Optimal Interpretable Clustering Using Oblique Decision Trees

Magzhan Gabidolla   Miguel Á. Carreira-Perpiñán

Dept. Computer Science & Engineering
University of California, Merced

28th SIGKDD Conference on
Knowledge Discovery and Data Mining

Introduction

- Due to
  - the widespread deployment of ML in practical applications
  - the black-box nature of the more accurate models (such as neural networks and random or boosted forests)
  - upcoming regulations in many jurisdictions that require model or algorithmic decisions to be explainable, so they can be trusted or audited (for bias, fairness, mistakes, etc.)

  the topic of **model interpretability/explainability** has achieved enormous prominence in recent years.
- While the vast majority of work in this area has focused on classification, interpreting clustering methods/results has received far less attention.

What is an “interpretable” clustering?

- We aim at explaining how an input instance $\mathbf{x} \in \mathbb{R}^D$ (not necessarily in the training set) is mapped or assigned 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. However, this mapping is not very helpful in explaining how the input features in $\mathbf{x}$ determine the cluster. Also, precisely characterizing the cluster regions (Voronoi cells in D dimensions!) is complicated.
- For other clustering methods (e.g. spectral clustering) a natural out-of-sample mapping is much harder to determine.
- For these reasons, 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.

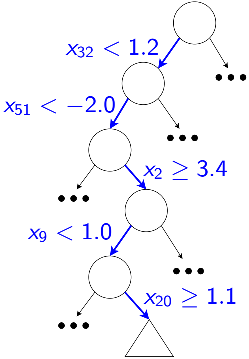
Decision trees as out-of-sample mapping

Decision trees have several attractive properties in this context:

- They naturally **handle multiple classes**.
- By using multiple leaves per class, they **can model nonconvex** and even disconnected classes.
- They **make the clustering hierarchical**, i.e., they define a nested set of clusters.
- As long as the number of nodes is not very large, they are **globally interpretable** by simple inspection of the nodes and the features they involve, without the need of any approximation or external explanation method.
- Each leaf **can be described by a rule** (given by the root-leaf path).

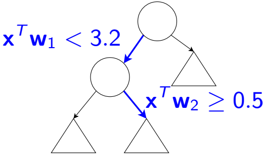
Modeling capacity:

Axis-aligned 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.

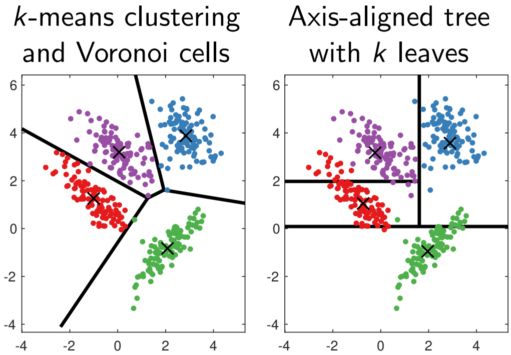
Oblique trees



$\mathbf{x}^T\mathbf{w}_1 < 3.2$
$\mathbf{x}^T\mathbf{w}_2 \geq 0.5$

- 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**.

Illustration on *k*-means for a toy example in 2D



k-means clustering and Voronoi cells

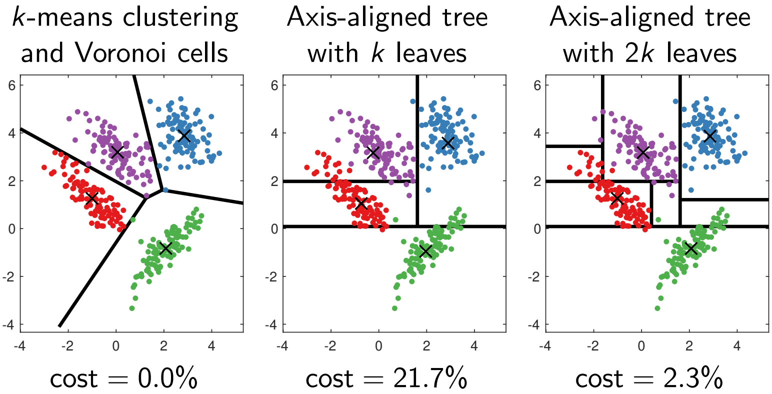
Axis-aligned tree with *k* leaves

cost = 0.0%

cost = 21.7%

Figure: Toy example in 2D with $K = 4$ clusters. Cost is defined as the percentage increase from the reference *k*-means clustering.

Illustration on *k*-means for a toy example in 2D



k-means clustering and Voronoi cells

Axis-aligned tree with *k* leaves

Axis-aligned tree with 2*k* leaves

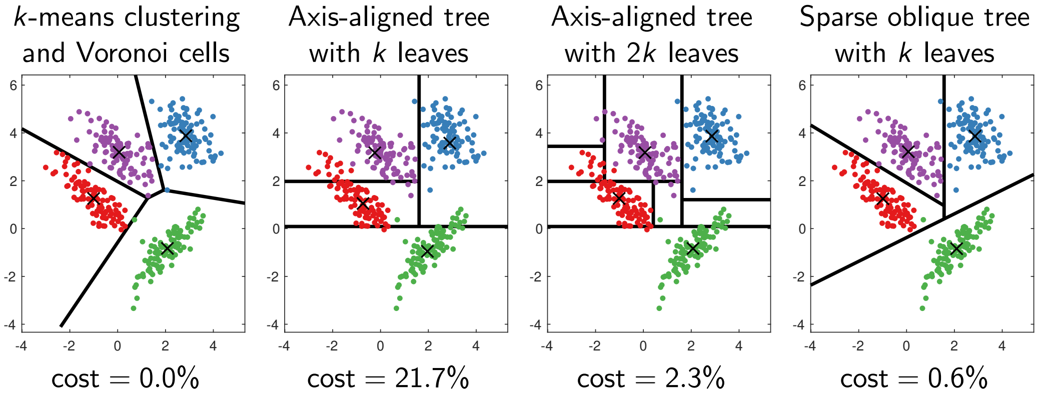
cost = 0.0%

cost = 21.7%

cost = 2.3%

Figure: Toy example in 2D with $K = 4$ clusters. Cost is defined as the percentage increase from the reference *k*-means clustering.

Illustration on k-means for a toy example in 2D



k-means clustering and Voronoi cells

Axis-aligned tree with k leaves

Axis-aligned tree with $2k$ leaves

Sparse oblique tree with k leaves

cost $= 0.0\%$

cost $= 21.7\%$

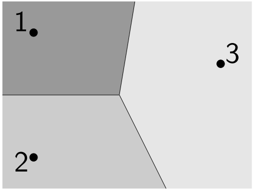
cost $= 2.3\%$

cost $= 0.6\%$

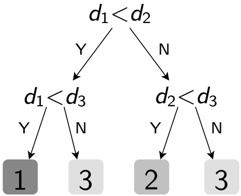
Figure: Toy example in 2D with $K = 4$ clusters. Cost is defined as the percentage increase from the reference k-means clustering.

k-means case: an exact oblique tree

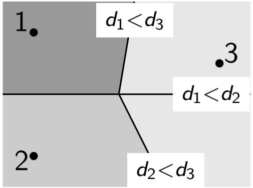
For the special case of *k*-means, an exact representation of the clustering out-of-sample mapping **can be done with a (deep) oblique tree, but not with an axis-aligned tree**.



Voronoi tessellation induced by $K = 3$ centroids



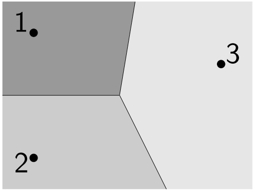
Oblique tree with 4 $(= 2^{K-1})$ leaves



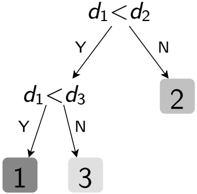
Its partition is equivalent to the Voronoi tessellation.

Trade off accuracy with the interpretability of the out-of-sample mapping

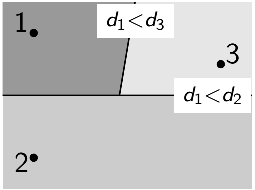
But our goal here is to trade off optimally the clustering accuracy (according to a specific clustering criterion) with the interpretability of the clustering out-of-sample mapping, having the form of a small oblique tree with sparse hyperplane splits.



Voronoi tessellation induced by $K = 3$ centroids



Oblique tree with 3 $(= K)$ leaves



Its partition induces distortion to the Voronoi tessellation.

Clustering problem formulation

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

- Input is a training set $\mathbf{X}_{D\times N} = (\mathbf{x}_1,\dots,\mathbf{x}_N)$ and seeking K clusters.
- The assignment variables $\mathbf{Z}_{K\times N} = (\mathbf{z}_1,\dots,\mathbf{z}_N)$ indicate which cluster each instance $\mathbf{x}_n$ is assigned to, encoded as one-hot vectors.
- The variables $\boldsymbol{\Psi}$ include any other variables learnt by the algorithm, for example the cluster centroids in k-means.
- $E(\mathbf{Z},\boldsymbol{\Psi})$ is a cost function defining the goodness of a clustering. For example:

$$E(\mathbf{Z},\boldsymbol{\Psi}) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{kn}\, d(\mathbf{x}_n,\psi_k), \tag{2}$$

  $\psi_1,\dots,\psi_K \in \mathbb{R}^D$ and d is a distance function, corresponds to centroid-based methods, such as k-means, spherical k-means, k-medoids, k-medians etc.

Interpretable clustering problem formulation

We now solve problem (1) but demand that the cluster assignments $\mathbf{z}_n$ be produced by an out-of-sample mapping $\mathbf{T}(\mathbf{x}_n; \boldsymbol{\Theta})$, a classification tree with parameters $\boldsymbol{\Theta}$. That is:

$$\min_{\boldsymbol{\Psi},\boldsymbol{\Theta}} E(\mathbf{T}(\mathbf{X}; \boldsymbol{\Theta}), \boldsymbol{\Psi}) + \lambda\, \phi(\boldsymbol{\Theta}) \tag{3}$$

- $\mathbf{T}(\cdot; \boldsymbol{\Theta})$: $\mathbb{R}^D \rightarrow \{1, \dots, K\}$ (one-hot encoded)
- We consider either an axis-aligned tree or an oblique tree.
- As for the leaf predictors, we consider either a class label or a histogram over classes (where the most frequent class is the final prediction).
- The regularization term $\phi(\boldsymbol{\Theta})$ with user parameter $\lambda \geq 0$ controls the tree complexity.

Interpretable clustering problem formulation

- This is a difficult optimization problem because the tree $\mathbf{T}$ is not a differentiable function of $\mathbf{\Theta}$, and appears as an argument of the nonlinear function E.
- We rewrite (3) as a constrained problem by introducing the assignment variables:

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{\Psi},\mathbf{\Theta}} E(\mathbf{Z},\mathbf{\Psi}) + \lambda\,\phi(\mathbf{\Theta}) \\ &\text{s.t.} \quad \mathbf{Z} = \mathbf{T}(\mathbf{X};\mathbf{\Theta}), \quad \mathbf{Z}^T\mathbf{1} = \mathbf{1}, \quad \mathbf{Z} \in \{0,1\}^{K\times N}. \end{aligned} \tag{4}$$

Interpretable clustering: optimization algorithm

- We apply a penalty method to the equality constraints in (4) that involve $\mathbf{T}$ (leaving the other constraints in place) and define the problem:

$$\begin{aligned} \min_{\mathbf{Z},\boldsymbol{\Psi},\boldsymbol{\Theta}} \; & E(\mathbf{Z},\boldsymbol{\Psi}) + \lambda\,\phi(\boldsymbol{\Theta}) + \mu\,P(\mathbf{Z},\mathbf{T}(\mathbf{X};\boldsymbol{\Theta})) \\ \text{s.t.} \quad & \mathbf{Z}^T\mathbf{1} = \mathbf{1}, \quad \mathbf{Z} \in \{0,1\}^{K\times N} \end{aligned} \tag{5}$$

- 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 $\mathbf{z} \neq \mathbf{z}'$. The notation $P(\mathbf{Z},\mathbf{T}(\mathbf{X};\boldsymbol{\Theta}))$ stands for $P(\mathbf{z}_1,\mathbf{T}(\mathbf{x}_1;\boldsymbol{\Theta})) + \dots + P(\mathbf{z}_N,\mathbf{T}(\mathbf{x}_N;\boldsymbol{\Theta}))$.
- If $\mu \to \infty$ then (4) and (5) have the same solutions.
- The objective of (5) becomes progressively ill-conditioned (hence harder to optimize numerically) as μ increases.
- Thus, rather than optimizing (5) directly for a very large value of μ, we follow a path of solutions starting from small μ, as is common with quadratic-penalty and other homotopy methods.

Interpretable clustering: Alternating Optimization

For a fixed value of μ, we optimize (5) using alternating optimization over the clustering variables $(\mathbf{Z}, \boldsymbol{\Psi})$ and the tree parameters $\boldsymbol{\Theta}$:

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

$$\min_{\mathbf{Z},\boldsymbol{\Psi}} E(\mathbf{Z}, \boldsymbol{\Psi}) + \mu \sum_{n=1}^{N} P(\mathbf{z}_n, \overline{\mathbf{z}}_n) \quad \text{s.t.} \quad \mathbf{Z}^T\mathbf{1} = \mathbf{1}, \quad \mathbf{Z} \in \{0,1\}^{K\times N} \tag{6}$$

  where $\overline{\mathbf{z}}_n = \mathbf{T}(\mathbf{x}_n; \boldsymbol{\Theta})$ is a constant vector for $n = 1, \ldots, N$. This can be seen as the original clustering problem (1) but with a regularization term that pulls the assignments $\mathbf{Z}$ towards $\overline{\mathbf{Z}}$. This step can be usually solved using a modified version of the original unconstrained clustering.

Interpretable clustering: Alternating Optimization

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

$$\min_{\boldsymbol{\Theta}} \sum_{n=1}^{N} P(\mathbf{z}_n, \mathbf{T}(\mathbf{x}_n; \boldsymbol{\Theta})) + \frac{\lambda}{\mu}\phi(\boldsymbol{\Theta}). \tag{7}$$

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.

Why Learning a Tree with Tree Alternating Optimization (TAO)?

- We use a recent Tree Alternating Optimization (TAO) to solve the tree step subproblem, because:
  - It can directly optimize the objective function (eq. (7)).
  - It can learn the structure of the tree and the parameters at the nodes.
  - It can take an initial tree and improve over it, so the tree step decreases the overall objective function in (5) (i.e., warm-start).
  - It is computationally efficient.
- The traditional, recursive partitioning algorithms, such as CART or C4.5, are inadequate because:
  - They grow a tree greedily from scratch rather than improving a given tree.
  - They are also quite suboptimal, particularly with oblique trees.
- "Optimal tree" algorithms (e.g. based on mixed-integer optimization and branch-and-bound) do not scale beyond toy datasets and tiny trees.

Overview of Tree Alternating Optimization (TAO)

- The underlying mechanism of TAO is to take a parametric tree of fixed structure (here, complete of depth Δ), and perform optimization steps in turn over the parameters of a single node (decision node or leaf) while keeping the rest of the parameters fixed.
- It works quite similar to how one would optimize a neural network, but instead of gradients (which do not apply) TAO uses alternating optimization on a fixed tree structure.

TAO is based on two theorems:

- Eq. (7) **separates over any subset of non-descendant nodes** (e.g. all the nodes at the same depth); this follows from the fact that the tree makes hard decisions.
- Optimizing over the parameters of a single node i simplifies to a well-defined **reduced problem** over the instances that currently reach node i (the **reduced set** $\mathcal{R}_i \subset \{1, \dots, N\}$).

Overview of Tree Alternating Optimization (TAO)

The form of the reduced problem depends on the type of node:

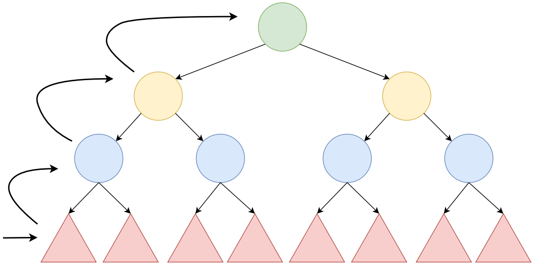
Decision node It is a **weighted 0/1 loss binary classification problem**, where the two classes correspond to the left and right child, and we want the decision node to learn to send points to the best child. For oblique nodes this problem is NP-hard but can be well approximated with a convex surrogate; we use ℓ_1-regularized logistic regression. For axis aligned splits the optimal solution is found by enumeration.

Leaf The reduced problem consists of optimizing the original loss but over the leaf classifier on its reduced set:

$$\min_{\boldsymbol{\theta}_i} \sum_{n \in \mathcal{R}_i} P(\mathbf{z}_n, \boldsymbol{\theta}_i). \tag{8}$$

For the two penalty functions that we consider (0-1 loss and squared error) the solution is either a majority label or a normalized histogram.

Overview of Tree Alternating Optimization (TAO)



- Given an initial tree structure with initial parameter values, the resulting algorithm repeatedly visits nodes in reverse breadth-first search order.
- Each iteration trains all nodes at the same depth (in parallel) from the leaves to the root, by solving either an ℓ_1-regularized logistic regression for oblique splits or by enumeration in axis-aligned case, or the exact solution at each leaf.

Pseudocode of the joint optimization framework for interpretable clustering

```
input $\mathbf{X}_{D\times N} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$, $\lambda \geq 0$, $a > 0$, $\mu_0 > 0$
      initial tree structure and random $\mathbf{\Theta}$
$\mathbf{Z}, \mathbf{\Psi} \leftarrow \arg\min E(\mathbf{Z}, \mathbf{\Psi})$  s.t.  $\mathbf{Z}^T\mathbf{1}=\mathbf{1}$, $\mathbf{Z}\in\{0,1\}^{K\times N}$          Free clustering
$\mathbf{\Theta} \leftarrow \begin{cases} \arg\min P(\mathbf{Z}, \mathbf{T}(\mathbf{X};\mathbf{\Theta})), & \lambda = 0 \\ 0, & \lambda > 0 \end{cases}$          Direct tree fit
$\mu \leftarrow \mu_0$
repeat
    $\mathbf{Z}, \mathbf{\Psi} \leftarrow \arg\min E(\mathbf{Z}, \mathbf{\Psi}) + \mu\, P(\mathbf{Z}, \mathbf{T}(\mathbf{X};\mathbf{\Theta}))$          Clustering step
            s.t.  $\mathbf{Z}^T\mathbf{1} = \mathbf{1}$, $\mathbf{Z} \in \{0,1\}^{K\times N}$
    $\mathbf{\Theta} \leftarrow P(\mathbf{Z}, \mathbf{T}(\mathbf{X};\mathbf{\Theta})) + \frac{\lambda}{\mu}\,\phi(\mathbf{\Theta})$          Tree step
    $\mu \leftarrow \mu \cdot a$
until $\mathbf{Z} = \mathbf{T}(\mathbf{X};\mathbf{\Theta})$ and no parameter change
return tree $\mathbf{T}(\cdot;\mathbf{\Theta})$ and $\mathbf{Z}, \mathbf{\Psi}$
```

Experiments: the benefit of joint optimization

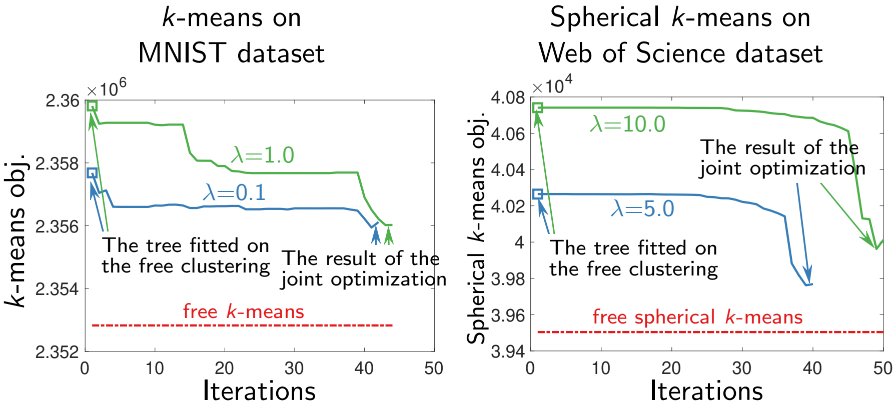
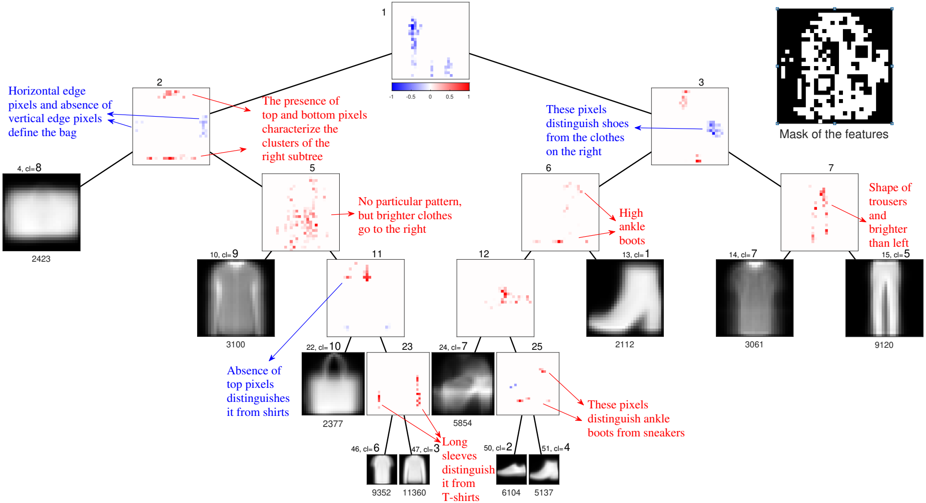


Figure: The plots of the k-means and spherical k-means objectives for MNIST and Web of Science datasets as our joint optimization algorithm progresses over the μ schedule. Each point shows the clustering objective when the cluster assignments are given by a sparse oblique tree. The initial points are the direct fit of TAO to the free clustering assignments.

Experiments: quantitative comparison

	Method	cost (%)	#parameters	#features/node	Δ	#leaves
MNIST (60k,784,10)	IMM	14.34	28	1	9	10
	Ex-Greedy	12.48	28	1	8	10
	CART	11.54	28	1	4	10
	TAO oblique	7.90	199	23	4	9
	CART	1.87	3070	1	16	1024
	ExKMC	1.81	3070	1	29	1024
	TAO oblique	1.50	753	66	5	12
	TAO oblique	0.94	1372	96	4	15
Letter (20k,16,26)	IMM	35.02	76	1	25	26
	CART	30.61	76	1	10	26
	ExGreedy	27.78	76	1	21	26
	TAO oblique	9.94	523	15	5	32
	TAO oblique	4.06	516	8	6	52
	CART	2.89	3070	1	25	1024
	ExKMC	2.91	3070	1	39	1024
	TAO oblique	2.75	858	12	6	64

Experiments: sparser tree on Fashion MNIST



Horizontal edge pixels and absence of vertical edge pixels define the bag

The presence of top and bottom pixels characterize the clusters of the right subtree

These pixels distinguish shoes from the clothes on the right

No particular pattern, but brighter clothes go to the right

High ankle boots

Shape of trousers and brighter than left

Absence of top pixels distinguishes it from shirts

Long sleeves distinguish it from T-shirts

These pixels distinguish ankle boots from sneakers

Figure: The tree results from $\lambda = 100.0$, $\Delta = 5$ and has a distortion of 5.22% from the free k-means clustering objective.

Experiments: denser tree on Fashion MNIST

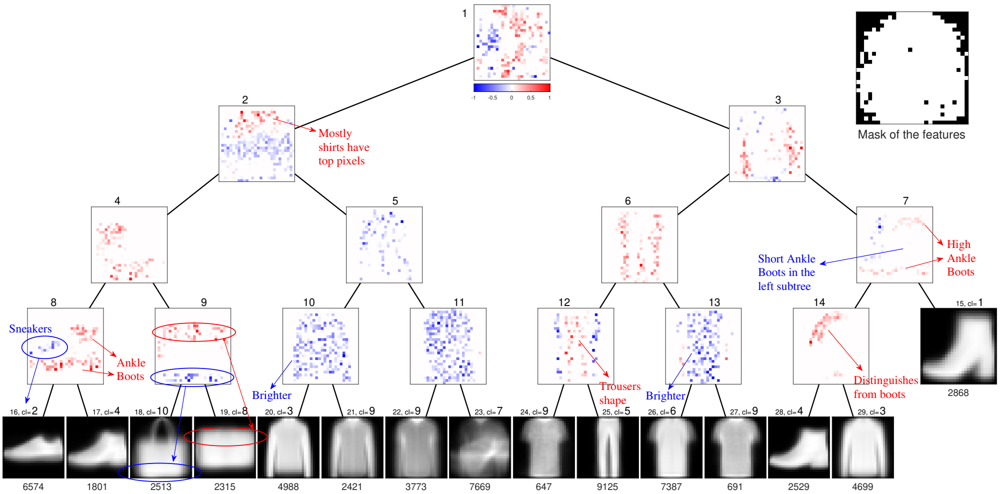


Figure: The tree results from $\lambda = 10.0$, $\Delta = 4$ and has a distortion of 0.44% from the free k-means clustering objective.

Experiments: clustering tree for documents

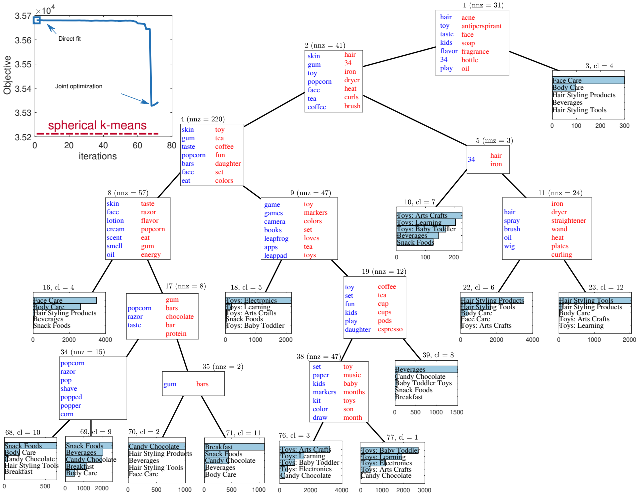


Figure: The tree results for spherical *k*-means clustering objective.

Conclusion

- We have proposed a way to redefine any clustering method defined by a cost function of the cluster assignments, by constraining the latter to be produced by an interpretable out-of-sample mapping.
- The mapping is given by a sparse oblique decision tree, which is far more powerful than the usual axis-aligned trees, particularly with high-dimensional data.
- The tree makes it possible to explain how a prediction was arrived at by simple inspection.
- In our experiments we have demonstrated this with k-means-type methods, but the approach applies to other clustering methods defined by a cost function.
- **Acknowledgments.** Work supported by NSF award IIS–2007147.