Lecture 14
clustering possibilities & impossibilities.
Intro to Gaussian Mixture Models

Outline:

big picture

- implications of Kleinberg's impossibility theorem
  - what's "wrong" with ~~partit~~ allowing partitions that refine each other?
  - how should we think about the clustering problem, given the impossibility result?

ended up skipping

- sidebar: proving a subpart of the impossibility theorem.
  reason: ~~the~~ [a souped-up version of] kind of reasoning that can help in A2: creating a dataset that forces a clustering algorithm to output a desired partition
  see previous lecture's notes for proof. (or the actual paper)

- ((Gaussian) mixture) models: ~~can information abo~~
  how does having prior information or assumptions change how we think about clustering?

---

pg 2 of handout: three properties of clustering functions we discussed last time.

not stated explicitly either last lecture or this one, which was a mistake...
placing here as a refresher/reference, altho ~~not~~ ~~show~~ in lecture should be presented when covering the theorems.

Ex: $X = \{1,2,3,4\}$

"clustering" = "partition"
"cluster" = "a bin in a partition"

~~Set of all~~ some ~~possible~~ partitions of X:

$\{1,2,3,4\}$

$\{1\}, \{2,3,4\}$ $\quad$ $\{2\}, \{1,3,4\}$ $\quad$ $\{3\}\{1,2,4\}$ $\quad$ $\{4\}\{1,2,3\}$

$\{1,2\}, \{3,4\}$ $\quad$ $\{1,3\}, \{2,4\}$

~~$\{1\}$ $\{2\}$~~

$\{1\}, \{2\}, \{3,4\}$ $\quad$ $\{1\}, \{3\}, \{2,4\}$ $\quad$ $\{1\}, \{4\}, \{2,3\}$

clicker q rc: "does $\{1\},\{2\},\{3\},\{4\}$ refine $\{1,2,3,4\}$", but phrased informally, was instructive. 75/25 split.

Partition P_2 refines partition P_1 if $\forall$ sets $s_2 \in P_2$, $\exists$ set $s_1 \in P_1$ s.t. $s_1 \subseteq s_2$.

In blue, we've ~~shown~~ let $P_2 = \{1\}, \{2\}, \{3,4\}$ and shown all the P_1's that P_2 refines.

"Refines" = ~~"splits up some~~ "achieved by breaking up some clusters".

Recall: clustering functions f take as input a ~~data~~ distance function d and output a partition of X.

Review the three properties on the handout:

Richness: ~~there is no division of X~~ "there's no clustering/partitioning of X that can't be achieved by f by getting the right d."

or,

"for any partition you might want a priori (w/out looking at the data) there's some distance function d that causes f to output that target."

- so, an "expressiveness" property

Scale-invariance:

~~"the distance func~~ "output shouldn't depend on the units (miles vs. mm) that the distances represent"

or,

"~~you do~~ the output doesn't change if you scale all the distances by the same amount".

- so, a "stability" property

Consistency:

~~"if you change the di~~

~~"suppose you have a d that gives~~

"suppose f produces a partition P when given d. If we alter d so that ~~the~~ each cluster of P gets even more "packed together", ~~and~~ and the clusters of P grow even farther apart from each other, then f doesn't change its output".

- so, a "stability" property.

thm 3.1 + thm 3.2 = (scale-invariance + consistency) ≤ "pseudo-richness"

$\lneq$ richness.

pseudo-rich: possible output sets are the antichains for X.

↳ sets of partitions where no partition refines another.

implications: ~~implication~~

clicker q: Can we have a fn f whose range is an anti-chain A where |A|>1 and X ∈ range(f)?

↑ X = all points in one big cluster

class went ~~70/30~~ ≈ 80/20 on this.

No. Every other ~~partio~~ partition of X refines X.

{1,2}, {3}, {4,5} refines {1,2,3,4,5}.

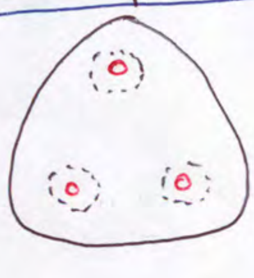
[skip, but for those reading these notes:
You also couldn't have |A|>1 and the partition $P_{singletons}$ where every point is in its own cluster,
because $P_{singletons}$ refines any other partition.]

You can have A ⊆ set of all ~~sized~~ k-partitionings
(2 different partitions of X that both consist of k clusters can't possibly refine each other).

... and hey, that ~~helps~~ coincides with the fixed-k-means & fixed-k single-link algs that are so famous!

(*) Why is refinement such a problem for scale-invariance and consistency?

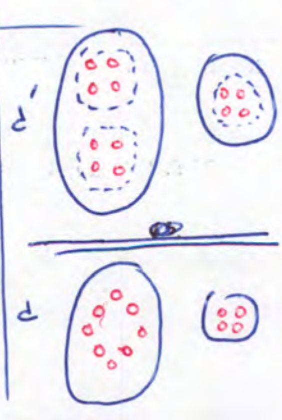
Some examples for intuition.



□ = a partition

⁚⁚ = another partition, refining □.

Whether all-in-one or each-in-own seems to be a judgment about whether the distances are "big" or "little" in an absolute sense.



□ : a given partition

⁚⁚ : another partition, refining □,

whether to ~~pick~~ pick □ or ⁚⁚ seems to depend on whether the top left-hand ~~subcluster~~ distances cause "subclusters" to ~~arise~~ arise.

So, should we just give up on clustering?
Don't think so, it's just too useful a task – it tells you about the structure of your data.

It's just: you either have to:

- give up on some of the properties
- formulate your own properties
- break the formulation

~~xxxxxx~~ ... ~~including~~, ~~asso~~

... [perhaps what the theorem says we need is more information than just a distance fn.]

(an example: ~~what k~~ predetermine k.)
↑ important

abbreviation of "function"

After all, when we are clustering, is our goal really:
"just put points that are close together in the same group" ?
or is it
** determine the underlying structure of the data" **

seems like a more fulfilling thing to do.

perhaps it has become time to consider a big question:

Where does ~~my~~ the data come from?

~~Where~~

When does our data come from? (the birds & bees talk)

What kind of generative story or account can we make up that's reasonable for how clusters underly the data?

ex generative story: parentage ~~phrentage~~ ("the apple doesn't fall far from the tree")

- ~~pick Mother nature selects~~

We have tree species: oaks, maples, and apple trees. ~~Model of population~~ Explain spatial location of the trees.

- Mother Nature picks a species,
- picks a pre-existing tree of that species to be a parent ← this intermediate variable is what distinguishes this from GMMs.
- plants a seed around the locat'n of that parent,

~~position~~

~~likeli~~ prob of ~~location~~ seed's location falls off w/ distance from parent.

ex: generative story: Gaussian sources <Gaussian mixture model>

~~Assume two types~~

We have species, oaks, maples & apple trees.

~~Have features regarding~~

Explain distribution of height, width, leaf lengths, leaf widths. → 4 #s

M.N. picks a species & grows a tree of that species, chooses the height & width values ~~as~~ from ~~appro~~ multivariate Gaussian for that species – i.e., each species has a template (with variance)

→ ~~think about both mixture models~~:

clustering = { given observed data, w/o-t identifying labels (so, you don't get to see the actual trees, just the lat-longs or the height/weight data) (don't see the species!!). "recover the underlying parameters of the "source" model," ~~go~~
– in this case, to figure out which trees belong to the same species.

?

→ ~~our data com~~

But there's a problem with thinking of our quest as to recover "the" source model, and that is: ~~that's~~ it's an impossible problem.

Ex: top ~~figure~~ figure of pg 4:

assume we're dealing with one-dimensional data.

black dots = data, all on one line.

and we tell you that the data was generated by two Gaussians, so one cluster should be the points generated by ~~the leftmost Gaussian~~ one of the Gaussians, and the other cluster should be the points generated by the other Gaussian.

So, one possibility is that we have ~~one Gaussian~~

{ one Gaussian centered under the left bump of the green curve,
{ one Gaussian centered " " right " " " " .

green = resulting combined density fn.

this would be a plausible way the black dots got generated; it does look like there's a rightmost clump of points and a leftmost clump.

But, it really could be that the true source had one Gaussian a mile to the left, and one Gaussian a mile to the right.
<so, both totally off the screen>.

We cannot say that that's not the source model.
it could be that that was the source model, and we just got a really unlikely draw of the data.

So, we ~~can~~ can't say that we can select the true source.

However, if we have to pick some choice of how to decide what model to settle on in order to create output,

then one idea is to use the ~~maximum~~ maximum-likelihood principle (IV on handout)

Assume fixed generating model class, where elements are indexed by parameter setting θ. (a vector)
~~Each possible source~~ P_θ

Ex: Gaussian mixture model w/ 2 Gaussians w/ fixed variances, one chosen w/ fixed prob 2/3, one w/ fixed prob 1/3 ← 1-d

=> possible θ's are the: the means of the two Gaussians.

ex: green ~~one~~ curve: $\theta = (-2, 2)$ ← right mean, ← left mean

resulting densities shown on handout

another possible choice would be our "miles-away" alternative

$$\theta = (-10{,}000,\ 10{,}000)$$

↳ mean of 1st Gaussian → mean of second Gaussian.

~~ML principle~~

~~Say for given dataset~~

↙ do write the English down first.

for given data X_{given}, pick model ~~w/ the θ that maximizes~~ $P(X_{given} \mid \theta)$

$$\hat{\theta}_{ML} = \underset{\theta}{\arg\max}\, P(X_{given} \mid \theta)$$

↖ function of θ.

thus, since the green-curve model thinks X_{given} (on handout, as black dots) is pretty likely,

whereas the X_{given} is very unlikely according to the "miles-away" model,

MLP says to prefer the green-curve model over the miles-away model.

How to find the $\hat{\theta}_{ML}$? ~~"Derivatize"~~: set $\frac{\partial}{\partial\theta[i]}$ ... param.

~~let $\theta = (a, b, c, \ldots)$ for notation's sake~~

let: $\theta = (\theta[1], \theta[2], \ldots \theta[\ell], \ldots)$

"Derivatize": $\frac{\partial}{\partial\theta[\ell]} P(X_{given} \mid \theta)$, set to 0, solve for $\theta[\ell]$.

but, a trick:

- almost always a good idea to use the log of the likelihood instead

(and since log is monotone increasing in its argument, makes no diff. in finding the argmax).

Handout: the log-likelihood function for our two-Gaussian model, 2/3 vs. 1/3 prob:

$\theta = (\mu_1, \mu_2)$ → μ_1 is shown as x axis, μ_2 shown as y axis.

$$p(x \mid \theta) = p(x \mid \mu_1, \mu_2) = \frac{1}{3} \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-\mu_1)^2} + \frac{2}{3} \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-\mu_2)^2}$$

↑ prob of choosing 1st Gaussian

prob of x if generated by 1st Gaussian.

$\log P(x \mid \mu_1, \mu_2)$ is the surface plot.

in orange: the application of an iterative method for trying to do gradient ascent to find a local max of the log-likelihood.

→ there are two trajectories, corresponding to two diff. starting points.

μ_a ; μ_b are the two places this search converged.

↳ on the contour plot below the surface. (in our notation, μ_a would be written θ_a, and μ_b as θ_b)

on the figure above, the densities for θ_a ; θ_b are shown:

θ_a has $\mu_1 \approx -2$, $\mu_2 \approx 2$

θ_b has $\mu_1 \approx -1$, $\mu_2 \approx 2$.

Note: the iterative method mentioned above and whose trajectories are depicted in the handout does not make use of latent variables. It instead uses the hideous equation whose existence is mentioned in the next lecture.