Fairness and Explainability

TOP: Data Clustering 076/091

Instructor: Sayan Bandyapadhyay

Portland State University

Outline

Factors of Importance

In ML/Algorithms, what are the most important factors that drive technological innovation?

Factors of Importance

In ML/Algorithms, what are the most important factors that drive technological innovation?

- Running time
- Query time
- Update time
- Storage and handling of big data
- Computational power of hardware
- Quality
- Privacy and security

Factors of Importance

In ML/Algorithms, what are the most important factors that drive technological innovation?

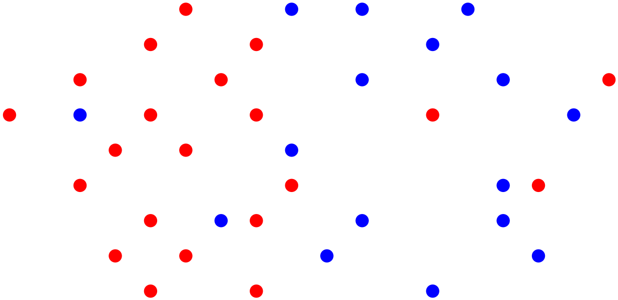
- Running time
- Query time
- Update time
- Storage and handling of big data
- Computational power of hardware
- Quality
- Privacy and security

Recent challenges

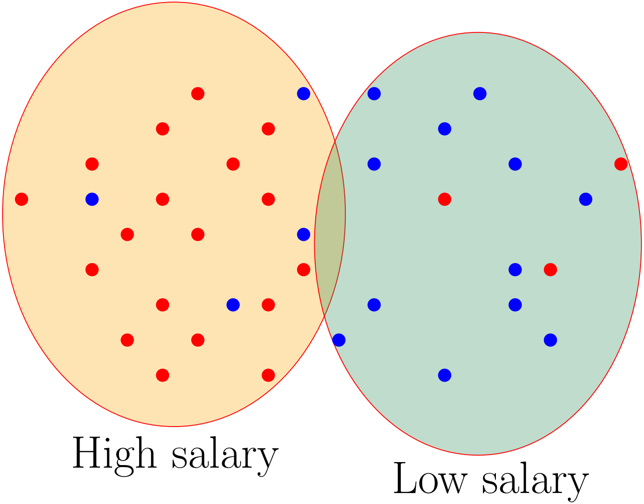
- Fairness
- Explainability

Recommendation of Jobs

Recommendation of Jobs



Recommendation of Jobs



Recommendation of Jobs

DE GRUYTER OPEN Proceedings on Privacy Enhancing Technologies 2015; 2015 (1):92–112

Amit Datta*, Michael Carl Tschantz, and Anupam Datta

Automated Experiments on Ad Privacy Settings

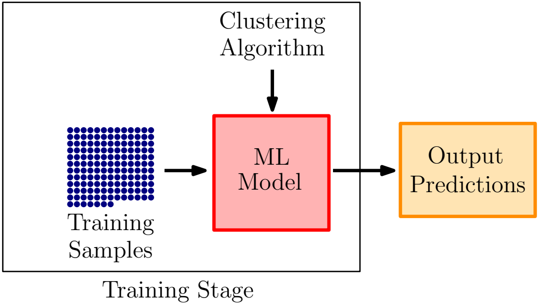
A Tale of Opacity, Choice, and Discrimination

ANITA BORG INSTITUTE
WOMEN TRANSFORMING TECHNOLOGY
GRACE HOPPER
CELEBRATION OF WOMEN IN COMPUTING

Recommending Dream Jobs in a Biased Real World

Nadia FAWAZ
LinkedIn Corporation
nfawaz@linkedin.com

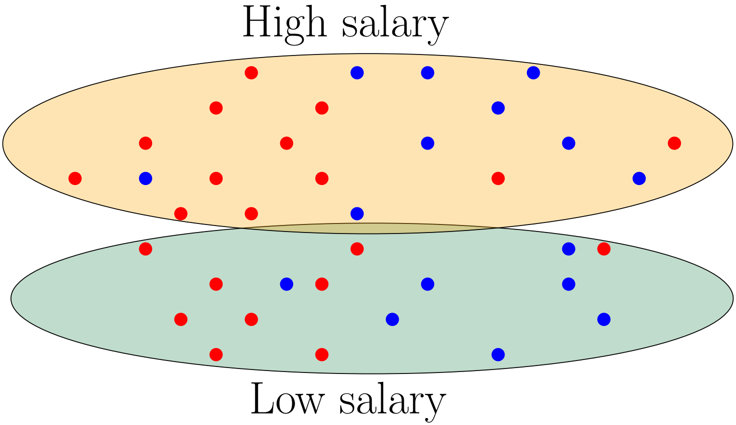
The ML Pipeline



Training the classifier: feature engineering/labeling of samples

Outline

Fair Clustering



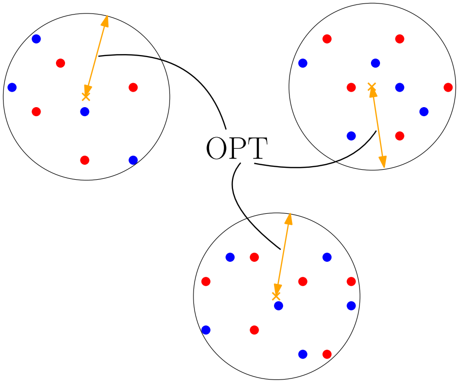
(Proportionally) Fair k-center Clustering

- Given colored points in $\mathbb{R}^d$
- Find a set C of k points (cluster centers) in $\mathbb{R}^d$ that
  - minimizes $\text{cost}(C)$; and
  - each cluster is fair or color-balanced

Fair k-center Clustering

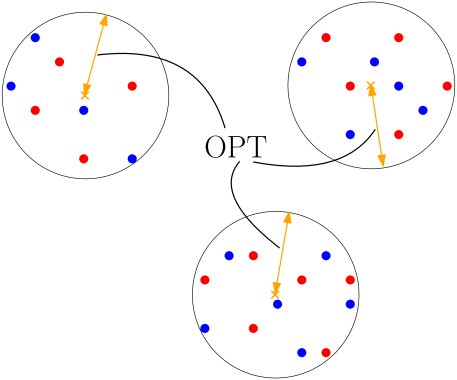
- Given n red and n blue points in $\mathbb{R}^d$
- Find a set C of k points (cluster centers) in $\mathbb{R}^d$ that
  - minimizes $\max_{c \in C}$ radius(c); and
  - each cluster has equal number of red and blue points.

Fair Clustering



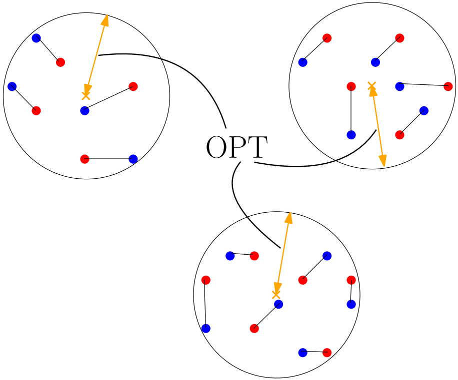
Outline

How to address fairness?



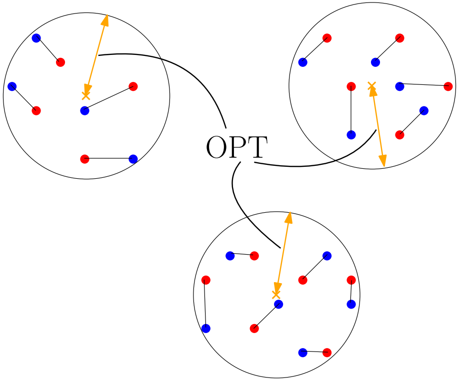
We know how to solve regular k-center problem

Observations



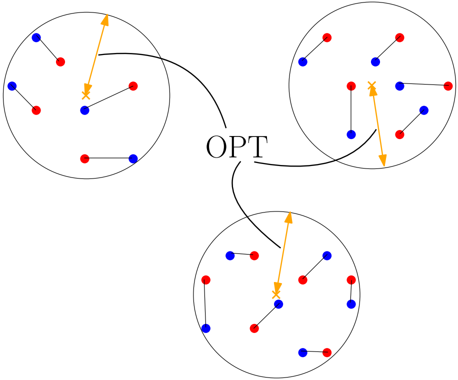
- Consider the bipartite graph between red and blue points (vertices)

Observations



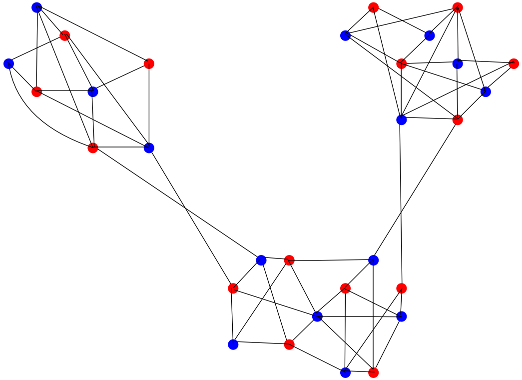
- Consider the bipartite graph between red and blue points (vertices)
- There is a perfect matching in this graph

Observations



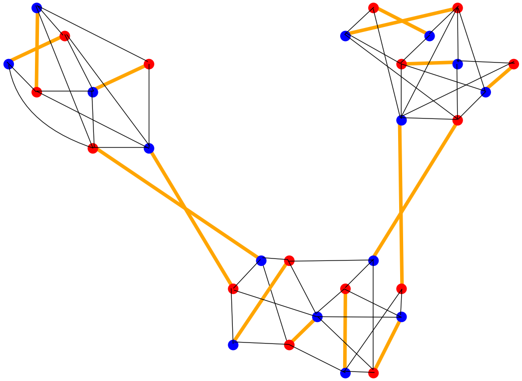
- Consider the bipartite graph between red and blue points (vertices)
- There is a perfect matching in this graph
- Each edge has length at most $2 \cdot \mathrm{OPT}$

The Algorithm



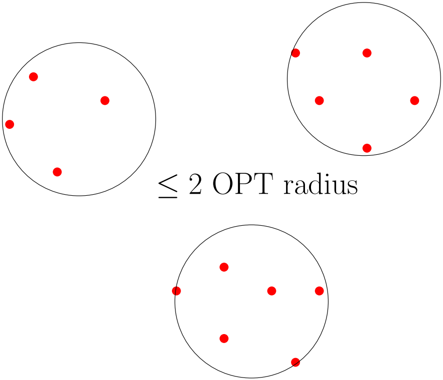
Compute a bipartite graph G between red and blue points (vertices) by adding edges of length at most $2 \cdot \mathrm{OPT}$

The Algorithm



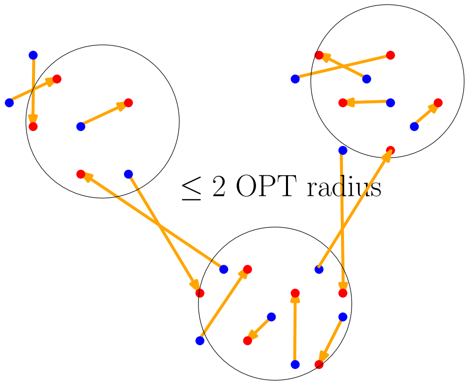
Compute a perfect matching in G (orange edges)

The Algorithm



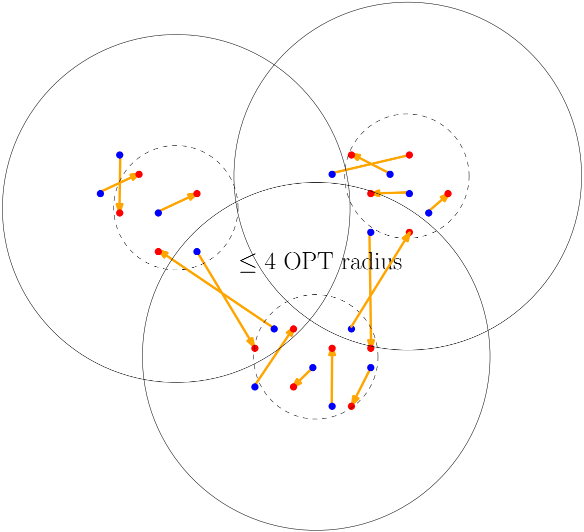
Compute a 2-approximate k-center clustering of the red points

The Algorithm



Assign each blue point to the cluster of the red point matched to it

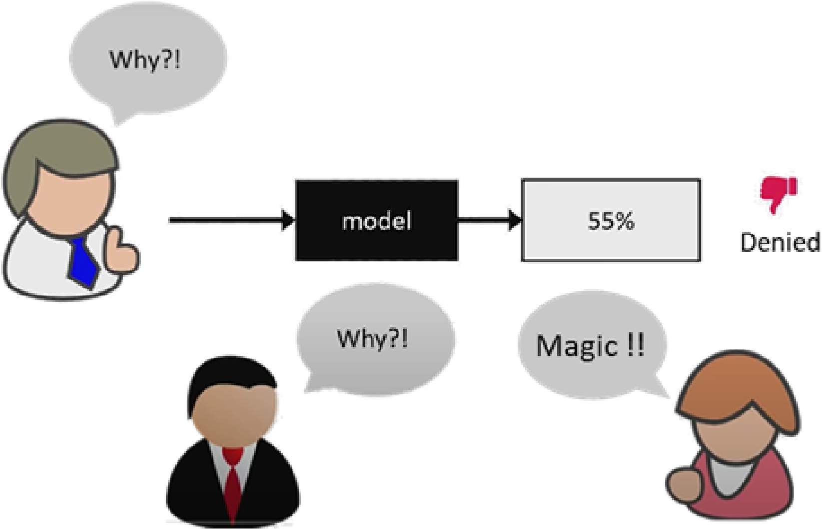
The Algorithm



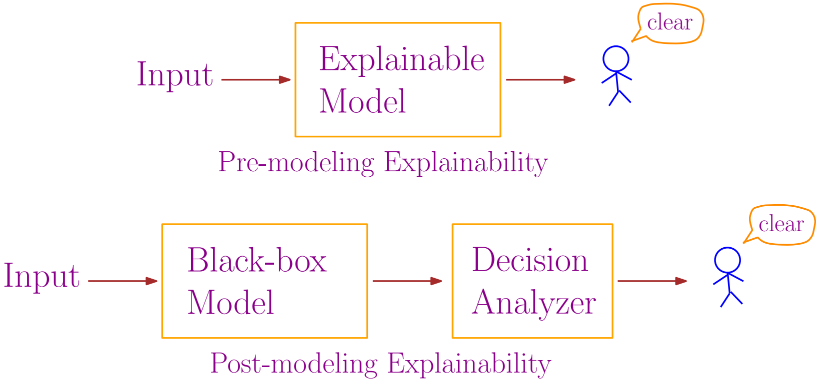
Expand each ball by 2OPT to cover all points assigned to the cluster

Outline

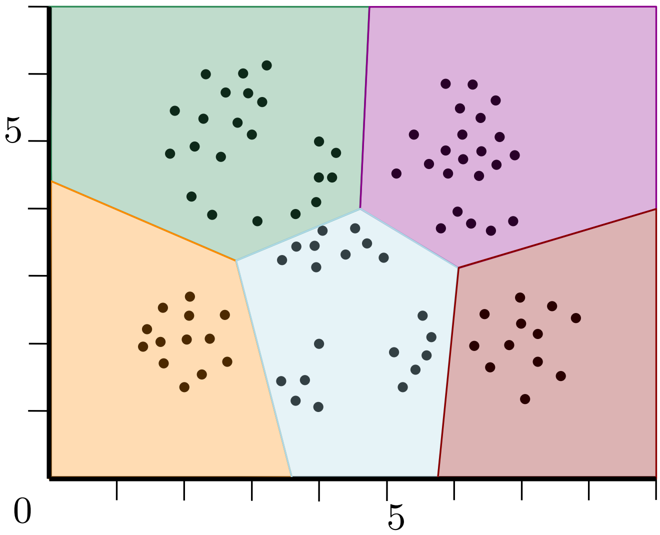
Black Magic of Black Boxes



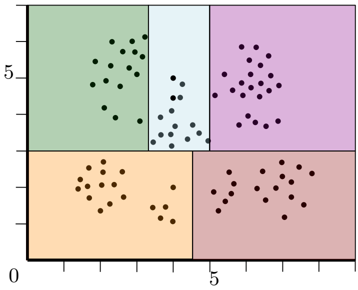
Explainable AI

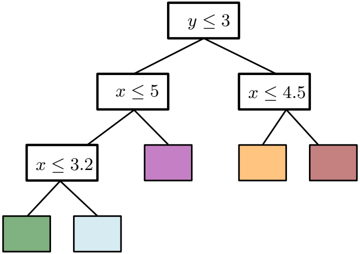


Explaining a Clustering



Pre-modeling Explainability[1]

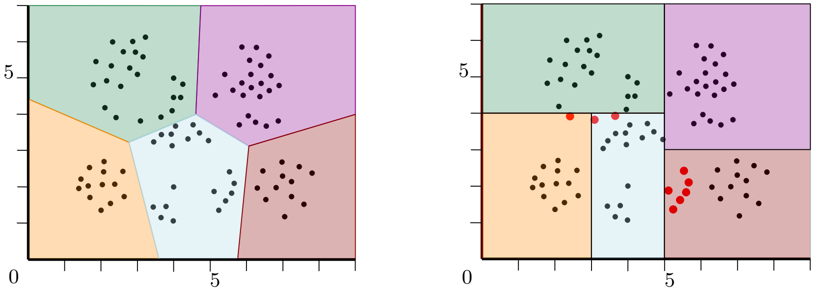




Inherent explainability via decision tree

[1]Moshkovitz et al. ICML 2020

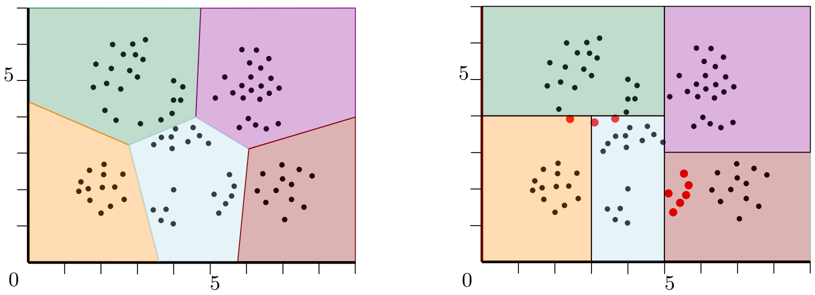
Post-modeling Explainability



Remove a subset of points to explain the given clustering

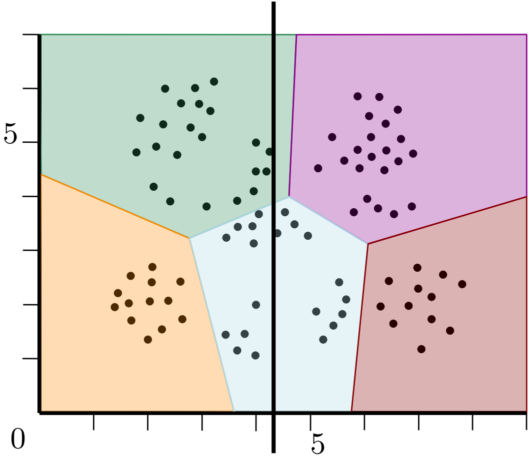
Outline

Clustering Explanation



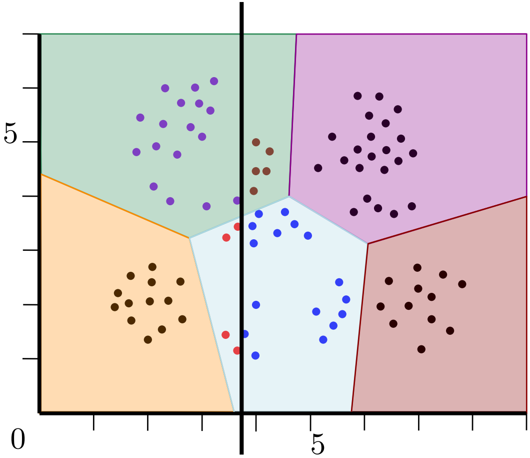
Remove s points to explain the given clustering

An Exact Algorithm for 2D



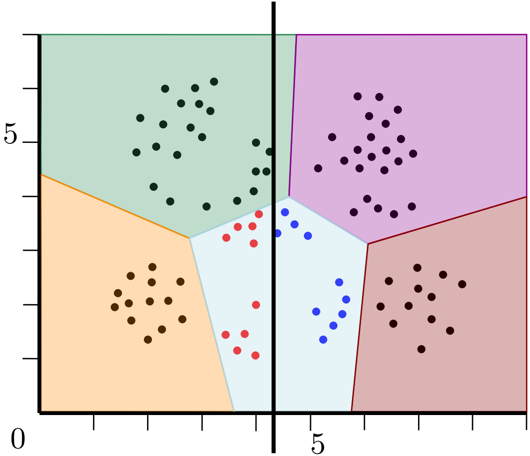
A cut might separate a cluster

An Exact Algorithm for 2D



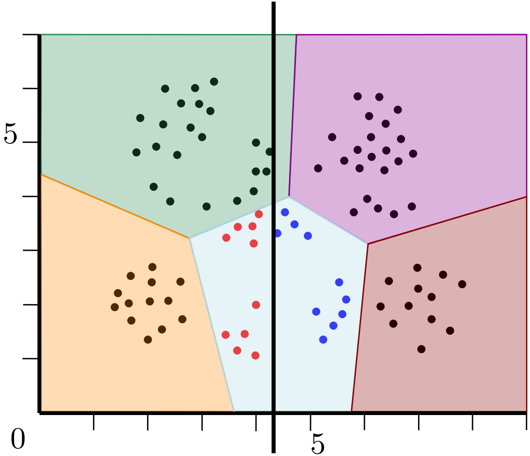
A cut might separate multiple clusters - 2^k choices to make

An Exact Algorithm for 2D



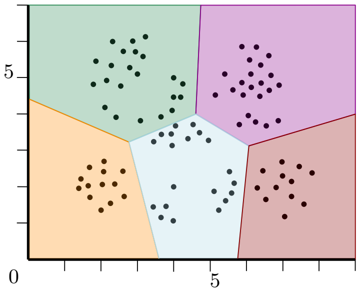
You have $2(n-1)$ choices for vertical and horizontal cuts -

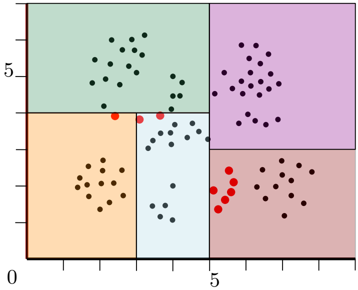
An Exact Algorithm for 2D



You have $2(n-1)$ choices for vertical and horizontal cuts -
$2(n-1)2^k =$ choices for selecting the first cut

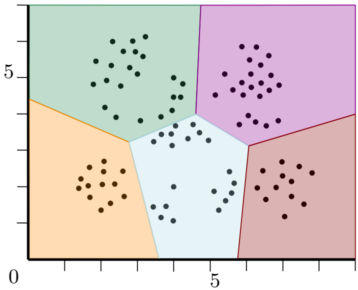
An Exact Algorithm for 2D

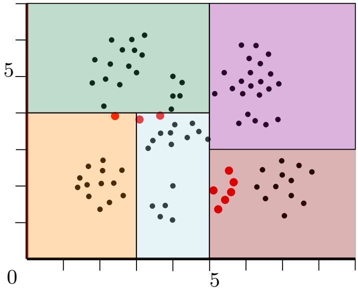




For k clusters, we need $k - 1$ cuts

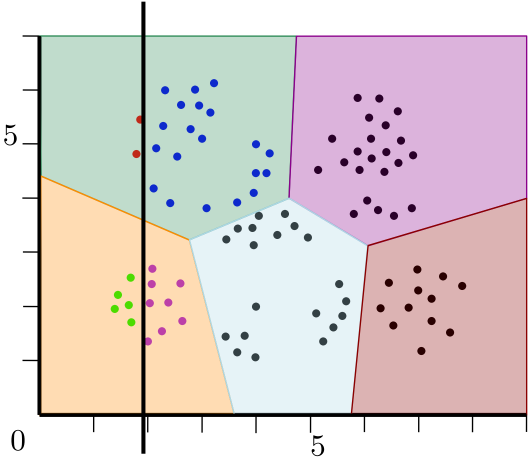
An Exact Algorithm for 2D





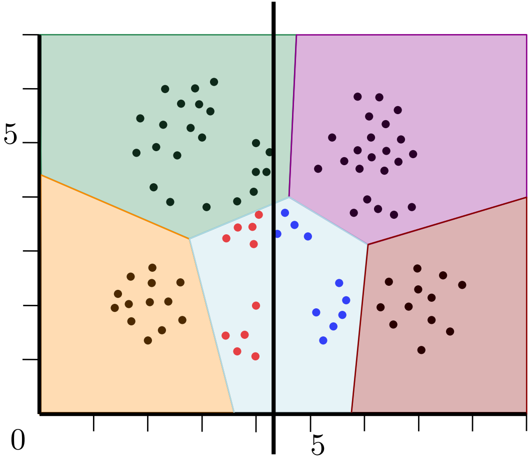
For k clusters, we need $k-1$ cuts –> $(2(n-1)2^k))^{k-1}$
$=O(n^k 2^{k^2})$ choices in total

An Approximation Algorithm



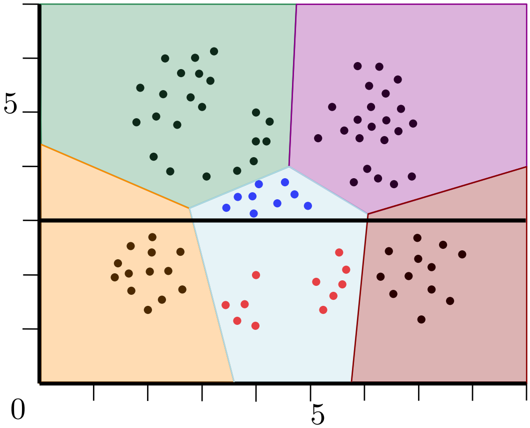
Several choices for cuts

An Approximation Algorithm



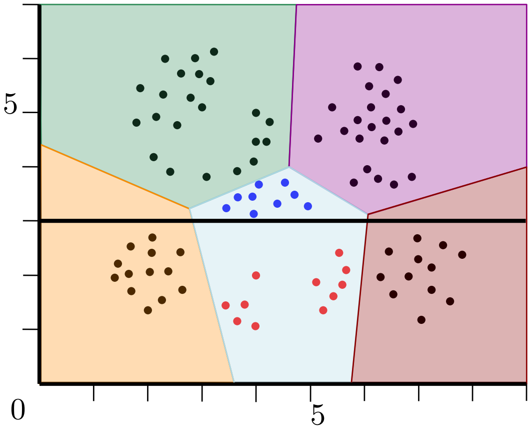
Several choices for cuts

An Approximation Algorithm



Greedily pick the cut that removes the minimum number of points: remove the smaller chunk for each cluster

An Approximation Algorithm



Runs in poly-time – $(k-1)$-approximation


THANK YOU