Reminder: Paper summary / Review due tomorrow

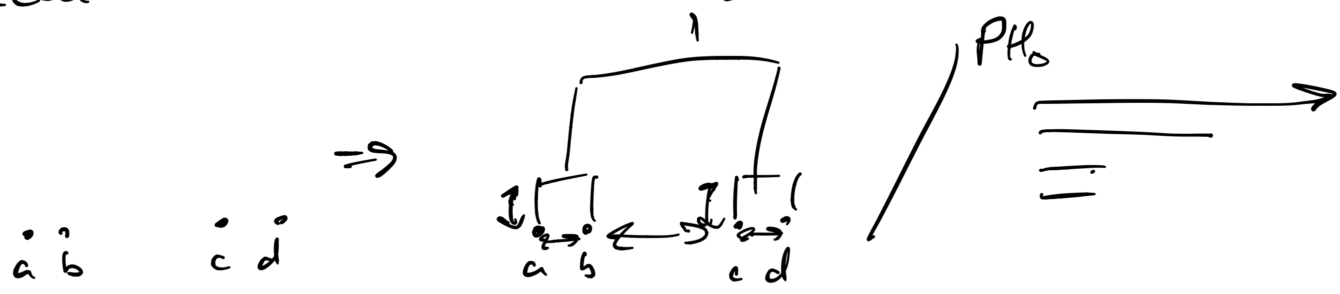
Thursday next week (3/11) project presentations

10 min/group

- say something abt setup
- details abt methodology/constructions
- any (partial) results/conclusions

Applications to Clustering & Regularization

Recall Hierarchical clustering (single-linkage)

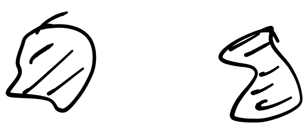


dendrogram

0-dimensional Rips barcode & dendrogram contain same information.

Applications of PH:

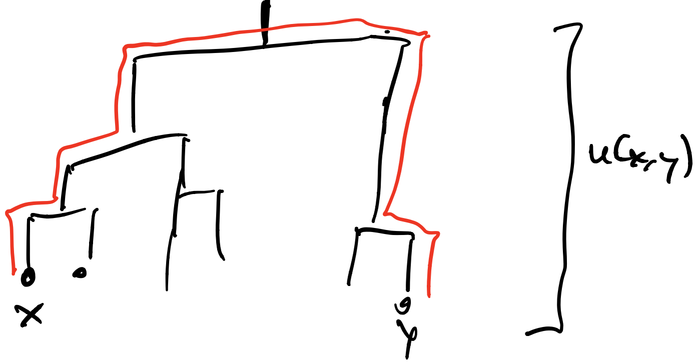
- "sweet range" of Rips complex implies "sweet range" for dendrogram.



Another nice result: Carlsson & Memoli 2010:

define ultrametric u on dendrogram.

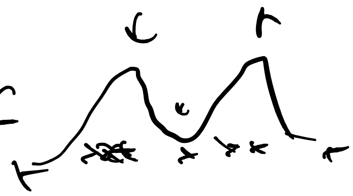
$u(x,y)$ = min height path connecting x & y on dendrogram



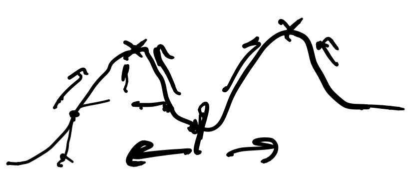
$\Rightarrow d_{GH}((P, u_P), (Q, u_Q))$
$\leq d_{GH}((P, d_P), (Q, d_Q))$

Problem w/ single linkage clustering (same as with Rips stability): sensitivity to outliers.

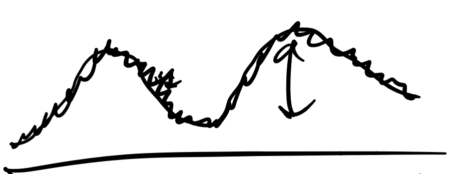
Alternative clustering methods:
$\underline{k}$-means, DBSCAN (density thresholding)
spectral clustering (weighted adjacency graphs)
Mode-seeking clustering:
data drawn from density fn f. $X \sim f$



find local maxima of f. partition by "basin of attraction"
some kind of "hill climbing" procedure



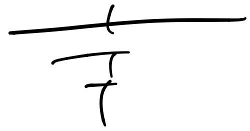
Common issue in practice
density estimation produce
many local maxima.



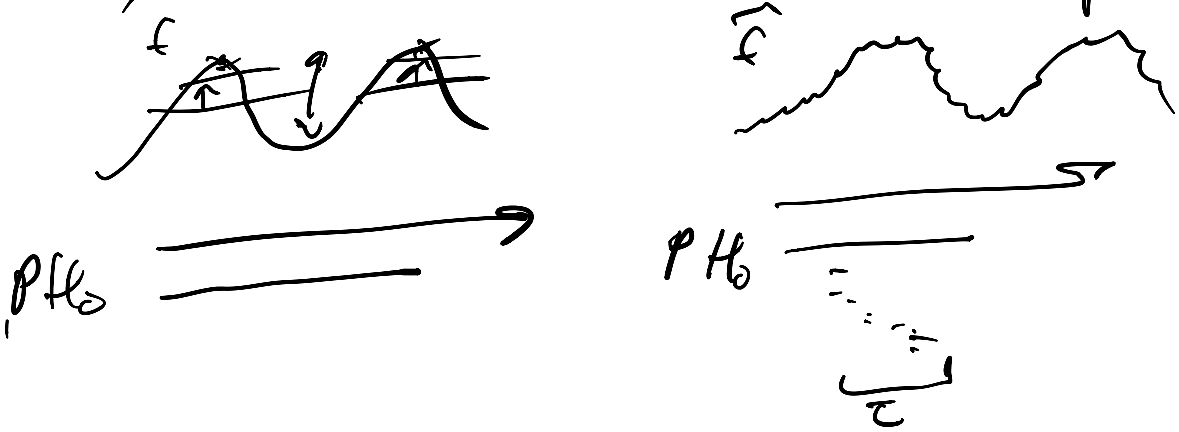
use some sort of smoothing

Idea of ToMATO (Chazal et al 2013)
"Topological Mode Analysis Tool" use persistence
instead of smoothing procedure.

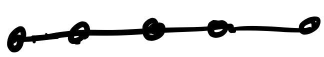
Clustering is discrete
Topology is discrete
persistence turns discrete into continuous.

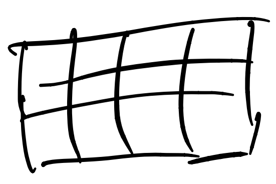


use super-levelset filtrations on density estimator
& merge peaks w/ prominence $\leq \tau$ to "parent peaks"
↑ parameter



ToMATo:
input: graph G, vertices are data points
edges if pts adjacent (proximity rule)

e.g. 



function $\hat{f}: V \to \mathbb{R}_+$ (estimated density of each vtx)
1) (Mode seeking): compute initial clusters:
iterating over vertices by decreasing $\hat{f}$ value.
Connect vertex i w/ neighbor with highest value $\hat{f}(j) \geq \hat{f}(i)$
if no such nbrs, then i is a "peak"
result: spanning forest of G, each tree is basin of attraction

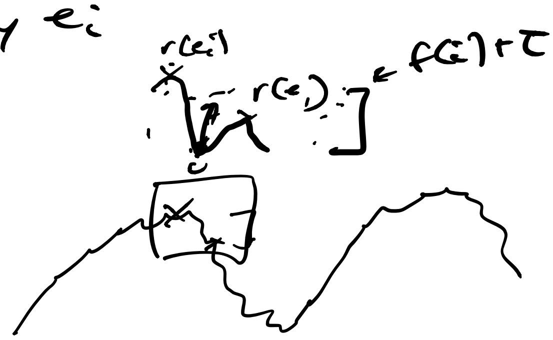
2) iterate over vertices by decreasing $\hat{f}$ value.
maintain union-find data structure, every entry a union of trees in spanning forest.

a) either vtx i is a peak, start a new tree

b) vtx i is not a peak. See if any of its nbrs are in same basin of attraction.
if not, and prominence of peaks $< \tau$, then combine basins of attraction (merge clusters)
smaller peak always merged into larger peak (elder rule)

Alg: sort indices $\hat{f}(1) \geq \hat{f}(2) \geq \dots \hat{f}(n)$
initialize union-find data structure $\mathcal{U}$.
two vectors g, r of length n.

for $i = 1 \dots n$
  N nbhd of i with $j < i$ $\forall i \in N$
  if $N = \emptyset$,
    create new entry e_i in $\mathcal{U}$
    $r(e_i) \leftarrow i$ // root of entry e_i
  else
    $g(i) \leftarrow \underset{j \in N}{\operatorname{argmax}} \hat{f}(j)$
    $e_i \leftarrow \mathcal{U}.\text{find}(j)$
    attach vtx i to e_i
    for $j \in N$
      $e \leftarrow \mathcal{U}.\text{find}(j)$
      if $e \neq e_i$ and $\min\{\hat{f}(r(e)), \hat{f}(r(e_i))\} < \hat{f}(i) + \tau$
        $\mathcal{U}.\text{union}(e, e_i)$
        $r(e \cup e_i) \leftarrow \underset{r(e), r(e_i)}{\operatorname{argmax}} \hat{f}$



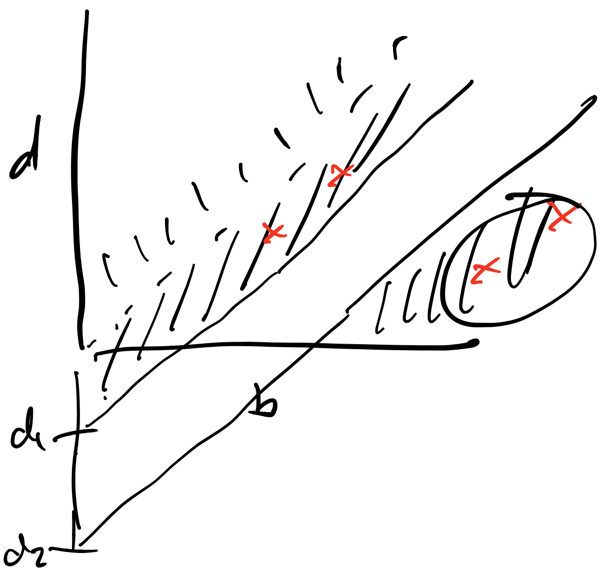
$e_i \cancel{\geq e} \cup e_i$

return collection of e st. $\hat{f}(r(e)) \geq \tau$

main result: when can we recover true # clusters?

Def: given $d_2 > d_1 > 0$, we say $dgm_0(f)$ is (d_1, d_2)-separated if every pt of $dgm_0(f)$ – lies either above $y = x - d_1$

or below $y = x - d_2$



super-level set PH has (b,d)-pairs below diagonal.

larger gap $d_2 - d_1$ makes it easier to separate "prominent" peaks

Thm: assume $dgm_0(f)$ is (d_1, d_2)-separated with $d_2 - d_1 > 5\eta$ ($\eta = \|f - \hat{f}\|_\infty$: $|f(x) - \hat{f}(x)| \leq \eta \ \forall x$)

Then for any δ, $0 < \delta < \min(\underbrace{\rho(x)}_{\text{cvx radius}}, \frac{d_2 - d_1 - 5\eta}{c})$

and any threshold $\tau \in (d_1 + 2(c\delta + \eta), d_2 - 3(c\delta + \eta))$

the # of clusters computed on n samples $x \overset{iid}{\sim} f$ is equal to # of peaks of f w/ prominence at least d_2 w/ probability $\geq 1 - e^{-\Omega(c(c\delta + \eta)n)}$

$n \to \infty$

Commentary: notion of persistence useful for making qualitative notions of "small peaks" quantitative.

Regularization:

Fairly new topic using PH in conjunction with areas of ML.

Basic premise: topology is good at capturing qualitative information. We'd like to use this to help w/ failure cases in ML.

E.g. wrong # CC in a GAN

How to use PH in optimization?

First, what is optimization?

$$\begin{cases} \underset{x}{\text{minimize}} & f(x) \\ \text{subject to} & c_I(x) \leq 0 \\ & c_E(x) = 0 \end{cases}$$

examples: Regression:

$$\underset{\beta}{\text{minimize}} \ \underbrace{\|X\beta - y\|_2^2}_{f(\beta)} + \underbrace{\underbrace{\lambda_1 \|\beta\|_1}_{\text{Lasso}} + \underbrace{\lambda_2 \|\beta\|_2^2}_{\text{Ridge}}}_{\text{Elastic net}}$$

regularization terms encode "priors"

$\|\cdot\|_1$ encourages sparsity

we'd like to have topological regularizers.

e.g. $\underset{\beta}{\text{minimize}} \; \|X\beta - y\|_2^2 + f(PH(\beta))$

e.g. penalize # maxima
penalize # clusters.

ideas for f: $\sum_{dgm} (d_i - b_i)^2$ // penalize long bars

$\sum_{i \geq k} (d_i - b_i)^2$ // penalize bars after first k-longest.

$\sum_i |d_i - b_i|$ // promote small # bars

$\sum_i \left(\frac{d_i + b_i}{2}\right)^2$ // penalize bars born late

These are all fns similar to features used by Adcock, Carlsson, Carlsson 2016 (can apply to any H_k)

Important question: can we differentiate?

e.g. what is $\nabla_X f(PH_k(X))$

A: yes (or at least subgradients)

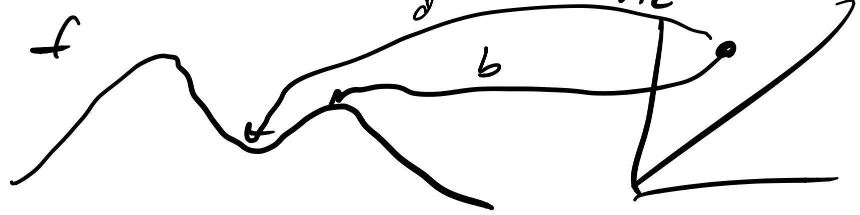
Recall Lemma: let X_t be a filtered complex. The addition of each $x \in X_t$ either creates or destroys homology.

Pf: suppose $\dim(x) = k$. Then effect is to add col. to ∂_k. Two cases:

a) $\partial_k(x) \in \mathrm{span}(\partial_k(X^{(k)} \setminus x)) \Rightarrow$
increase dim ker ∂_k by 1.
$\Rightarrow$ dim H_k = dim ker ∂_k − $\underline{\text{dim img } \partial_{k+1}}$
increases by 1

b) not in span:
increases dim img ∂_k by 1
$\rightarrow H_{k-1}$ = dim ker ∂_{k-1} − dim img ∂_k
decreases by 1.

More importantly, there is a map btw birth + death parameters to individual simplices.
TIE



in lower-star filtrations, can map back to fn on vertices.
in Rips/flag filtrations, can map back to edge parameters.

$$\frac{\partial f}{\partial x} \sum_{\substack{i \in I_k \\ \{(b_i, d_i)\}}} \frac{\partial f}{\partial b_i} \mathbb{1}_{\pi_f(k)(b_i) = x} + \frac{\partial f}{\partial d_i} \mathbb{1}_{\pi_f(k)(d_i) = x}$$

---

why is this a subgradient?
if multiple simplices appear at same filtration value, mapping is not unique.

Brüel-Gabrielsson et al 2020

Chen et al 2019