# Université Mohamed Boudiaf – M’sila

**Département Informatique Master Intelligence Artificielle 2020-2021**

Chapitre 4 Clustering

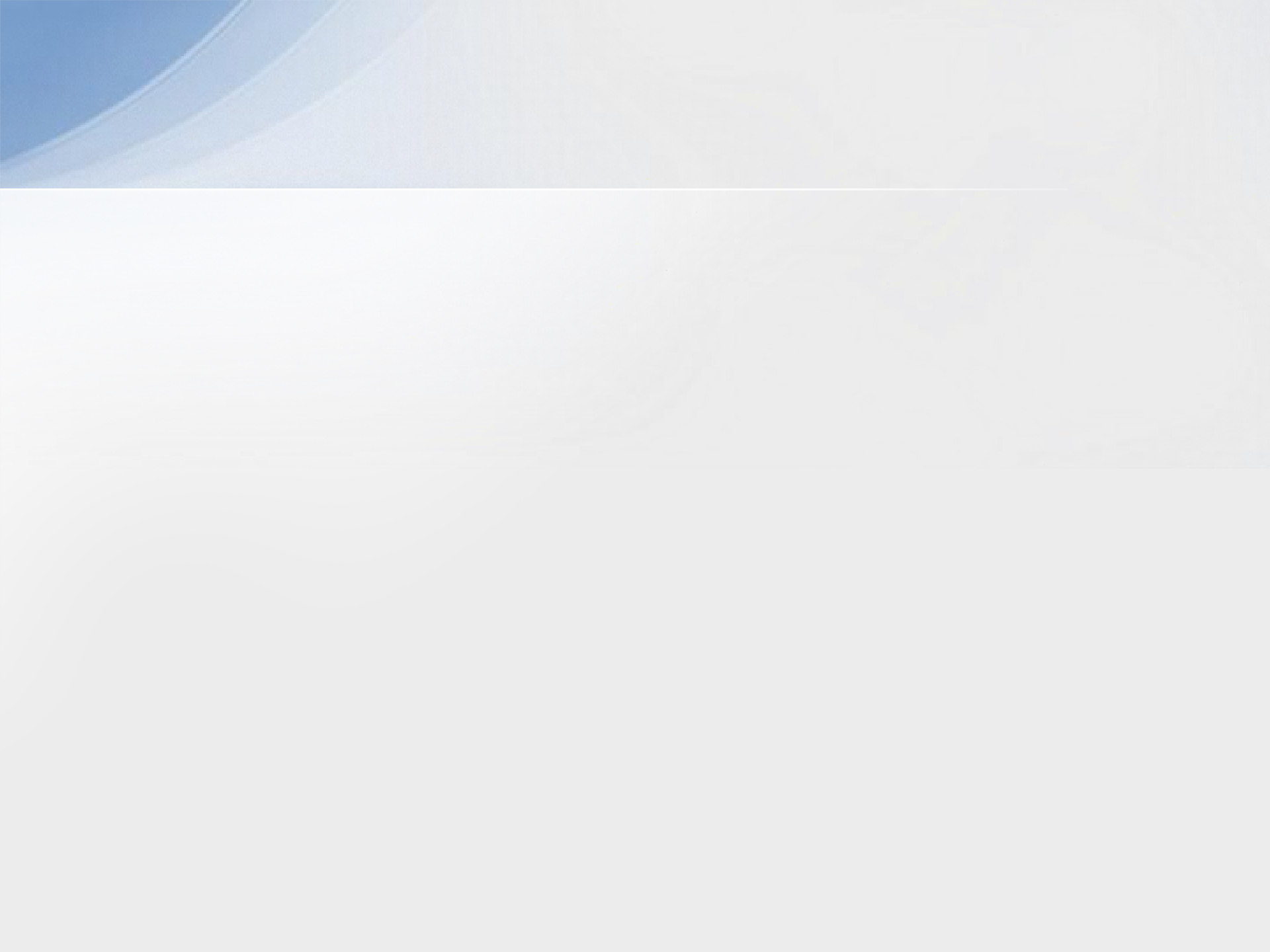
Dr. Mehenni Tahar

## Clustering

* **La Classification est un apprentissage supervisé. La supervision est faite en nommant les classes des instances d’apprentissage.**
* **Le Clustering est un apprentissage non supervisé. Il n’y a pas une connaissance apriori des classes, ni un ensemble**

**d’apprentissage.**

* **L’algorithme de clustering nécessite une affectation de chaque instance à un groupe ou classe (cluster) de telle façon que tous les objets d’un même groupe sont plus semblables que les autres.**

**Clustering**

* **Trouver des groupes (classes) d’objets tels que chaque objet d’un groupe est similaire qu’un autre objet du même groupe et différent des autres objets des autres groupes**
* L’objectif est de trouver un groupement le plus naturel possible des instances.
  + A l’intérieur d’un groupe: Maximiser la similarité entre instances.
  + Entre les groupes: Minimiser la similarité entre les instances.

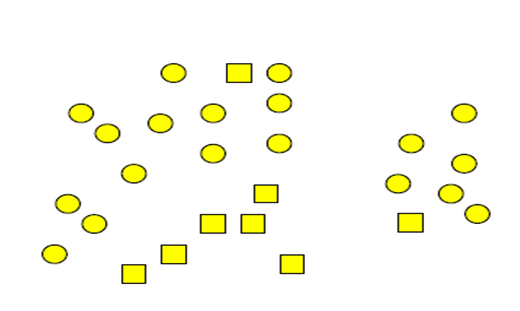
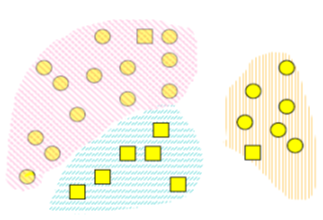


**Distances Intra- classes sont minimisées**

**Distances Inter-classes sont**

**maximisées**

## Clustering

* **Par exemple, soit l’ensemble de figures suivant:**
* **Un algorithme de clustering peut trouver les clusters suivants:**
* **Bien que certaines figures différentes coexistent dans un cluster.**

## Le problème du Clustering

* **Etant donnée une base de données D={t1,t2,…,tn} de tuples et une valeur entière k, le *Clustering est de définir une application* f:D🡒{1,..,k} où chaque ti est affecté à un seul cluster (groupe ou classe) Kj, 1<=j<=k.**

#### Un *Cluster*, Kj, contient exactement les tuples qui lui sont affectés.

* **Contrairement au problème de classification, les clusters ne sont pas connus apriori.**

# Qu’est ce qu’un bon regroupement?

* + Une bonne méthode de regroupement permet de garantir
    - Une grande similarité intra-groupe
    - Une faible similarité inter-groupe
  + La qualité d’un regroupement dépend donc de la mesure de similarité utilisée par la méthode et de son implémentation

**Structures de données**

* + Matrice de données



 ***x11***

***...***

***x1f***

***...***

***x1p*** 

 ***...***



***...***

***...***

***...***

***...*** 

 ***xi1***



***...***

***xif***

***...***

***xip*** 

 ***...***





***...***

***...***

***...***

***...*** 

* + Matrice de similarité

 ***xn1***

***...***

***xnf***

***...***

***xnp*** 

 ***0*** 

 ***d(2,1)***



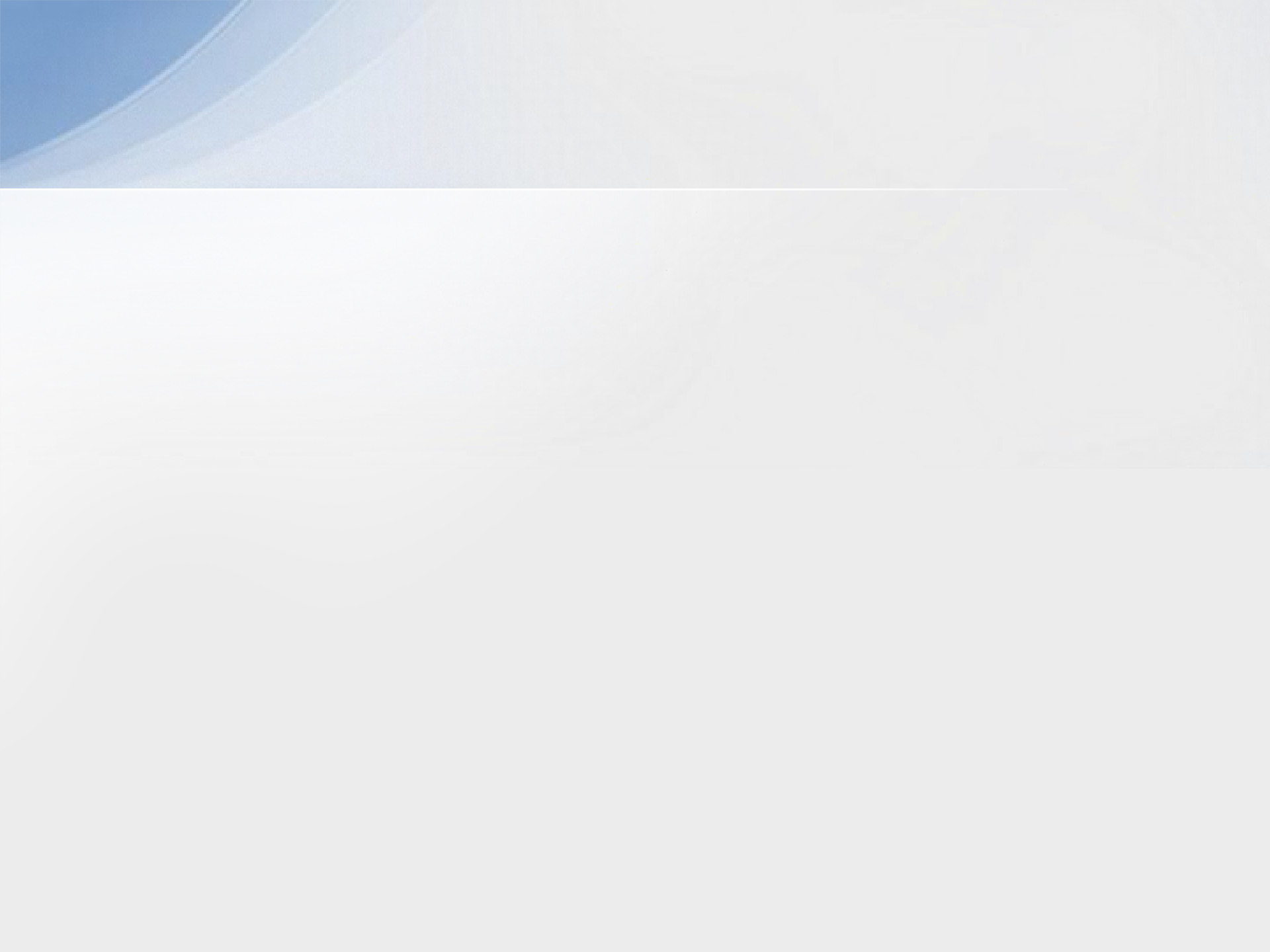
 ***d(3,1*)**





***0***

***d* (3,2) *0*** 

 



**:**

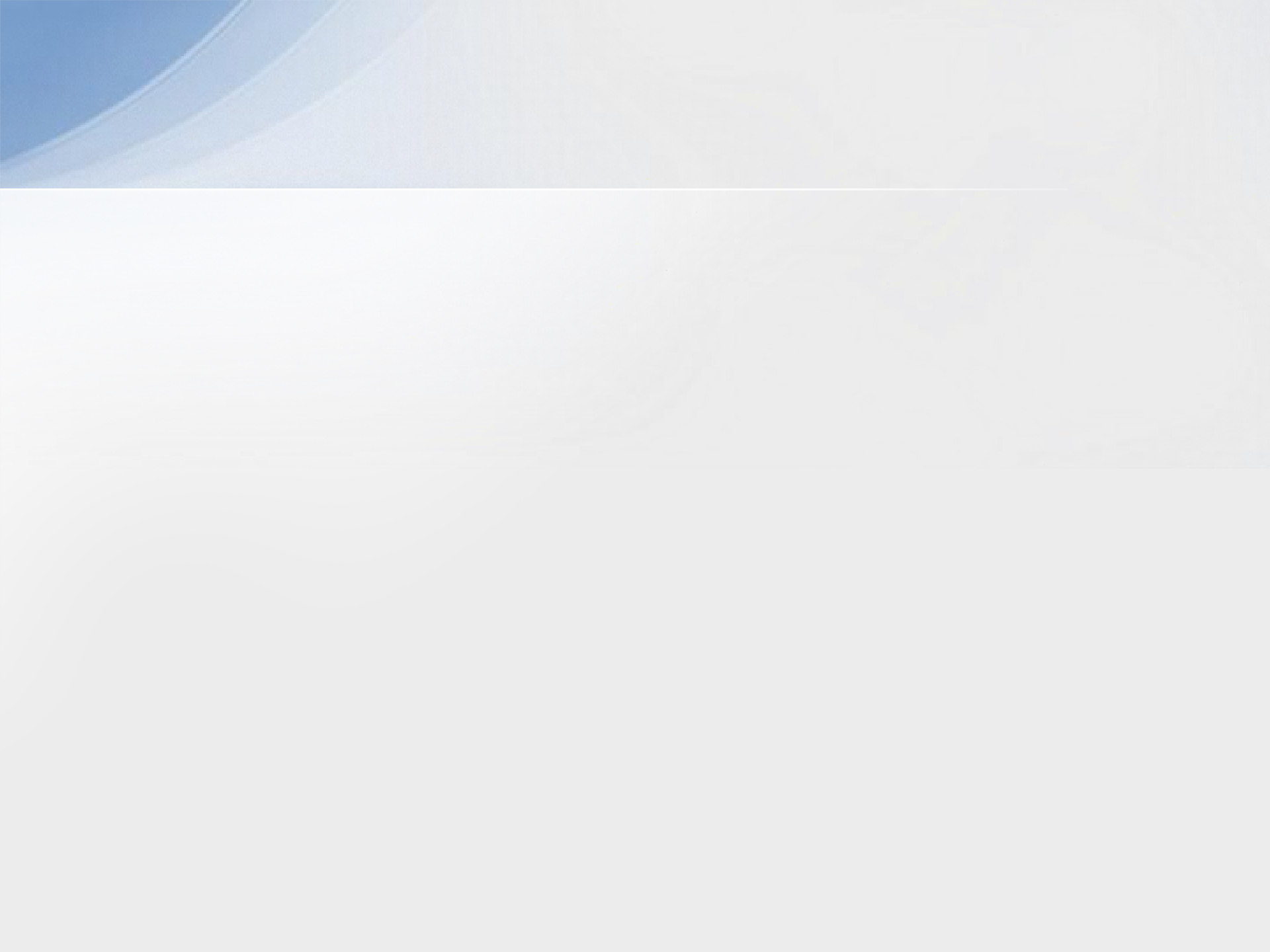
**:**

**:**

 

***d* (*n*,1) *d* (*n*,2) 0**

## Types des variables

**Mesurer la qualité d’un clustering**

* + - Métrique pour la similarité: La similarité est exprimée par le biais d’une **mesure de distance**
    - Une autre fonction est utilisée pour la mesure de la qualité
    - Les définitions de distance sont très différentes que les variables soient des intervalles **(continues), catégories, booléennes ou ordinales**
    - En pratique, on utilise souvent une pondération des variables

## Similarité entre objets

* + - * Les distances expriment une similarité
      * Ex: la *distance de Minkowski* :

*q* (| *x*  *x*

*i*1

*j*1

|*q* | *x*  *x*

*i*2

*j*2

|*q* ...| *x*  *x*

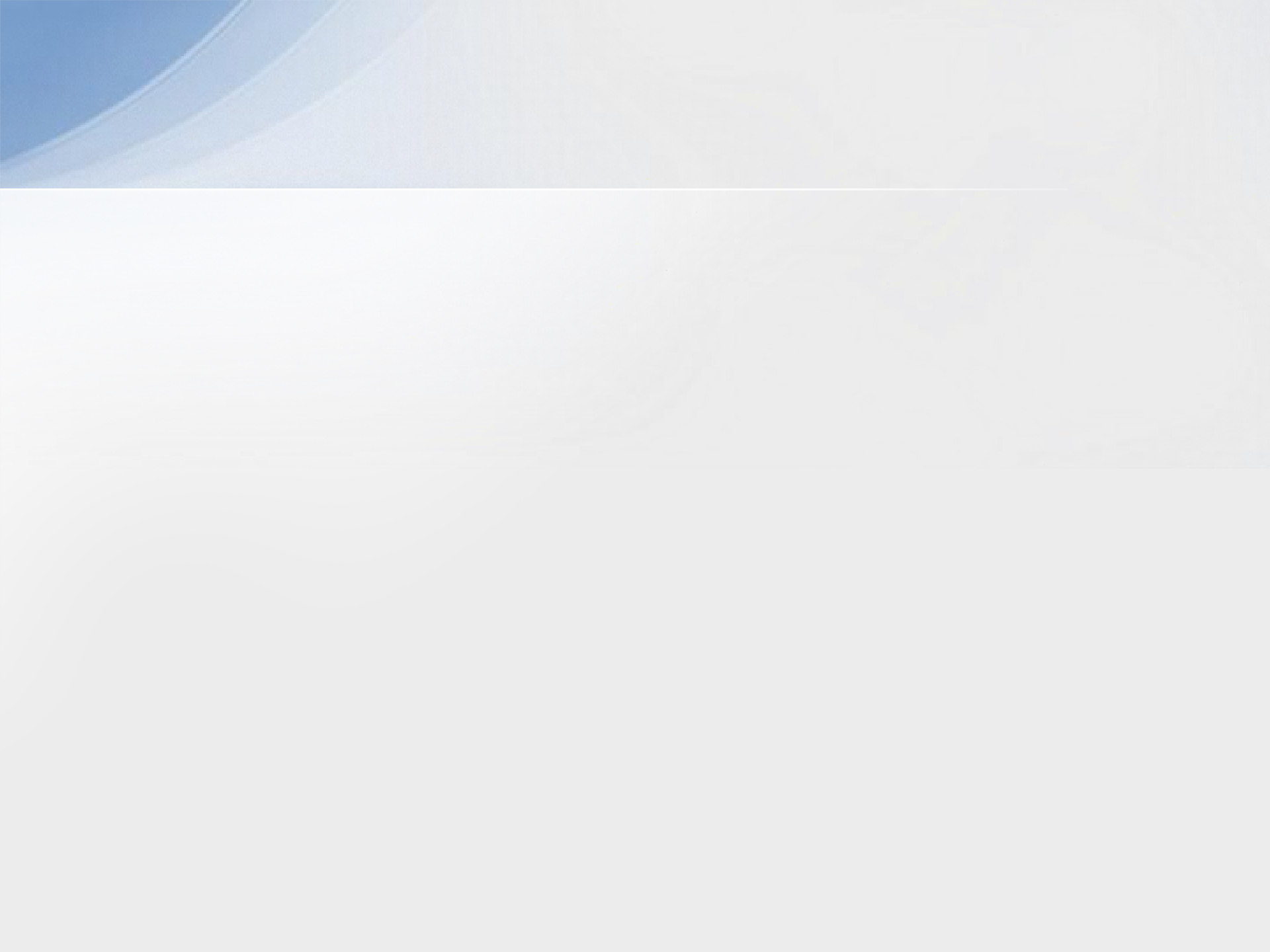
*ip*

*q*

*jp* | )

*d*(*i*,

1. 



où *i* = (*x*i1, *x*i2, …, *x*ip) et *j* = (*x*j1, *x*j2, …, *x*jp) sont deux objets *p*- dimensionnels et *q* un entier positif

* + Si *q* = *1*, *d* est la distance de Manhattan

*d*(*i*, *j*) | *xi*1  *x j*1 || *xi*2  *x j*2 |...| *xip*  *x jp* |

**Similarité entre objets(I)**

##### Si q = 2, d est la distance Euclidienne :

*d*(*i*,

###### 

(| *x*  *x*

|2 | *x*  *x*

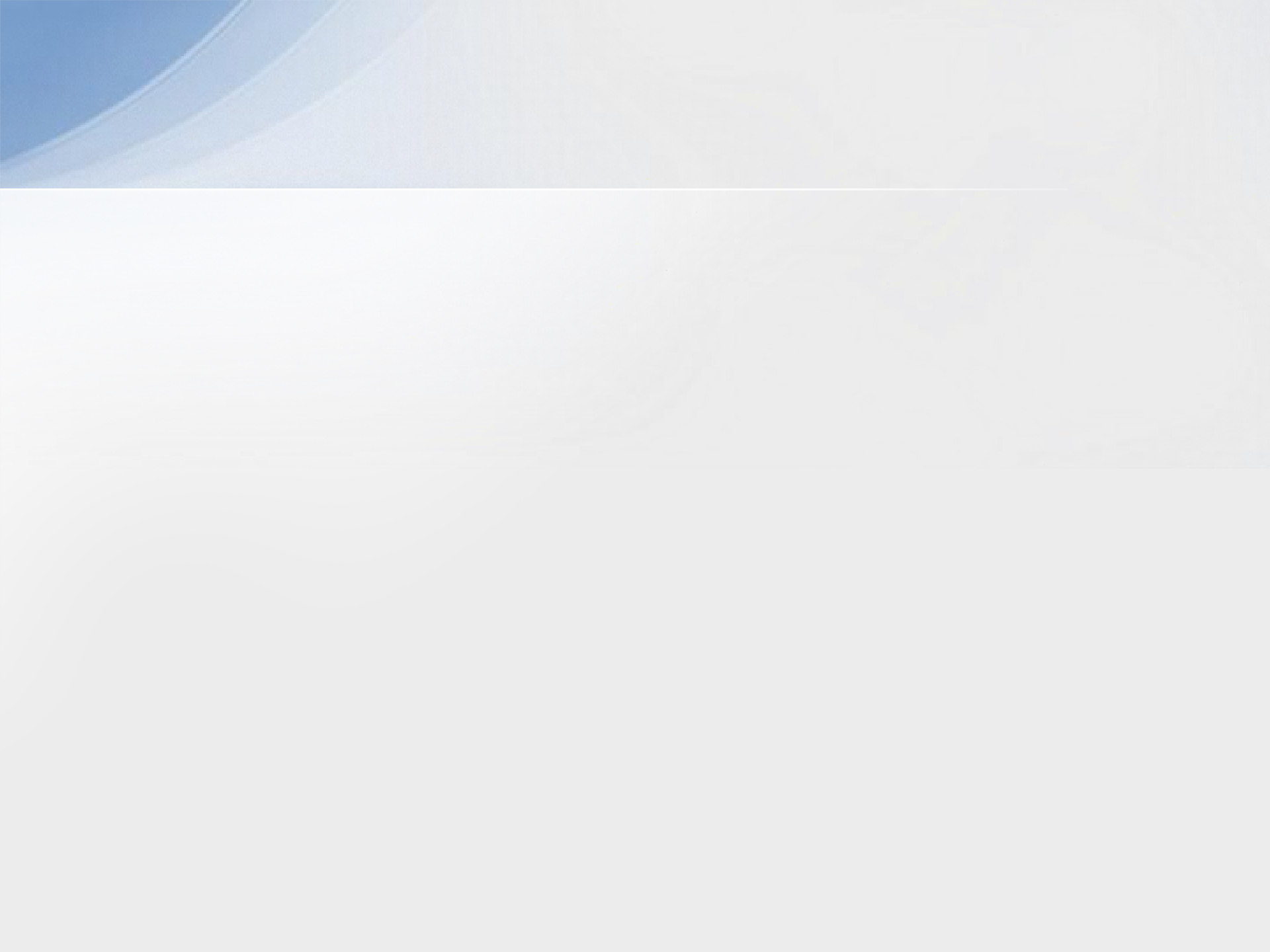
|2 ...| *x*  *x*

###### |2 )

*i*1 *j*1

*i*2 *j*2

*ip jp*



* Propriétés
  + *d(i,j)*  0
  + *d(i,i)* = 0

##### d(i,j) = d(j,i)

* + *d(i,j)*  *d(i,k)* + *d(k,j)*

# Variables binaires

* + Une table de contingence pour données binaires

**Objet *j*** a= nombre de positions

**Objet *i***

1

0

*sum*

1

*a c*

*a*  *c*

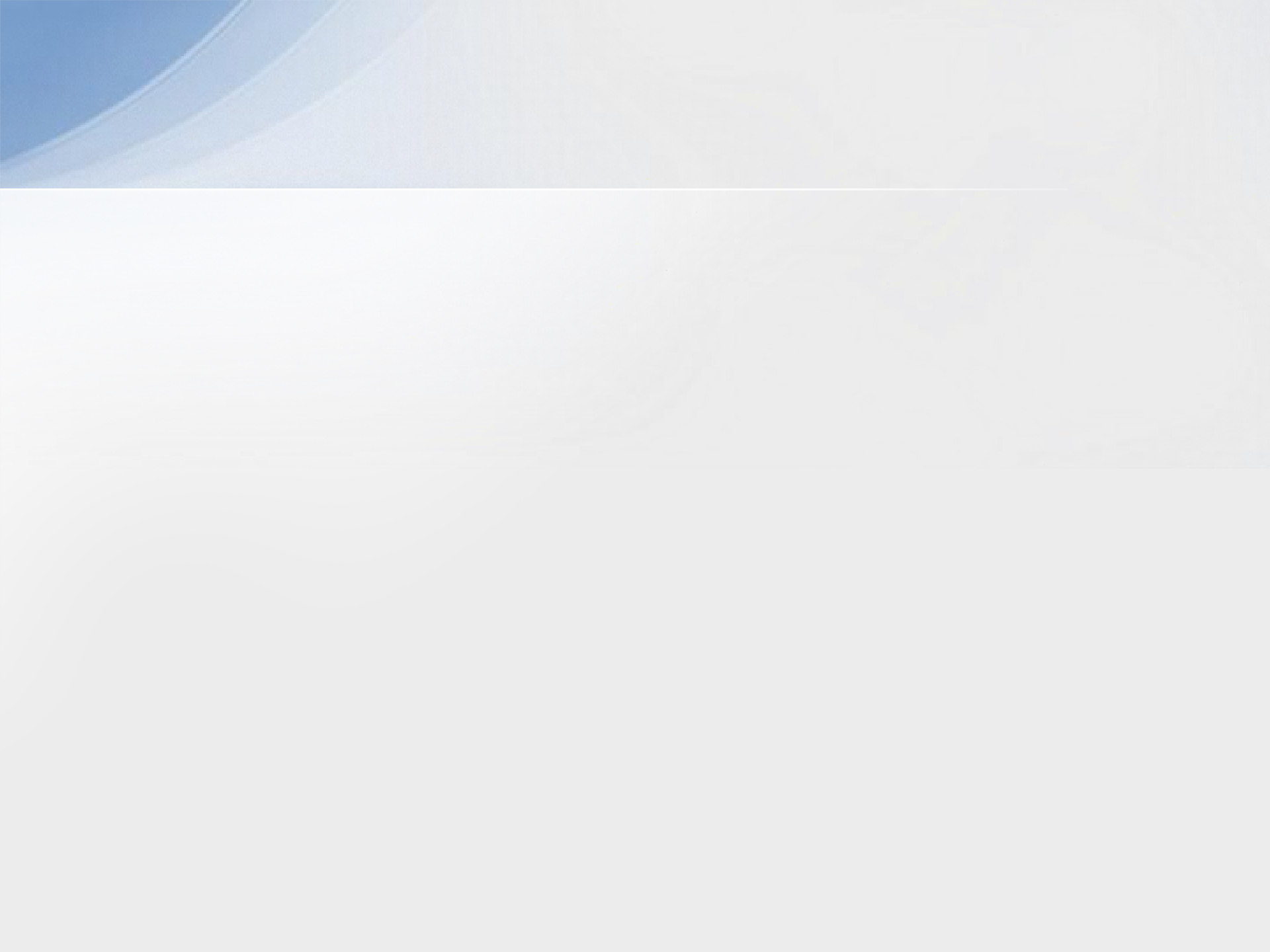
0

*b d*

*b*  *d*

*sum a* *b c*  *d p*

où i a 1 et j a 1



* + Exemple oi=(1,1,0,1, 0) et

oj=(1,0,0,0,1) a=1, b=2, c=1, d=1

**Mesures de distances**

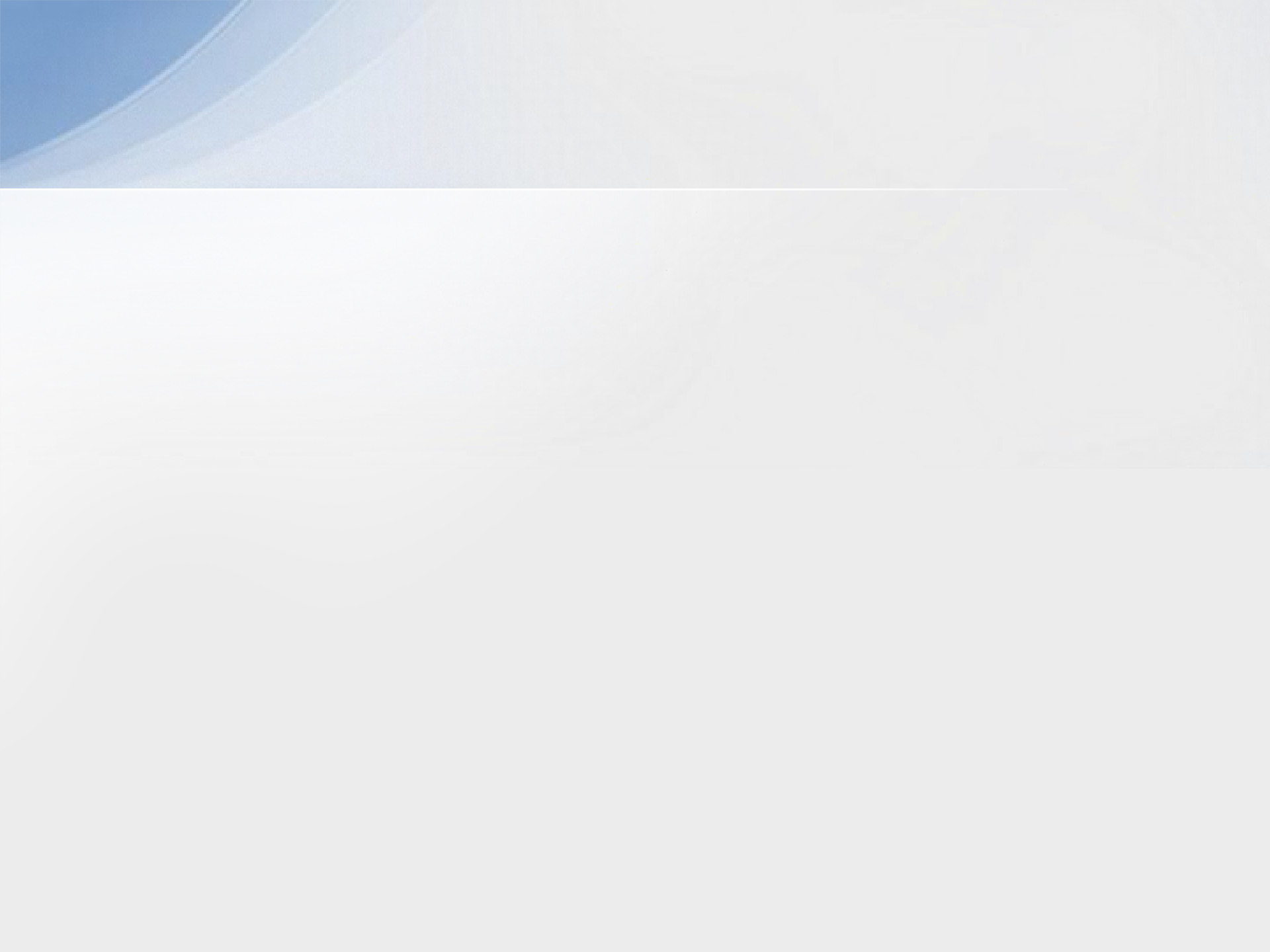
* Coefficient d’appariement (matching) simple (invariant pour variables symétriques):

*d*(*i*,

1. 

*b*  *c a* *b*  *c*  *d*

Exemple oi=(1,1,0,1,0) et oj=(1,0,0,0,1) d(oi, oj)=3/5

* + Coefficient de Jaccard d(oi, oj)=3/4

*d*(*i*,

1. 

*b*  *c*

*a* *b*  *c*

## Variables binaires (I)

* + Variable symétrique: Ex. le sexe d’une personne, i.e coder masculin par 1 et féminin par 0 c’est pareil que le codage inverse
  + Variable asymétrique: Ex. Test HIV. Le test peut être positif ou négatif (0 ou 1) mais il y a une valeur qui sera plus

présente que l’autre. Généralement, on code par 1 la

modalité la moins fréquente

* + - 2 personnes ayant la valeur 1 pour le test sont *plus similaires* que 2 personnes ayant 0 pour le test
* Exemple

### Variables binaires(II)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Nom | Sexe | Fièvre | Toux | Test-1 | Test-2 | Test-3 | Test-4 |
| Jack  Mary Jim | M  F M | Y  Y Y | N  N P | P  P N | N  N N | N  P N | N  N N |

* Sexe est un attribut symétrique
* Les autres attributs sont asymétriques
* Y et P  1, N  0, la distance n’est mesurée que sur les asymétriques

*d*(*jack*,

*mary*) 

0  1

2  0  1 

1  1

0.33

*d*(*jack*, *ji m*)

 1  1  1 

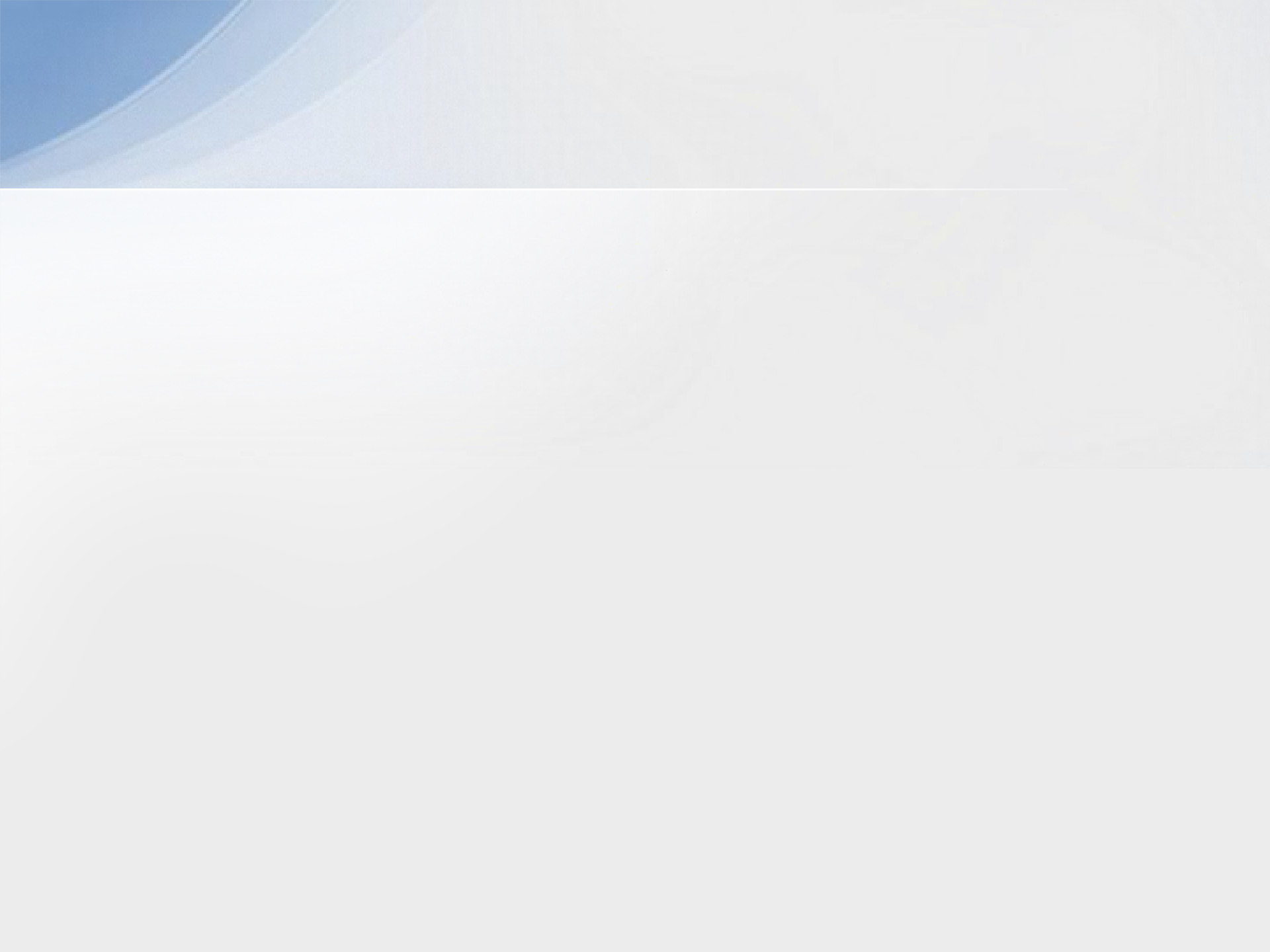
1  2

0.67

*d*(*ji m*, *mary*)

 1  1  2

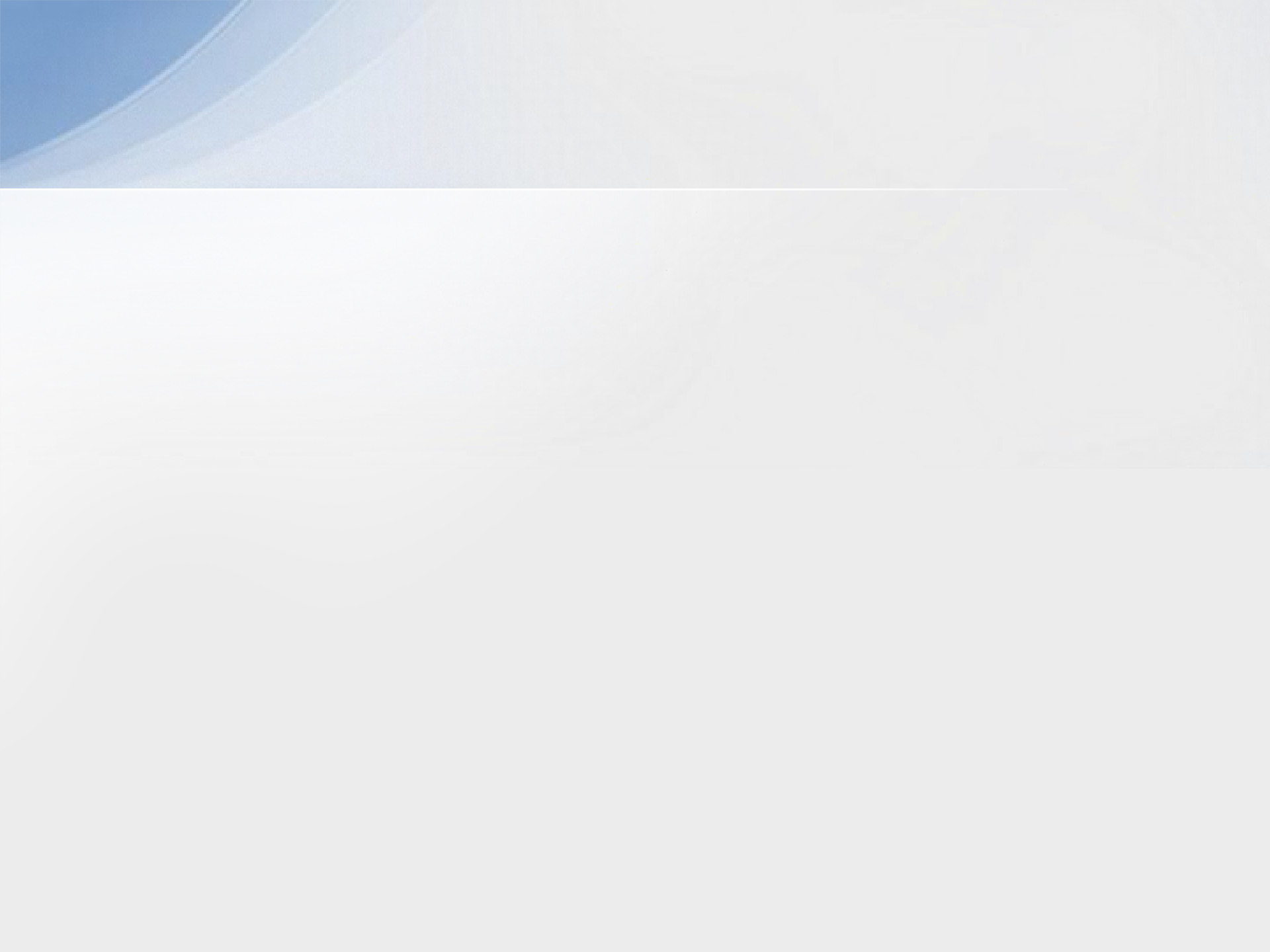
 0.75



Les plus similaires sont Jack et Maryatteints du même mal

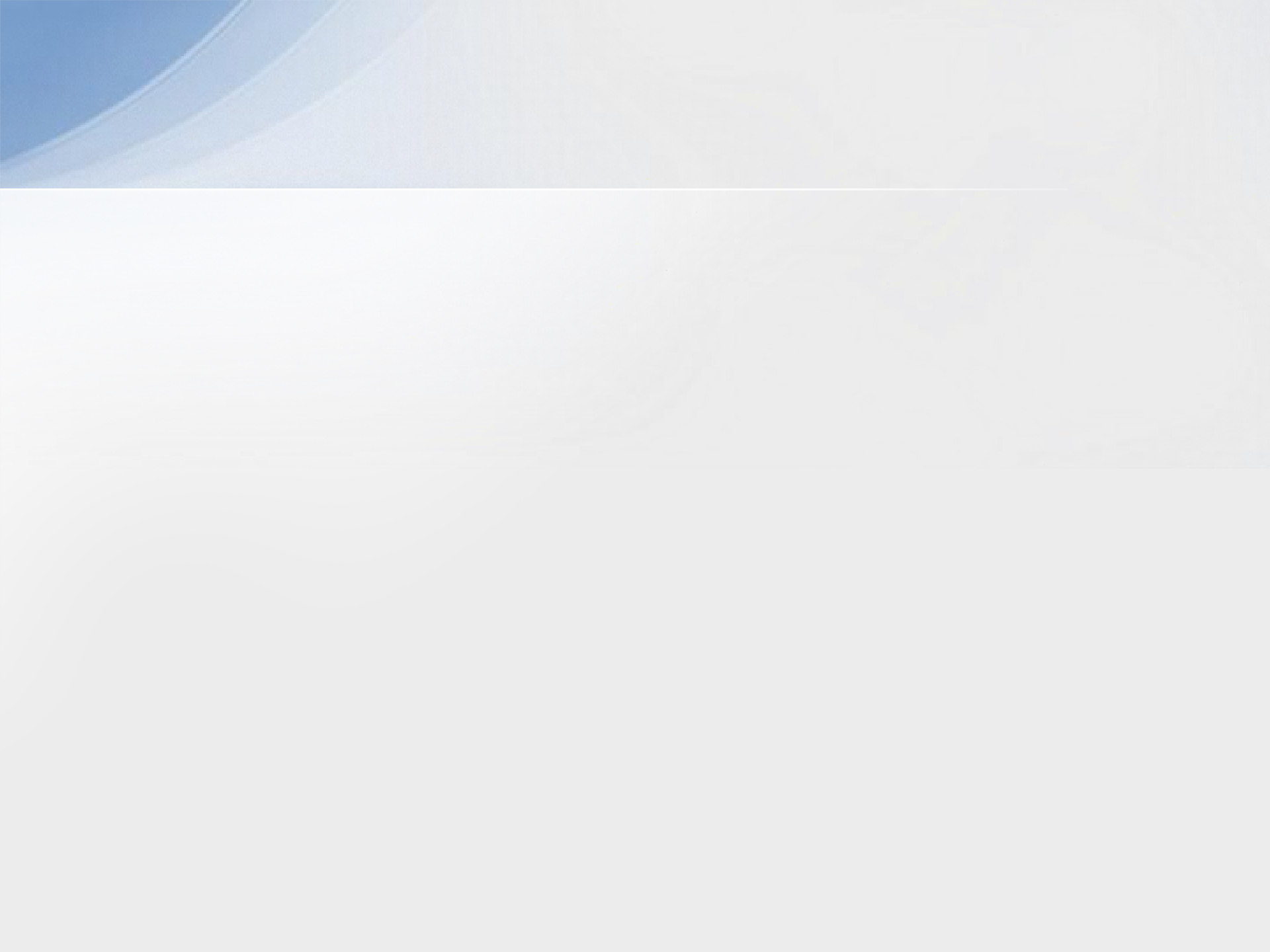
## Approches de Clustering

* Algorithmes de Partitionnement: Construire plusieurs partitions puis les évaluer selon certains critères
* Algorithmes hiérarchiques: Créer une décomposition hiérarchique des objets selon certains critères
* Algorithmes basés sur la densité: basés sur des notions de connectivité et de densité
* Algorithmes de grille: basés sur un structure à multi-niveaux de granularité
* Algorithmes à modèles: Un modèle est supposé pour chaque cluster ensuite vérifier chaque modèle sur chaque groupe pour choisir le meilleur



## Algorithmes à partitionnement

* + Construire une partition à ***k*** clusters d’une base ***D*** de ***n*** objets
  + Les *k clusters doivent* optimiser le critère choisi
    - Global optimal: Considérer toutes les k-partitions
    - Heuristic methods: Algorithmes *k-means* et *k-medoids*
      * *k-means* (MacQueen’67): Chaque cluster est représenté par son centre
      * *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw’87): Chaque cluster est représenté par un de ses objets

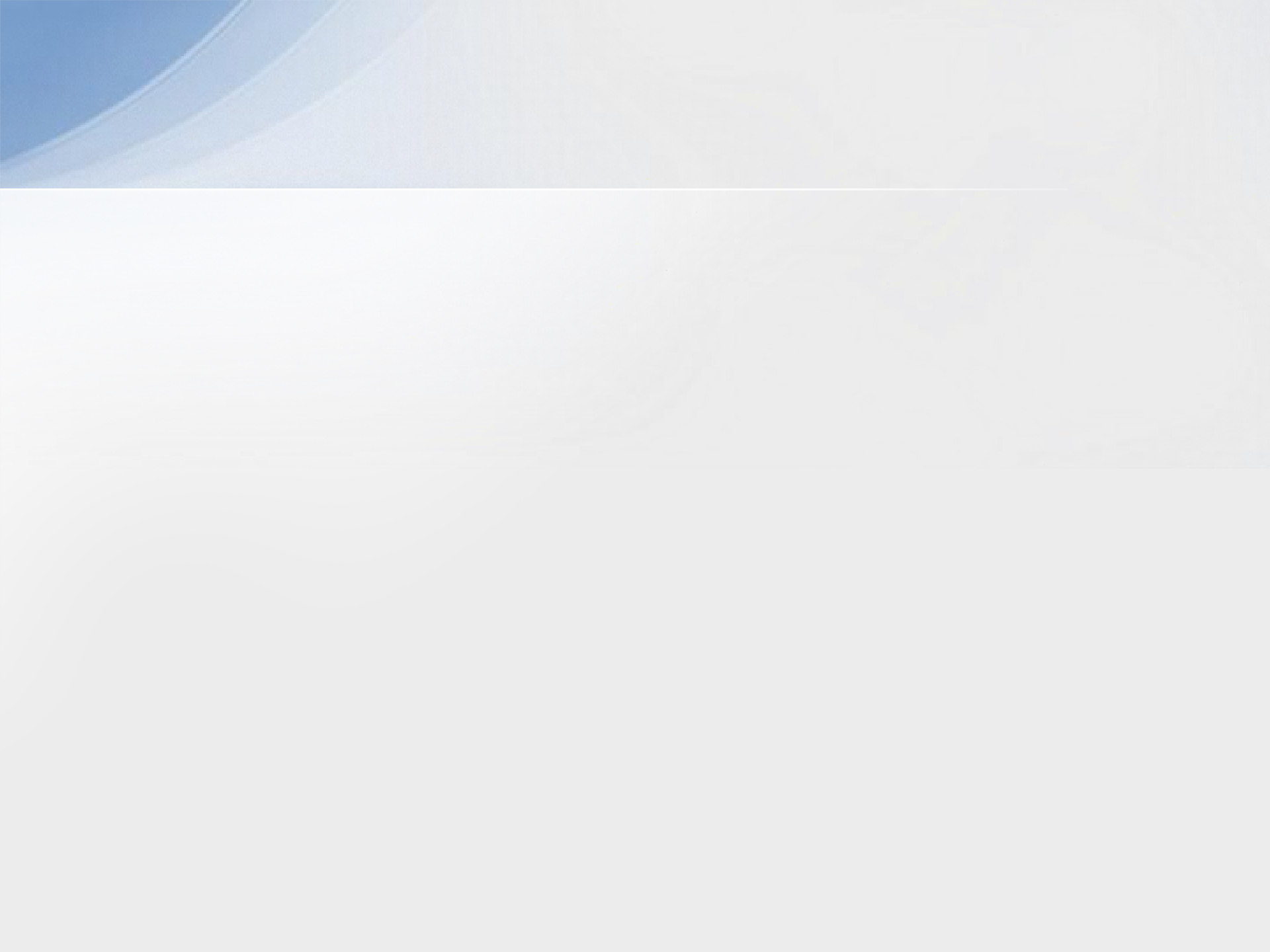
**La méthode des k-moyennes (*K-Means)***

* L’algorithme *k-means* est en 4 étapes :

1. Choisir k objets formant ainsi k clusters
2. (Ré)affecter chaque objet O au cluster Ci de centre

Mi tel que dist(O,Mi) est minimal

1. Recalculer Mi de chaque cluster (le barycentre)
2. Aller à l’étape 2 si on vient de faire une affectation



## K-Means :Exemple

* A={1,2,3,6,7,8,13,15,17}. Créer 3 clusters à partir de A
* On prend 3 objets au hasard. Supposons que c’est 1, 2 et 3. Ca donne C1={1}, M1=1, C2={2}, M2=2, C3={3} et M3=3
* Chaque objet O est affecté au cluster au milieu duquel, O est le plus proche. 6 est affecté à C3 car dist(M3,6)<dist(M2,6) et dist(M3,6)<dist(M1,6)

On a C1={1}, M1=1,

C2={2}, M2=2

C3={3, 6,7,8,13,15,17}, M3=69/7=9.86

## K-Means :Exemple (suite)

* dist(3,M2)<dist(3,M3)3 passe dans C2. Tous les autres objets ne bougent pas. C1={1}, M1=1, C2={2,3}, M2=2.5,C3={6,7,8,13,15,17} et M3= 66/6=11
* dist(6,M2)<dist(6,M3)6 passe dans C2. Tous les autres objets ne bougent pas.

C1={1}, M1=1, C2={2,3,6}, M2=11/3=3.67, C3={7,8,13,15,17}, M3= 12

* dist(2,M1)<dist(2,M2)2 passe en C1. dist(7,M2)<dist(7,M3) 7 passe en C2. Les autres ne bougent pas. C1={1,2}, M1=1.5, C2={3,6,7}, M2=5.34, C3= {8,13,15,17}, M3=13.25
* dist(3,M1)<dist(3,M2)3 passe en 1. dist(8,M2)<dist(8,M3)8 passe en 2

C1={1,2,3}, M1=2, C2={6,7,8}, M2=7, C3={13,15,17}, M3=15

Plus rien ne bouge

### Commentaires sur la méthode des *K-Means*

* Force
  + *Relativement efficace*: *O*(*tkn*), où *n* est # objets, *k* est #

clusters, et *t est* # itérations. N*k*

ormalement, *k*, *t* << *n*.

2

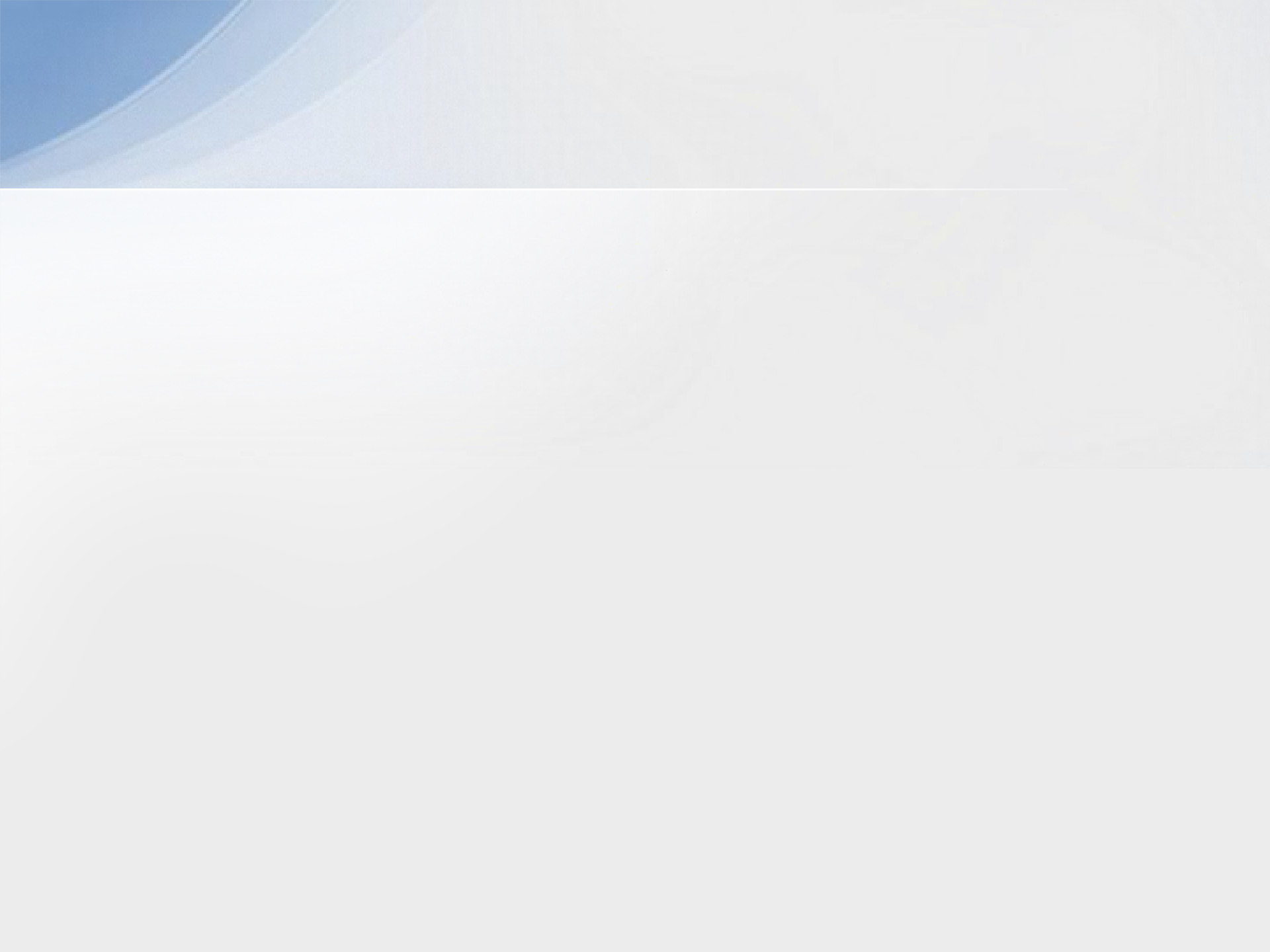
* + Tend à réduire
* Faiblesses

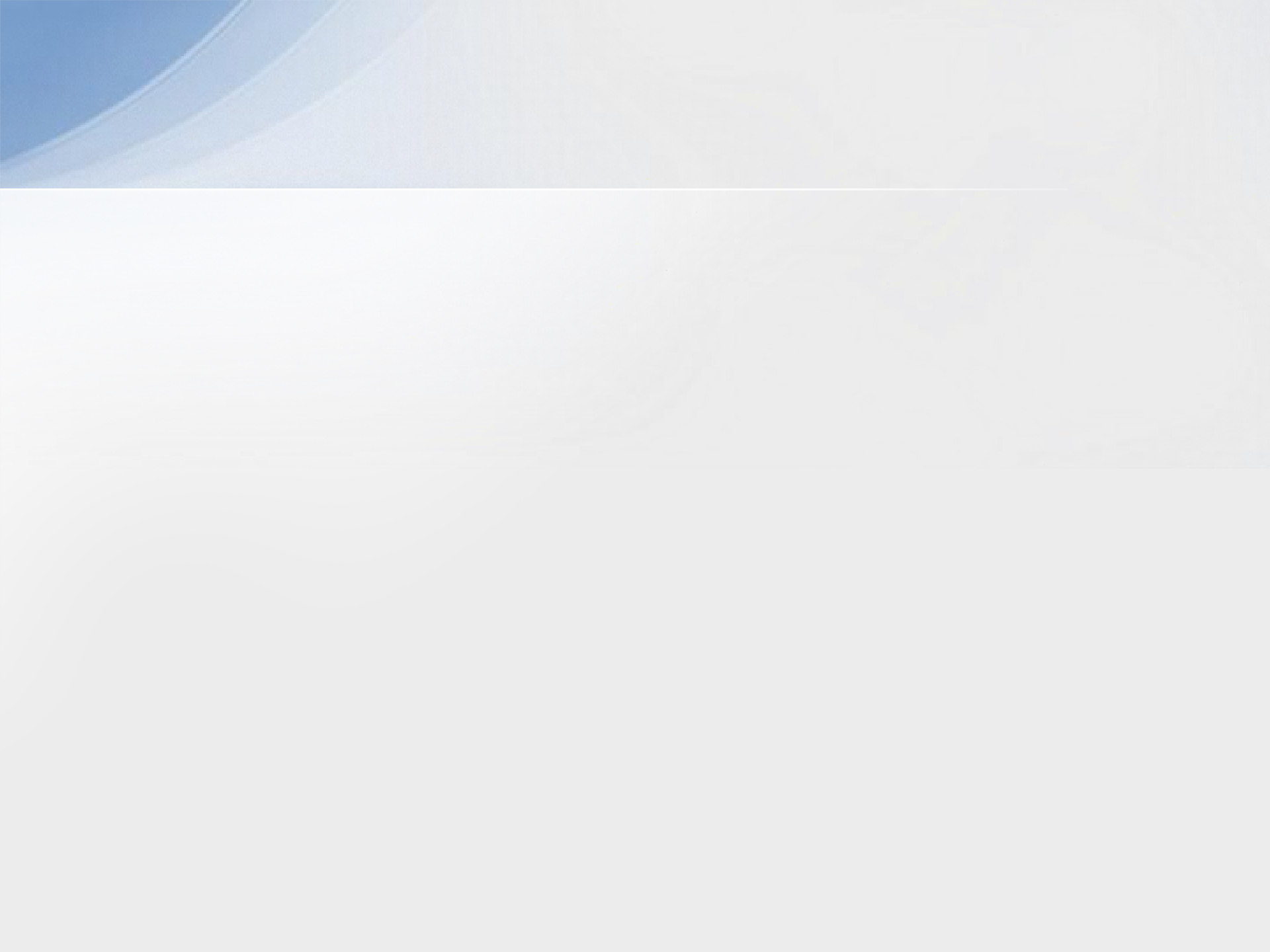
*E*   

*i* 1

*p**Ci*

*p*  *mi*

* + N’est pas applicable en présence d’attributs qui ne sont pas du type intervalle (moyenne=?)
  + On doit spécifier k (nombre de clusters)
  + Les clusters sont construits par rapports à des objets inexistants (les milieux)
  + Ne peut pas découvrir les groupes *non-convexes*

**La méthode des *K*-*Medoids (PAM)***

* Trouver des objets représentatifs (medoïdes) dans les clusters (au lieu de la moyenne)
* Principe
  + Commencer avec un ensemble de medoïdes puis itérativement remplacer un par un autre si ça permet de réduire la distance globale
  + Efficace pour des données de petite taille

## Algorithme des k-Medoides

Choisir arbitrairement k medoides Répéter

affecter chaque objet restant au medoide le plus proche Choisir aléatoirement un non-medoide Or

Pour chaque medoide Oj

Calculer le coût TC du remplacement de Oj par Or Si TC < 0 alors

Remplacer Oj par Or

Calculer les nouveaux clusters

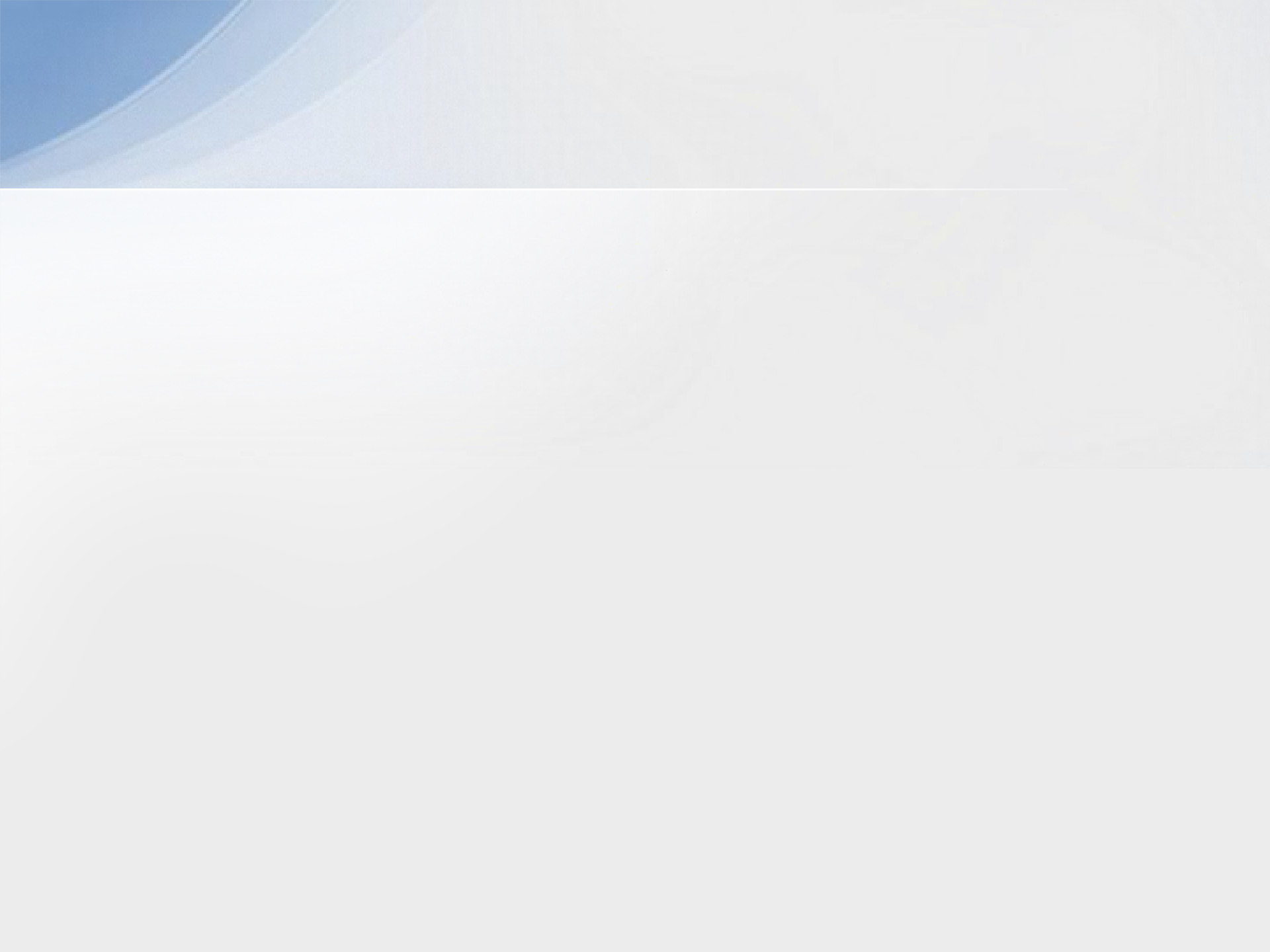
Finsi FinPour

Jusqu’à ce ce qu’il n’y ait plus de changement

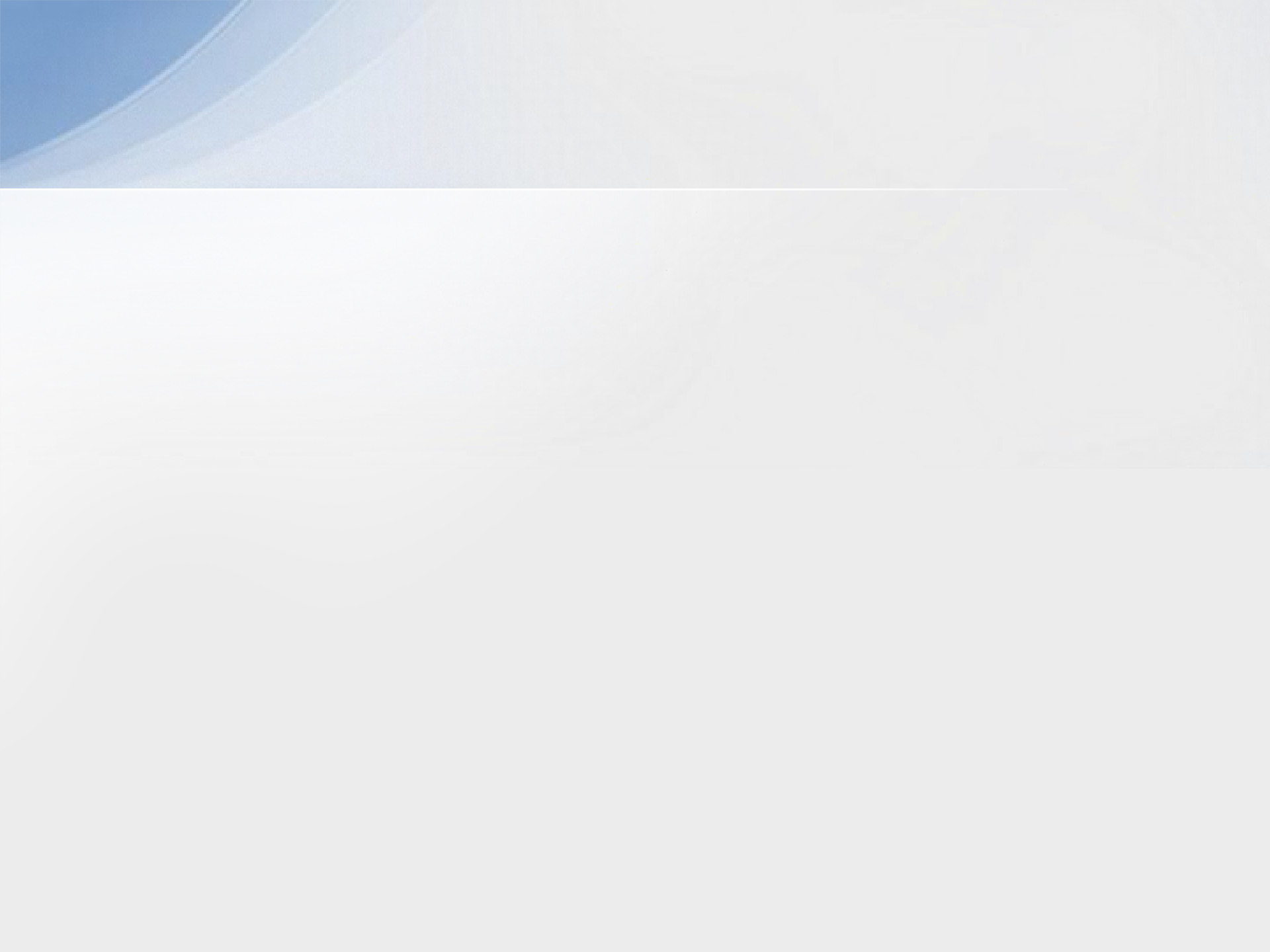
## PAM (Partitioning Around Medoids) (1987)

Choisir arbitrairement ***k*** objets représentatifs

* Pour toute paire (h,j) d’objets t.q h est choisi et j non, calculer le coût ***TCjh*** du remplacement de j par h
  + Si *TCih* < 0, ***j*** est remplacé par ***h***
  + Puis affecter chaque objet non sélectionné au medoïde qui lui est le plus similaire
* Répéter jusqu’à ne plus avoir de changements

**La méthode des *K*-*Medoids***

* TCjh représente le gain en distance globale que l’on va avoir en remplaçant h par j
* Si TCjh est négatif alors on va perdre en distance. Ca veut dire que les clusters seront plus compacts.
* TCjh=i dist(j,h)-dist(j,i)= i Cijh

**La méthode des *K*-*Medoids:* Exemple**

* + Soit A={1,3,4,5,8,9}, k=2 et M={1,8} ensemble des medoides

C1={1,3,4} et C2={5,8,9}

E{1,8}=dist(3,1)2+dist(4,1)2+dist(5,8)2+dist(9,8)2=23

* + Comparons 1 et 3M={3,8}C1={1,3,4,5} et C2={8,9} E{3,8} =dist(1,3)2+dist(4,3)2+dist(5,3)2+dist(9,8)2=10

E {3,8} - E{1,8}= -13 <0 donc le remplacement est fait.

* + Comparons 3 et 4 M={4,8} C1 et C2 inchangés et E{4,8}=dist(1,4)2+dist(3,4)2+dist(5,4)2+dist(8,9)2= 12 3 n’est pas remplacé par 4
  + Comparons 3 et 5M={5,8} C1 et C2 inchangés et E{5,8}>E{3,8}

## Clustering Hiérarchique

* + Utiliser la matrice de distances comme critère de regroupement. ***k*** n’a pas à être précisé, mais a besoin d’une condition d’arrêt

**agglomerative (AGNES)**

Etape 0

Etap

e 1

Etap

e 2

Etap

e 3

Etap

e 4

a

a b

b

a b c d e

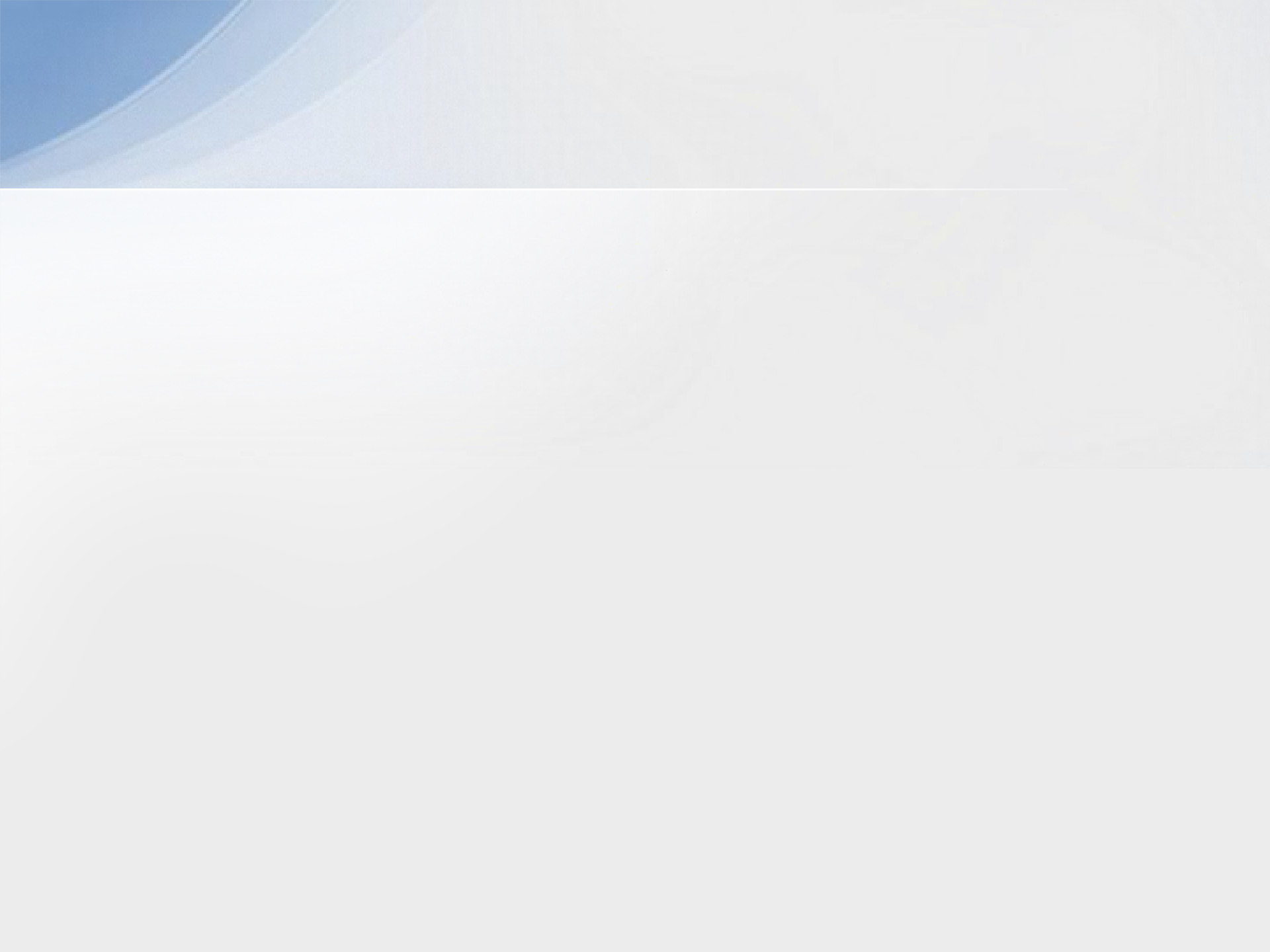
c

c d e

d

d e

e



Etap

e 4

Etap e 3

Etap e 2

Etap e 1

Etap e 0

**divisive (DIANA)**

### AGNES (Agglomerative Nesting)

* + Utilise la matrice de dissimilarité.
  + Fusionne les nœuds qui ont la plus faible dissimilarité
  + On peut se retrouver dans la situation où tous les nœuds sont dans le même groupe



10

9

8

7

6

5

4

3

2

1

0

0 1 2 3 4 5 6 7 8 9 10



10

9

8

7

6

5

4

3

2

1

0

0 1 2 3 4 5 6 7 8 9 10



10

9

8

7

6

5

4

3

2

1

0

0 1 2 3 4 5 6 7 8 9 10

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

### DIANA (Divisive Analysis)

* + - L’ordre inverse de celui d’AGNES
    - Il se peut que chaque objet forme à lui seul un groupe



10

9

8

7

6

5

4

3

2

1

0

0

1 2

3 4

5 6

7 8

9 10



10

9

8

7

6

5

4

3

2

1

0

0

1 2

3 4

5 6

7 8

9 10



10

9

8

7

6

5

4

3

2

1

0

0

1 2

3 4

5 6

7 8

9 10

## Critères de fusion-éclatement

* Exemple: pour les méthodes agglomératives, C1 et C2 sont fusionnés si
  + il existe o1  C1 et o2 C2 tels que dist(o1,o2)  seuil, ou
  + il n’existe pas o1  C1 et o2 C2 tels que dist(o1,o2) 

seuil, ou

* + distance entre C1 et C2  seuil avec

*dist*

*C*1

, *C*2  

1

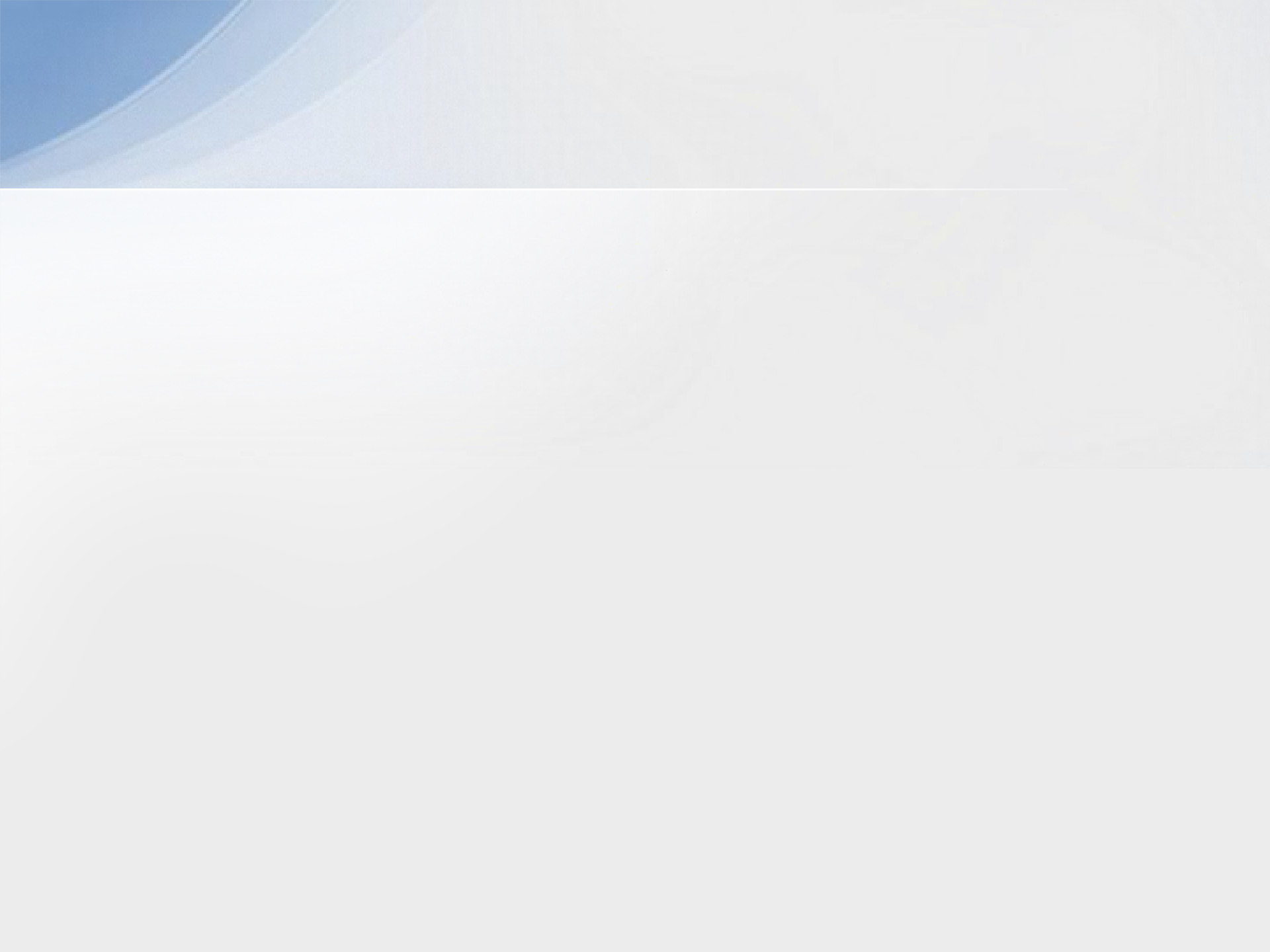
*n*1 \*



*n*2 *o*1*C*1,*o*2*C*2

*dist*(*o*1,

*o*2)

et n1=|C1|.

* Ces techniques peuvent être adaptées pour les méthodes divisives

# Méthodes d’agrégation

#### Lien minimum

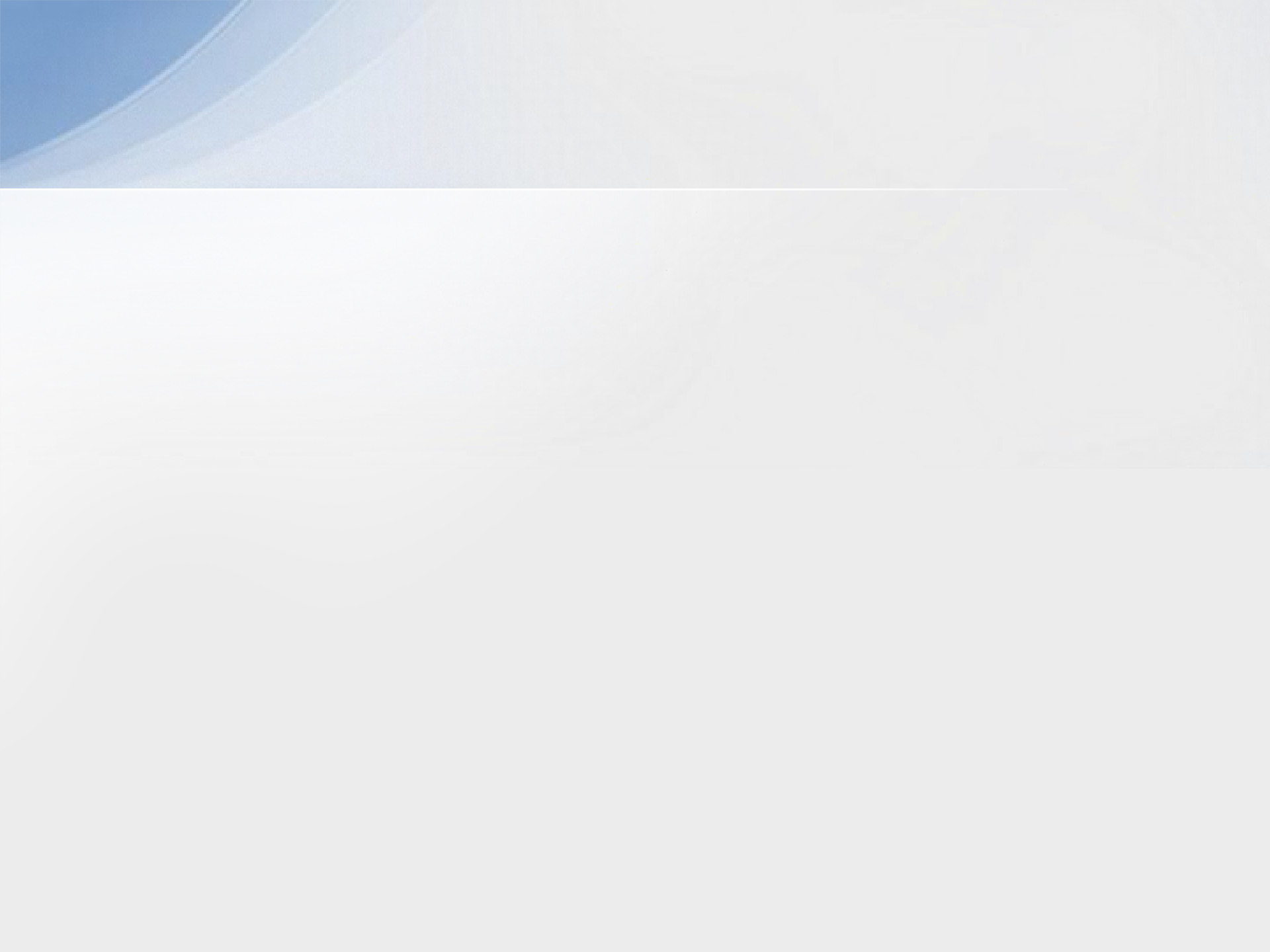
* + δ(A, B) = min{d(a, b), a∈A, b∈B}

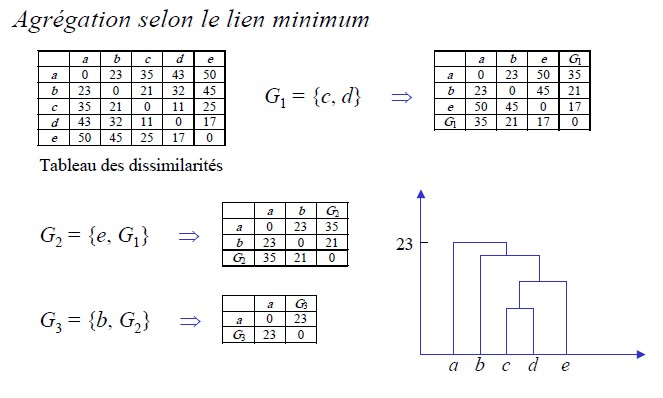
#### Lien maximum

* + δ(A, B) = max{d(a, b), a∈A, b∈B}

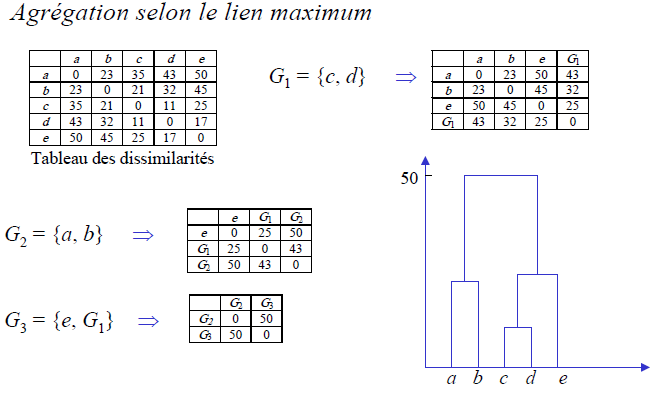
#### Distance des centres de gravité

* + δ(A, B) = d(ga, gb)

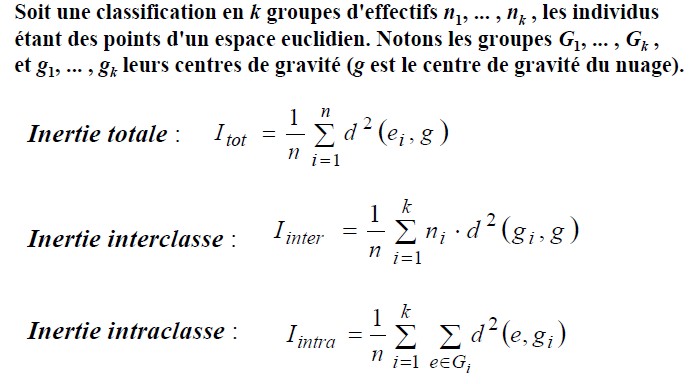
**Exemple**

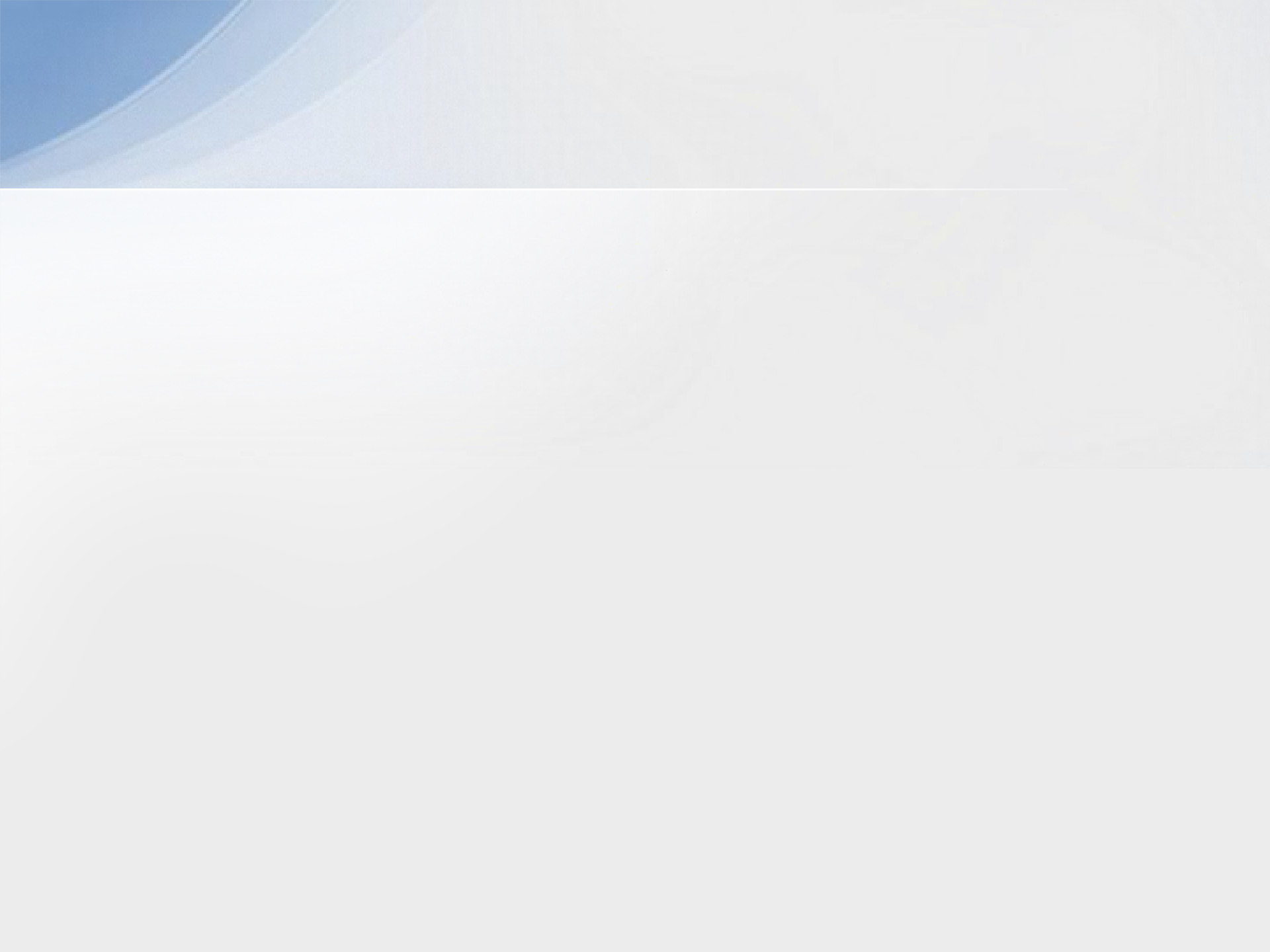


# Exemple (suite)



## Inerties interclasse et intraclasse



**Critère d’agrégation selon l’inertie**

#### Théorème de Huygens :

* + **Inertie totale = Inertie inter-classe + Inertie intra- classe**
  + Au fur et à mesure que les regroupements sont effectués, l'inertie intra-classe augmente et l'inertie interclasse diminue, car leur somme est une constante liée aux données analysées.