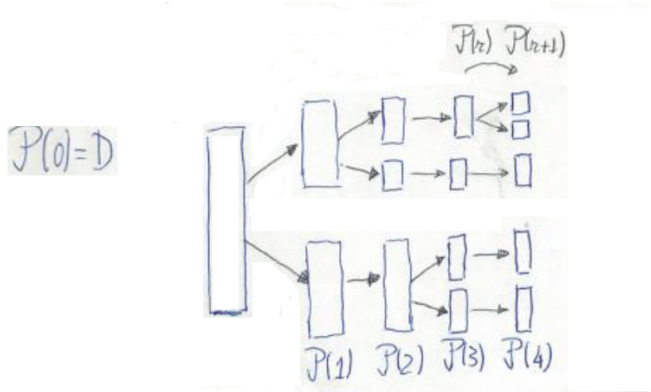
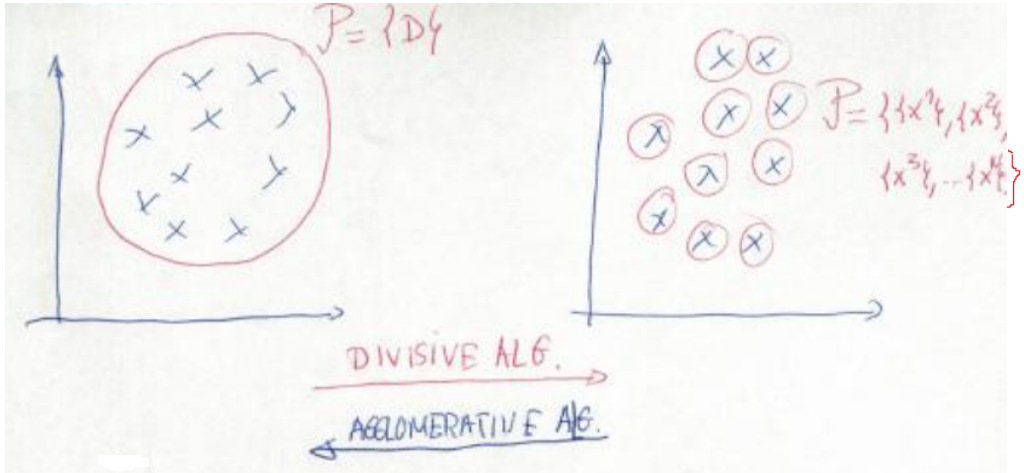
Métricas em Machine Learning

Hierarchical Clustering:



Nested Partitioning - partição aninhada



Divisive versus Agglomerative Algorithm

Métricas em Machine Learning

Definição : $\mathcal{P}$ é uma K-partição de D se,

$\mathcal{P} = \{C^1, C^2, ..., C^K\}$ onde,

a) $C^K \subseteq D$ com $C^k \neq \emptyset$;
b) $C^K \cap C^L = \emptyset$ para $K \neq L$;
c) $\bigcup_{k=1}^{K} C^k = D$.

Uma partição de um conjunto D é uma família de conjuntos composta por subconjuntos não vazios de D, disjuntos 2 a 2 e cuja união é D.

Definição : partição aninhada (nested partition)

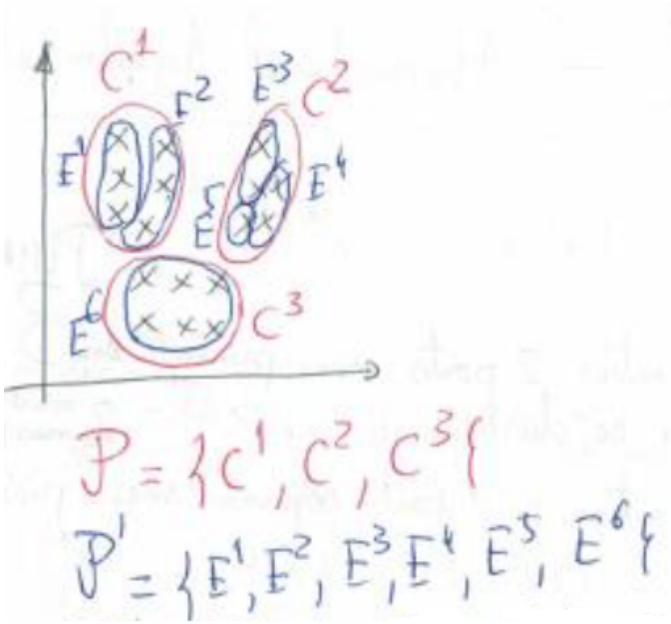
$\mathcal{P}$ e $\mathcal{P}'$ são partições de D. $\mathcal{P}'$ diz-se **partição aninhada** de $\mathcal{P}$ se,

$$\forall_{C' \in \mathcal{P}'} \exists_{C \in \mathcal{P}} : \ C' \subseteq C$$

e escreve-se $\mathcal{P}' \sqsubseteq \mathcal{P}$.

Exemplo de nested partition (contexto HDA)

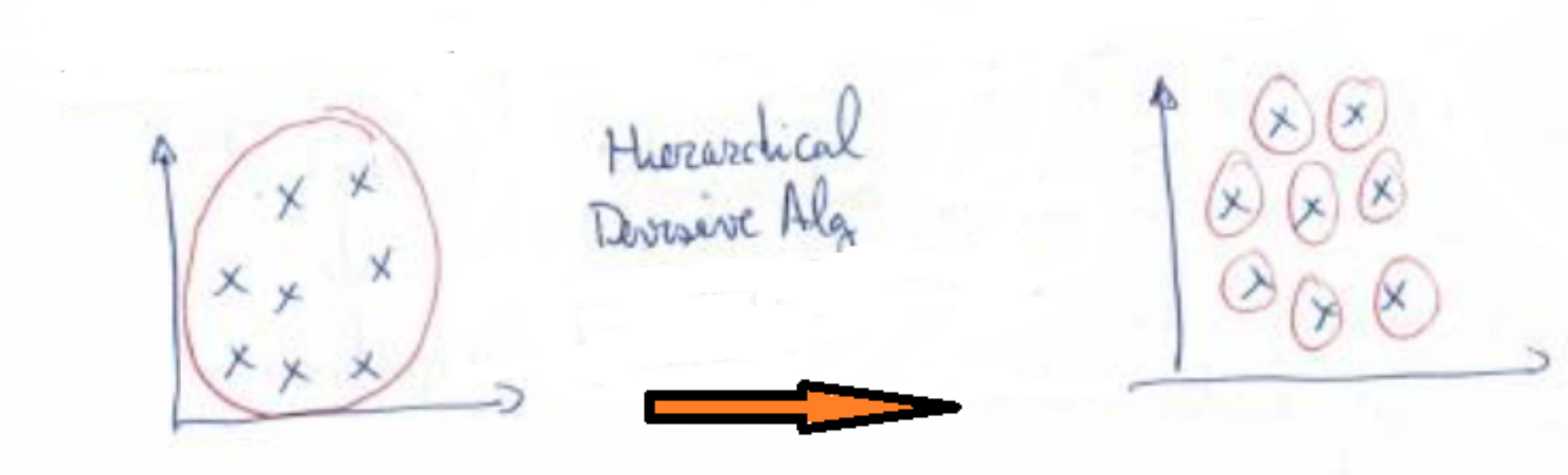
$\underline{\mathcal{P}(D)} = \mathcal{P}(0) = \{\{1,2,3,4,5\}\}$ (muito grosseira)
$\mathcal{P}(1) = \{\{1,2,3\},\{4,5\}\}$
$\mathcal{P}(2) = \{\{1\},\{2,3\},\{4,5\}\}$
$\mathcal{P}(3) = \{\{1\},\{2,3\},\{4\},\{5\}\}$
$\overline{\mathcal{P}(D)} = \mathcal{P}(4) = \{\{1\},\{2\},\{3\},\{4\},\{5\}\}$ (muito fina)



Métricas em Machine Learning

Hierarchichal Divisive clustering (HDA):

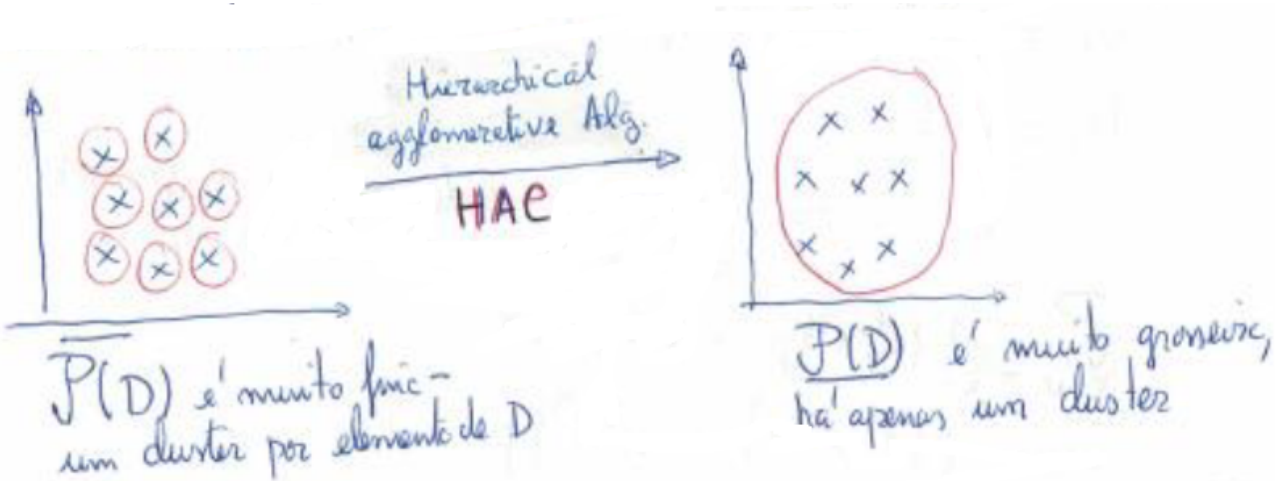
$\underline{\mathcal{P}(D)} = \mathcal{P}(0) \sqsupseteq \mathcal{P}(1) \sqsupseteq \mathcal{P}(2) \sqsupseteq ... \sqsupseteq \mathcal{P}(r) = \overline{\mathcal{P}(D)}$



Nas nossas notas vamos descrever o procedimento do Hierarchical Agglomerative Algorithm.

Métricas em Machine Learning

Um algoritmo do tipo **Hierarchical clustering** consiste em encontrar uma Lista ordenada de nested partitions.



Hierarchichal Agglomerative clustering (HAC):

$$\overline{\mathcal{P}(D)} = \mathcal{P}(0) \sqsubseteq \mathcal{P}(1) \sqsubseteq \mathcal{P}(2) \sqsubseteq \ldots \sqsubseteq \mathcal{P}(r) = \underline{\mathcal{P}(D)}$$

onde $\overline{\mathcal{P}(D)}$ é muito fina e $\underline{\mathcal{P}(D)}$ é muito grosseira.

Hierarchichal Agglomerative Clustering (o algoritmo):

Seja $D = \{x^1, x^2, ..., x^N\}$

$\mathcal{P}(0) = \{\{x^1\}, \{x^2\}, ..., \{x^N\}\}$, i=0, $Energia(0) = 0$

for i=1: N-1

$N_{i-1} = |\mathcal{P}(i-1)|$

[k,p,LKG]= **encontra par clusters com linkage minimo** $(\mathcal{P}(i-1), N_{i-1})$

$\mathcal{P}(i)$= **faz fusão de clusters**$(k\ , p,\ \mathcal{P}(i-1))$.

Energia(i)=LKG (linkage entre cluster k e cluster p)

end for

Nota: Se D tem N eventos, a partição inicial $\mathcal{P}(0)$ tem N subconconjuntos, e a partição $N-1$ tem apenas 1 suconjunto. Há $N-1$ passos de procura de linkage mínimo. Notar que em aulas anteriores, definimos vários linkages entre subconjuntos.

Métricas em Machine Learning

Exemplo: Seja $D = \{(0,0), (1/2,0), (2,1), (2,2.5), (0,3)\}$ uma base de dados com 5 eventos de $\mathbb{R}^2$. Usando a métrica de Manhattan e Single Linkage, aplicar o algoritmo hierárquico aglomerativo (HAC).

Resolução:

Deve considerar $\mathcal{P}(0) = \{\{(0,0)\}, \{(0.5,0)\}, \{(2,1)\}, \{(2,2.5)\}, \{(0,3)\}\}$

T = tabela com valores de linkage entre cada para de subconjuntos de $\mathcal{P}(0)$

Encontrar entre que subconjuntos o linkage é mínimo - sejam k e p.

$\mathcal{P}(1)$ resulta de fundir suconjuntos k e p com energia de fusão E(1)

repetir procedimento até ter $\mathcal{P}$ ter apenas 1 subconjunto.

Tabela de Linkage

Nas linhas e nas colunas de T temos os subconjuntos da partição que estamos a analisar.

T(0)	C1={x1}	C2={x2}	C3={x3}	C4={x4}	C5={x5}
C1={x1}	**0**	0.5	3	4.5	3
C2={x2}	*	**0**	2.5	4	3.5
C3={x3}	*	*	**0**	1.5	4
C4={x4}	*	*	*	**0**	2.5
C5={x5}	*	*	*	*	**0**

Calcula-se single linkage(dd) entre todos os pares de conjuntos na tabela.

$$dd(C^1, C^2) = |x_1^1 - x_1^2| + |x_2^1 - x_2^2| = |0 - 1/2| + |0 - 0| = 1/2$$

$$dd(C^1, C^3) = |x_1^1 - x_1^3| + |x_2^1 - x_2^3| = |0 - 2| + |0 - 1| = 3$$

Métricas em Machine Learning

$$dd(C^1, C^4) = |x_1^1 - x_1^4| + |x_2^1 - x_2^4| = |0 - 2| + |0 - 2.5| = 4.5$$

$$dd(C^1, C^5) = |x_1^1 - x_1^5| + |x_2^1 - x_2^5| = |0 - 0| + |0 - 3| = 3$$

$$dd(C^2, C^3) = |x_1^2 - x_1^3| + |x_2^2 - x_2^3| = |0.5 - 2| + |0 - 1| = 2.5$$

$$dd(C^2, C^4) = |x_1^2 - x_1^4| + |x_2^2 - x_2^4| = |0.5 - 2| + |0 - 2.5| = 4$$

$$dd(C^2, C^5) = |x_1^2 - x_1^5| + |x_2^2 - x_2^5| = |0.5 - 0| + |0 - 3| = 3.5$$

$$dd(C^3, C^4) = |x_1^3 - x_1^4| + |x_2^3 - x_2^4| = |2 - 2| + |1 - 2.5| = 1.5$$

$$dd(C^3, C^5) = |x_1^3 - x_1^5| + |x_2^3 - x_2^5| = |2 - 0| + |1 - 3| = 4$$

$$dd(C^4, C^5) = |x_1^4 - x_1^5| + |x_2^4 - x_2^5| = |2 - 0| + |2.5 - 3| = 2.5$$

O valor mínimo dos linkage é 0.5. Logo fazemos a **fusão** entre C^1 e C^2.

A partição obtida é : $\mathcal{P}(1) = \{\{x^1, x^2\}, \{x^3\}, \{x^4\}, \{x^5\}\}$.

A energia de fusão é $e(1) = 0.5$.

Métricas em Machine Learning

Construimos nova tabela de linkage.

T(1)	C1={x1,x2}	C2={x3}	C3={x4}	C4={x5}
C1={x1,x2}	**0**	2.5	4	3
C2={x3}	*	**0**	1.5	4
C3={x4}	*	*	**0**	2.5
C4={x5}	*	*	*	**0**

Notar que agora o primeiro subconjunto tem 2 elementos, pelo que o linkage já não é apenas a distância ponto a ponto.

$dd(C^1, C^2) = min(d(x^1, x^3), d(x^2, x^3)) = min(3, 2.5) = 2.5.$

$d(x^1, x^3) = |x_1^1 - x_1^3| + |x_2^1 - x_2^3| = |0 - 2| + |0 - 1| = 3$

$d(x^2, x^3) = |x_1^2 - x_1^3| + |x_2^2 - x_2^3| = |0.5 - 2| + |0 - 1| = 2.5$

Métricas em Machine Learning

$dd(C^1, C^3) = min(d(x^1, x^4), d(x^2, x^4)) = min(4.5, 4) = 4.$

$d(x^1, x^4) = |0 - 2| + |0 - 2.5| = 4.5$

$d(x^2, x^4) = |0.5 - 2| + |0 - 2.5| = 4$

$dd(C^1, C^4) = min(d(x^1, x^5), d(x^2, x^5)) = min(3, 3.5) = 3.$

$d(x^1, x^5) = |0 - 0| + |0 - 3| = 3$

$d(x^2, x^5) = |0.5 - 0| + |0 - 3| = 3.5$

Continuando, vem,

$dd(C^2, C^3) = d(x^3, x^4) = 1.5$

$dd(C^2, C^4) = d(x^3, x^5) = 4$

$dd(C^3, C^4) = d(x^4, x^5) = 2.5$

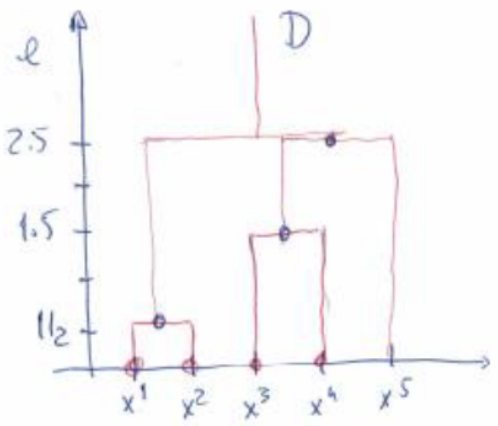
O valor menor é 1.5. Ocorre para $dd(C^2, C^3)$, logo faz-se a fusão entre C^2 e C^3.

A partição obtida é : $\mathcal{P}(2) = \{\{x^1, x^2\}, \{x^3, x^4\}, \{x^5\}\}$. A energia de fusão é $e(2) = 1.5$.

Métricas em Machine Learning

O procedimento continua até que numa partição só haja um conjunto. Tal acontece para $\mathcal{P}(4) = \{\{x^1, x^2, x^3, x^4, x^5\}\}$.

Para analisar as partições obtidas é comum usar um **dendrograma**. Um dendrograma representa o historial em termos de energia de fusão desde a partição inicial até à final.



O objectivo não é ter a partição final. Há que arranjar um critério para obter a partição ideal.

Métricas em Machine Learning

Como decidir qual a partição a considerar?

- Poderá passar por considerar um número de subconjuntos.
- Ter subconjuntos com um determinado número mínimo ou máximo de elementos.
- A energia não ser maior que determinado valor (clusters que não devem ser fundidos).
- Outros critérios ou métricas poderiam ser usados. Devemos ter em conta os dados e o objectivo a atingir.