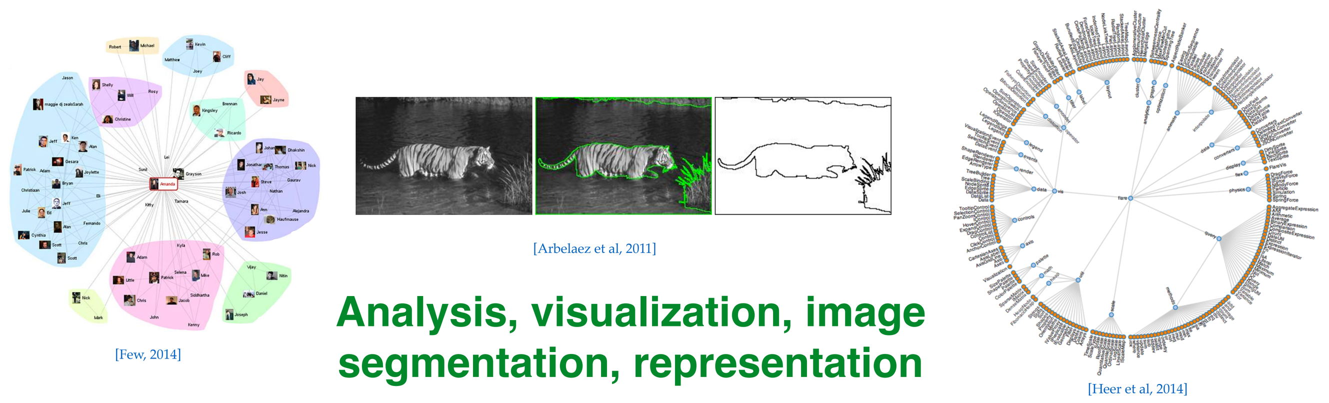
A Hierarchical Algorithm for Extreme Clustering

Ari Kobren, Nicholas Monath*, Akshay Krishnamurthy, Andrew McCallum*
University of Massachusetts Amherst

Extreme Clustering

Clustering: partitioning a dataset into a set of disjoint subsets

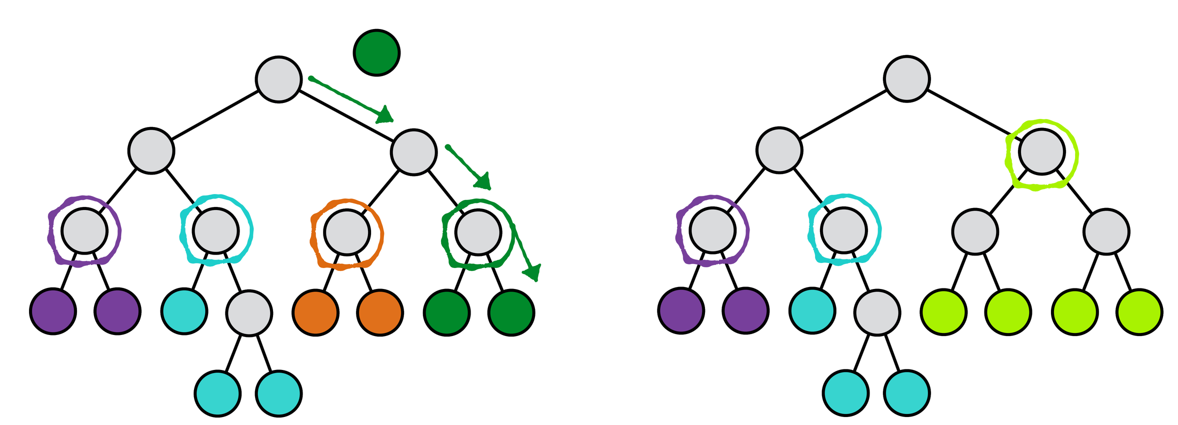


Extreme Clustering: large N and large K

IMAGENET ~14M images, ~21k classes

Cluster Trees

- Insert/search scales with log(n)
- Number of clusters unnecessary a priori
- Online updates and construction
- Represents multiple alternative clusterings



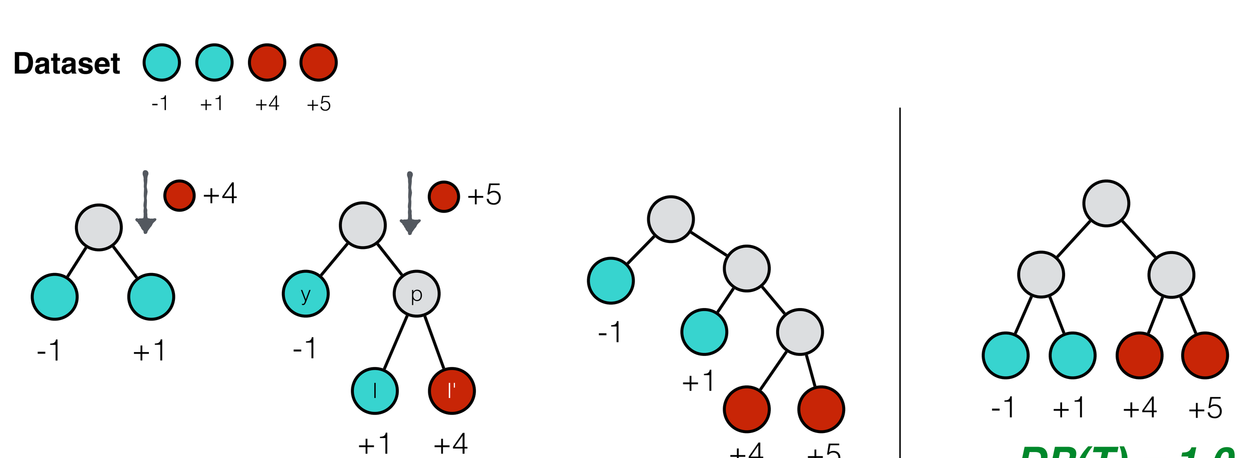
Dendrogram Purity

Holistic measure of tree's clustering quality

$$\mathrm{DP}(\mathcal{T}) = \frac{1}{|\mathcal{P}^\star|}\sum_{k=1}^{K}\sum_{x_i,x_j\in C_k^\star}\mathtt{pur}(\mathtt{lvs}(\mathtt{LCA}(x_i,x_j)), C_k^\star)$$

Greedy Algorithm

Definition 1 (Masking). A node v with sibling v' and aunt a in a tree $\mathcal{T}$ is **masked** if there exists a point $x \in \mathtt{lvs}(v)$ such that $\max_{y\in\mathtt{lvs}(v')}\|x-y\| > \min_{z\in\mathtt{lvs}(a)}\|x-z\|$.



Greedy insertion. *DP(T) = 0.75* *DP(T) = 1.0* *Ground truth.*

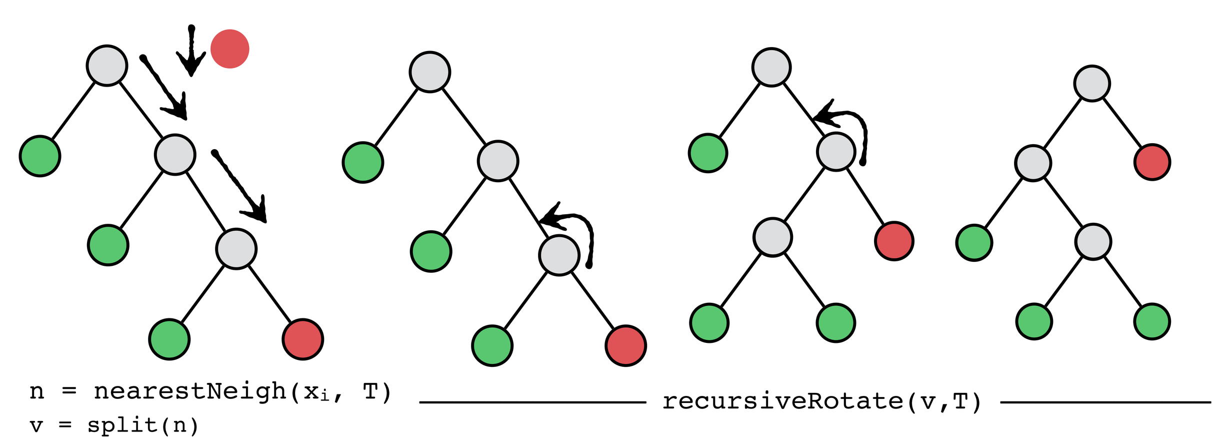
PERCH

Purity **E**nhancing **R**otations for **C**luster **H**ierarchies

```
def perch(x1…xN,T):
  for xi in x1…xN:
    n = nearestNeigh(xi,T)
    v = split(n)
    recursiveRotate(v,T)
    balanceRotate(v,T)
    collapse(v,T)
```

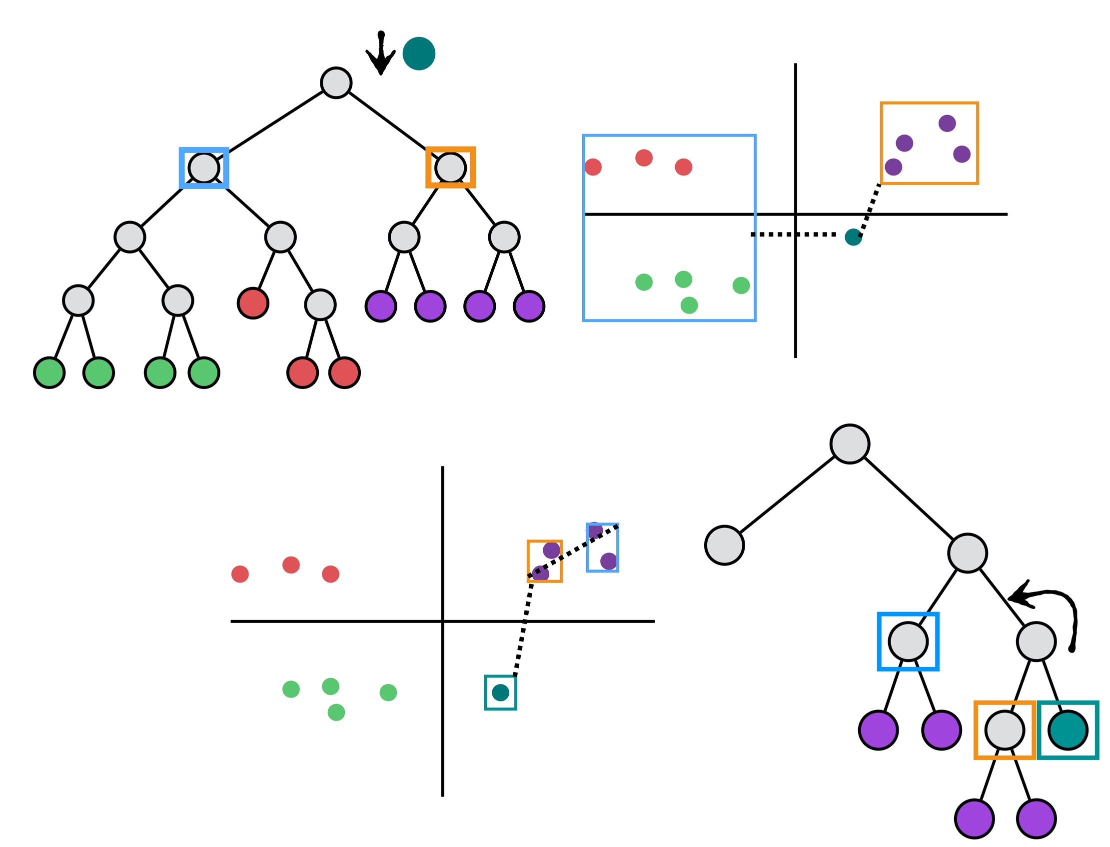
Masking-Based Rotations

Alleviate masking via *masking-based rotations*

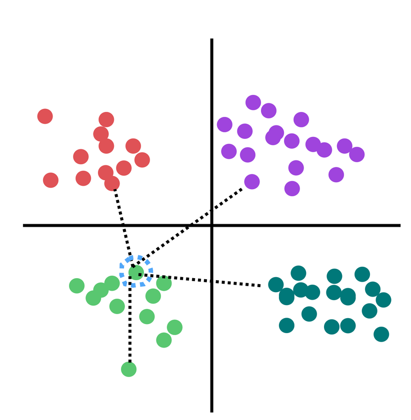


Bounding Boxes for Efficient Computation

Nearest Neighbor Search & Rotation Check



Separated Data



Assumption 1 (Separability). A dataset X is **separable** with respect to a clustering $\mathcal{C}^\star$ if

$$\max_{(x,y)\in\mathcal{P}^\star}\|x-y\| < \min_{(x',y')\notin\mathcal{P}^\star}\|x'-y'\|,$$

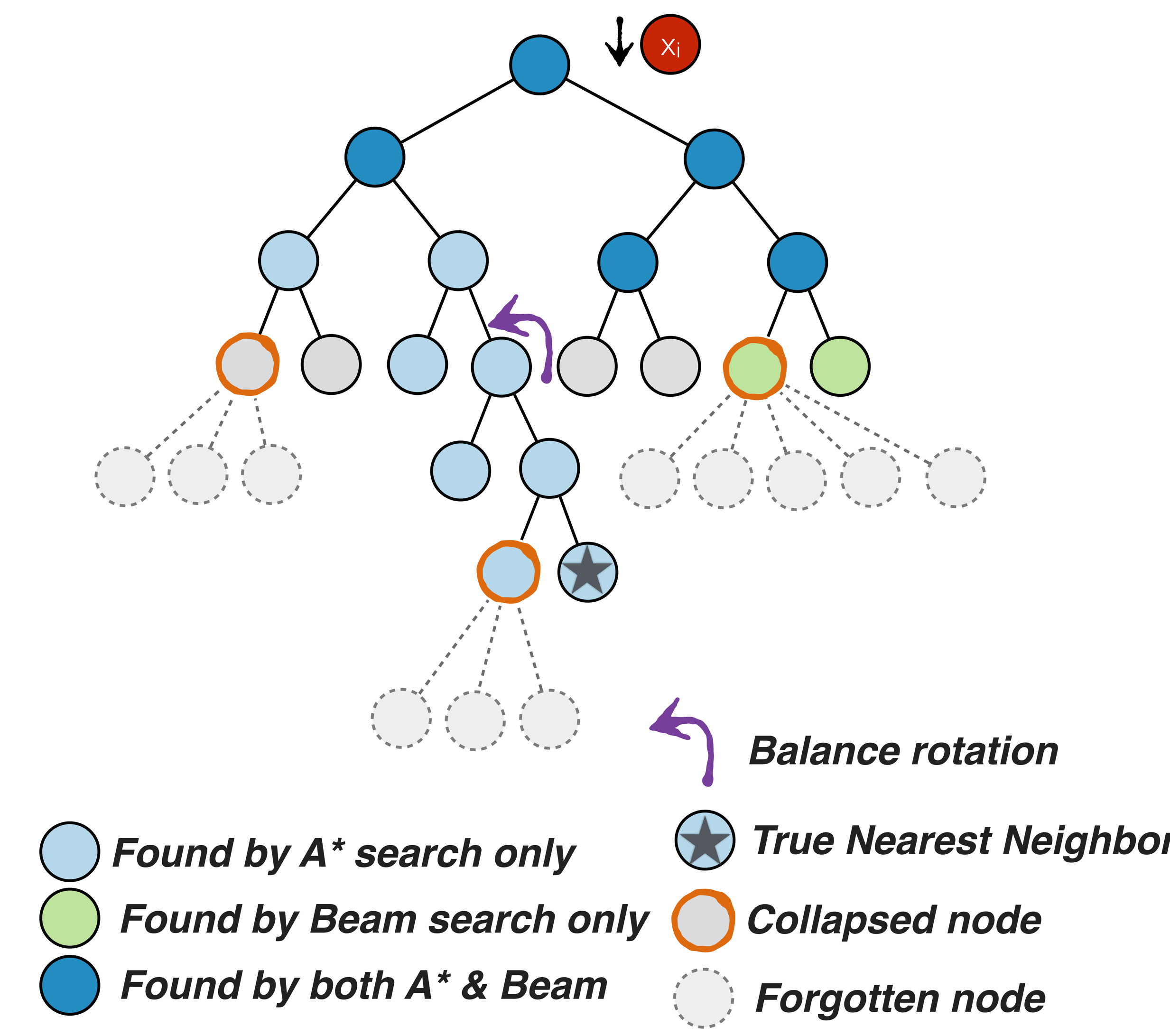
where $\|\cdot\|$ is the Euclidean norm.

Theorem 1. If X is separated w.r.t. $\mathcal{C}^\star$,the greedy algorithm with masking-based rotations constructs a cluster tree with dendrogram purity 1.0.

Beam, Balance & Collapsing

- For additional speed invoke *balance-based rotations*
- *Collapse mode* enforces maximum leaf constraint
- Use approximate nearest neighbor search with beam

PERCH + balance & collapse provably optimal.



Accuracy Experiments

- Compared 10 algorithms on 9 datasets.
- Evaluate pairwise F1 and dendrogram purity.

	Name	Clusters	Points	Dim.
Large Data sets	ImageNet (100K)	17K	100K	2048
	Speaker	4958	36,572	6388
	ILSVRC12	1000	1.3M	2048
	ALOI	1000	108K	128
	ILSVRC12 (50K)	1000	50K	2048
	CoverType	7	581K	54
Small Benchmarks	Digits	10	200	64
	Glass	6	214	10
	Spambase	2	4601	57

Table 1: Dataset statistics.

Method	CovType	ILSVRC12 (50k)	ALOI	ILSVRC 12	Speaker	ImageNet (100k)
Perch	***0.45 ± 0.004***	**0.53 ± 0.003**	***0.44 ± 0.004***	—	**0.37 ± 0.002**	***0.07 ± 0.00***
Perch-BC	***0.45 ± 0.004***	0.36 ± 0.005	0.37 ± 0.008	***0.21 ± 0.017***	0.09 ± 0.001	0.03 ± 0.00
BIRCH (incremental)	0.44 ± 0.002	0.09 ± 0.006	0.21 ± 0.004	0.11 ± 0.006	0.02 ± 0.002	0.02 ± 0.00
MB-HAC-Com.	—	0.43 ± 0.005	0.15 ± 0.003	—	0.01 ± 0.002	—
MB-HAC-Cent.	0.44 ± 0.005	0.02 ± 0.000	0.30 ± 0.002	—	—	—
HKMmeans	0.44 ± 0.001	0.12 ± 0.002	***0.44 ± 0.001***	0.11 ± 0.003	0.12 ± 0.002	0.02 ± 0.00
BIRCH (rebuild)	0.44 ± 0.002	0.26 ± 0.003	0.32 ± 0.002	—	0.22 ± 0.006	0.03 ± 0.00
HAC-Avg	—	*0.54*	—	—	*0.55*	—
HAC-Complete	—	0.40	—	—	0.40	—

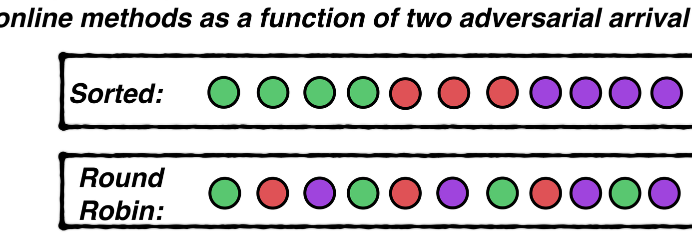
(a) Dendrogram purity for hierarchical clustering.

Method	CoverType	ILSVRC 12 (50k)	ALOI	ILSVRC 12	Speaker	ImageNet (100K)
Perch	22.96 ± 0.7	**54.30 ± 0.3**	***44.21 ± 0.2***	—	**31.80 ± 0.1**	***6.178 ± 0.0***
Perch-BC	22.97 ± 0.8	37.98 ± 0.5	37.48 ± 0.7	25.75 ± 1.7	1.05 ± 0.1	4.144 ± 0.04
SKM++	23.80 ± 0.4	28.46 ± 2.2	37.53 ± 1.0	—	—	—
BICO	***24.53 ± 0.4***	45.18 ± 1.0	32.984 ± 3.4	—	—	—
MB-KM	24.27 ± 0.6	51.73 ± 1.8	40.84 ± 0.5	***56.17 ± 0.4***	1.73 ± 0.141	5.642 ± 0.00
DBSCAN	—	16.95	—	—	22.63	—
Kmeans	24.42 ± 0.00	*60.40 ± 0.5*	39.311 ± 0.3	—	32.185 ± 0.01	—
HAC-Avg	—	—	—	—	40.258	—
HAC-Complete	—	18.28	—	—	*44.297*	—

(b) Pairwise F1 for flat clustering.

Adversarial Orderings

Compare the performance of PERCH and other incremental and online methods as a function of two adversarial arrival orders:



Method	Round.	Sort.
Perch	0.446	0.351
MB-HAC (5K)	0.299	0.464
MB-HAC (2K)	0.171	0.451

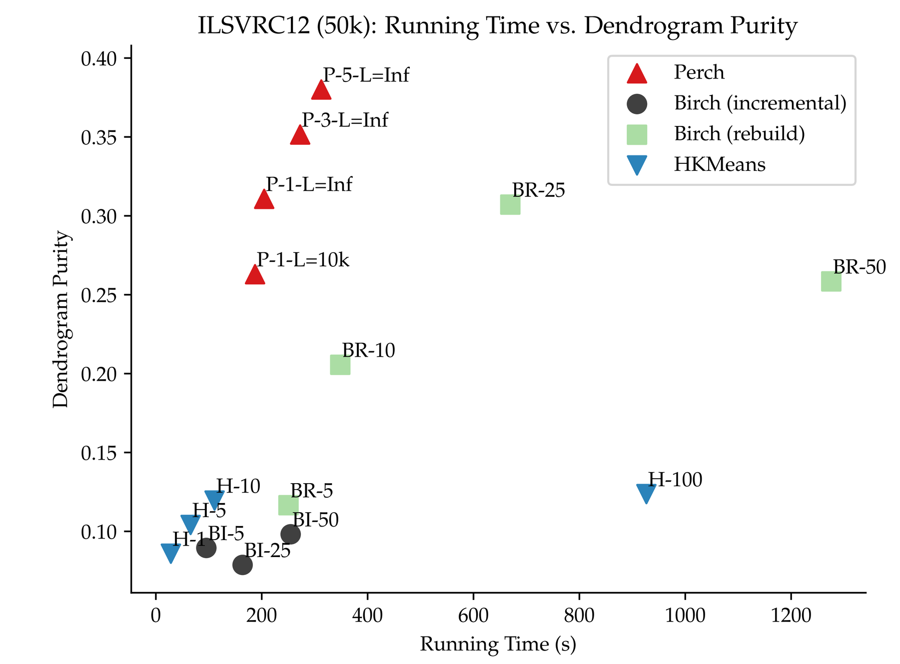
(c) Dendrogram purity on adversarial input orders for ALOI.

Method	Round.	Sort.
Perch	44.77	35.28
o-MB-KM	41.09	19.40
SKM++	43.33	46.67

(d) Pairwise F1 on adversarial input orders for ALOI.

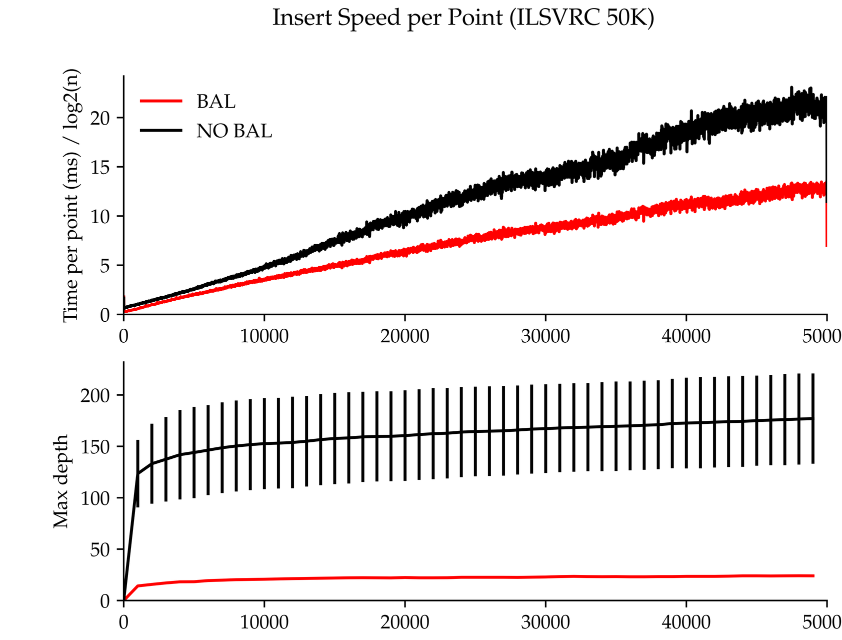
Speed Experiments

- PERCH-BC: Beam width, collapse threshold
- BIRCH: branching factor
- HKMeans: number of iterations per level.



- PERCH produces purer trees in less time
- Others algorithms are faster but low purity

Impact of Balance



- Balance rotations improve running time by reducing tree depth.

Conclusion

- PERCH scales well with both N and K
- Also performant on traditional clustering problems
- Provably optimal on separated data

code: http://github.com/iesl/xcluster

paper: http://dl.acm.org/citation.cfm?id=3098079