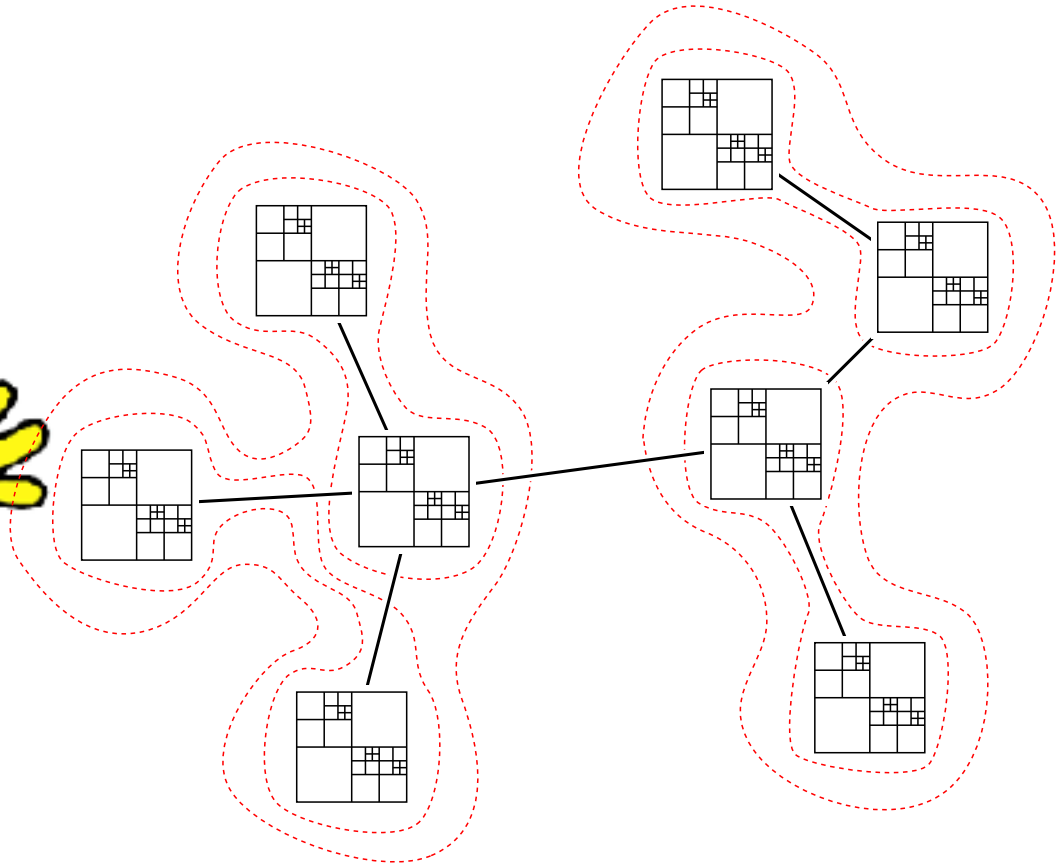
Squarepants in a Tree: Sum of Subtree Clustering and Hyperbolic Pants Decomposition

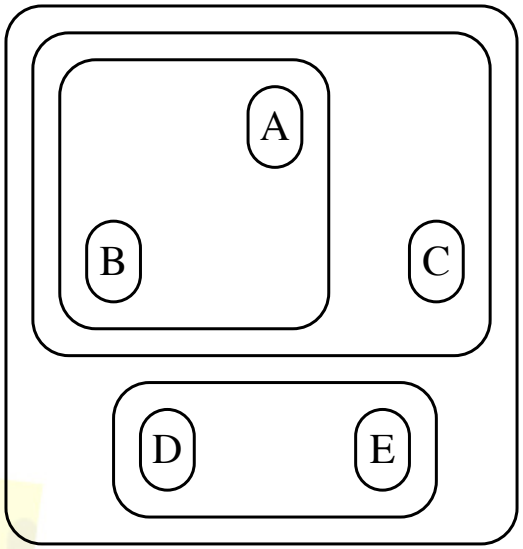


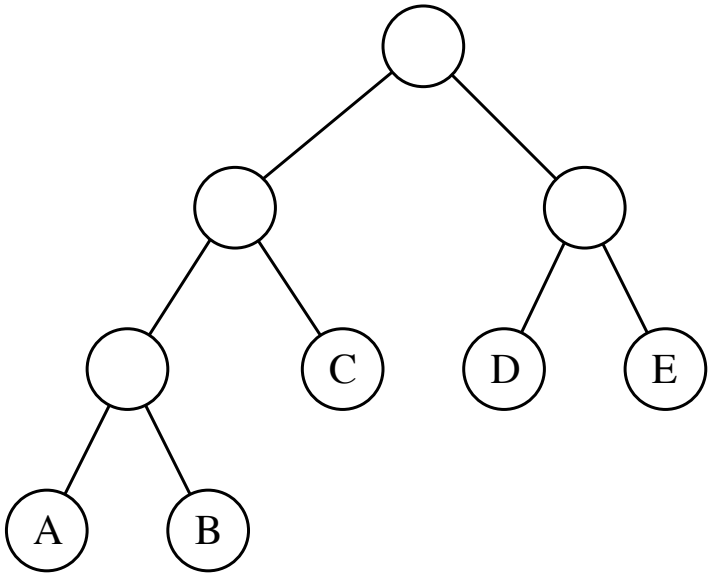
David Eppstein

Three related concepts of hierarchy:

1. Maximal family of sets, each two disjoint or subsets of each other
{}, {A}, {B}, {C}, {D}, {E}, {A,B}, {D,E}, {A,B,C}, {A,B,C,D,E}

2. Family of nested curves such that each region between them has three boundaries "pants decomposition"





3. Binary tree
having elements as its leaves
(evolutionary tree, phylogeny, dendrogram)

Squarepants in a Tree

How to hierarchically cluster a data set?

Heuristic approach:

Build tree by sequence of locally plausible decisions
Agglomerative (bottom up): merge pairs of clusters
Divide and conquer (top down): split clusters into pairs

Statistical approach:

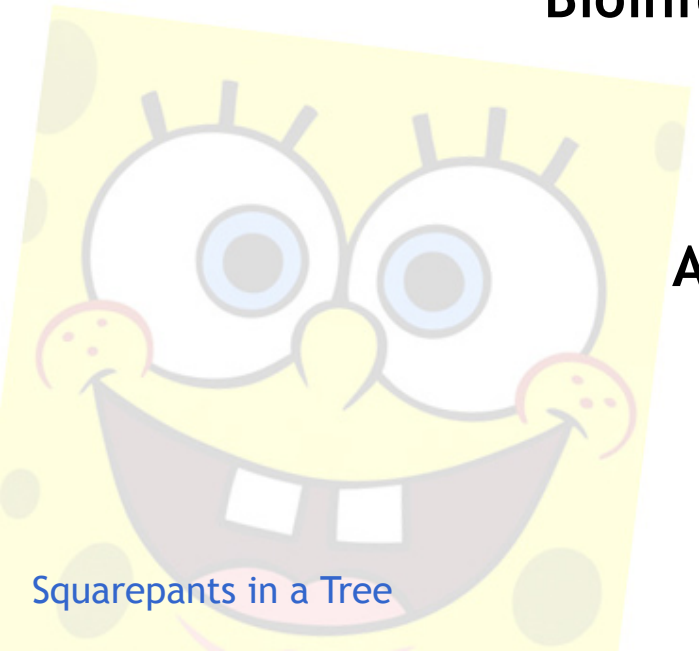
Carefully model Bayesian likelihoods of trees
Heuristic search (e.g. Metropolis) for high likelihood

Bioinformatic approach:

There is a correct evolutionary tree
Find methods that converge to it with enough data

Algorithmic approach:

Define simple quality metrics for trees
Find fast algorithms with guaranteed quality



How to hierarchically cluster a data set?

Heuristic approach:

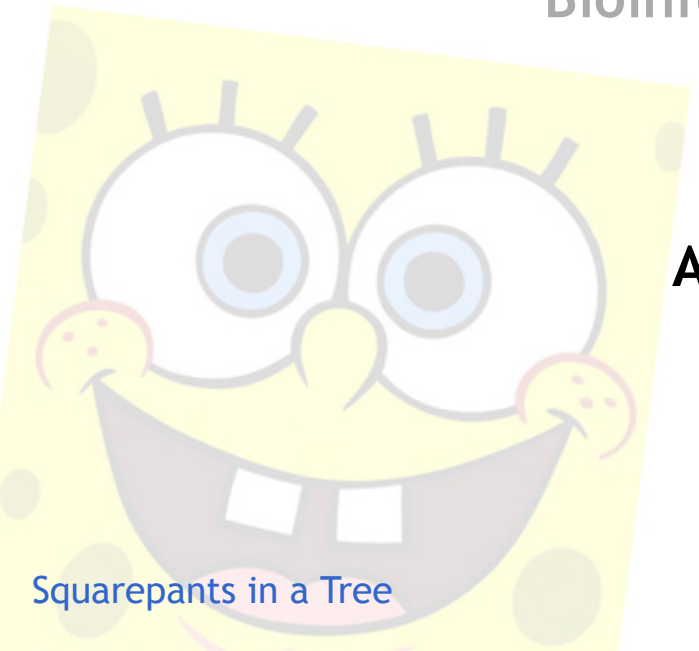
Build tree by sequence of locally plausible decisions
Agglomerative (bottom up): merge pairs of clusters
Divide and conquer (top down): split clusters into pairs

Statistical approach:

Carefully model Bayesian likelihoods of trees
Heuristic search (e.g. Metropolis) for high likelihood

Bioinformatic approach:

There is a correct evolutionary tree
Find methods that converge to it with enough data



Algorithmic approach:

Define simple quality metrics for trees
Find fast algorithms with guaranteed quality

Quality measures:
Sum of sizes of clusters

For points in arbitrary metric spaces:

size = diameter (not studied here)
or
size = length of minimum spanning tree

For points in Euclidean or hyperbolic planes:

size = convex hull perimeter
(proportional to diameter for Euclidean points)
or
size = length of curve in pants decomposition

Outline

Introduction

Hardness of clustering for general metric spaces

Approximate clustering for general metric spaces

Approximate clustering for Euclidean planes

Approximate clustering for hyperbolic planes

Squarepants in a Tree

Outline

Introduction

Hardness of clustering for general metric spaces

Approximate clustering for general metric spaces

Approximate clustering for Euclidean planes

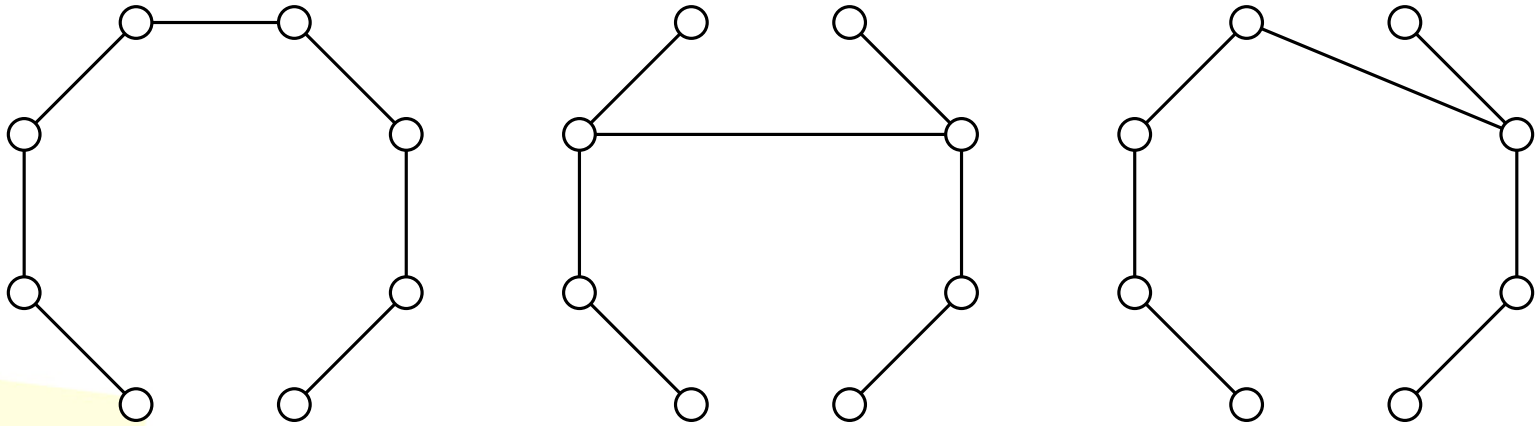
Approximate clustering for hyperbolic planes

Squarepants in a Tree

Bisectable Trees

Define a tree to be i-bisectable if
it is formed by joining two (i-1)-bisectable trees by an edge

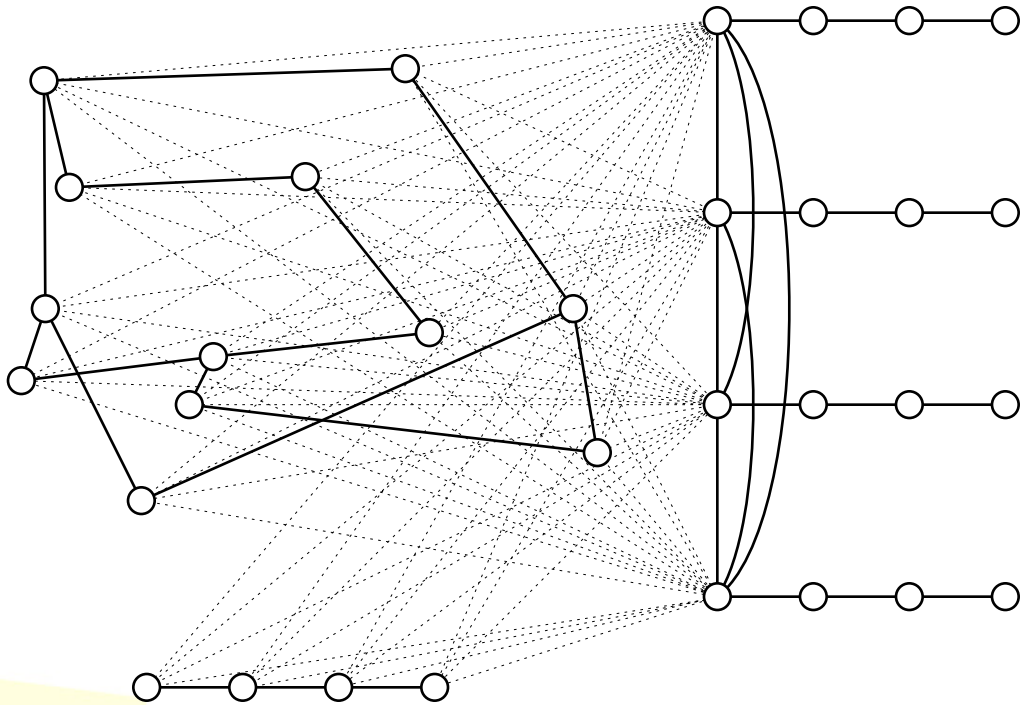
0-bisectable: single vertex
i-bisectable: has 2^i vertices



Claim:
For the metric of distances in any unweighted graph G
there is a hierarchical clustering with
$\sum$(cluster MST lengths) = $i \bullet 2^i$
if and only if G has a bisectable spanning tree

NP-Completeness Proof

The subgraph in the upper left can be covered by disjoint paths of length two if and only if...



...the graph formed by adding the vertices at the left and bottom has a bisectable spanning tree

Conclusion:

Testing whether a metric space has a clustering with small total spanning tree length is NP-complete

Outline

Introduction

Hardness of clustering for general metric spaces

Approximate clustering for general metric spaces

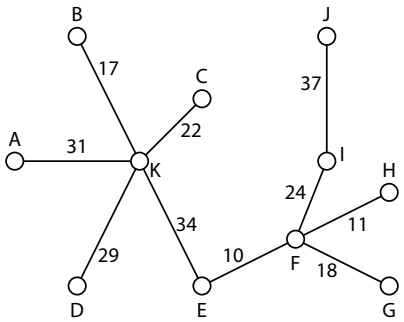
Approximate clustering for Euclidean planes

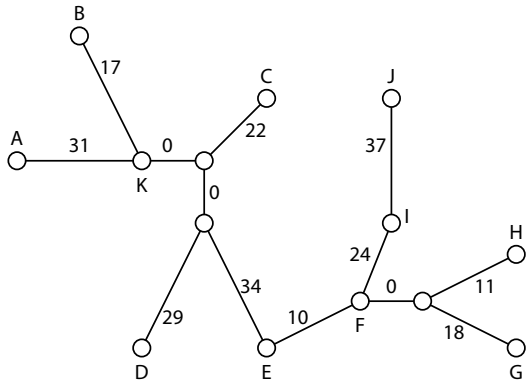
Approximate clustering for hyperbolic planes

Squarepants in a Tree

Algorithm for Approximate Clustering

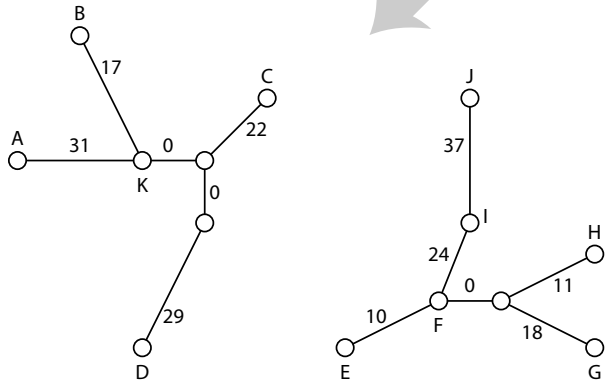
1. Construct a minimum spanning tree of the metric

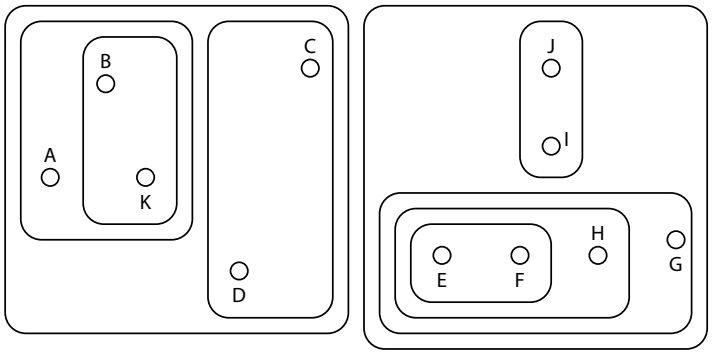




2. Split vertices by length-0 edges so all degrees ≤ 3

3. Remove an edge, form two subtrees with weight ≤ 2/3 the weight of the whole tree



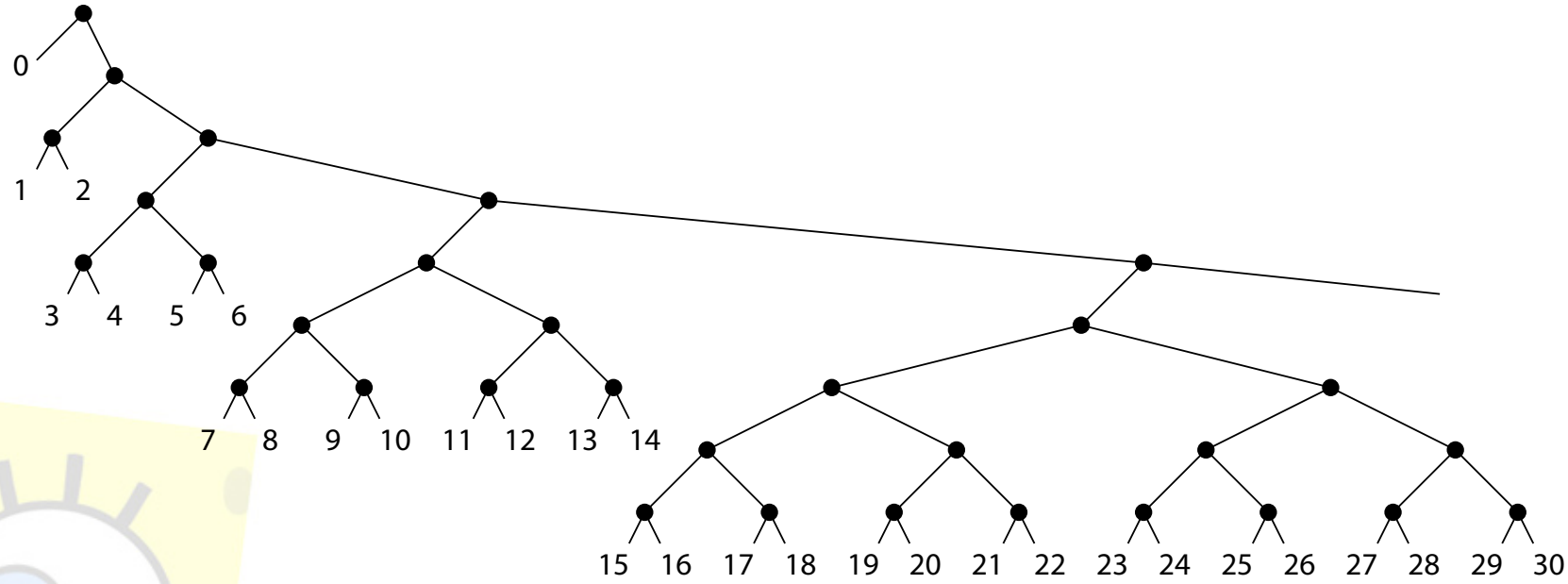


4. Continue splitting subtrees recursively until remaining clusters are single points

Digression: Some Coding Theory

"Prefix-free binary code": binary tree with items as leaves
(in our application, items = MST edges, not points!)

Shannon: if items have probabilities p_i, average path length $\geq \sum p_i \log_2 1/p_i$



This tree has average path length $1 + \sum p_i (2 \text{ floor}(\log_2(i+1))$

Conclusion: $\frac{1}{2} + \sum p_i \text{ floor}(\log_2(i+1)) \geq \frac{1}{2} \sum p_i \log_2 1/p_i$

Quality Guarantees for our Approximation Algorithm

Lower Bound:

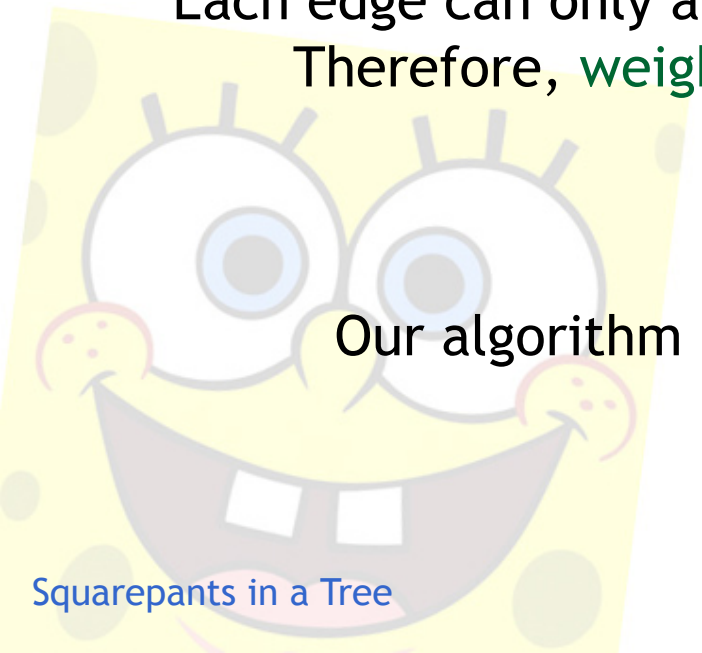
Best possible level-i clustering formed by removing 2^i-1 largest edges from MST
Therefore, summing over all levels in the clustering, with $W = \sum w_i$,
weight of best hierarchy $\geq \sum w_i \, (1 + \text{floor}(\log_2(i+1))) \geq \frac{1}{2} \sum w_i \, (1 + \log_2(W/w_i))$

Upper Bound:

Each time we split a tree, weight of subtrees goes down by 2/3 factor
Each edge can only appear in $\log_{3/2}(W/w_i)$ subtrees before being split
Therefore, weight of our clustering $\leq \sum w_i \, (1 + \log_{3/2}(W/w_i))$

Conclusion:

Our algorithm has approximation ratio $\leq 2 \log_{3/2} 2 \approx 3.42$



Outline

Introduction

Hardness of clustering for general metric spaces

Approximate clustering for general metric spaces

Approximate clustering for Euclidean planes

Approximate clustering for hyperbolic planes

Squarepants in a Tree

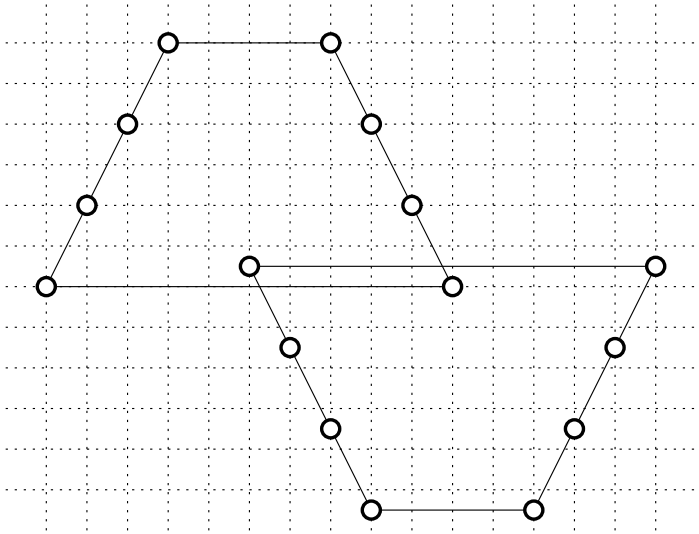
Two different Euclidean clustering problems

Minimum Sum of Convex Hulls

Cluster boundaries are the convex hulls of each cluster, may cross each other

Pants Decomposition

Cluster boundaries can be nonconvex curves, must not cross each other



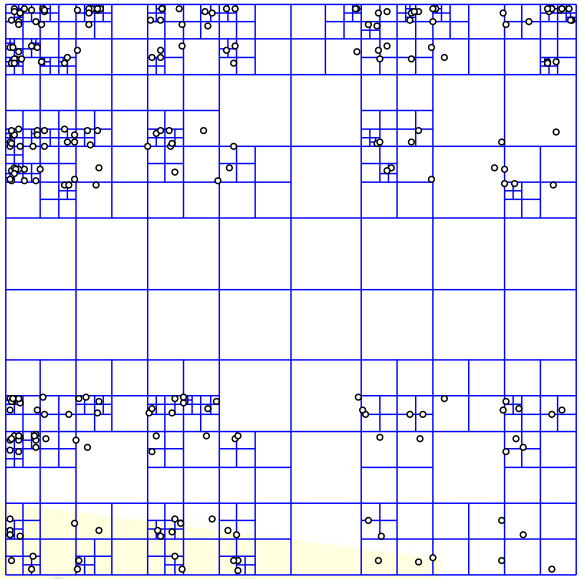
For this point set, the optimal min-sum clustering has overlapping clusters

So the two problems can have different solutions

We provide a single approximation algorithm for both problems

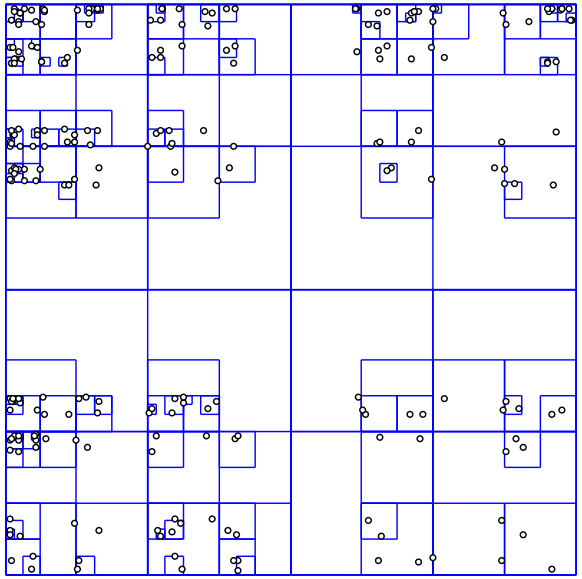
Quadtree:

Repeatedly subdivide square
into four qudrants
until each quadrant contains one point



Compressed Quadtree:

Keep only quadtree squares
that have points in ≥ 2 quadrants



Our clustering algorithm:

Form a cluster for each compressed quadtree square
(or pairs of sibling squares when parent has ≥ 3 children)

Quadtree Clustering Analysis

Local feature size lfs(x) = distance from x to 2nd nearest input point

Total length of (compressed) quadtree ≈ integral 1/lfs

Proof idea: perimeter of unsubdivided square is O(integral 1/lfs)
charge subdivided squares with no unsubdivided children equally among children
charge subdivided squares with a subdivided child to that child

Total hull perimeter of any clustering ≥ const • integral 1/lfs

Proof idea: sum(cluster perimeter) ≈ sum(integral within cluster of 1/perimeter)
= integral (sum of 1/perimeter for containing clusters)
< integral (1/perimeter for smallest containing cluster) ≈ integral 1/lfs

Conclusion: quadtree is constant factor approximation to optimal

Based on an idea of Jim Ruppert for counting triangles in meshes using integral $1/lfs^2$

Outline

Introduction

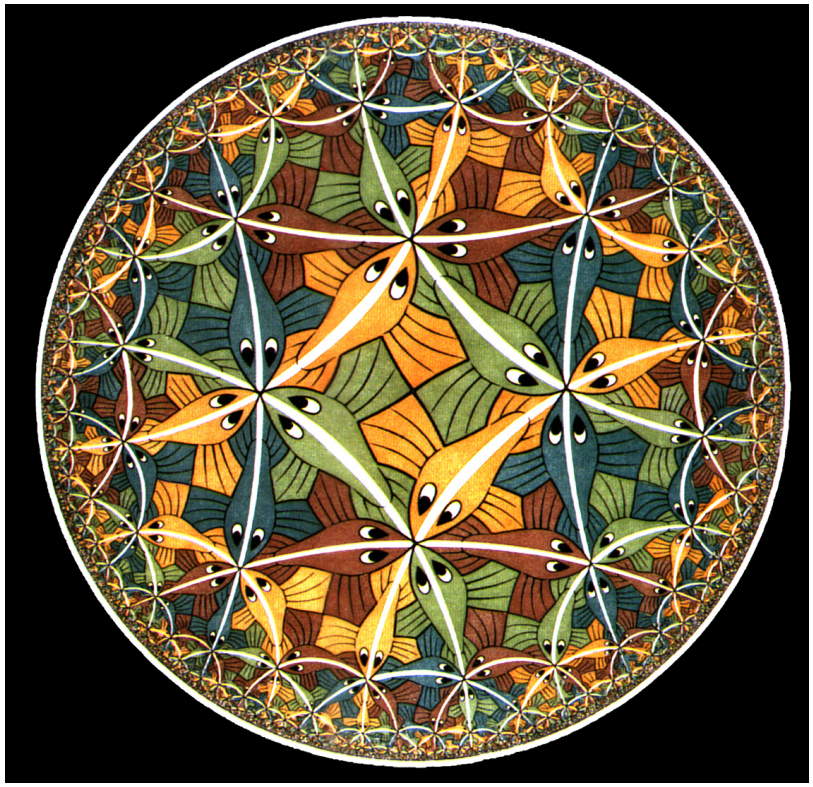
Hardness of clustering for general metric spaces

Approximate clustering for general metric spaces

Approximate clustering for Euclidean planes

Approximate clustering for hyperbolic planes

Squarepants in a Tree



Hyperbolic Geometry

Uniform space with
constant negative curvature

Infinitely many parallel lines
to a given line through any point

Small patches are
approximately Euclidean
but larger areas are not

Triangles have area ≤ π
but circle area ≈ exp(radius)

For any simple closed curve
area ≤ const • perimeter

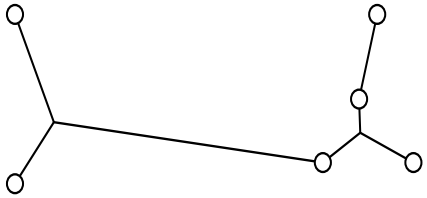
Squarepants in a Tree

Key property:

If a point set has all points at least constant distance apart
Then convex hull perimeter ≈ minimum spanning tree length

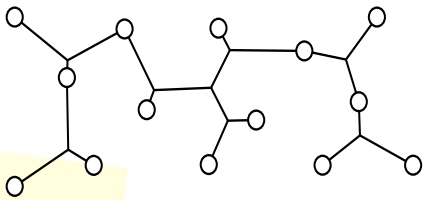
Proof idea:
Show each of the following quantities is at most proportional to the next

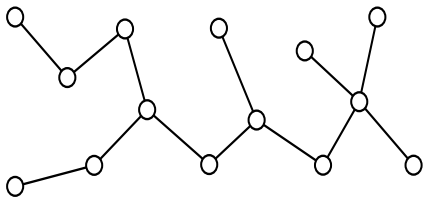
MST length for original points



Steiner tree of original points

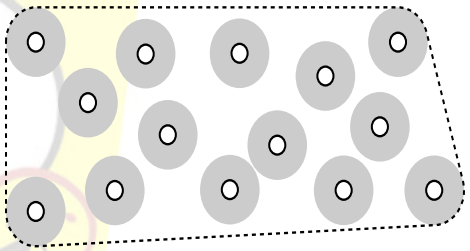
Steiner tree of maximal ∂-separated set

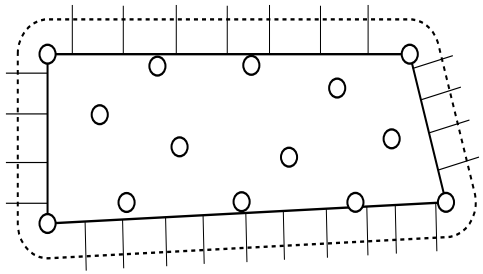




MST of maximal ∂-separated set

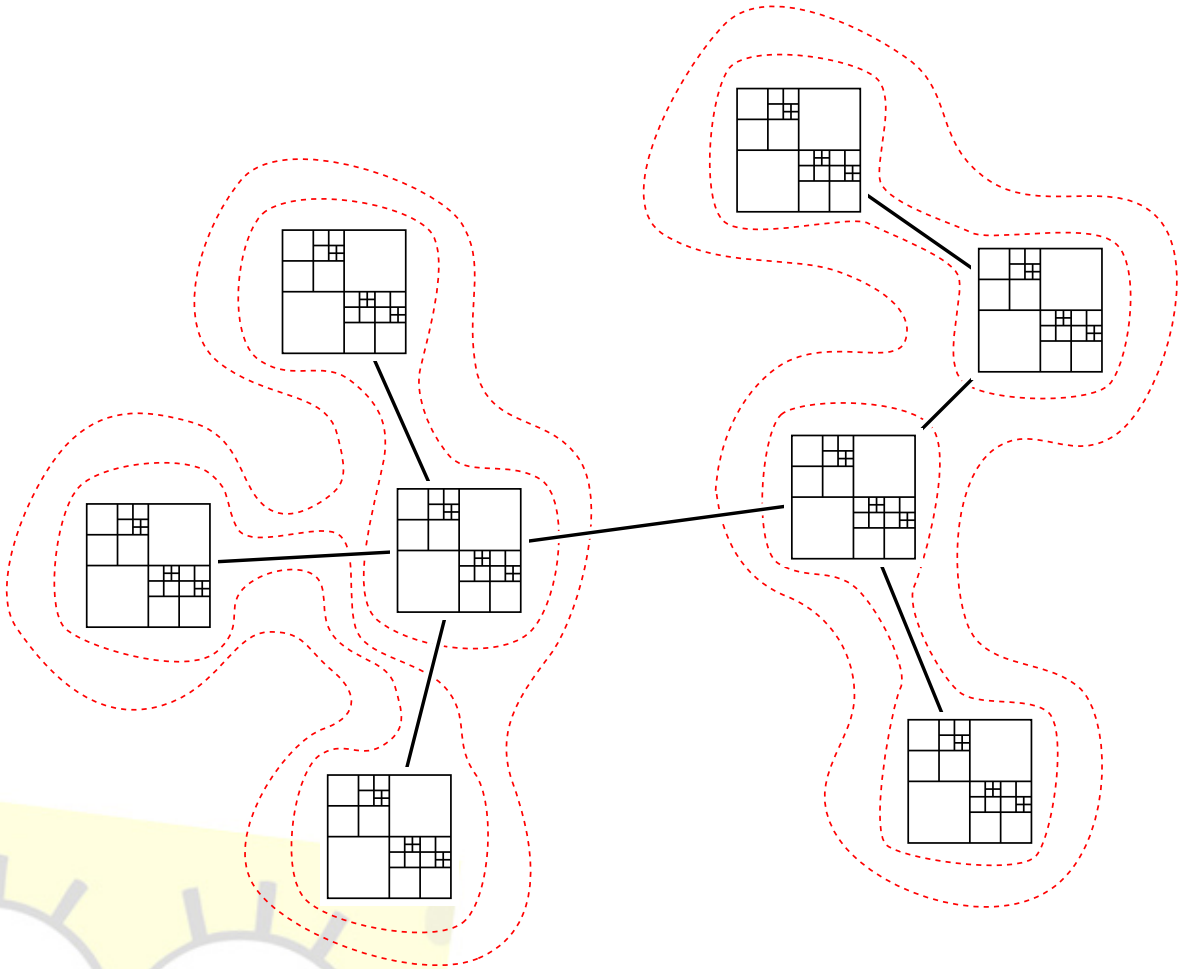
Area of dilated convex hull





Perimeter of original convex hull

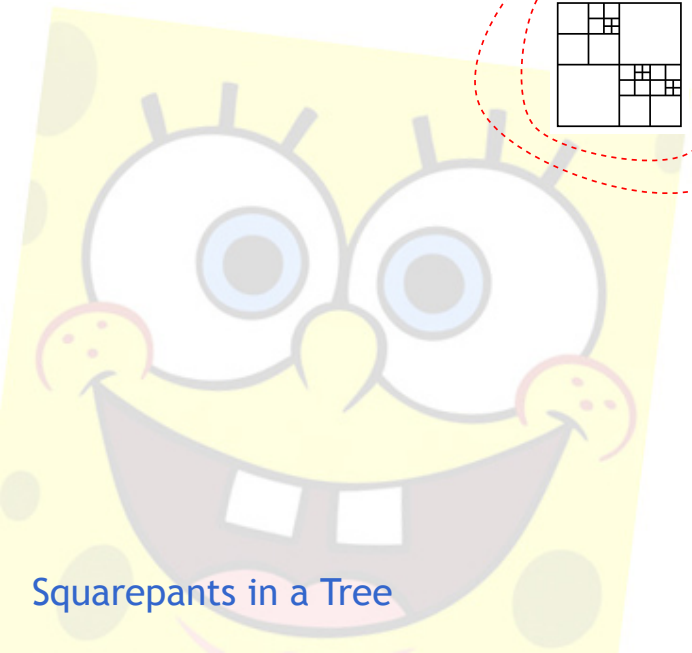
Hyperbolic Approximation Algorithm



1. Find maximal ∂-separated subset, group other points into clusters around them

2. Approximate clusters by Euclidean planes and apply quadtree method in each cluster

3. Use the MST-splitting algorithm for general metric spaces to group unions of clusters



Conclusions

Fast constant-factor approximations to several natural clustering problems

Still open:

Computational complexity of exact solution for geometric problems?

Better approximations?

Other objective functions?

Squarepants in a Tree