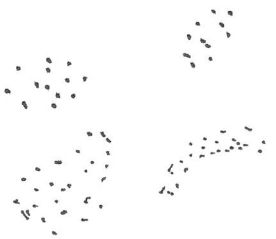
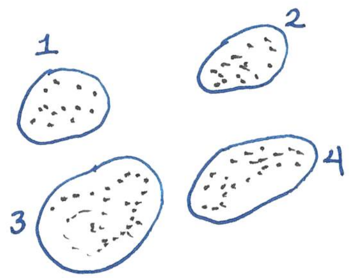
An Energy/Cost function for Clustering.

The idea behind clustering is to group data into distinct groupings such that the data within a group share some common feature or property.

Example: set of 2D points



expected outcome



visual expectation based on proximity of data points to each other

how should we encode mathematically what we intuit visually?

- proximity or nearness equivalent to distance

⇒

perhaps energy should be based on distance metric (or its squared value).

- clustering is equivalent to assigning each data point to a common label.

⇒

should formalize clustering/grouping process as a label assignment problem.

- center of each cluster, or mean value, seems to be a useful point to use.

Formalizing the clustering process to derive algorithms.

Let $\mathcal{X} = \{x_i\}_1^N$ be ~~collection~~ set of N points in $\mathbb{R}^n$, indexed by i.

Let $\mathcal{L} = \{l_i\}_1^N$ be set of N labels taking values from set $\underline{\{1, \ldots, M\}}$, indexed by i.

↑ assignment set, $\mathcal{A}$.

$i \in \mathcal{I} = \{1, \ldots, N\}$

↑ index set

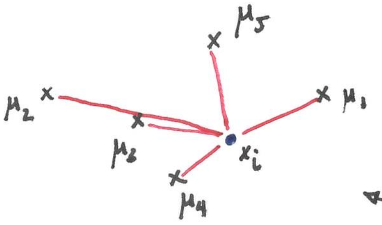
goal: assign label l_i for each x_i.

The problem is somewhat nebulous, so let's start to incorporate our intuitions.

1. define an exemplar or centroidal vector μ_l for each label l in the assigment set $\mathcal{A}$.

   ⇒ get set of means $\mathcal{M} = \{\mu_l\}_1^M$

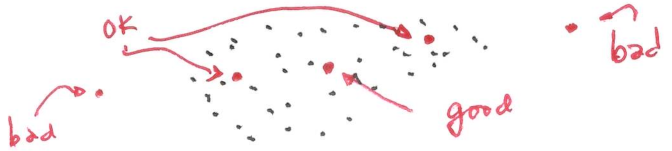
2. fitness of a point x_i relative to the centroidal vector μ_l is determined by distance.



$[\text{dist}(x_i, \mu_l)]^2$ is fitness.

← most fit label is the one that is associated to the closest mean.

3. fitness of an exemplar vector μ_l determined by ~~spread~~ spread of data w/same label around it.



OK, so if we define a labeling of the data, $\mathcal{L}$, then we are implicitly also defining the exemplar.

let $X_\ell = \{ x_i \mid \ell_i = \ell \}$ ← set of x_i such that its label ℓ_i is equal to ℓ.

↑ ℓ takes values from $\mathcal{A} = \{1, \dots, M\}$

then define μ_ℓ = mean ~~value~~ vector of $X_\ell = \frac{1}{n_\ell} \sum_{x \in X_\ell} x$ where n_ℓ = # elements in X_ℓ.

ideally $\mathrm{dist}(x_i, \mu_{\ell_i})$ is small for the chosen label ℓ_i of x_i.

an energy for the assigned cluster ℓ and its resulting set X_ℓ can be:

$$\sum_{x \in X_\ell} \cancel{\mathrm{dist}\,(x_i, \mu} \; \mathrm{dist}^2(x, \mu) = \sum_{x \in X_\ell} \| x - \mu \|$$

we can sum over all possible cluster assignments

$$\mathcal{E}(\mathcal{L}) = \sum_{\ell=1}^{M} \sum_{x \in X_\ell} \| x - \mu_\ell \|^2 = \sum_{i=1}^{N} \| x_i - \mu_{\ell_i} \|^2$$

to get our assignment energy or cost functional.

BRUTE FORCE SOLUTION: Testing all possible label assignments is expensive. There are M^N possible assignments. Clearly trying them all out for large datasets is costly.

A GRADIENT-BASED OR ITERATIVE SOLUTION:

We can't test all options / or don't want to.

So, what about a gradient-based solution? ← well, we need a clean way to derive one.

The stated energy equations are no good

$$\sum_{i=1}^{N} \|x_i - \mu_{l_i}\|^2 = \sum_{l=1}^{M} \sum \|x - \mu_l\|^2$$

Why? i) label assignment is discrete

ii) the mean / exemplar vectors are implicitly computed from the assignment.

try cost function w/ explicit call out of the μ_l vectors.

define (new) cost function that depends on the label assignments $\mathcal{L}$ & the exemplar ~~vectors~~ vectors $\mathcal{M}$, conditioned on the data.

$$\mathcal{E}(\mathcal{M}, \mathcal{L} ; \mathcal{X}) = \sum_{l=1}^{M} \sum_{x \in \mathcal{X}_l} \|x - \mu_l\|^2$$

[where $\mathcal{X}_l = \{ x_i \mid l_i = l \}$

How do we specify & solve as an optimization problem defined on the set of exemplars & the set of label assignments?

Specification of problem: $\underset{(\mathcal{M},\mathcal{L})}{\arg\min} \; \mathcal{E}(\mathcal{M},\mathcal{L};\mathcal{X})$

$\Rightarrow$

$$\underset{(\mathcal{M},\mathcal{L})}{\arg\min} \; \sum_{l=1}^{M} \sum_{x \in \mathcal{X}_l} \|x-\mu_l\|^2$$

problem: $\mathcal{M}$ consists of continuous variables

$\mathcal{L}$ consists of discrete variables

cannot optimize jointly.

must optimize separately, preferably in an alternating fashion

($\mathcal{M}$ values, then $\mathcal{L}$ values, then ...)

or

($\mathcal{L}$ values, then $\mathcal{M}$ values, then ...)

let's derive the strategy for the alternating variables, iterative solver.

① Given ~~mea~~ exemplar set $\mathcal{M}$, optimize label set.

$$\underset{\mathcal{L}}{\arg\min} \; \mathcal{E}(\mathcal{L};\mathcal{M},\mathcal{X})$$

Assertion: with ~~value~~ vectors in $\mathcal{M}$ fixed, the assigment problem becomes decoupled.

$$\cancel{\sum_{l=1}^{M}\sum_{x\in X_l}\|x-\mu_l\|^2 =} \left(\sum_{l=1}^{M}\sum_{x\in X_l\setminus\{x_i\}}\|x-\mu_l\|^2\right) + \|x_i-\mu_{l_i}\|^2$$

$$\sum_{i=1}^{N}\|x_i-\mu_{l_i}\|^2 = \left(\sum_{j=1,\,j\neq i}^{N}\|x_j-\mu_{l_j}\|^2\right) + \|x_i-\mu_{l_i}\|^2$$

① take the i^{th} energy out of the summation

② note that the summation cost is independent of l_i since the i^{th} element (x_i) is explicitly excluded.

⇒ optimizing l_i is independent of l_j for $j\neq i$.

⇒ just optimize each label assignment on its own.

$$\arg\min_{\mathcal{L}} \mathcal{E}(\mathcal{L};\mathcal{M},X) = \cancel{\arg\min \mathcal{E}(\{l_i\}}$$

$$\approx \arg\min \mathcal{E}(l_i;\mathcal{M},x_i) \quad \forall i\in\{1,\dots,N\}$$

optimize one at a time, for all data points so, we just need to solve

$$l_i = \arg\min_{l} \; \| x_i - \mu_{l} \|^2 \qquad \text{where } l = \{1, \dots, M\}$$

this is a discrete optimization problem.

just substitute each μ_l for $l = \{1, \dots, M\}$, record the distance, and pick the l giving the smallest distance.

easy.

② Given a (supposedly) optimal labeling set $\mathcal{L}$, optimize the exemplars

$$\arg\min_{\mathcal{M}} \; \mathcal{E}(\mathcal{M}\,;\, \mathcal{L}, \mathcal{X})$$

Assertion: with the labels fixed, the exemplar optimization becomes decoupled by label

$$\sum_{l=1}^{M} \sum_{x \in \mathcal{X}_l} \|x - \mu_l\|^2 = \underbrace{\left(\sum_{l' \neq l,\, l'=1}^{M} \; \sum_{x \in \mathcal{X}_{l'}} \|x - \mu_{l'}\|^2 \right)}_{\text{does not involve any labels equal to } l.} + \underbrace{\sum_{x \in \mathcal{X}_l} \|x - \mu_l\|^2}_{\text{depends on labels equal to } l.}$$

Solve for μ_ℓ by solving for criticality of the energy:

$$\frac{\partial \mathcal{E}}{\partial \mu_\ell} \cdot \delta\mu_\ell = 0 \qquad \forall\ \delta\mu_\ell \in \mathbb{R}^n$$

$\Rightarrow$

$$\frac{\partial \mathcal{E}}{\partial \mu_\ell} = 0$$

$$\frac{\partial \mathcal{E}}{\partial \mu_\ell} = \frac{\partial}{\partial \mu_\ell}\left[\cancelto{0}{\left(\sum_{\ell'\neq\ell,\,\ell'=1}^{M}\ \sum_{x\in X_\ell}\|x-\mu_{\ell'}\|^2\right)} + \underbrace{\sum_{x\in X_\ell}\|x-\mu_\ell\|^2}_{\text{need differential.}}\right]$$

independent of μ_ℓ
so vanishes under
differentiation by μ_ℓ

$$= 0 + \frac{\partial}{\partial \mu_\ell}\sum_{x\in X_\ell}\|x-\mu_\ell\|^2$$

$$= \frac{\partial}{\partial \mu_\ell}\sum_{x\in X_\ell}(x-\mu_\ell)^T(x-\mu_\ell)$$

$$\frac{\partial \mathcal{E}}{\partial \mu_\ell} = 2\sum_{x\in X_\ell}(x-\mu_\ell)$$

$\Rightarrow$ enforce criticality

$$2\sum_{x\in X_\ell}(x-\mu_\ell) = 0$$

$\Rightarrow$

$$\cancel{2}\sum_{x\in X_\ell} x = \cancel{2}\sum_{x\in X_\ell} \mu_\ell$$

recall $n_\ell = |X_\ell| =$ cardinality of X_ℓ.

$\uparrow$

Constant for all $x \in X_\ell$
so just a straight summation.

$\Rightarrow$

$$n_\ell \cdot \mu_\ell = \sum_{x\in X_\ell} x$$

$\Rightarrow$

$$\mu_\ell = \frac{1}{n_\ell}\sum_{x\in X_\ell} x$$ = mean of the data in X_ℓ

= mean of all data w/the label ℓ.

so, for each label ℓ, set the ~~mean~~ exemplar vector equal to the mean value of all data w/that label.

easy.

so, optimization iterates between

① optimize assignments given ~~the~~ exemplar vectors.

② optimize exemplar vectors given assignments.

initial condition / guess is the set of exemplar vectors. (don't need to guess labels)

Because the solution involves setting the exemplar vectors to the mean vectors for the label sets, the algorithm is called k-means.

The k stands for the # of exemplars to use.

(equivalent to the variable M used in the setup).

WHAT IS THE TIME COST?

step 1 compares each point's distances to M exemplars ⇒ M·N operations

step 2 computes the means using the data. ⇒ N operations.
each set is disjoint

~~the number~~
⇒

$$\Theta(MN + N) = \underbrace{\Theta(MN)}_{\text{rough cost}} \text{ operations per iteration.}$$

let n_i = # of iterations

let n = dimension of vector data

⇒

detailed cost $\Theta(n_i \cdot M \cdot N \cdot \underset{\uparrow}{n})$

if dimension is large, then n can be problematic.

better than $\Theta(M^N)$ of brute force method.

DOES IT WORK?

There is a proof that k-means will converge to a critical value. That's good.

Bad news: not guaranteed to be global critical value. only a local optima.

The set of possibilities is huge M^N, so we should not expect impressive performance. Nevertheless, it works well enough to be a go-to solution for many procedures that elect to have a clustering step involving a known quantity of clusters.

WHAT ABOUT THE VALUE k?

Oh, that's a tough one. Totally problem dependent.

Usually the geometry / distribution of data will be such that bigger k is better. Idea being to over cluster the data, then have downstream processes collect outcomes so that # clusters reduces if a specific set of clusters is required.

Many times, the way k-means is to be used, one will naturally select a larger k than # of clusters.

Only for very specific data arrangements & guesses can k-means get the chosen K clusters exactly.
(minimal)