DMML, 6 Feb 2020

Naive Bayes text classification

- Boolean document model

  $d \in D$ is a set of words over V

  $|V| = n \rightarrow d$ is a $\{0,1\}$ vector of length n

- Multiset model - "Bag" of words model

  $d \in D$ - has a length - count occurrences of each word

Set of words

- Pick c with $P(c)$
- For each $w \in V$, include w with $P(w|c)$

$$\sum_w P(w|c) = 1$$

Bag of words

- Parameters are same: $P(c)$, $P(w|c)$
- $P(\ell)$ – length of document – assume indep of c
- Pick c with $P(c)$, Pick ℓ with $P(\ell)$

  For i in 1 to ℓ, generate w_i with $P(w_i|c)$
  - $|V|$-sided die

$d = w_1 w_2 \cdots w_{|d|}$

$$P(d_k|c_j) = P(|d|)\,|d|!\prod_{i=1}^{|d|}\frac{P(w_i|c_j)}{N_{ik}!} \cdot \prod_{w_i \in V}\frac{P(w_i|c_j)^{N_{ik}}}{N_{ik}!}$$

N_{ik} = # of times w_i appears in d_k

Want $P(c_j|d_k)$

What is $P(w_i|c_j)$ = $\dfrac{\#\text{ of occurrences of } w_i \text{ in } c_j}{\text{total \# words in } c_j}$

Recall N_{ik} = # of times w_i appears in d_k

1 if $d_k \in c_j$
0 if $d_k \notin c_j$

$$P(w_i|c_j) = \frac{\sum_{d_k \in c_j} N_{ik}}{\sum_{w_t \in V} \sum_{d_k \in c_j} N_{tk}} = \frac{\sum_{d_k \in D} N_{ik} \cdot P(c_j|d_k)}{\sum_{w_t \in V} \sum_{d_k \in D} N_{tk} P(c_j|d_k)}$$

$$P(c_j|d_k) = \frac{P(d_k|c_j) \cdot P(c_j)}{P(d_k)}$$

$$P(d_k) = \sum_{c_m} P(c_m) \cdot P(d_k|c_m)$$

$$P(|d|)\,|d|! \prod_{w_i \in V} \frac{P(w_i|c_j)^{N_{ik}}}{N_{ik}!}$$

$$\sum_{c} P(c)\, P(d_k|c)\, P(|d|) \cdot |d|! \prod_{w_i} P(w_i|c)$$

Unsupervised Learning

"Find patterns" without training data

Typically - find groups of related items

- Market segmentation

Notion of similarity between data items

- Alternatively - define distance between items

Simplest case - numerical data, $\sqrt{\Delta x_1^2 + \Delta x_2^2 + \dots \Delta x_n^2}$

Euclidean distance

Given distances - identify groups — clusters

Top down clustering — divide population into clusters

Bottom up — combine nearby clusters to form larger ones

Top down — how many clusters?

Fix the number of clusters in advance — K

Target: Separate data items into K clusters in "best possible" way

How to describe a cluster?

Thresholds for each attribute – how to find them?

Represent a cluster by an "average" point

– mean in each attribute

– geometrically – centroid

K centroids

– Each data item maps to nearest centroid

Achieve this iteratively

K-Means Algorithm

K-Mean = K-Centroid

Initially, choose K random centroids $c_1, \ldots, c_k$

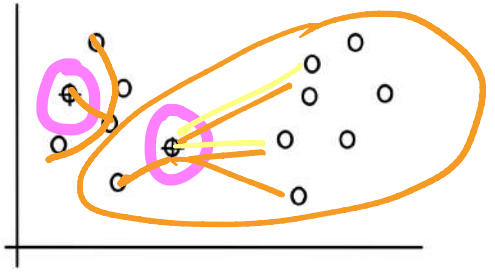
For each data item d_j,

assign d_j to nearest c_m

After one pass → $C_1, C_2, \ldots, C_k$ – clusters

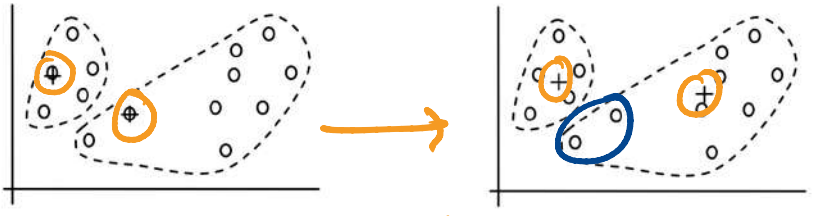
Recompute centroids of $C_1, C_2, \ldots, C_k$ – $c_1', c_2', \ldots, c_k'$

Ideally, converge to stable set $c_1, \ldots, c_k$



(A). Random selection of k seeds (or centroids)

Useful to pick random centroids from among existing data points



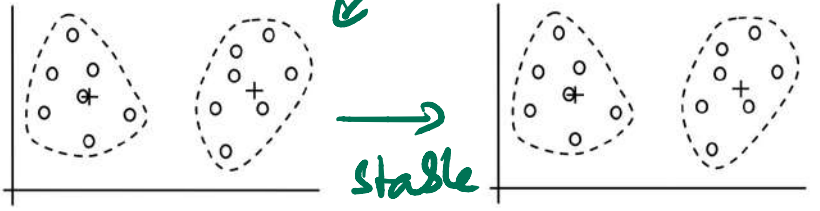
Iteration 1: (B). Cluster assignment

(C). Re-compute centroids

Iteration 2: (D). Cluster assignment

same

(E). Re-compute centroids



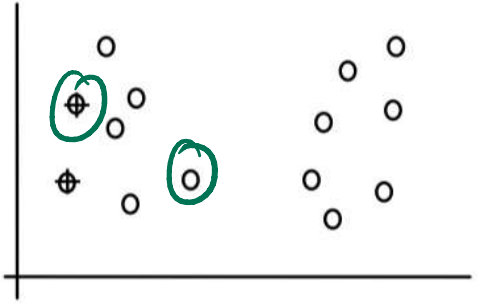
Iteration 3: (F). Cluster assignment

stable

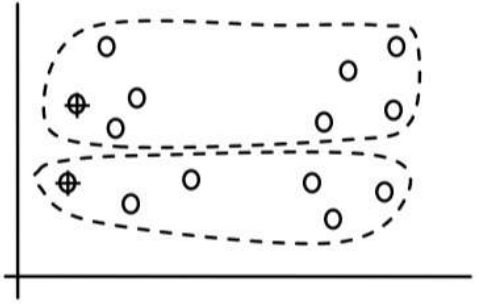
(G). Re-compute centroids

Choice of initial centroids

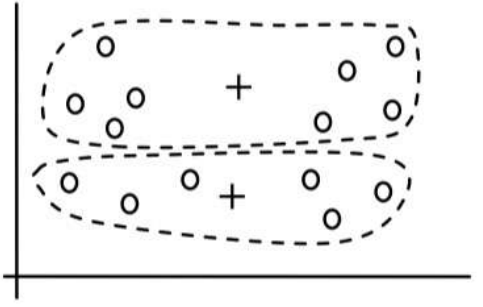
How to fix?



(A). Random selection of seeds (centroids)

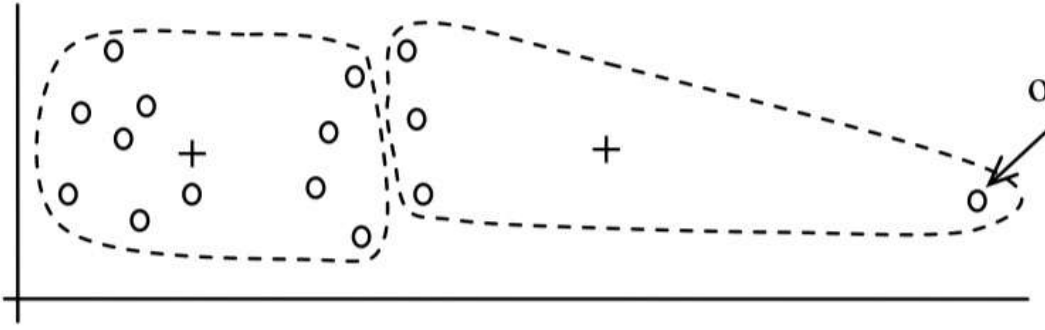


(B). Iteration 1



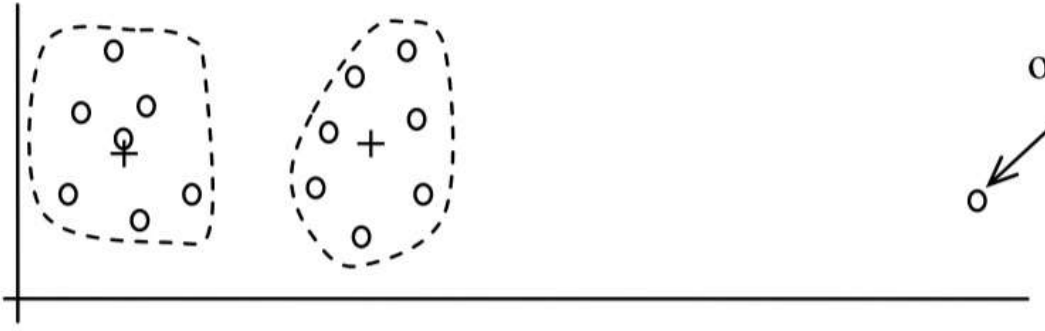
(C). Iteration 2

Outliers



(A): Undesirable clusters

Identifying outliers



(B): Ideal clusters