Aprendizaje no supervisado

Lección 18

Dr. Pablo Alvarado Moya

CE5506 Introducción al reconocimiento de patrones
Área de Ingeniería en Computadores
Tecnológico de Costa Rica

II Semestre, 2019

Contenido

Motivación

- Aprendizaje **supervisado**: tenemos tanto datos de entrada $\underline{\mathbf{x}}^{(i)} \in \mathbb{R}^n$ como etiquetas $y^{(i)}$ correspondientes.
- En un problema de **regresión** el algoritmo tiene que aprender a predecir un valor real a la salida
- En un problema de **clasificación** el algoritmo tiene que aprender a asignar una etiqueta discreta a cada entrada.
- En aprendizaje **no supervisado** el objetivo es descubrir *estructura* en un conjunto no etiquetado de datos $\underline{\mathbf{x}}^{(i)} \in \mathbb{R}^n$, **sin** contar con las etiquetas $y^{(i)}$.
- *Estructura* puede ser:
  - Conglomerados por distancia, similitud, densidad, ...
  - Distribución probabilística subyacente

Aplicaciones

- Fase inicial en minería de datos: agrupar datos similares
- En biología: se agrupan genes, neuronas, proteínas, etc.
  - Ejemplo: agrupación de neuronas de *Drosophila melanogaster*
- Análisis de mercado, para identificar segmentos
  - Agrupar clientes de acuerdo a su historia de compra
  - Agrupar productos basados en los clientes que los compran
- Análisis de información (news.google.com)
- Segmentación de imágenes

Aglomeración
Clustering

- Dado un conjunto de puntos $\{\underline{\mathbf{x}}^{(i)} | i = 1, \ldots, m\}$, el objetivo de **aglomeración** es encontrar un conjunto de conglomerados coherentes:
    - Puntos en un mismo conglomerado son similares entre sí
    - Puntos en conglomerados distintos son disímiles
- Puntos $\underline{\mathbf{x}}^{(i)} \in \mathbb{R}^n$.
- Concepto central es el de **distancia** entre puntos
    - Distancias de Minkowski (Euclídea, Manhattan, Máxima, etc.)
    - Distancias genéticas (biología)
    - Distancias de texto
    - Distancias probabilísticas
    - ...

Estrategias de aglomeración

Existen varias estrategias de aglomeración:

1. Basada en particiones
   - *k*-medias
2. Basada en densidad
   - DBSCAN
3. Basada en jerarquías
4. Basada en distribuciones
5. Basada en teoría de grafos
6. Basada en modelos
7. ...

k-medias

Algoritmo *k*-medias
k-Means

Inicialice los centroides $\underline{\boldsymbol{\mu}}_1, \underline{\boldsymbol{\mu}}_2, \ldots, \underline{\boldsymbol{\mu}}_k \in \mathbb{R}^n$

repeat

  foreach *dato* $i = 1, \ldots, m$ **do**

    $c^{(i)} := \arg\min_j d(\underline{\mathbf{x}}^{(i)}, \underline{\boldsymbol{\mu}}_j)$

  end

  foreach *conglomerado* $j = 1, \ldots, k$ **do**

$$\underline{\boldsymbol{\mu}}_j := \frac{\sum_{i=1}^{m} 1\left\{c^{(i)} = j\right\} \underline{\mathbf{x}}^{(i)}}{\sum_{i=1}^{m} 1\left\{c^{(i)} = j\right\}}$$

  end

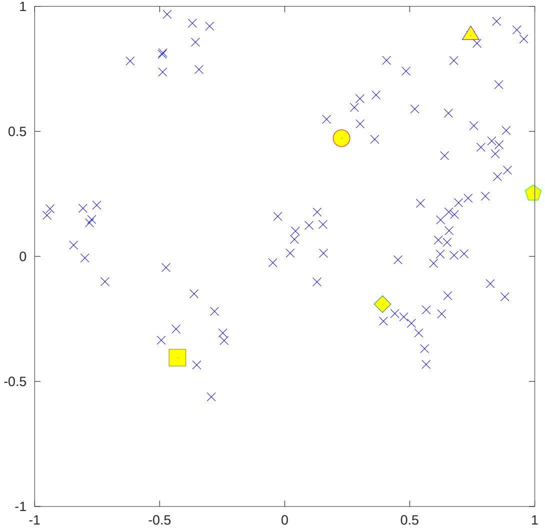
until *convergencia*

Características de las k-medias (1)

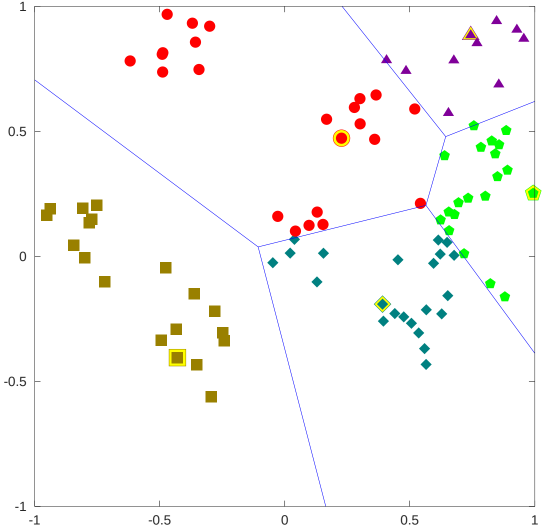
- Existen varias estrategias de inicialización de los $\underline{\boldsymbol{\mu}}_j$
  - Completamente aleatoria
  - Selección aleatoria de k de los datos
  - Algoritmo `kmeans++`
  - ...
- Hay dos pasos fundamentales:
  1. Asignación de los puntos a un conglomerado
  2. Corrección de los centroides
- Distancia $d(\underline{\mathbf{x}}^{(i)}, \underline{\boldsymbol{\mu}}_j)$ depende de aplicación

  Usualmente se usa $d(\underline{\mathbf{x}}^{(i)}, \underline{\boldsymbol{\mu}}_j) = \|\underline{\mathbf{x}}^{(i)} - \underline{\boldsymbol{\mu}}_j\|_2^2$ (euclídea)
- Criterio de convergencia: número de cambios en $c^{(i)} < \tau$

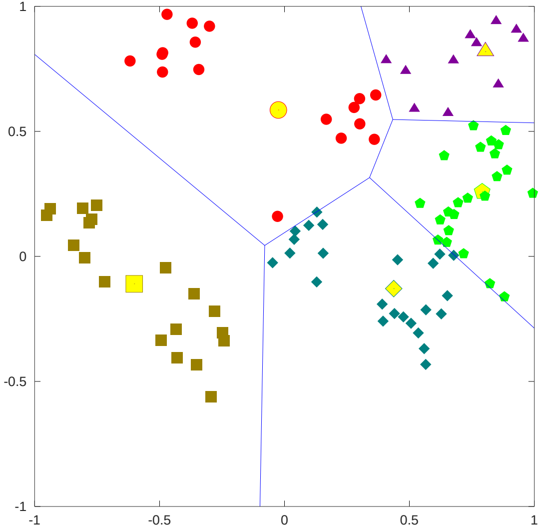
Ejemplo de *k*-medias



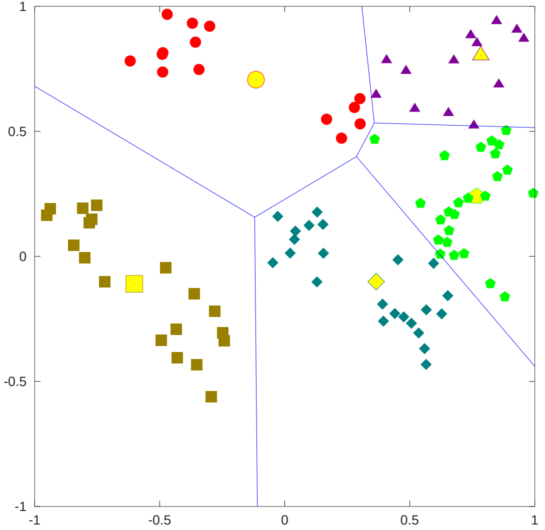
Ejemplo de *k*-medias



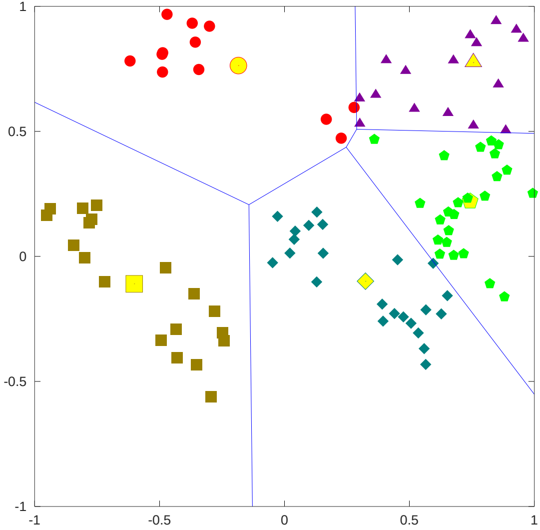
Ejemplo de *k*-medias



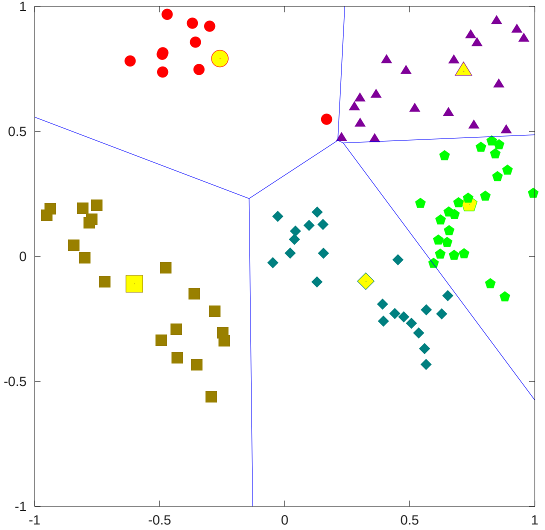
Ejemplo de *k*-medias



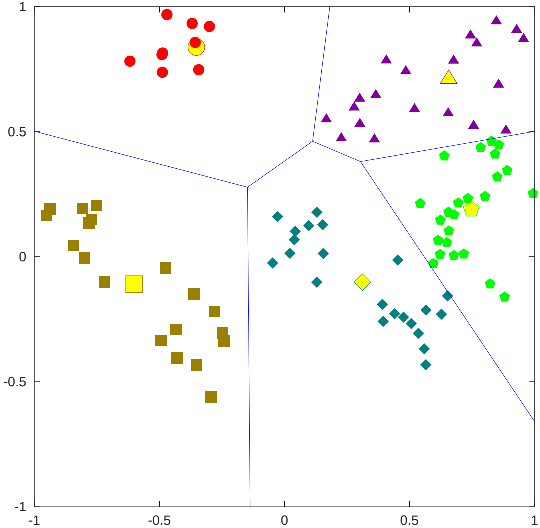
Ejemplo de *k*-medias



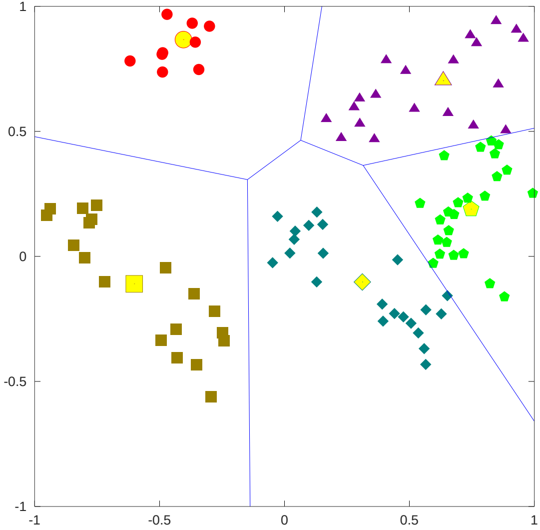
Ejemplo de *k*-medias



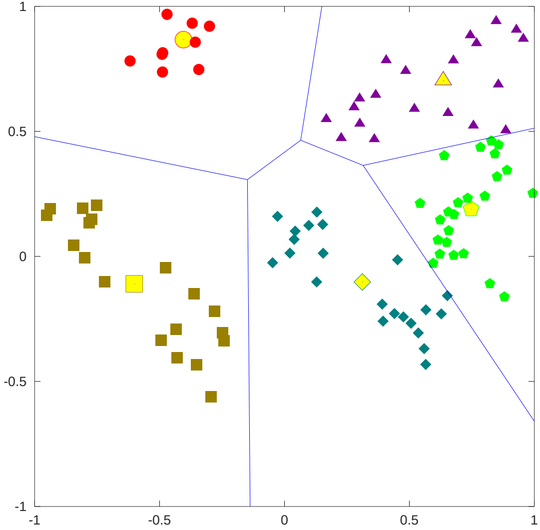
Ejemplo de *k*-medias



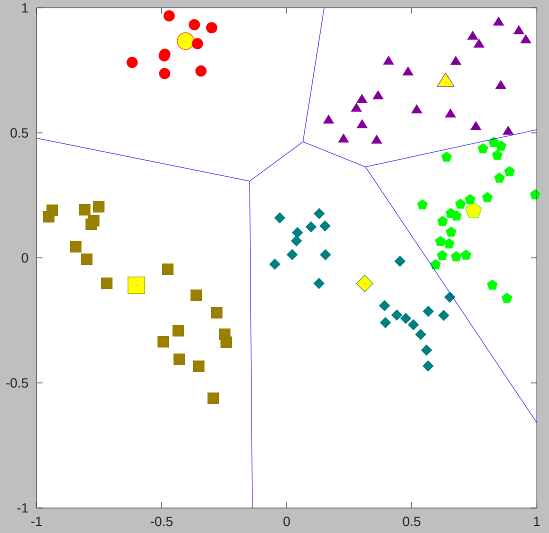
Ejemplo de *k*-medias



Ejemplo de *k*-medias



Ejemplo de *k*-medias

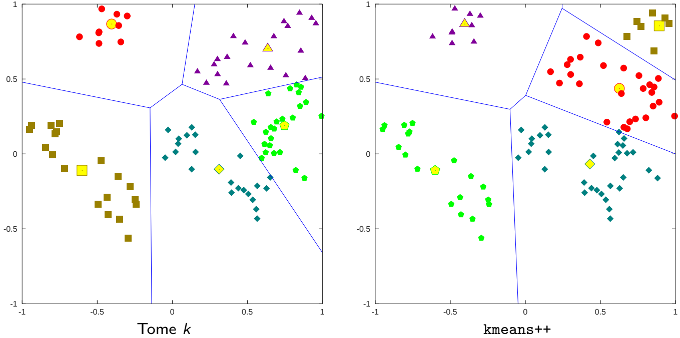


Partición de Voronoi

- Algoritmo particiona espacio de entrada
- Entre cada par de centroides, hiperplano mediatriz indica frontera de decisión de pertenencia a conglomerado
- Combinación de fronteras de decisión produce una partición de Voronoi
- Por particionar el espacio de entrada es que k-medias entra en la categoría de algoritmos de partición.
- Note que un nuevo punto puede ser agrupado al conglomerado con el centroide más cercano.

Convergencia

- El algoritmo de *k*-medias siempre converge
- Sin embargo, conglomerados dependen de inicialización



Función de distorsión

- La **función de distorsión** se define como

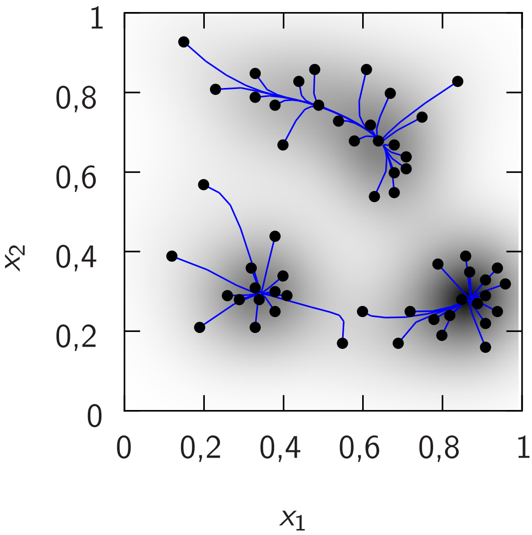
$$J(\underline{\mathbf{c}},\underline{\boldsymbol{\mu}}) = \sum_{i=1}^{m} \|\underline{\mathbf{x}}^{(i)} - \underline{\boldsymbol{\mu}}_{c^{(i)}}\|_2^2$$

- $J(\underline{\mathbf{c}},\underline{\boldsymbol{\mu}})$ mide la suma de cuadrados de distancias de los puntos $\underline{\mathbf{x}}^{(i)}$ a sus centroides $\underline{\boldsymbol{\mu}}_{c^{(i)}}$
- Se puede demostrar que k-medias realiza un descenso de coordenadas en J
    - Primero minimiza J respecto a $\underline{\mathbf{c}}$ (con $\underline{\boldsymbol{\mu}}$ constante)
    - Luego minimiza J respecto a $\underline{\boldsymbol{\mu}}$ (con $\underline{\mathbf{c}}$ constante)
- J no es convexa y por tanto la convergencia no se garantiza que sea a un mínimo global
- Si mínimo global es requerido, se puede inicializar varias veces y seleccionar aglomeración con menor J.

Métodos basados en densidad

- El *k*-medias requiere como entrada el número *k* de conglomerados
- En bastantes aplicaciones no se sabe cuántos *k* se requieren
- Otra estrategia se basa en usar la **densidad** de puntos
- Los conglomerados son regiones densas separadas por regiones de baja densidad.
- No requieren saber el número de conglomerados.
- La forma de los clusters es arbitraria (no son celdas de Voronoi)
- Métodos usuales:
  - Aglomeración por desplazamiento de medias
  - DBSCAN

Desplazamiento de medias



DBSCAN

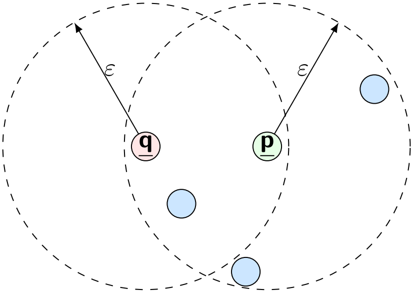
DBSCAN

- Density-based spatial clustering of applications with noise (DBSCAN, Ester, 1996)
- Algoritmo premiado en 2014, basado en densidad, más eficiente que el desplazamiento de medias
- Parte de definición de Vecindad-ε:

$$\mathcal{N}_\varepsilon(\underline{\mathbf{p}}) = \{\underline{\mathbf{q}} \mid d(\underline{\mathbf{p}},\underline{\mathbf{q}}) \leq \varepsilon\}$$

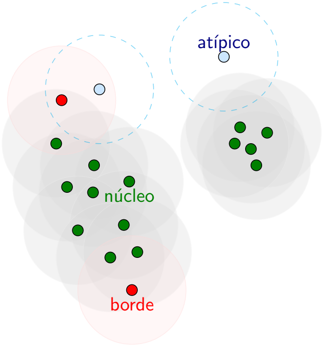
- Un punto se dice estar en región densa si su vecindad-ε tiene al menos `minPts` puntos
- Algoritmo tiene entonces dos parámetros:
  - eps (ε): radio de vecindario
  - `minPts`: mínimo número de puntos para formar región densa

Región densa



- $|\mathcal{N}_\varepsilon(\underline{\mathbf{p}})| = 5$
- $|\mathcal{N}_\varepsilon(\underline{\mathbf{q}})| = 3$
- Con `minPts`=4 entonces $\underline{\mathbf{p}}$ está en región densa y $\underline{\mathbf{q}}$ no.

Categorías de puntos



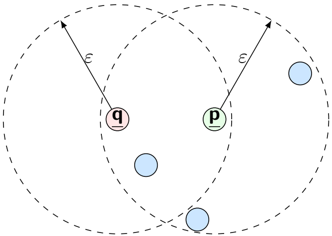
Ejemplo con `minPts`$= 4$ y $\varepsilon = 1$

Los puntos caen en tres categorías:

- Es **núcleo** si $\mathcal{N}_\varepsilon$ tiene más o `minPts`
- Es **borde** si $\mathcal{N}_\varepsilon$ tiene menos de `minPts`, pero $\mathcal{N}_\varepsilon$ tiene al menos un punto núcleo
- Es **atípico** si no es ni núcleo ni borde

Denso-alcance directo

- Un punto $\underline{\mathbf{q}}$ es **directamente denso-alcanzable** desde el objeto $\underline{\mathbf{p}}$ si $\underline{\mathbf{p}}$ es núcleo y $\underline{\mathbf{q}}$ está en su vecindad-ε



En ejemplo con `minPts`$=4$:

- $\underline{\mathbf{q}}$ es directamente denso-alcanzable desde $\underline{\mathbf{p}}$
- $\underline{\mathbf{p}}$ no es directamente denso-alcanzable desde $\underline{\mathbf{q}}$
- El denso-alcance **no** es simétrico

Denso-alcance indirecto

- Un punto $\underline{\mathbf{q}}$ es **indirectamente denso-alcanzable** desde un punto $\underline{\mathbf{p}}$ si existe una cadena de puntos

$$\underline{\mathbf{q}} \leftarrow \underline{\mathbf{p}}_1 \leftarrow \underline{\mathbf{p}}_2 \leftarrow \cdots \leftarrow \underline{\mathbf{p}}_k \leftarrow \underline{\mathbf{p}}$$

  tales que

  - $\underline{\mathbf{p}}_k$ es directamente denso-alcanzable desde $\underline{\mathbf{p}}$
  - $\underline{\mathbf{p}}_{i-1}$ es directamente denso-alcanzable desde $\underline{\mathbf{p}}_i$
  - $\underline{\mathbf{q}}$ es directamente denso-alcanzable desde $\underline{\mathbf{p}}_1$

Entradas y salidas del algoritmo

- Entrada al algoritmo será la matriz de diseño $\mathbf{X}$ con datos en cada fila, y parámetros `minPts` y ε.
- El DBSCAN retorna la asignación de un conglomerado a cada punto
- Los puntos pueden ser asignados a un pseudo-conglomerado `ruido`
- El algoritmo no prevé predicción de nuevos puntos
- El método `BusqueVecinos`($\mathbf{X}$,$\underline{\mathbf{x}}$,ε) retorna los índices de las filas en $\mathbf{X}$ que se encuentran a una distancia no mayor a ε de $\underline{\mathbf{x}}$ (los puntos denso-alcanzables desde $\underline{\mathbf{x}}$)

Pseudocódigo DBSCAN (versión corta)

foreach *dato* $i = 1, \ldots, m$ **do**
  if $\underline{\mathbf{x}}^{(i)}$ *sin clasificar* **then**
    if $\underline{\mathbf{x}}^{(i)}$ *es núcleo* **then**
      Recolecte todos los puntos denso-alcanzables desde $\underline{\mathbf{x}}^{(i)}$
      Asigne todos esos puntos a un nuevo conglomerado
    else
      Asigne $\underline{\mathbf{x}}^{(i)}$ a `ruido`
    end
  end
end

Pseudocódigo DBSCAN (versión larga) (1)

$C := 0$ */* Conglomerado actual */*
for $i = 1, \ldots, m$ **do**
    if $c^{(i)} \neq$ sin_definir **then**
        continúe   */* Dato ya fue asignado */*
    end if
    $\mathcal{N}$:=BusqueVecinos($\mathbf{X}$,$\underline{\mathbf{x}}^{(i)}$,$\varepsilon$)
    if $|\mathcal{N}|$ <minPts **then**
        $c^{(i)}$:=ruido   */* Dato aislado */*
        continúe
    end if
    $C := C + 1$
    $c^{(i)} := C$
    $\mathcal{S} := \mathcal{N} \setminus \{i\}$ */* Semillas aun por evaluar */*
    for $j \in \mathcal{S}$ **do**

Pseudocódigo DBSCAN (versión larga) (2)

if $c^{(j)}$ =ruido **then**
  $c^{(j)} := C$ $\quad$ /* *Reasigne a conglomerado actual* */
end if
if $c^{(j)} \neq$sin_definir **then**
  continúe $\quad$ /* *Ya fue asignado previamente* */
end if
$c^{(j)} := C$
$\mathcal{N}$:=BusqueVecinos($\mathbf{X}$,$\underline{\mathbf{x}}^{(j)}$,$\varepsilon$)
if $|\mathcal{N}| \geq$minPts **then**
  $\mathcal{S} := \mathcal{S} \cup \mathcal{N}$ $\quad$ /* *Solo asigne si suficientes vecinos* */
end if
end for
end for

Resumen

1. Aglomeración
   - *k*-medias
   - DBSCAN

Este documento ha sido elaborado con software libre incluyendo LaTeX, Beamer, GNUPlot, GNU/Octave, XFig, Inkscape, GNU-Make y Subversion en GNU/Linux