Affinity Propagation

Egor Malkov

University of Minnesota
FRB Minneapolis

Machine Learning and Big Data Workshop
November 2, 2020

General Overview

Affinity Propagation (AP) was proposed by Frey and Dueck (2007).

General idea:

1. AP takes real-valued dissimilarity measures between pairs of data points as input.
2. Real-valued messages are exchanged between data points until a high-quality set of *exemplars* and corresponding clusters gradually emerges.

AP simultaneously considers all data points as potential exemplars & no need to specify the number of clusters beforehand.

- K-means/K-medoids is quite sensitive to the initial selection of exemplars.
- K-means/K-medoids requires to specify K.

AP can take unusual measures of dissimilarity as input.

- K-means requires distances as input.

AP selects clusters with much lower error and much faster than K-medoids.

- Vlasblom and Wodak (2009): Markov clustering works better than AP on protein interaction graph partitioning.

Algorithm Components

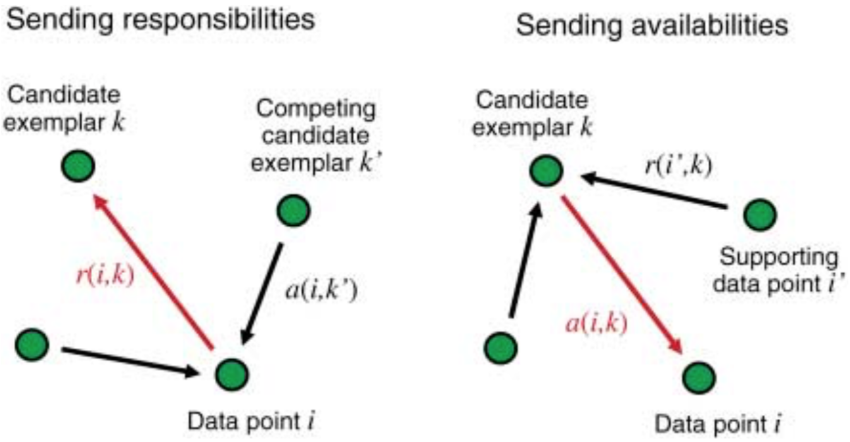
Input: Similarity $s(i,k)$.

- $s(i,j) > s(i,k)$ means *"x_i is more similar to x_j than to x_k"*.
- Euclidean distance: $s(i,k) = -||x_i - x_k||^2$. *Criterion can be more general!*
- "Preferences" $s(k,k)$: Larger values $\rightarrow$ more likely to be chosen as exemplars.
- How to choose a common value for $s(k,k)$ for all k? Median, minimum over $s(i,k)$...

Message Passing

- "Responsibility" $r(i,k)$: How well-suited point k is to serve as the exemplar for point i, taking into account other potential exemplars for point i.
- "Availability"" $a(i,k)$: How appropriate it would be for point i to choose point k as its exemplar, taking into account other points' preference for point k as an exemplar.
- View $r(i,k)$ and $a(i,k)$ as log-probability ratios.
- Combine $r(i,k)$ and $a(i,k)$ to monitor the exemplar decisions (*algorithm termination*).
- To avoid oscillation, add noise to the similarities or use damping factor.

Message Passing



Source: Frey and Dueck (2007).

Algorithm

1. To initialize, set all availabilities to zero, $a(i,k) = 0$.
2. Iterate until either the cluster boundaries remain unchanged over a number of iterations, or some # of iterations is reached:
   2.1 Update responsibility: $\underbrace{r(i,k)}_{\text{“responsibility”}} \leftarrow s(i,k) - \max_{k' \neq k}\{\underbrace{a(i,k')}_{\text{“availability”}} + s(i,k')\}$.
   2.2 Update availability:

$$a(i,k) \leftarrow \min\left\{0, r(k,k) + \sum_{i' \notin \{i,k\}} \max\{0, r(i',k)\}\right\}, \quad \text{for } i \neq k$$

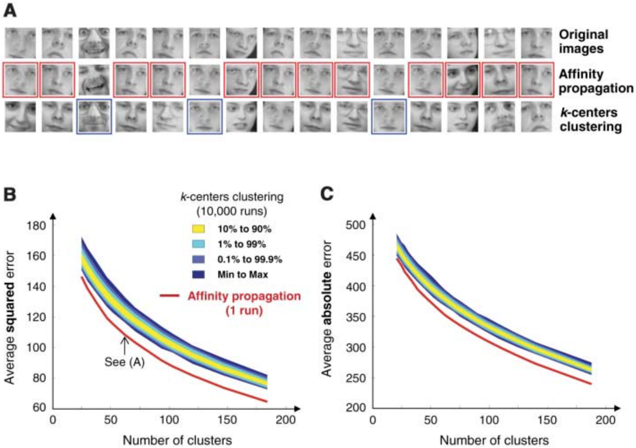
$$a(k,k) \leftarrow \sum_{i' \neq k} \max\{0, r(i',k)\}$$

Positive $r(i,k)$ means that k is a good exemplar to explain i.

When point i is effectively assigned to exemplar k', then $a(i,k)$ is negative.

For point i, the value of k that maximizes $a(i,k) + r(i,k)$ either identifies point i as an exemplar if $k = i$, or identifies the data point that is the exemplar for point i.

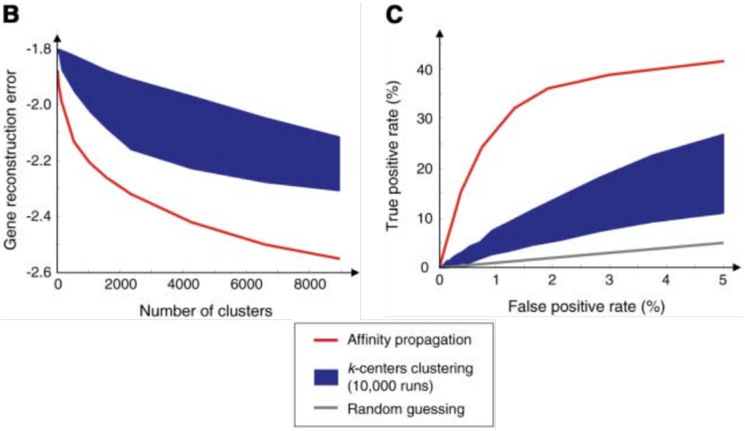
Example: Face Recognition



Source: Frey and Dueck (2007).

Main lesson: Affinity propagation works better and faster than K-centers clustering.

Example: Genes Detection



Source: Frey and Dueck (2007).

Again, affinity propagation works better and faster *(6 minutes vs. 208 hours)*.