Tema 8. Agrupamento (clustering)

- Obxectivo: agrupar patróns en grupos (clusters) usando unha medida de similaridade ou de distancia
- Aprenden a estructura (de grupos) dos datos. Espérase que:

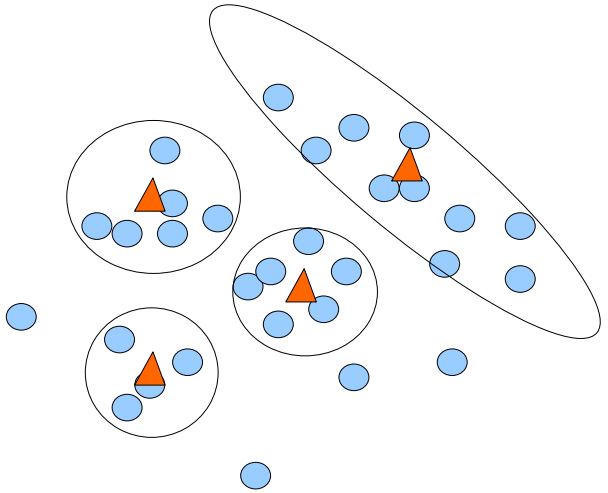
a) Patróns similares (distancias baixas) se asignen ao mesmo grupo e sexan usados para crear un representante do grupo (prototipo)

b) Patróns distintos (distancias altas) se asignen a grupos distintos

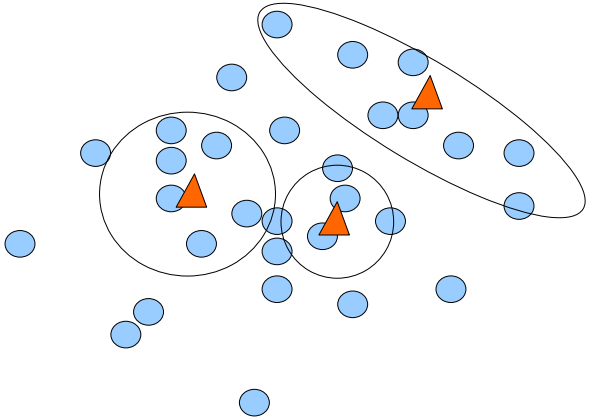
- O “umbral” que separa as distancias entre patróns similares e as distancias entre patróns diferentes depende do nº de grupos
- Moitos grupos: cada grupo terá patróns moi similares
- Poucos grupos: cada grupo pode ter patróns moi distintos

Agrupamento (I)

Exemplo de agrupamento claro



Exemplo de agrupamento difícil (datos pouco concentrados sen espazos baleiros)



- A distribución dos patróns determina a dificultade do agrupamento
- O agrupamento aprendido terá unha certa calidade: en que medida os prototipos representan aos patróns?
- Existen rexións libres de patróns entre os distintos grupos?
- En dimensións altas, todo se complica porque as distancias entre patróns son moi similares

Agrupamento (II)

- Resultado do agrupamento: apréndense prototipos ou representantes de grupos
- Actúan como un “resumo” dos datos completos
- É esperábel que os prototipos se sitúen nas rexións do espazo de entrada con maior densidade de patróns
- Estimación da probabilidade de ocurrencia de patróns: aprendizaxe da densidade de probabilidade dos datos
- Aprendizaxe non supervisada: non existe unha etiqueta de patrón verdadeira (discreta ou continua)
- So hai unha medida de distancia (ou similaridade), que indica o lonxe (ou perto) que están dous patróns entre si ou dun prototipo

Agrupamento k-medias (I)

- Hai que coñecer o nº G de grupos que se desexa crear
- Sitúanse aleatoriamente G prototipos (medias) $\{\mathbf{p}_i\}_{i=1}^{G}$ en G patróns (hai G prototipos, cada un representando a un grupo)
- Para cada patrón $\{\mathbf{x}_i\}_{i=1}^{N}$:

1) Calcúlanse as distancias dende $\mathbf{x}_i$ até os G prototipos

2) Selecciónase o prototipo máis cercano

- Cada prototipo $\{\mathbf{p}_i\}_{i=1}^{G}$ actualízase usando so os patróns para os que resultou o máis cercano
- Repítese o cálculo das distancias entre patróns e prototipos, xa que éstos variaron
- Repítese o proceso ata que os prototipos case non cambien ou ata un número pre-determinado de veces

Agrupamento k-medias (II)

Datos: G>1, $\{\mathbf{x}_i\}_{i=1}^{N}$, $\varepsilon>0$

Saídas: $\{\mathbf{p}_j,E_j\}_{j=1}^{G}$: prototipos e lista de patróns de cada prototipo

$\{\mathbf{p}_j=\mathbf{x}_{s(j)}\}_{j=1}^{G}$ #s(j): nº aleatorio en {1..N}

repeat

  P=$\{\mathbf{p}_j\}_{j=1}^{G}$; $\{N_j\}_{j=1}^{G}$=0; $\{E_j=\emptyset\}_{j=1}^{G}$

  for i=1:N

    $$k=\underset{j=1..G}{argmin}\left\{\left|\boldsymbol{x}_i-\boldsymbol{p}_j\right|\right\}$$

    $N_k=N_k+1$; $E_k=E_k\cup\{i\}$

  endfor

  for j=1:G

    $$\boldsymbol{p}_j'=\frac{1}{N_j}\sum_{i\in E_j}\boldsymbol{x}_i$$

  endfor

  P'=$\{\mathbf{p}_j'\}_{j=1}^{G}$; $\{\mathbf{p}_j=\mathbf{p}_j'\}_{j=1}^{G}$

until $|\mathbf{P'}-\mathbf{P}|<\varepsilon$

Medidas de distancia entre patróns e prototipos

- Distancia euclídea: $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{\sum_{k=1}^{N} (x_{ik} - x_{jk})^2}$
- Distancia de Mahalanobis: $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)\Sigma^{-1}(\boldsymbol{x}_i - \boldsymbol{x}_j)}$
- Similaridade: $s(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{\boldsymbol{x}_i^T \boldsymbol{x}_j}{|\boldsymbol{x}_i||\boldsymbol{x}_j|}$
- Similaridade de Tanimoto: $s(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{\boldsymbol{x}_i^T \boldsymbol{x}_j}{|\boldsymbol{x}_i^2| + |\boldsymbol{x}_j^2| - \boldsymbol{x}_i^T \boldsymbol{x}_j}$

Σ é a matriz de covarianza dos datos

Medidas de erro/calidade do agrupamento (I)

- Minimizar o erro cadrático entre prototipos e patróns asociados:

$$J=\sum_{k=1}^{G}\sum_{i\in E_k}|\boldsymbol{x}_i-\boldsymbol{p}_k|^2$$

- Minimizar o índice de Davies-Bouldin:

$$DB=\frac{1}{G}\sum_{k=1}^{G}\max_{l\neq k}\left\{\frac{\bar{d}_k+\bar{d}_l}{|\boldsymbol{p}_k-\boldsymbol{p}_l|}\right\}$$

$\bar{d}_k$: distancia media entre patróns do grupo k

- Minimizar o índice de Dunn:

$$D=\frac{\min\limits_{1\leq l<k\leq G}|\boldsymbol{p}_k-\boldsymbol{p}_l|}{\max\limits_{k=1,\ldots G}\{d'_k\}}$$

d_k': distancia máxima entre patróns do grupo k

- Post-procesamento: os grupos de menor calidade (p.ex. con meirande erro cadrático) pódense dividir en grupos máis homoxéneos para reducir o erro

O número G de grupos

- Hiper-parámetro que debe establecerse antes do proceso de agrupamento
- Nalgúns problemas pódese sintonizar optimizando algunha medida da calidade dos grupos aprendidos
- Ás veces está pre-definido polo problema
- Pódese usar análise en compoñentes principais (PCA) para visualizar os datos en 2D e facer G igual ao nº de agrupamentos detectados visualmente
- Pode ser máis operativo aplicar un umbral D á distancia patrón-prototipo que usar un número G de grupos:

a) Se $d_{ij}=|\mathbf{x}_i-\mathbf{p}_j|<D$: $\mathbf{x}_i$ asígnase ao grupo j con $d_{ij}<D$ mínima e actualiza o prototipo $\mathbf{p}_j$

b) Se $d_{ij}>D$ para tódolos grupos, créase un grupo novo

Kmeans++

- Kmeans falla algunhas veces. P. ex., 4 patróns nas esquinas dun rectángulo máis ancho que alto, G=2. Se os 2 prototipos iniciais están nas metades dos lados horizontais, os grupos son arriba e abaixo, e deberían ser esquerda e dereita
- Kmeans++ inicializa os prototipos evitando estos problemas:

1) Escolle aleatoriamente un patrón como prototipo $\mathbf{p}_1$
2) Para cada $\mathbf{x}_i$ non escollido, calcula as distancias $d_{ij}=|\mathbf{x}_i-\mathbf{p}_j|$ a tódolos prototipos e selecciona o prototipo con $d_i=\max_j\{d_{ij}\}$ máxima
3) Selecciona un novo prototipo aleatoriamente con probabilidade proporcional a d_i (canto maior d_i, máis probábel é que $\mathbf{x}_i$ sexa seleccionado)
4) Repite os pasos 2 e 3 ata seleccionar G prototipos iniciais
5) Executa kmeans partindo destos prototipos

Outros algoritmos de agrupamento

- Kmedias por lotes (mini-batch kmeans)
- K-medianas (K-medoids): usa medianas no canto de medias
- CLARA: clustering for large applications, extensión de K-medoids.
- K-medias xerárquico: parte de G=2 e continúa creando grupos en cada pola da árbore.
- Cobweb: crea unha árbore de nodos, permite crear nodos, xuntar e dividir nodos, e propagar un patrón árbore abaixo.
- Classit: extensión de Cobweb para valores numéricos usando gausianas.

Agrupamento xerárquico

- Agrupamento **aglomerativo**: comézase con N grupos (tantos como patróns)

1) Os pares de grupos máis cercanos fusiónanse progresivamente

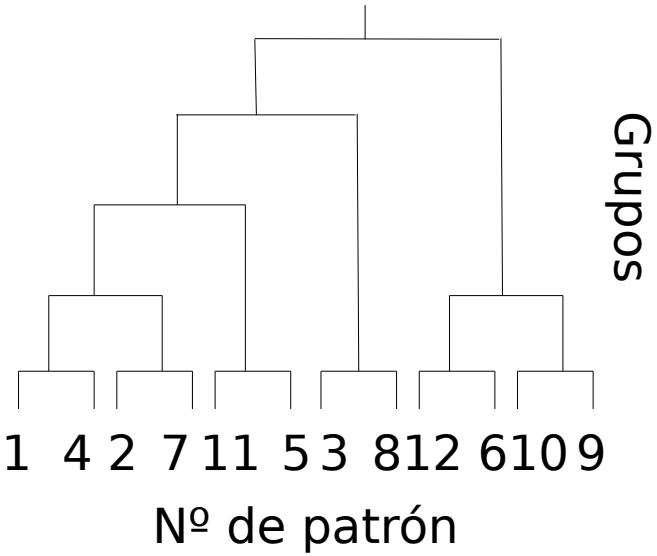
2) A distancia entre dous grupos pode ser:

    a) Mínima distancia entre dous patróns dos dous grupos (agrupamento por enlace simple)

    b) Máxima distancia (enlace dobre)

3) Dendrograma: árbore onde as follas son patróns agrupados na orde en que se fusionan os grupos

- Agrupamento **divisivo**: parte dun único grupo que se vai dividindo iterativamente en varios máis pequenos ata obter N grupos



Desprazamento medio (mean-shift)

- Descubre grupos de patróns cercanos e sitúa os prototipos nas súas medias
- Cada candidato a prototipo $\mathbf{p}_i(t)$ actualízase a $\mathbf{p}_i(t+1)=\mathbf{m}_i[\mathbf{p}_i(t)]$

$$\boldsymbol{m}_i(\boldsymbol{p}_i)=\frac{\sum_{j\in N_i} K(\boldsymbol{x}_j-\boldsymbol{p}_i)\boldsymbol{x}_j}{\sum_{j\in N_i} K(\boldsymbol{x}_j-\boldsymbol{p}_i)}$$

Suavizado usando un cerne gausiano

$K(\mathbf{x}_j-\mathbf{p}_i)=\exp(-|\mathbf{x}_j-\mathbf{p}_i|^2/\sigma^2)$: cerne gausiano con ancho σ (bandwidth)

N_i é o entorno de centro $\mathbf{x}_i$ con radio σ

$\mathbf{m}_i(\mathbf{p}_i)$ é o vector desprazamento medio (mean-shift), dirixido cara a rexión con meirande densidade de patróns

- Mean-shift fixa automáticamente o nº G de grupos a partir de σ

Mestura de gausianas (I)

- Cada grupo k=1..G é unha distribución gausiana dada polo seu prototipo $\mathbf{p}_k$ (media) e covarianza $\mathbf{\Sigma}_k$ (ambos a determinar)

$$f_k(\boldsymbol{x}) = \mathrm{e}^{-(\boldsymbol{x}-\boldsymbol{p}_k)\mathbf{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{p}_k)^T}$$

- Algoritmo esperanza-maximización (**EM**):

1) Paso E (**esperanza**): estima ($\mathbf{p}_k$, $\mathbf{\Sigma}_k$, k=1..G) e a partir deles calcula a esperanza w_{ik} de que cada patrón $\mathbf{x}_i$ pertenza ao grupo k:

$$w_{ik} = \frac{f_k(\boldsymbol{x}_i)}{\sum_{k=1}^{G} f_k(\boldsymbol{x}_i)}$$

$$\boldsymbol{p}_k(t+1) = \frac{\sum_{i=1}^{N} w_{ik}\boldsymbol{x}_i}{\sum_{i=1}^{N} w_{ik}}$$

2) Paso M (**maximización**): actualiza os valores ($\mathbf{p}_k$,$\mathbf{\Sigma}_k$) para maximizar esta esperanza

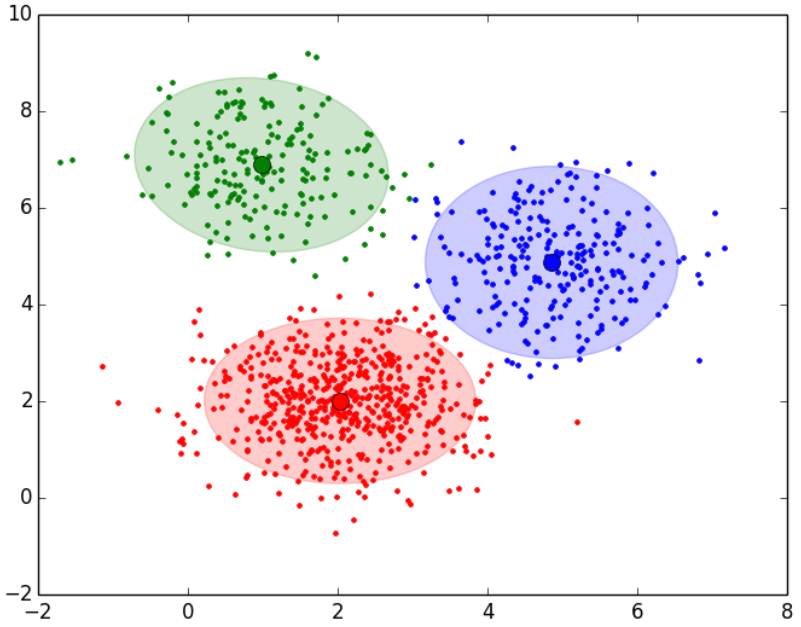
$$\mathbf{\Sigma}_k(t+1) = \frac{\sum_{i=1}^{N} w_{ik}(\boldsymbol{x}_i-\boldsymbol{p}_k)(\boldsymbol{x}_i-\boldsymbol{p}_k)^T}{\sum_{i=1}^{N} w_{ik}}$$

Mestura de gausianas (II)

- Os pasos EM repítense até a converxencia
- En cada paso, aumenta a verosimilitude (credibilidade) V:

$$V = \prod_{i=1}^{N} \sum_{k=1}^{G} w_{ik}$$

- Converxe cando V deixa de aumentar
- Sempre converxe, pero pode atopar mínimos locais
- É aconsellábel repetir o proceso varias veces seleccionando o mellor resultado, que maximiza V
- Unha vez determinados ($\mathbf{p}_k$, Σ_k, k=1..G), un patrón **x** asígnase ao grupo k que maximiza $f_k(\mathbf{x})$
- É un método de agrupamento baseado en probabilidades



https://jarroba.com/expectation-maximization-python-scikit-learn-ejemplos/

Propagación de afinidades

- Sexa s_{ik} a similaridade entre o patrón $\mathbf{x}_i$ e o grupo k ($\mathbf{p}_k$)
- Responsabilidade r_{ik}: mide se grupo k debe apropiarse do patrón i

$$r_{ik} = s_{ik} - \max_{k \neq l} \left\{ a_{il} + s_{il} \right\}$$

- Dispoñibilidade a_{ik}: mide se o patrón i debe asignarse ao grupo k

$$a_{ik} = min\left(0 , r_{kk} + \sum_{j \neq i,k} r_{jk} \right)$$

- Inicialmente, $r_{ik}(t=0)=a_{ik}(t=0)=0$. En cada iteración t, actualízanse $r_{ik}(t)$ e $a_{ik}(t)$. Para evitar oscilacións numéricas, faise:

$$\left.\begin{aligned} r_{ik}(t+1) &= \alpha r_{ik}(t) + (1-\alpha) r_{ik}(t+1) \\ a_{ik}(t+1) &= \alpha r_{ik}(t) + (1-\alpha) r_{ik}(t+1) \end{aligned}\right\} \; i=1\ldots N;\; k=1\ldots G$$

- Converxe cando r_{ik} e a_{ik} non cambian apreciábelmente
- Os prototipos son os G patróns $\mathbf{x}_k$ con máis $\Sigma_{i=1}^{N} r_{ik}$
- Cada patrón $\mathbf{x}_i$ asígnase ao prototipo $\mathbf{p}_k$ con meirande r_{ik}

Agrupamento espectral

- Parte da matriz de afinidades entre patróns
- Busca o subespazo de baixa dimensión xerado polos seus autovectores principais (maiores autovalores da matriz de afinidade)
- Proxecta os patróns de entrenamento a este espazo
- Aplica kmedias sobre os patróns proxectados
- Require coñecer G, é lento para G elevado
- Este método con G=2 realiza un corte normalizado do grafo de similaridades
- O peso do arco do corte é pequeno comparado cos pesos dos arcos en cada grupo
- En imaxes, este peso dos arcos no grafo de similaridades é o gradente da imaxe

Dbscan

- Baseado en densidades. Considera os grupos como áreas de alta densidade separadas por áreas despoboadas
- Crea grupos con formas xeométricas variábeis
- Cada grupo ten patróns principais (situados en áreas densas) e patróns non principais (cercanos aos principais)
- Patrón principal: ten M ou máis patróns (veciños) do mesmo grupo nun entorno de radio ε
- Constrúe o grupo recursivamente:

1) Atopa un primeiro patrón
2) Atopa tódolos seus veciños principais
3) Repite o proceso ata que tódolos patróns do grupo (que están a ε dun patrón principal) sexan non principais

- Hiper-parámetros: M (min_samples), ε (epsilon, moi importante) seleccionado usando un codo do diagrama de veciños máis cercanos

Optics

- Baseado en Dbscan
- Substitúe un único ε por un rango de valores
- Cada patrón ten un ε (alcance) e unha posición (índice) distintos no grupo
- Permite grupos de distintos tamanos: grandes e pequenos
- O hiper-parámetro ε_{max} define o seu comportamento:

a) Se $\varepsilon_{max}=\infty$, funciona como Dbscan

b) Se ε_{max} é baixo, crea grupos de tamanos variábeis, cada un cos patróns dentro do seu radio de alcance

Birch

- Balanced iterative reducing and clustering using hierarchies
- Crea unha árbore de grupos, de altura equilibrada
- Dous hiper-parámetros:

1) Factor de ramificación (B): limita o nº de subgrupos por nodo
2) Umbral: limita a distancia entre un patrón e os subgrupos

- Cada nodo ten ata B nodos fillos e ata B subgrupos
- Os grupos finais son os subgrupos dos nodos terminais
- Cada patrón propágase dende o nodo raíz até acadar un nodo terminal, asignándose ao subgrupo máis cercano do nodo terminal, respectando factor e umbral
- Se non respecta algún deles, créase un novo subgrupo no nodo terminal seleccionado, ou un novo nodo se xa ten B grupos

Exemplo de aplicación (I): segmentación de imaxes

- Segmentación: extracción de rexións de apariencia visual homoxénea a partir dunha imaxe
- Cada píxel da imaxe é un vector de 3 valores (intensidades no espazo RGB)
- O método K-medias mostra a imaxe cunha paleta de G cores

C. Bishop. Pattern recognition and machine learning

Exemplo de aplicación (II): compresión de imaxes

- Imaxe orixinal: N píxeles, $\mathbf{x}_i=(R_i,G_i,B_i)$, i=1,...,N. Ocupa $T_1=3N$ bytes, porque cada valor V=R,G ou B é 1 byte ($1\leq V\leq 255$)
- Agrupamento: G ($\ll$N) prototipos $\{\mathbf{p}_j\}_{j=1}^{G}$, con $\mathbf{p}_j=(R_j,G_j,B_j)$.

a) Os G prototipos ocupan 3G bytes

b) A imaxe ocupa $N\log_2 G$ bits=$N\log_2 G/8$ bytes, porque cada píxel almacena o índice do seu prototipo, que ocupa $\log_2 G$ bits

- Imaxe comprimida: ocupa $T_2=3G+N\log_2 G/8$ bytes
- Aforro α (en %):

$$\alpha=100\frac{T_1-T_2}{T_1}=100\left(1-\frac{G}{N}-\frac{\log_2 G}{24}\right)$$

- Se G=10, N=800x600: $T_1=11.52\cdot 10^6$ bytes, $T_2=1.59\cdot 10^6$ bytes, α=86.1% de aforro