x}_n; \boldsymbol{\Theta})) + \frac{\lambda}{\mu}\phi(\boldsymbol{\Theta}). \tag{5}$$

This takes the form of a classification problem with loss P, tree classifier $\mathbf{T}$ and regularization ϕ, which we can solve using the Tree Alternating Optimization (TAO) algorithm.

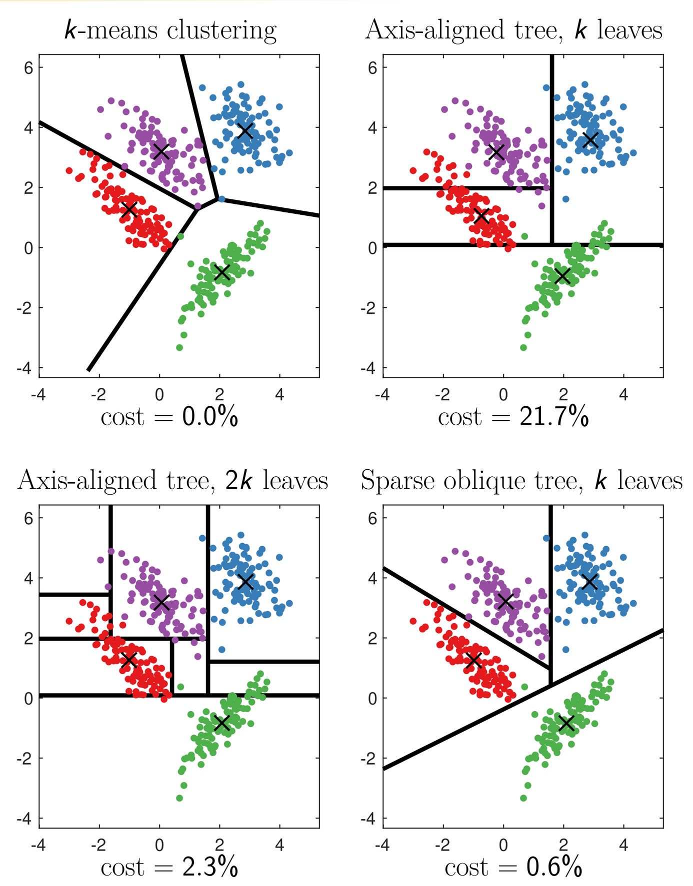
Axis-Aligned vs Oblique trees



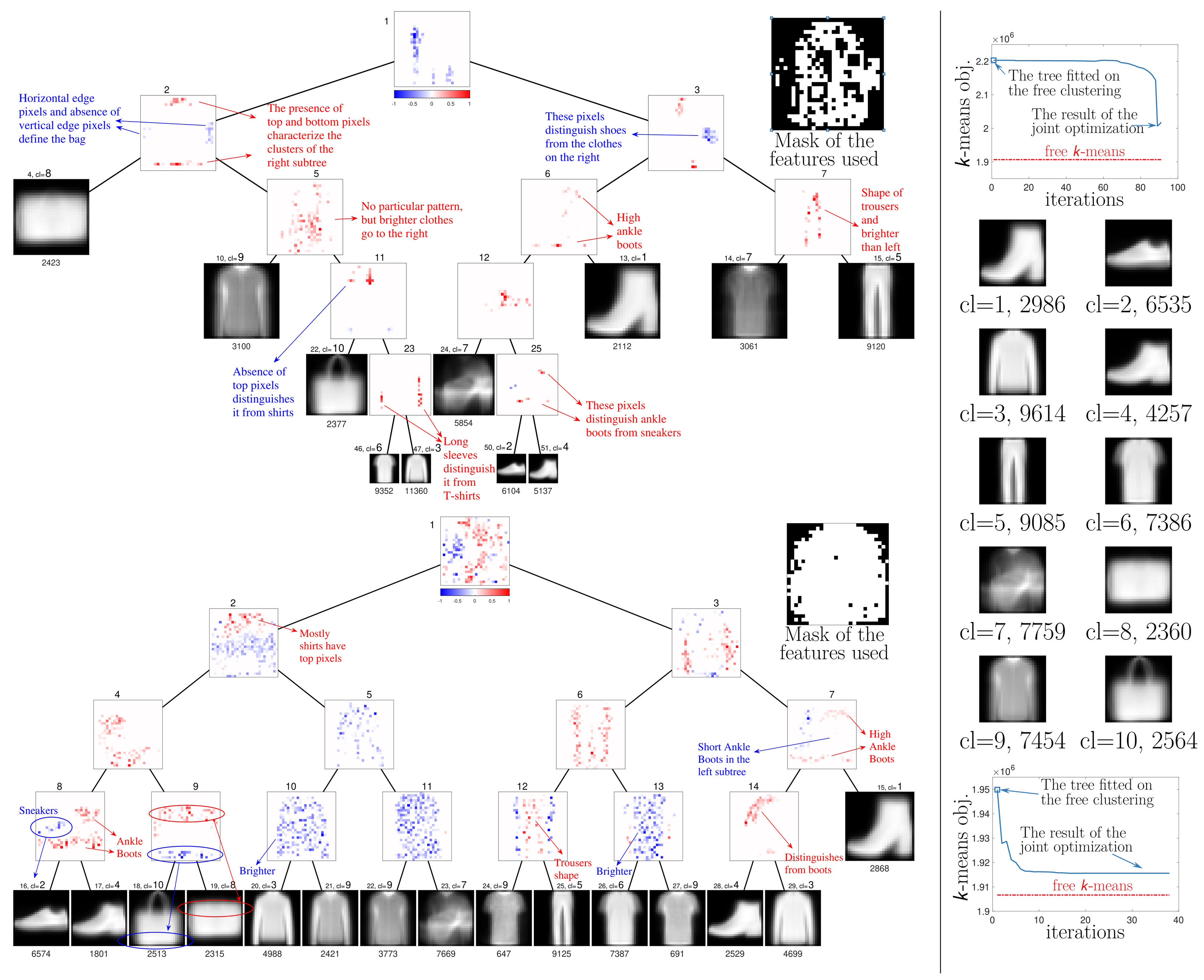
- Only 5 features participate in the routing function of the above leaf.
- Max order of feature interactions is limited by the depth Δ in axis-aligned trees.
- Can model only axis-aligned boundaries.



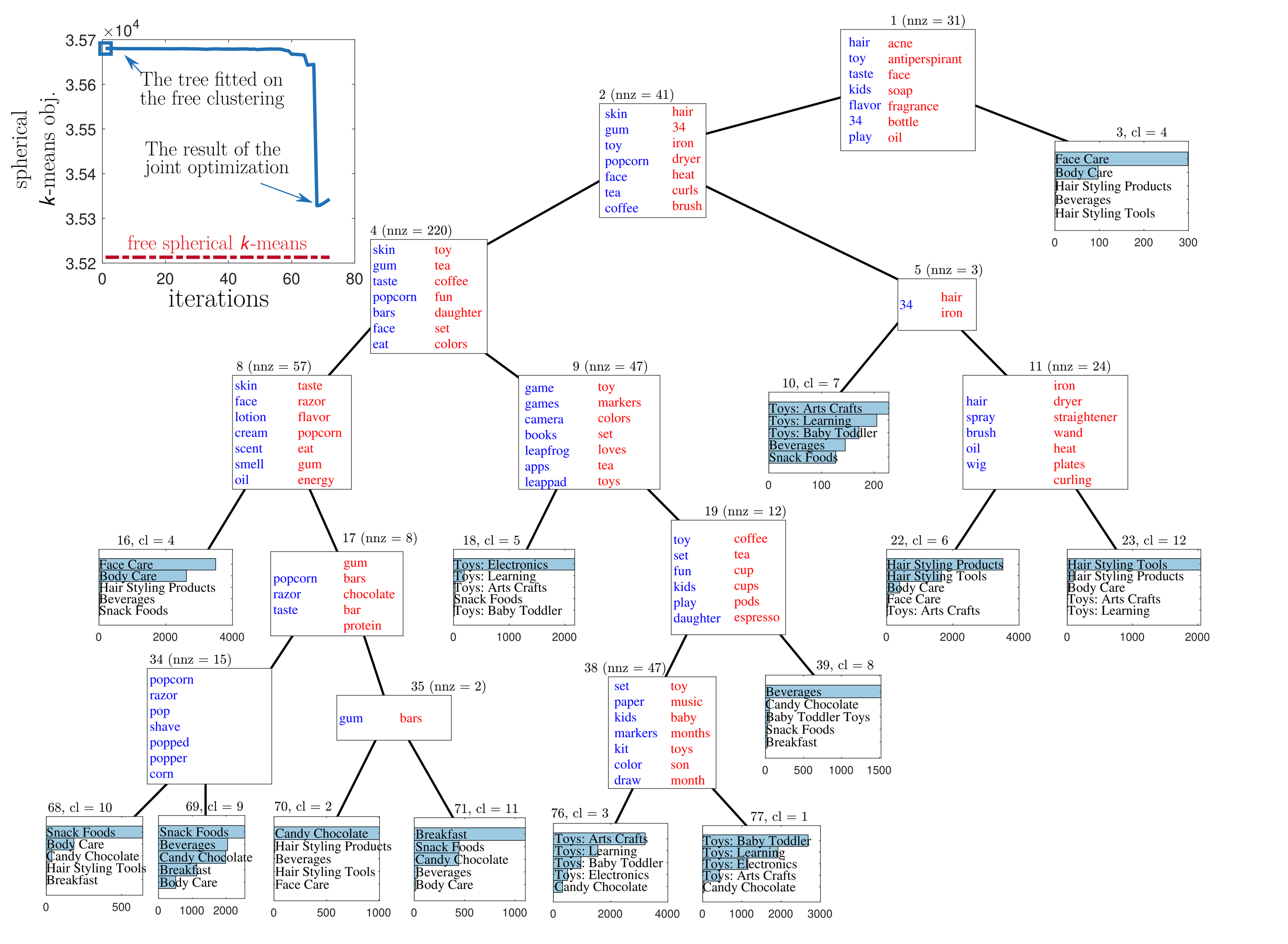
- Each decision node is a function of all the features.
- Their non-linear combination is a much more complex order-D interaction.
- As out-of-sample mapping, sparse oblique trees should have better modeling capacity while remaining small and interpretable.



Experiment Results



- Trees for k-means on FashionMNIST with different sparsity.
- Top tree: $\lambda = 100.0$, depth $= 5$, cost $= 5.22\%$.
- Bottom tree: $\lambda = 10.0$, depth $= 4$, cost $= 0.44\%$.
- Decision nodes: weight vector visualized as a 28×28 image.
- Red/blue pixels contribute sending points to the right/left.
- Leaves: visualizes the mean of images that reach it.
- The plots on the right: k-means objective during the algorithm run.
- Right side: free clustering (the mean image and number of points).



- A tree for spherical k-means on the subset of Amazon Reviews.
- Top left plot: the clustering objective during the algorithm run.
- Decision nodes: the words corresponding to the most positive/negative top 7 weights in red/blue (sorted by weight magnitude).
- nnz: number of nonzero weights (shown at the top of a node).
- Leaf: the histogram of product categories (shown horizontally).