Clustering by Passing Messages Between Data Points

Aldana Zarate

Basado en el paper Clustering by Passing Messages Between Data Points por Brendan J. Frey y Delbert Dueck

Motivación

Durante el curso se vieron métodos de clustering divisivos:

- K-means: el objetivo es encontrar una partición de los datos en k grupos, tal que la distancia media dentro de los puntos de cada grupo sea mínima.

Motivación

Durante el curso se vieron métodos de clustering divisivos:

- K-means: el objetivo es encontrar una partición de los datos en k grupos, tal que la distancia media dentro de los puntos de cada grupo sea mínima.
- Problemas:
  - El punto inicial de búsqueda al azar lleva a un mínimo local. Se puede remediar con múltiples corridas, pero ¿Qué pasa si el número de clusters no es pequeño y el azar no acompaña?

Motivación

Durante el curso se vieron métodos de clustering divisivos:

- K-means: el objetivo es encontrar una partición de los datos en k grupos, tal que la distancia media dentro de los puntos de cada grupo sea mínima.
- Problemas:
  - El punto inicial de búsqueda al azar lleva a un mínimo local. Se puede remediar con múltiples corridas, pero ¿Qué pasa si el número de clusters no es pequeño y el azar no acompaña?
  - Depende fuertemente de los outliers al usar la media

Motivación

Durante el curso se vieron métodos de clustering divisivos:

- K-means: el objetivo es encontrar una partición de los datos en k grupos, tal que la distancia media dentro de los puntos de cada grupo sea mínima.
- Problemas:
  - El punto inicial de búsqueda al azar lleva a un mínimo local. Se puede remediar con múltiples corridas, pero ¿Qué pasa si el número de clusters no es pequeño y el azar no acompaña?
  - Depende fuertemente de los outliers al usar la media
  - Solo vale en espacios vectoriales

Resumen del método

- Approach diferente: todos los puntos son potenciales centros de clusters y se transmiten iterativamente mensajes hasta que un conjunto aceptable de centros y sus clusters emerge

Resumen del método

- Approach diferente: todos los puntos son potenciales centros de clusters y se transmiten iterativamente mensajes hasta que un conjunto aceptable de centros y sus clusters emerge
- En todo momento, la magnitud de cada mensaje refleja la afinidad actual que tiene un punto para elegir otro como su centro de cluster.

Resumen del método

- Approach diferente: todos los puntos son potenciales centros de clusters y se transmiten iterativamente mensajes hasta que un conjunto aceptable de centros y sus clusters emerge
- En todo momento, la magnitud de cada mensaje refleja la afinidad actual que tiene un punto para elegir otro como su centro de cluster.
- Este método se llama "Affinity propagation"

Conceptos de propagación de afinidad

- **Similaridades** entre los puntos (input): s(i,k) manifiesta que tan indicado es que el punto k sea centro de i. (Se puede usar el criterio estándar de distancia euclidea ¡y más!)

Conceptos de propagación de afinidad

- **Similaridades** entre los puntos (input): s(i,k) manifiesta que tan indicado es que el punto k sea centro de i. (Se puede usar el criterio estándar de distancia euclidea ¡y más!)
- **Preferencias de centros:** caso particular s(k,k). No se necesita saber la cantidad de clusters!

Conceptos de propagación de afinidad

- Hay 2 tipos de mensajes intercambiados entre los puntos:
  - La **responsabilidad** r(i,k). Refleja la evidencia acumulada de qué tan buen candidato es k para ser centro de i, teniendo en cuenta otros candidatos a centro para i.

Conceptos de propagación de afinidad

- Hay 2 tipos de mensajes intercambiados entre los puntos:
  - La **responsabilidad** r(i,k). Refleja la evidencia acumulada de qué tan buen candidato es k para ser centro de i, teniendo en cuenta otros candidatos a centro para i.
  - La **disponibilidad** a(i,k). Refleja la evidencia acumulada de qué tan apropiado sería que el punto i elija a k como su centro, teniendo en cuenta el apoyo de otros puntos de que k sea su centro.

Algoritmo

- $a(i, k) = 0$ para todos inicialmente

Algoritmo

- $a(i,k) = 0$ para todos inicialmente
- $r(i,k) = s(i,k) - max\{a(i,k') + s(i,k')\} \quad \forall k' \neq k$

Algoritmo

- $a(i,k) = 0$ para todos inicialmente
- $r(i,k) = s(i,k) - max\{a(i,k') + s(i,k')\} \quad \forall k' \neq k$
- $a(i,k) = min\{0, r(k,k) + \sum_{i'/i' \notin \{i,k\}} max\{0, r(i',k)\}\}$

Algoritmo

- $a(i,k) = 0$ para todos inicialmente
- $r(i,k) = s(i,k) - max\{a(i,k') + s(i,k')\} \quad \forall k' \neq k$
- $a(i,k) = min\{0, r(k,k) + \sum_{i'/i' \notin \{i,k\}} max\{0, r(i',k)\}\}$
- En cualquier punto durante AP, disponibilidades y responsabilidades pueden ser combinadas para identificar centros.
  - Para el punto i, el valor de k que maximiza a(i,k) + r(i,k) identifica al punto i como un centro si k=i, o identifica el punto centro para el punto i.

Algoritmo

- $a(i,k) = 0$ para todos inicialmente
- $r(i,k) = s(i,k) - max\{a(i,k') + s(i,k')\} \quad \forall k' \neq k$
- $a(i,k) = min\{0, r(k,k) + \sum_{i'/i' \notin \{i,k\}} max\{0, r(i',k)\}\}$
- En cualquier punto durante AP, disponibilidades y responsabilidades pueden ser combinadas para identificar centros.
  - Para el punto i, el valor de k que maximiza a(i,k) + r(i,k) identifica al punto i como un centro si k=i, o identifica el punto centro para el punto i.
- Criterio de parada
  - Nro fijo iteraciones
  - Cambios en los mensajes caigan debajo de un umbral
  - que las decisiones del item anterior se mantengan constantes por algún número de iteraciones.

Actualizacion de la responsabilidad

$$r(i,k) = s(i,k) - max\{a(i,k') + s(i,k')\} \quad \forall k' \neq k$$

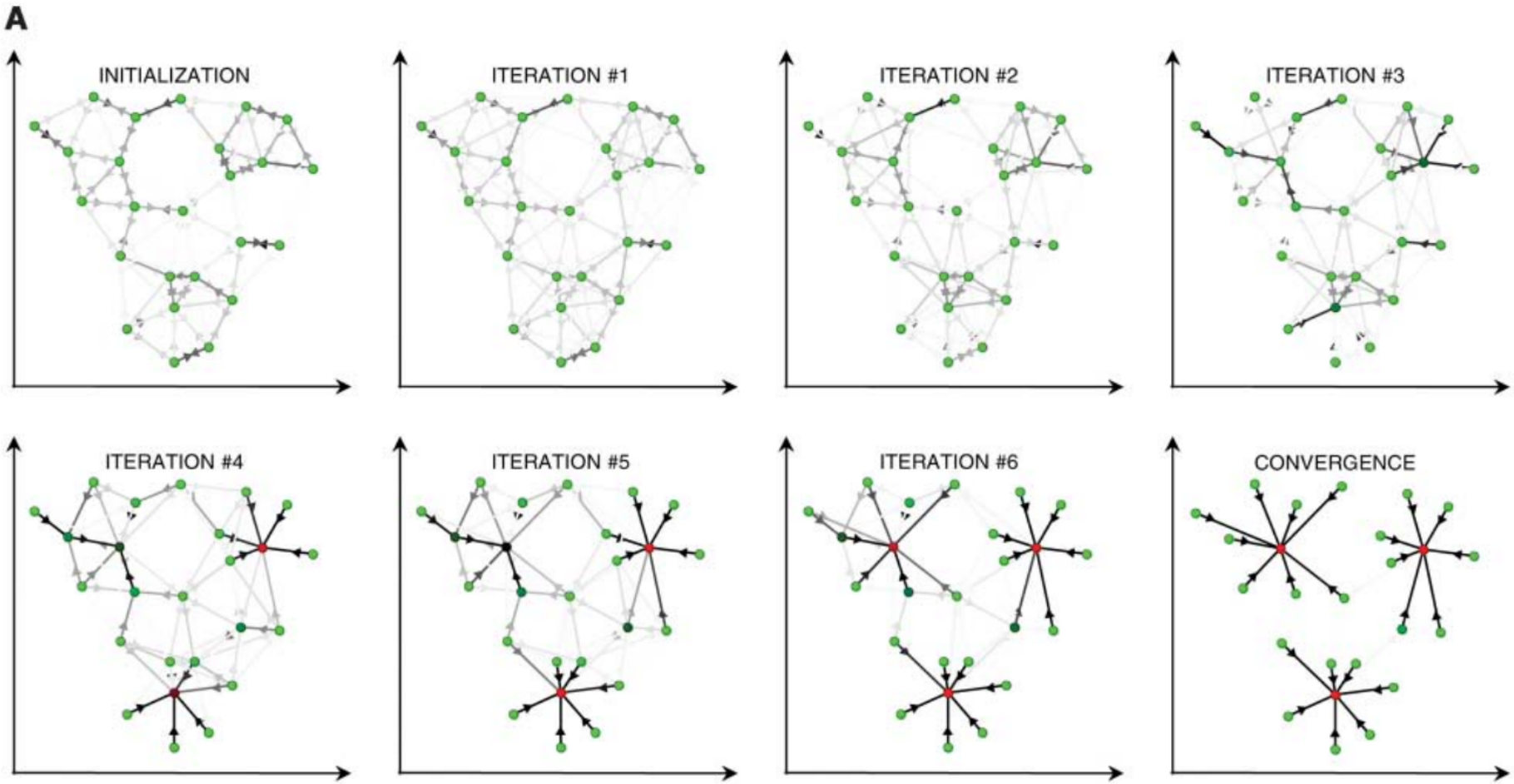
- Actualización guiada por los datos en la primera iteración
- Disponibilidades negativas elimina candidatos de la contienda
- Caso i=k, "responsabilidad propia" de moderar la preferencia segun la posibilidad de ser asignado a otro candidato a centro.

Actualizacion de la disponibilidad

$$a(i,k) = min\{0, r(k,k) + \sum_{i'/i' \notin \{i,k\}} max\{0, r(i',k)\}\}$$

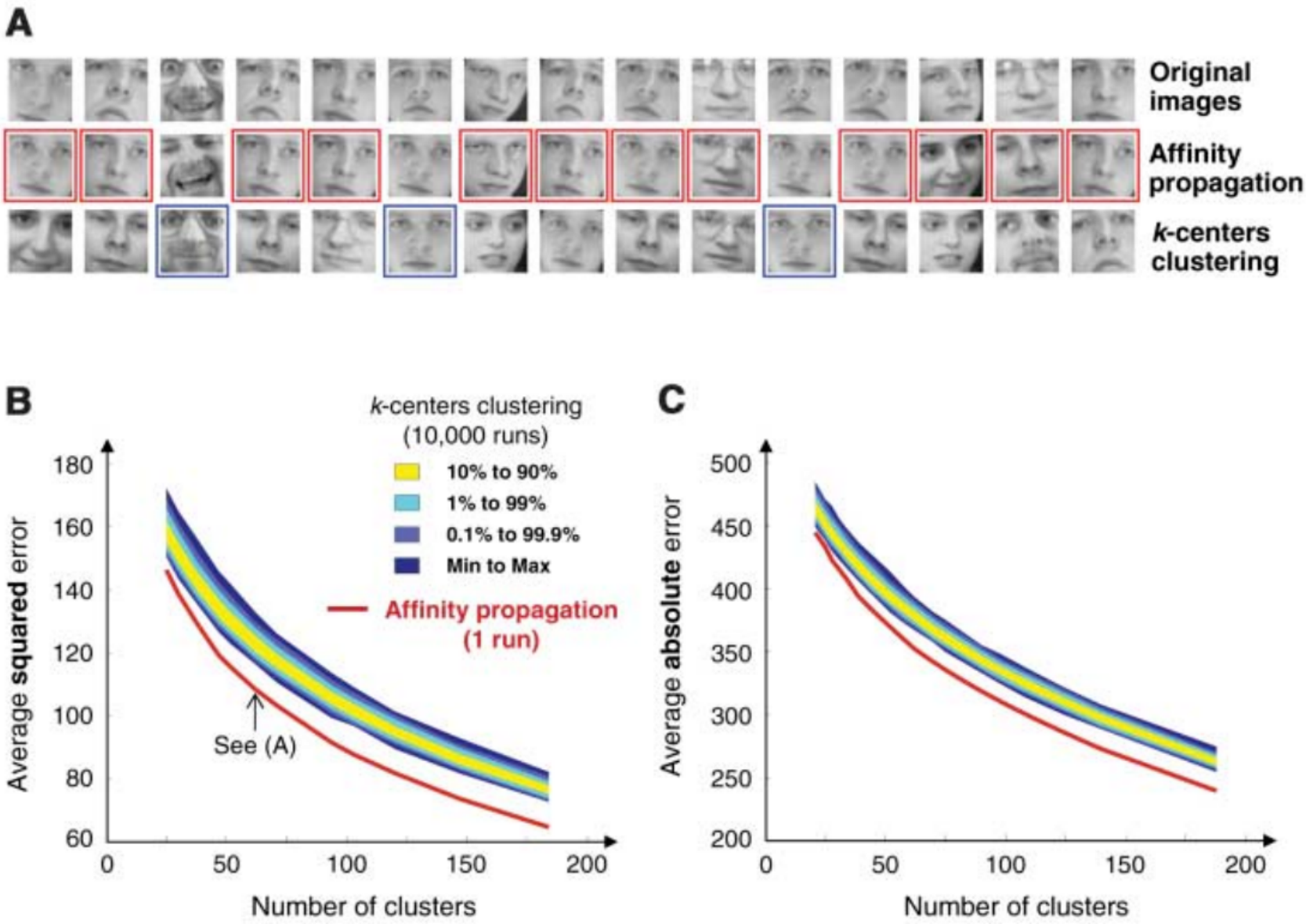
- Solo las responsabilidades positivas entrantes son sumadas
- Caso r(k,k) < 0, necesidad de limitar influencia de responsabilidades entrantes fuertes

Dinámica del algoritmo: Ejemplo

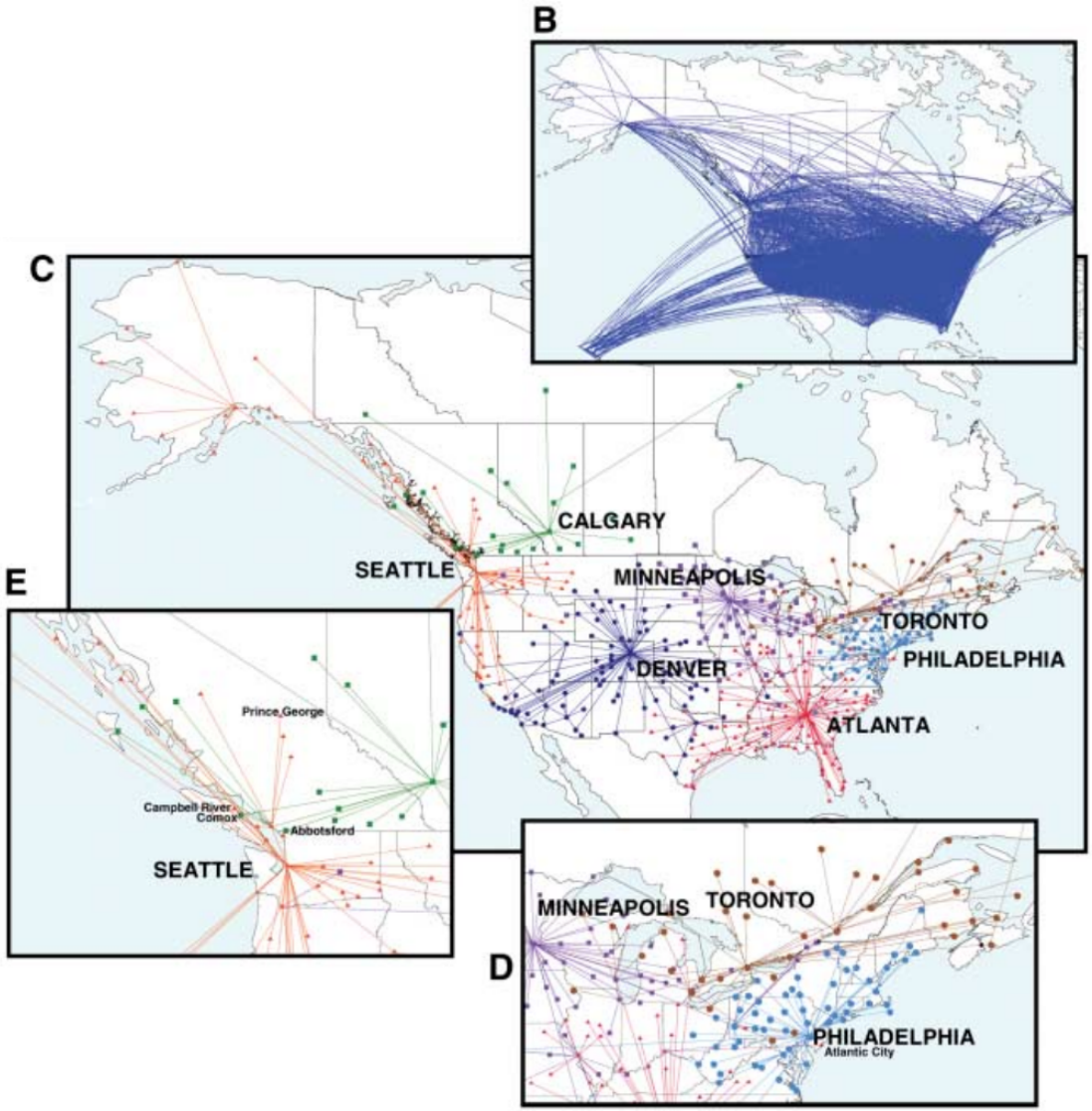


1. How affinity propagation works. ffinity propagation is illustrated for

Resultados que apoyan al método: Clustering de imágenes



Resultados que apoyan al método: Clustering de ciudades



Resultados que apoyan al método: Clustering de ciudades

- s(i,k) = - tiempo que toma viajar de la ciudad i a la ciudad k en avión, teniendo en cuenta paradas.
- Particularidades:
  - Debido al viento, muchas similaridades fueron no fueron simétricas ($s(i,k) \neq s(k,i)$).
  - En muchos casos no se cumplió la desigualdad triangular $s(i,k) < s(i,j) + s(j,k)$, ya que para ir de i a k había que hacer una parada larga en la ciudad j, con lo cual toma más tiempo que la suma de ir de i a j y de j a k.

Conclusiones

- Al considerar simultáneamente todos los puntos como candidatos a centros e ir gradualmente identificando los clusters, AP puede mitigar algunos de los problemas planteados.
- No es necesario brindar una cantidad de clusters por la naturaleza del algoritmo.
- AP puede tomar como input similaridades no métricas (negativas, no simétricas o que no satisfagan la desigualdad triangular)
- Tiene una performance ampliamente superior, *al menos en los casos de uso ilustrados en el paper presentado*