Clustering (Agrupamento)

- *Clustering* é uma técnica de aprendizado não-supervisionado, ou seja, quando não há uma classe associada a cada exemplo
- Os exemplos são colocados em clusters (grupos), que normalmente representam algum mecanismo existente no processo do mundo real que gerou os exemplos, fazendo com que alguns exemplos sejam mais similares entre si do que aos restantes

José Augusto Baranauskas
Departamento de Física e Matemática – FFCLRP-USP
E-mail: augusto@usp.br
URL: http://dfm.ffclrp.usp.br/~augusto

O que é Clustering?

- Dado um conjunto de objetos, colocar os objetos em grupos baseados na similaridade entre eles
- Utilizado para encontrar padrões inesperados nos dados
- Inerentemente é um problema não definido claramente

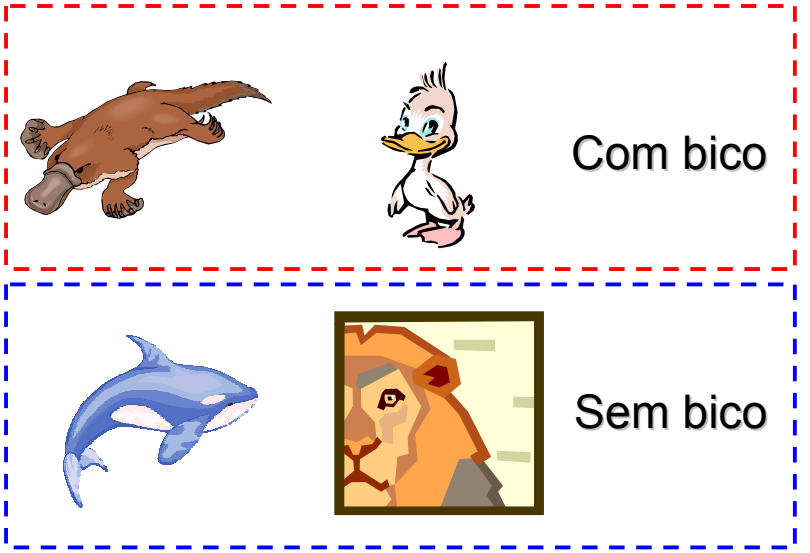
- Como agrupar os animais seguintes?




O que é Clustering?

- Dado um conjunto de objetos, colocar os objetos em grupos baseados na similaridade entre eles
- Utilizado para encontrar padrões inesperados nos dados
- Inerentemente é um problema não definido claramente

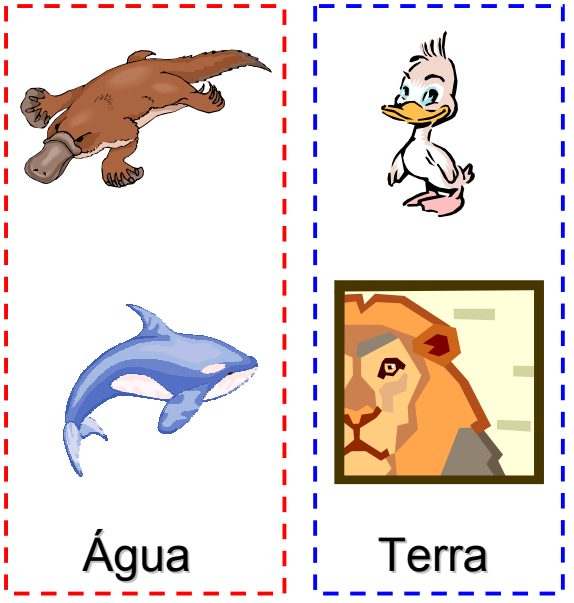
- Como agrupar os animais seguintes?



O que é Clustering?

- Dado um conjunto de objetos, colocar os objetos em grupos baseados na similaridade entre eles
- Utilizado para encontrar padrões inesperados nos dados
- Inerentemente é um problema não definido claramente

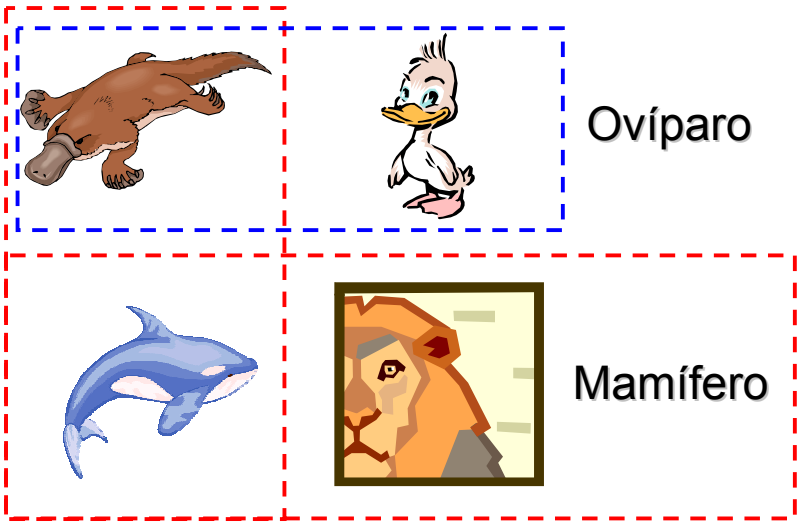
- Como agrupar os animais seguintes?



O que é Clustering?

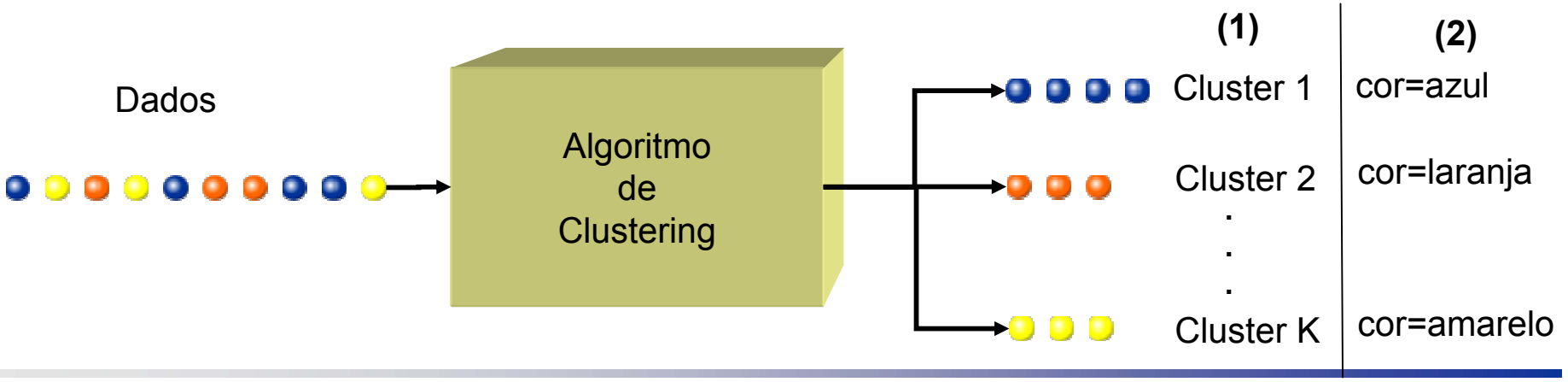
- Dado um conjunto de objetos, colocar os objetos em grupos baseados na similaridade entre eles
- Utilizado para encontrar padrões inesperados nos dados
- Inerentemente é um problema não definido claramente

- Como agrupar os animais seguintes?

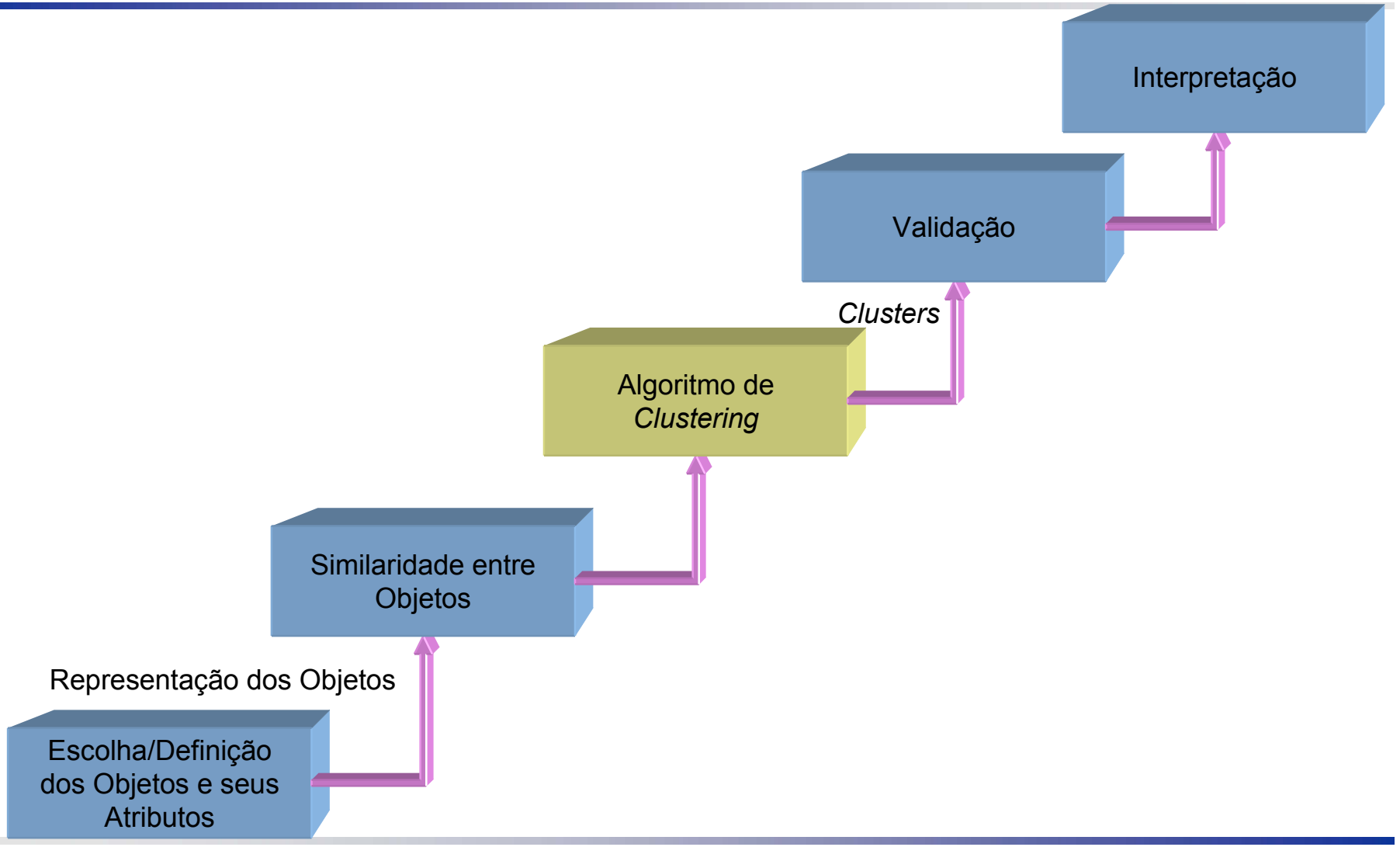


Descrição do Problema

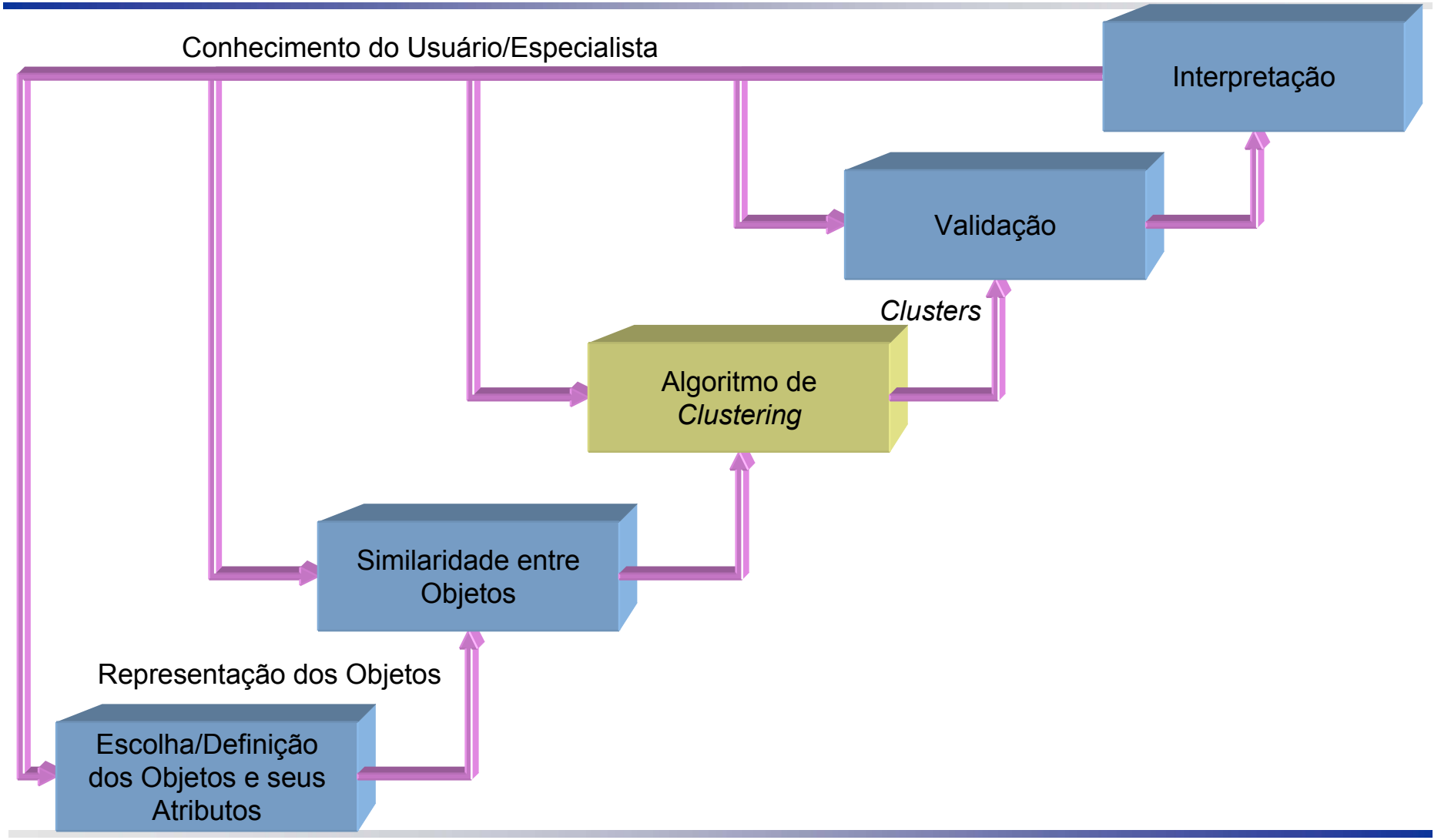
- Clustering (Agrupamento): Aprendizado não Supervisionado
- Dado um conjunto de objetos descritos por múltiplos valores (atributos)
  - (1) atribuir grupos (clusters) aos objetos particionando-os objetivamente em grupos homogêneos de maneira a:
    - Maximizar a similaridade de objetos dentro de um mesmo cluster
    - Minimizar a similaridade de objetos entre clusters distintos
  - (2) atribuir uma descrição para cada cluster formado



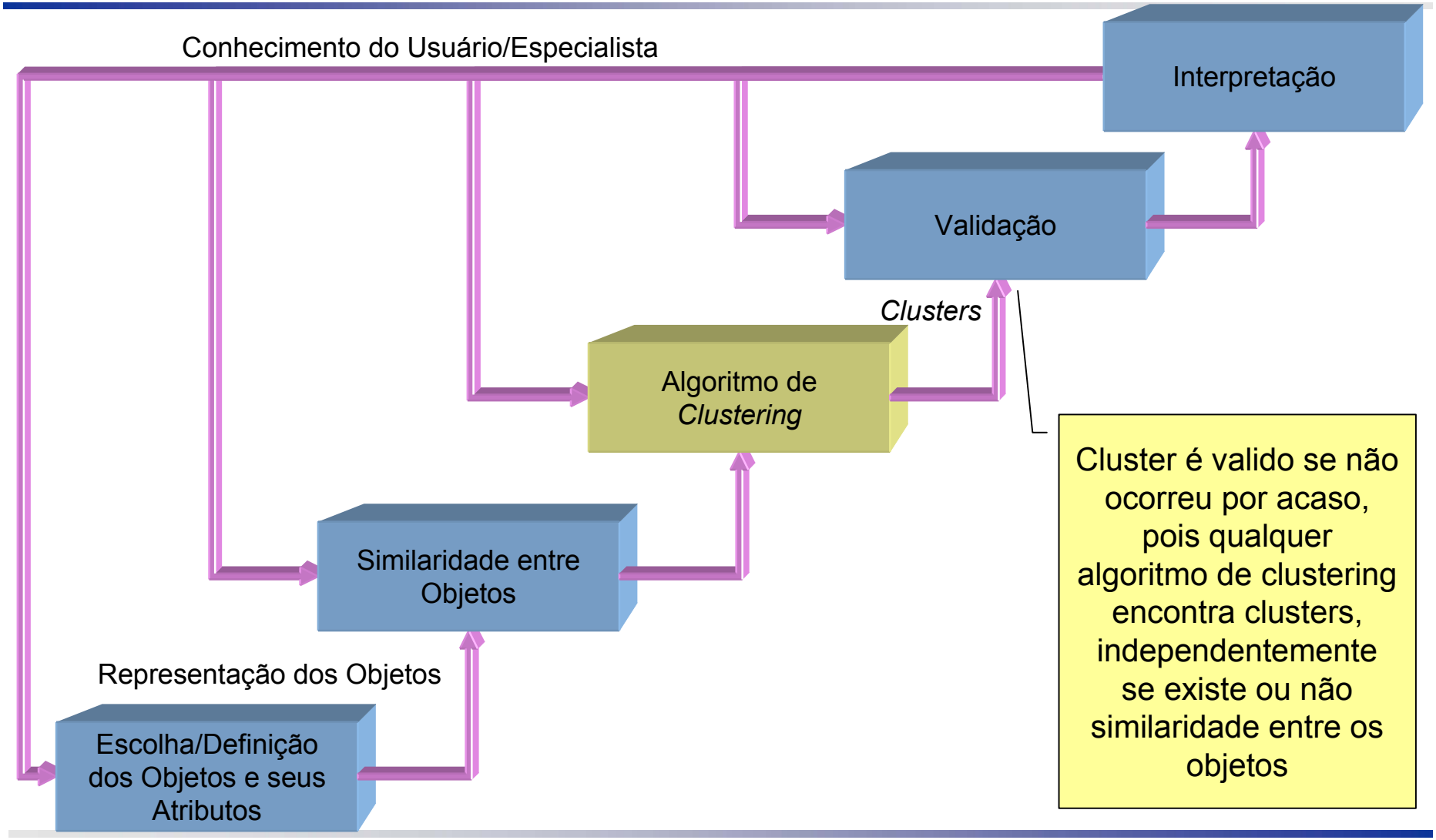
Descrição do Problema



Descrição do Problema



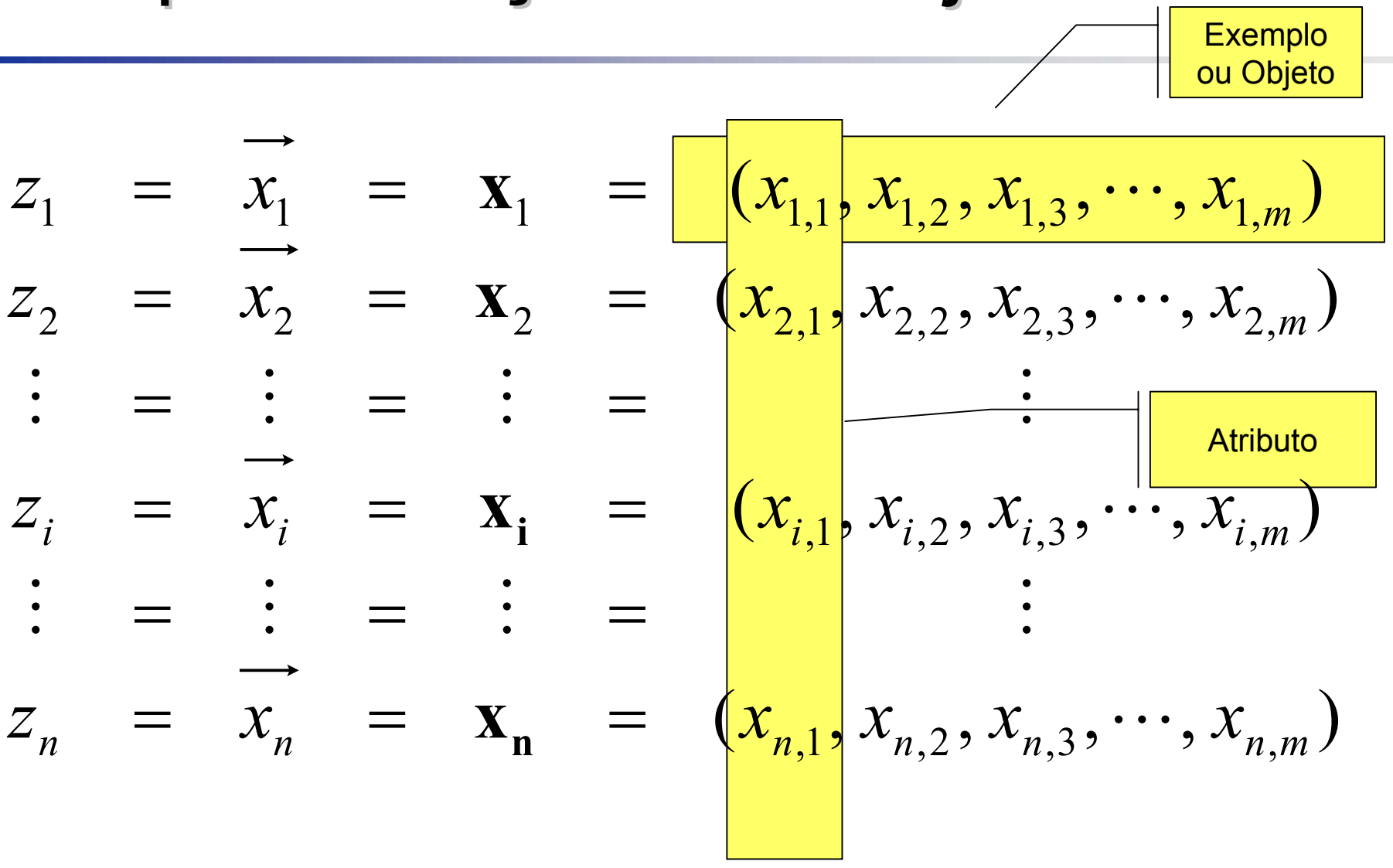
Descrição do Problema



Representação dos Objetos

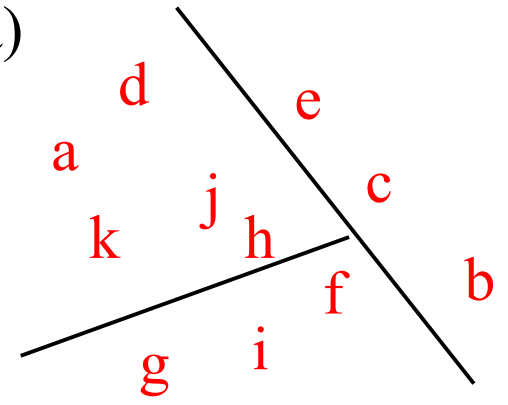
- Cada objeto (dado) de entrada para o algoritmo é representado usualmente por um vetor de **atributos**
  (objeto = dado = exemplo = tupla = registro)
- Tipos de Atributos
  - Numérico (inteiro, real)
  - Categórico (booleano, conjunto de valores)
- Por exemplo: Amostra de dados clínicos (Objeto: Paciente)
  - Idade (atributo numérico: inteiro)
  - Peso (atributo numérico: real)
  - Sexo (atributo categórico: masculino, feminino)
  - Cor da pele (atributo categórico: branca, marrom, amarela, preta)
  - Doente? (atributo booleano: Sim, Não)
- Deve também incluir um método para calcular a similaridade (ou a distância) entre os objetos

Representação dos Objetos

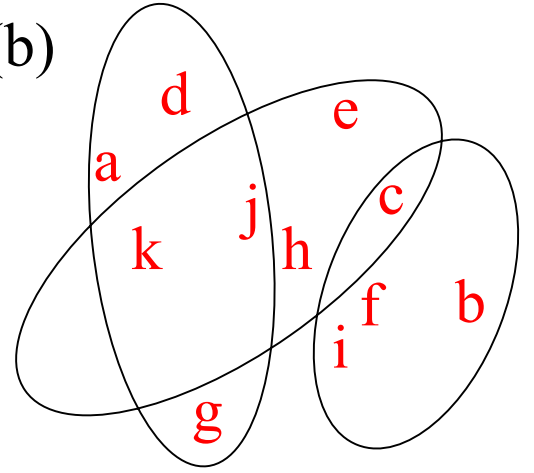


Representação de Clusters

(a)



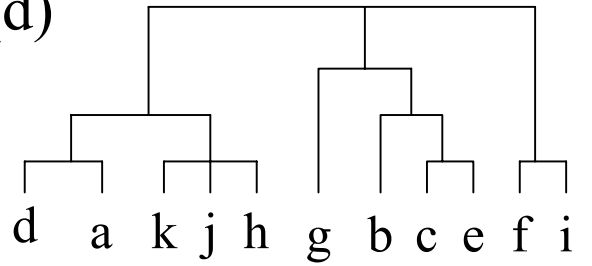
(b)



(c)

	1	2	3
a	0.6	0.1	0.3
b	0.1	0.8	0.1
c	0.3	0.3	0.4
...			

(d)



Avaliação de Clusters

- Avaliação Tradicional:

$$\text{Qualidade do Cluster} = \frac{\text{Distância Inter} - \text{Cluster}}{\text{Distância Intra} - \text{Clusters}}$$

  - Não aplicável a domínios hierárquicos
- Avaliação para Clusters Hierárquicos
  - Poucos clusters
    - Cobertura grande → boa generalidade
  - Descrição de clusters grandes
    - Mais atributos → maior poder de inferência
  - Mínima (nenhuma) sobreposição (intersecção) entre clusters
    - Clusters mais distintos → conceitos melhor definidos

Calculando a Distância

- A distância é o método mais natural para dados numéricos
- Valores pequenos indicam maior similaridade
- Métricas de Distância
  - Euclideana
  - Manhattan
  - Etc.
- Não generaliza muito bem para dados não numéricos
  - Qual a distância entre “masculino” e “feminino”?

Normalização

- Considerando a distância Euclidiana, mais utilizada nas aplicações, um problema ocorre quando um dos atributos assume valores em um intervalo relativamente grande, podendo sobrepujar os demais atributos
- Por exemplo, se uma aplicação tem apenas dois atributos A e B e A varia entre 1 e 1000 e B entre 1 e 10, então a influência de B na função de distância será sobrepujada pela influência de A
- Portanto, as distâncias são freqüentemente **normalizadas** dividindo a distância de cada atributo pelo intervalo de variação (i.e. diferença entre valores máximo e mínimo) daquele atributo
- Assim, a distância para cada atributo é **normalizada** para o intervalo [0,1]

Normalização

- De forma a evitar ruídos, é também comum:
  - dividir pelo desvio-padrão ao invés do intervalo ou
  - “cortar” o intervalo por meio da remoção de uma pequena porcentagem (e.g. 5%) dos maiores e menores valores daquele atributo e somente então definir o intervalo com os dados remanescentes
    - Também é possível mapear qualquer valor fora do intervalo para os valores mínimo ou máximo para evitar valores normalizados fora do intervalo [0,1]
- Conhecimento do domínio pode freqüentemente ser utilizada para decidir qual método é mais apropriado

Métricas

$$\mathbf{x_i} = (x_{i,1}, x_{i,2}, x_{i,3}, \cdots, x_{i,m})$$

$$\mathbf{x_j} = (x_{j,1}, x_{j,2}, x_{j,3}, \cdots, x_{j,m})$$

- Minkowski (L_p): escolha de *p* depende da ênfase que se deseja dar a grandes diferenças entre dimensões

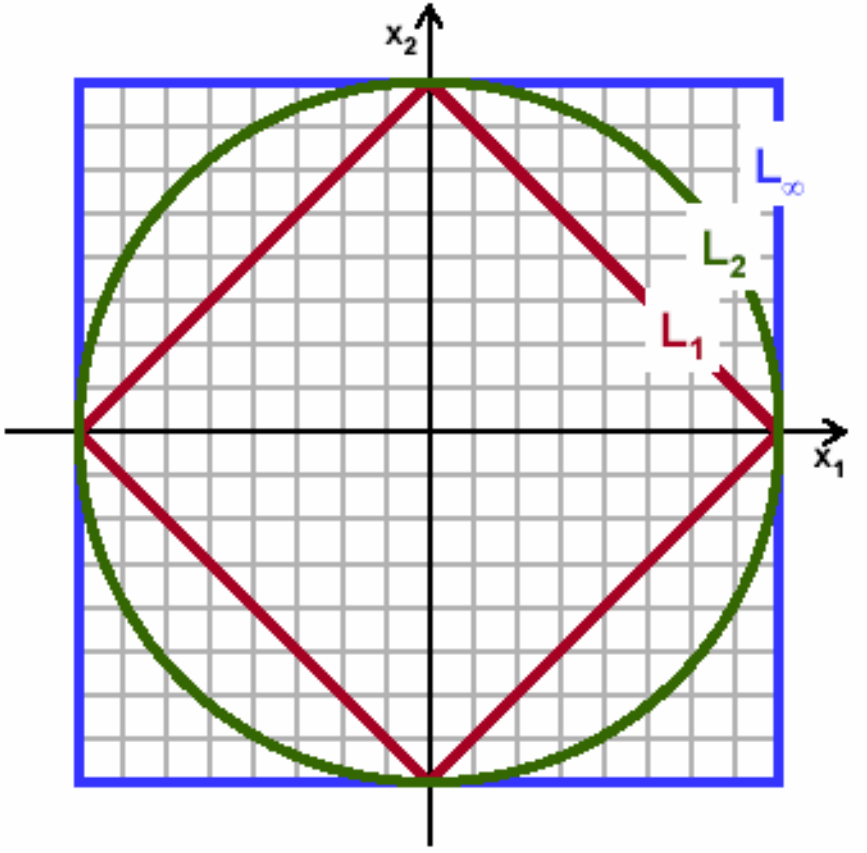
$$dist_p(\mathbf{x_i}, \mathbf{x_j}) = \left[\sum_{r=1}^{m} (x_{i,r} - x_{j,r})^p \right]^{1/p}$$

- Manhattan/City-Block (L_1): se atributos binários, é conhecida como distância Hamming

$$dist_M(\mathbf{x_i}, \mathbf{x_j}) = \sum_{r=1}^{m} \left| x_{i,r} - x_{j,r} \right|$$

- Euclidiana (L_2)

$$dist_2(\mathbf{x_i}, \mathbf{x_j}) = \left[\sum_{r=1}^{m} (x_{i,r} - x_{j,r})^2 \right]^{1/2}$$



Contornos de distâncias iguais

Métricas

$$\mathbf{x_i} = (x_{i,1}, x_{i,2}, x_{i,3}, \cdots, x_{i,m})$$
$$\mathbf{x_j} = (x_{j,1}, x_{j,2}, x_{j,3}, \cdots, x_{j,m})$$

- Camberra

$$dist_{\mathrm{Ca}}(\mathbf{x_i}, \mathbf{x_j}) = \sum_{r=1}^{m} \frac{\left| x_{i,r} - x_{j,r} \right|}{\left| x_{i,r} + x_{j,r} \right|}$$

- Chebychev

$$dist_{\mathrm{Ch}}(\mathbf{x_i}, \mathbf{x_j}) = \max_{r=1}^{m} \left| x_{i,r} - x_{j,r} \right|$$

- Correlação

$$dist_{\mathrm{Co}}(\mathbf{x_i}, \mathbf{x_j}) = \frac{\sum_{r=1}^{m} (x_{i,r} - \bar{x}_i)(x_{j,r} - \bar{x}_j)}{\sqrt{\sum_{r=1}^{m} (x_{i,r} - \bar{x}_i)^2 \sum_{r=1}^{m} (x_{j,r} - \bar{x}_j)^2}}$$

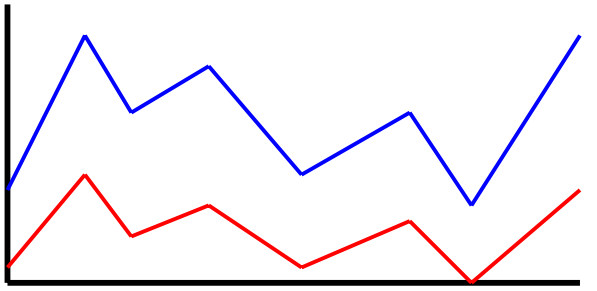
$\bar{x}_i = \bar{x}_j$ média dos valores do atributo X_r

Métricas

$$\mathbf{x_i} = (x_{i,1}, x_{i,2}, x_{i,3}, \cdots, x_{i,m})$$

$$\mathbf{x_j} = (x_{j,1}, x_{j,2}, x_{j,3}, \cdots, x_{j,m})$$

- Correlação Pearson:
  - Remove efeitos de magnitude; intervalo [-1.0, 1.0]
  - -1.0 = inversamente correlacionado, 0.0 = sem correlação, 1.0 = perfeitamente correlacionado
- No exemplo, as linhas azul e vermelha têm alta correlação, mesmo que a distância entre as linhas seja significante



$$dist_{\text{Pearson}}(\mathbf{x_i}, \mathbf{x_j}) = \frac{\sum_{r=1}^{m} x_{i,r} x_{j,r} - \left(\sum_{r=1}^{m} x_{i,r} \sum_{r=1}^{m} x_{j,r} \right) / m}{\sqrt{\left(\sum_{r=1}^{m} x_{i,r}^2 - (\sum_{k=1}^{m} x_{i,r})^2 / m \right) \left(\sum_{r=1}^{m} x_{j,r}^2 - (\sum_{r=1}^{m} x_{j,r})^2 / m \right)}}$$

Métricas

$$\mathbf{x_i} = (x_{i,1}, x_{i,2}, x_{i,3}, \cdots, x_{i,m})$$
$$\mathbf{x_j} = (x_{j,1}, x_{j,2}, x_{j,3}, \cdots, x_{j,m})$$

- O método mais simples para atributos categóricos é o seguinte

$$overlap(x_{i,r}, x_{j,r}) = \begin{cases} 1 & \text{se } x_{i,r} \text{ ou } x_{j,r} \text{ são desconhecidos} \\ 1 & \text{se } x_{i,r} \neq x_{j,r} \\ 0 & \text{se } x_{i,r} = x_{j,r} \end{cases}$$

$$dist_{\text{Cat}}(\mathbf{x_i}, \mathbf{x_j}) = \sum_{r=1}^{m} overlap(x_{i,r}, x_{j,r})$$

Métrica Heterogênea

- Heterogeneous Euclidean-Overlap Metric: HEOM
- Utiliza normalização no intervalo [0,1]
- Uma forma de lidar com aplicações com atributos nominais e contínuos consiste em utilizar uma função de distância heterogênea que utiliza funções diferentes para tipos de atributos diferentes

$$dist_H(x_{i,r}, x_{j,r}) = \begin{cases} overlap(x_{i,r}, x_{j,r}) & \text{se atributo } X_r \text{ é nominal} \\ \dfrac{|x_{i,r} - x_{j,r}|}{máx(X_r) - mín(X_r)} & \text{se atributo } X_r \text{ é contínuo} \end{cases}$$

$$dist_{HEOM}(\mathbf{x_i}, \mathbf{x_j}) = \sqrt{\sum_{r=1}^{m} dist_H(x_{i,r}, x_{j,r})^2}$$

Calculando Similaridade Booleana

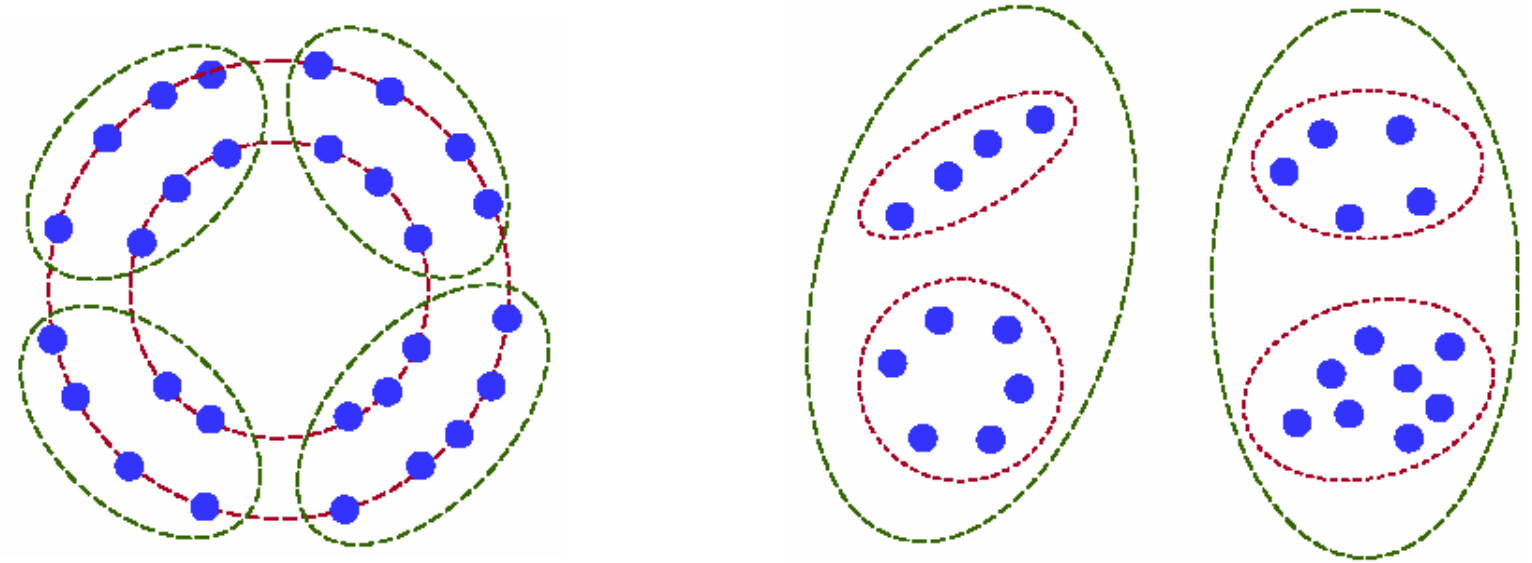
- Dados dois vetores booleanos X e Y, seja A o número de atributos onde ambos vetores assumem 1, etc. como mostrado abaixo
- Dois métodos para similaridade são dados ao lado
- Podem ser generalizados para dados categóricos

		Y[j]	
		1	0
X[j]	1	A	B
	0	C	D

- Correlação = (A+D)/(A+B+C+D)
- Coef. Jaccard = A / (A+B+C+D)
  - Utilizado quando a ausência de um valor verdadeiro não significa similaridade
  - **Exemplo:**
    - Suponha que estamos realizando um trabalho de filogenética estrutural e X[j] é verdadeiro se o organismo tem asas
    - Dois organismos não são mais similares se ambos não têm asas
    - Dessa forma, o coeficiente de Jaccard é mais natural que o coeficiente de correlação neste caso

Impacto da Escolha da Métrica

- A escolha da métrica de distância tem grande impacto no cluster final produzido
  - Note que a validade do cluster final é altamente subjetiva
  - Exemplo
    - Quais os cluster significativos nestes casos?
    - Quantos clusters devem ser considerados?

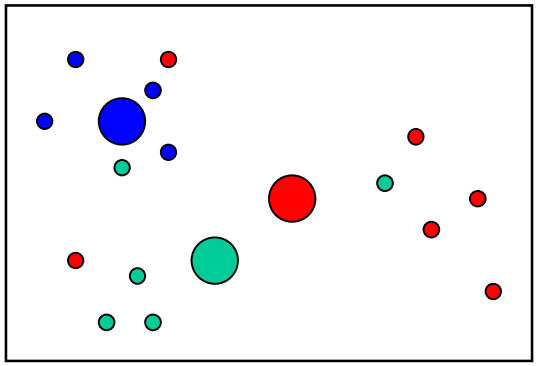


K-means: Algoritmo

- Dado um conjunto de pontos numéricos no espaço D-dimensional e um inteiro K
- O algoritmo gera K (ou menos) clusters da seguinte maneira:

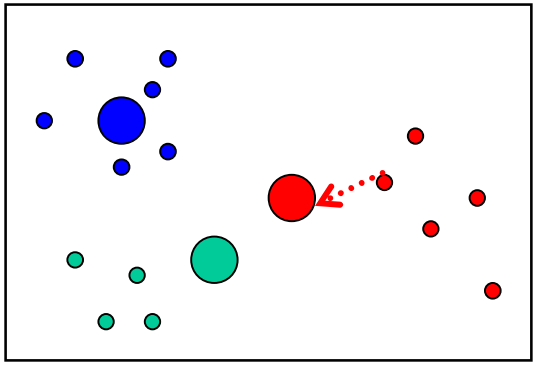
```
Escolha K clusters aleatoriamente
Calcule o centróide para cada cluster
Repita
   Atribua cada ponto ao centróide mais próximo
   Recalcule o centróide para cada cluster
Até estabilidade
```

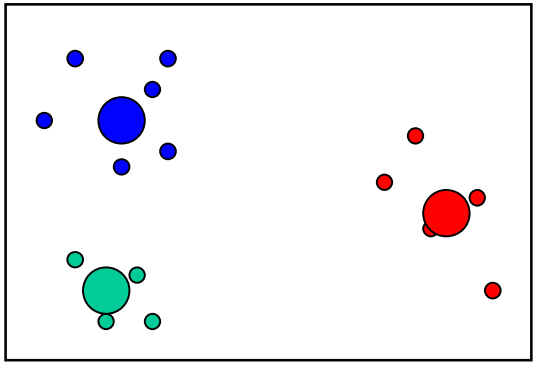
K-means: Exemplo, K = 3



Passo 1: Escolha aleatória de clusters e cálculo dos centróides (círculos maiores)

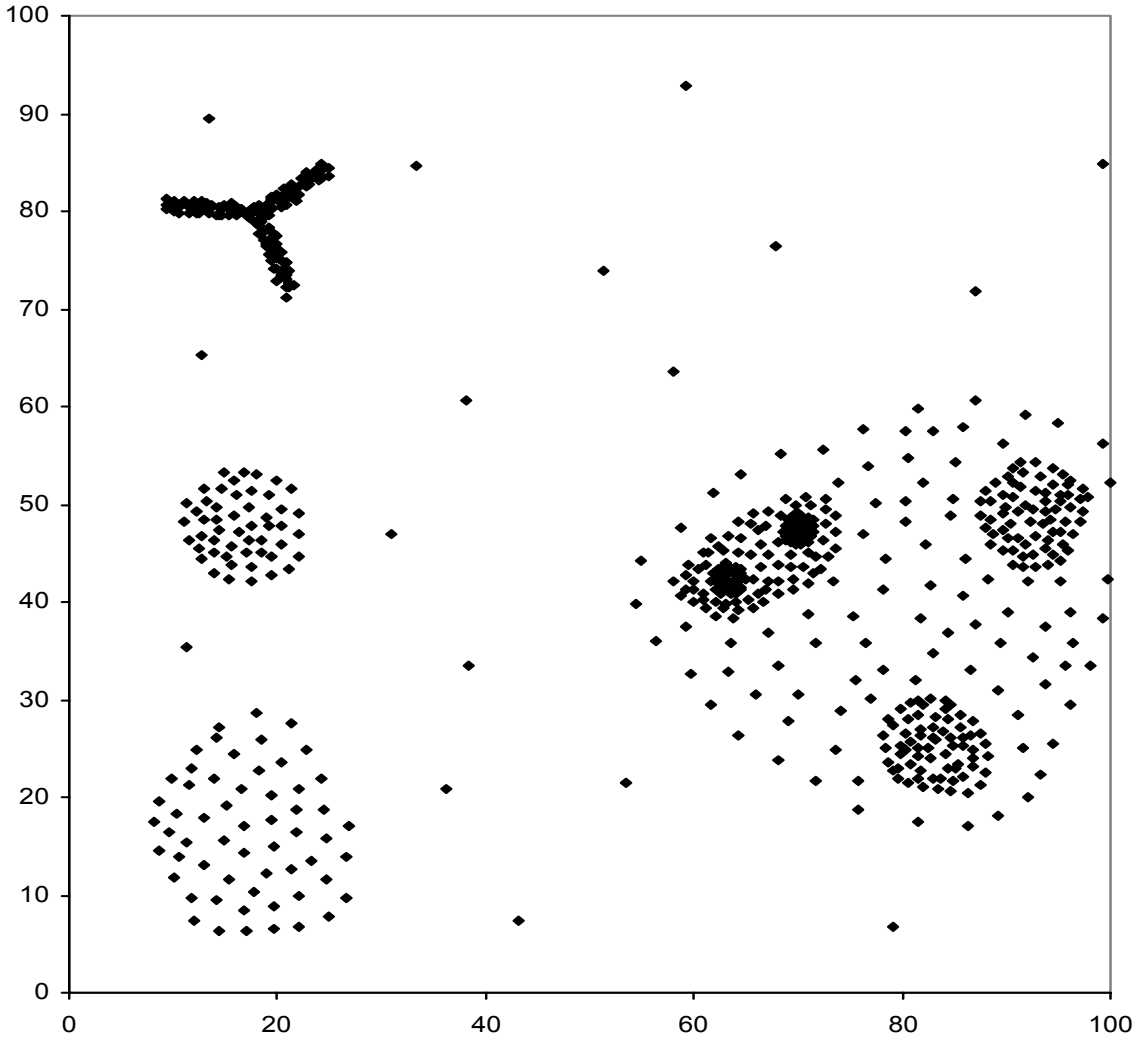
Passo 2: Atribua cada ponto ao centróide mais próximo



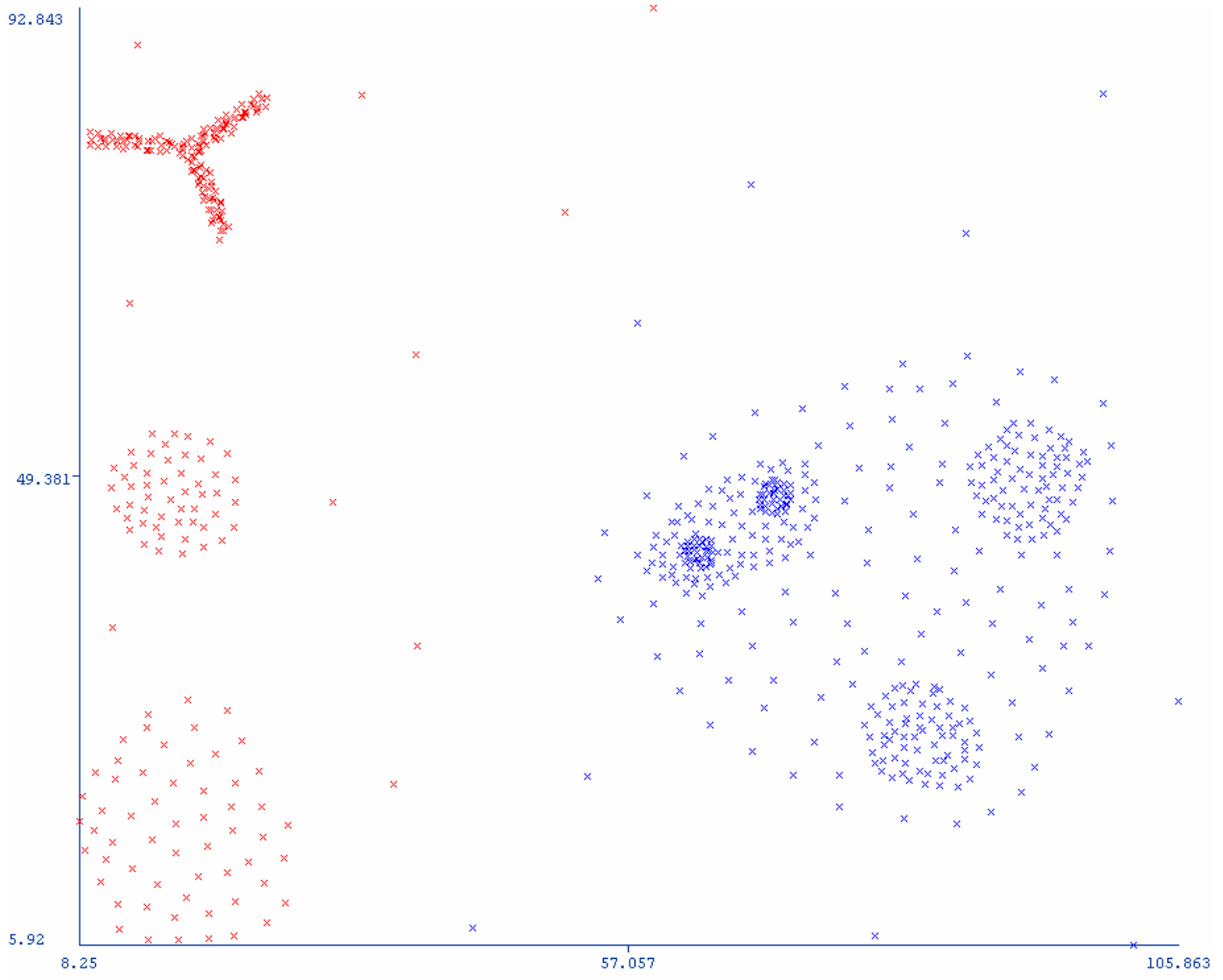


Passo 3: Recalcule centróides (neste exemplo, a solução é agora estável)

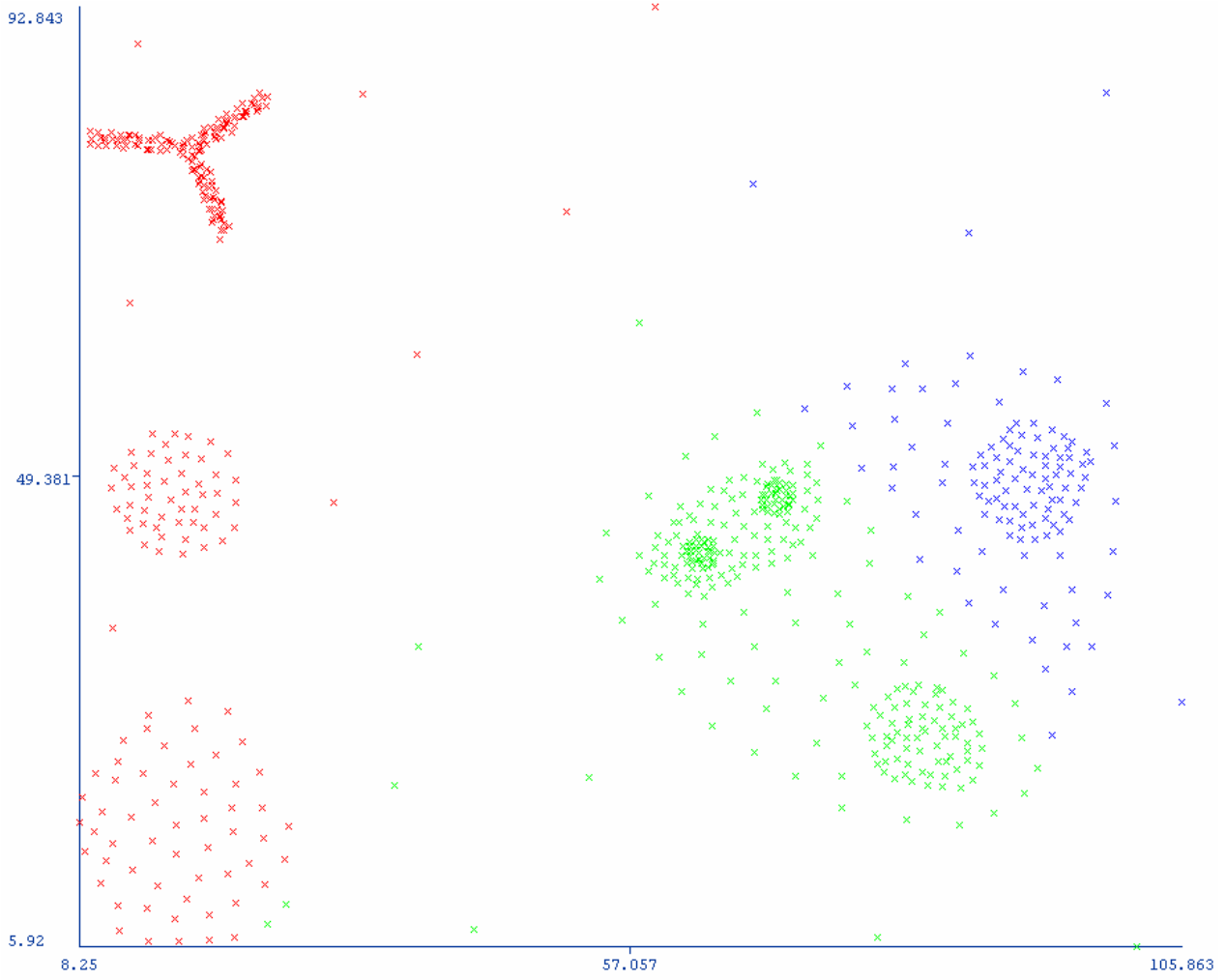
K-means: Exemplo



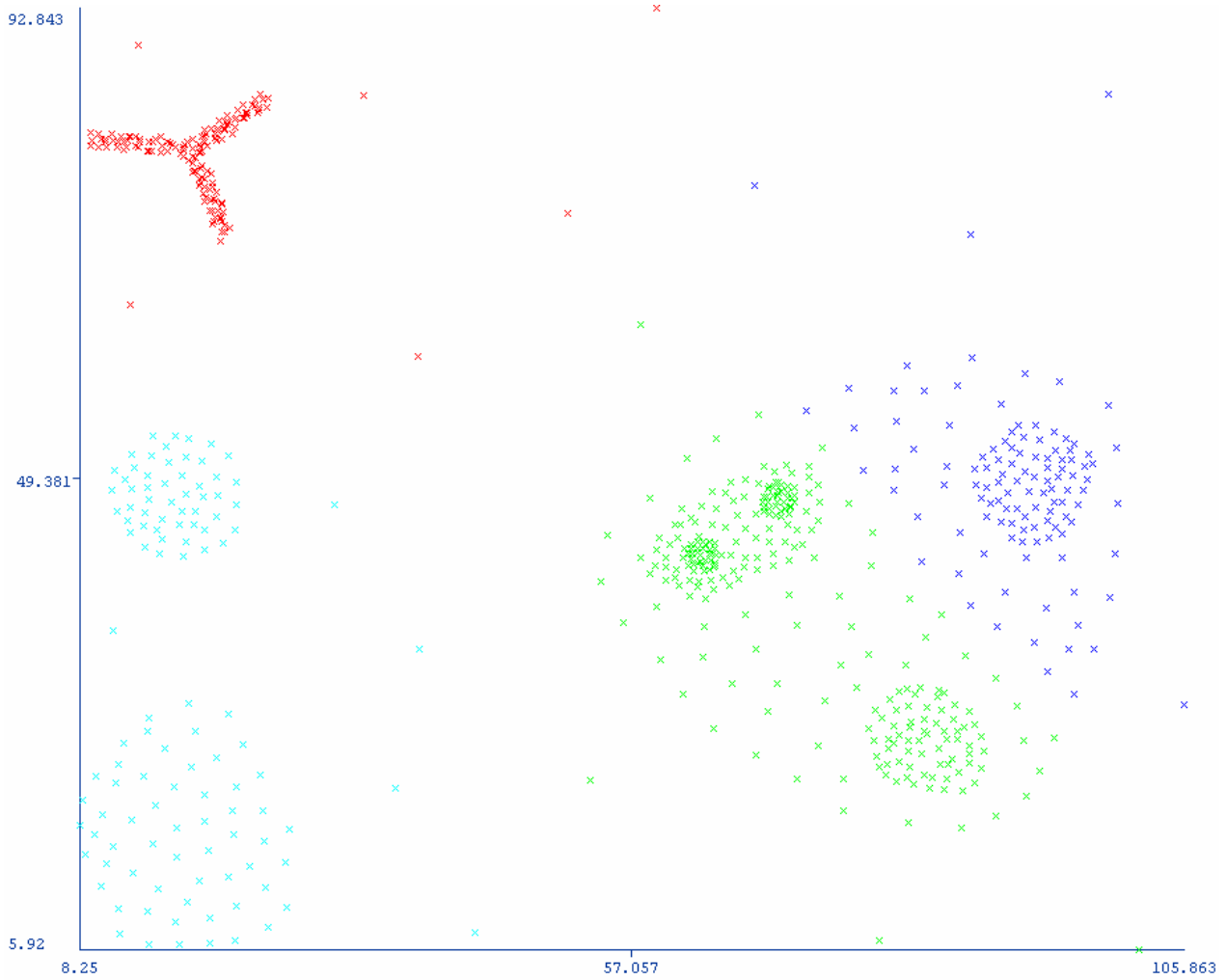
K-means: Exemplo, K=2



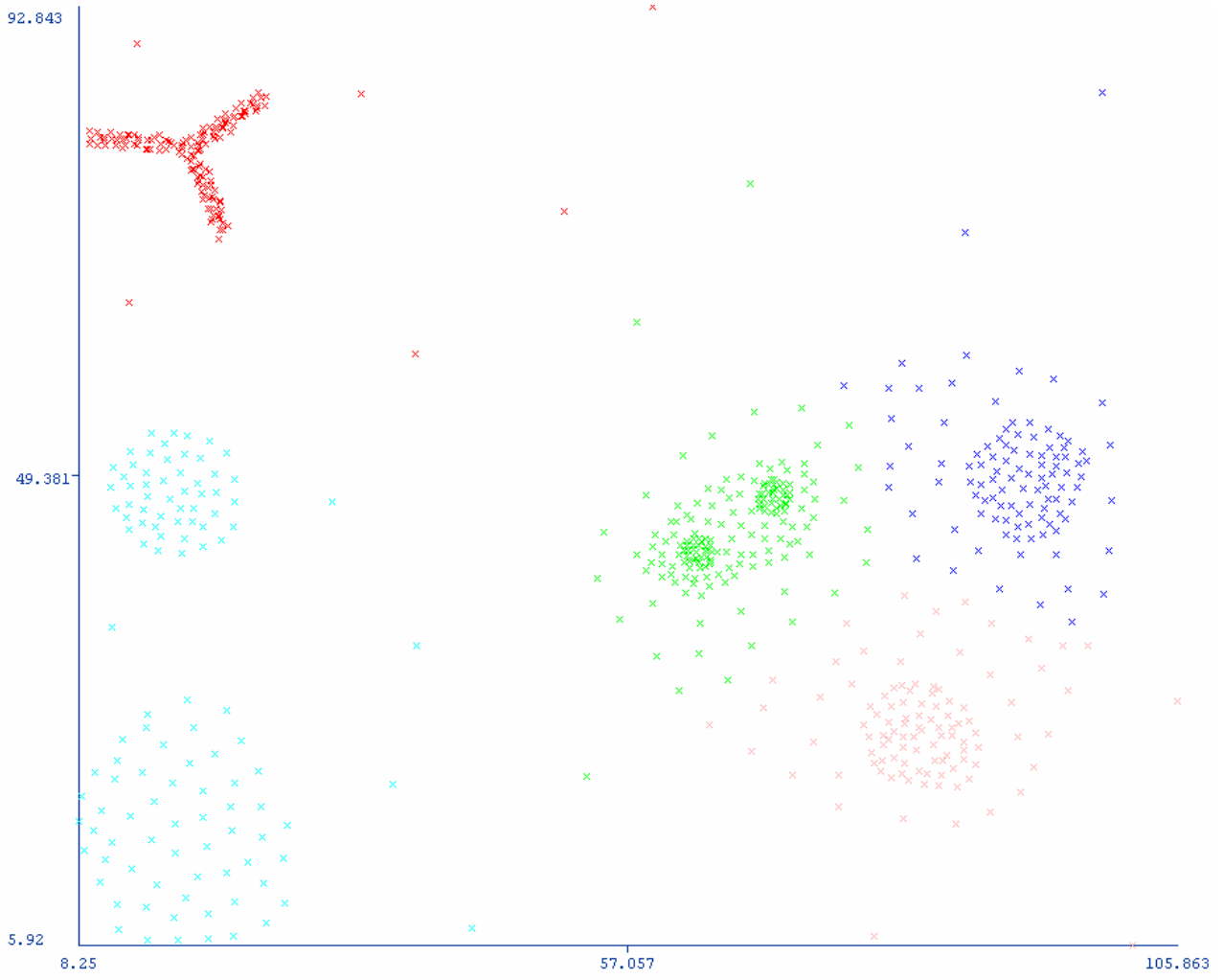
K-means: Exemplo, K=3



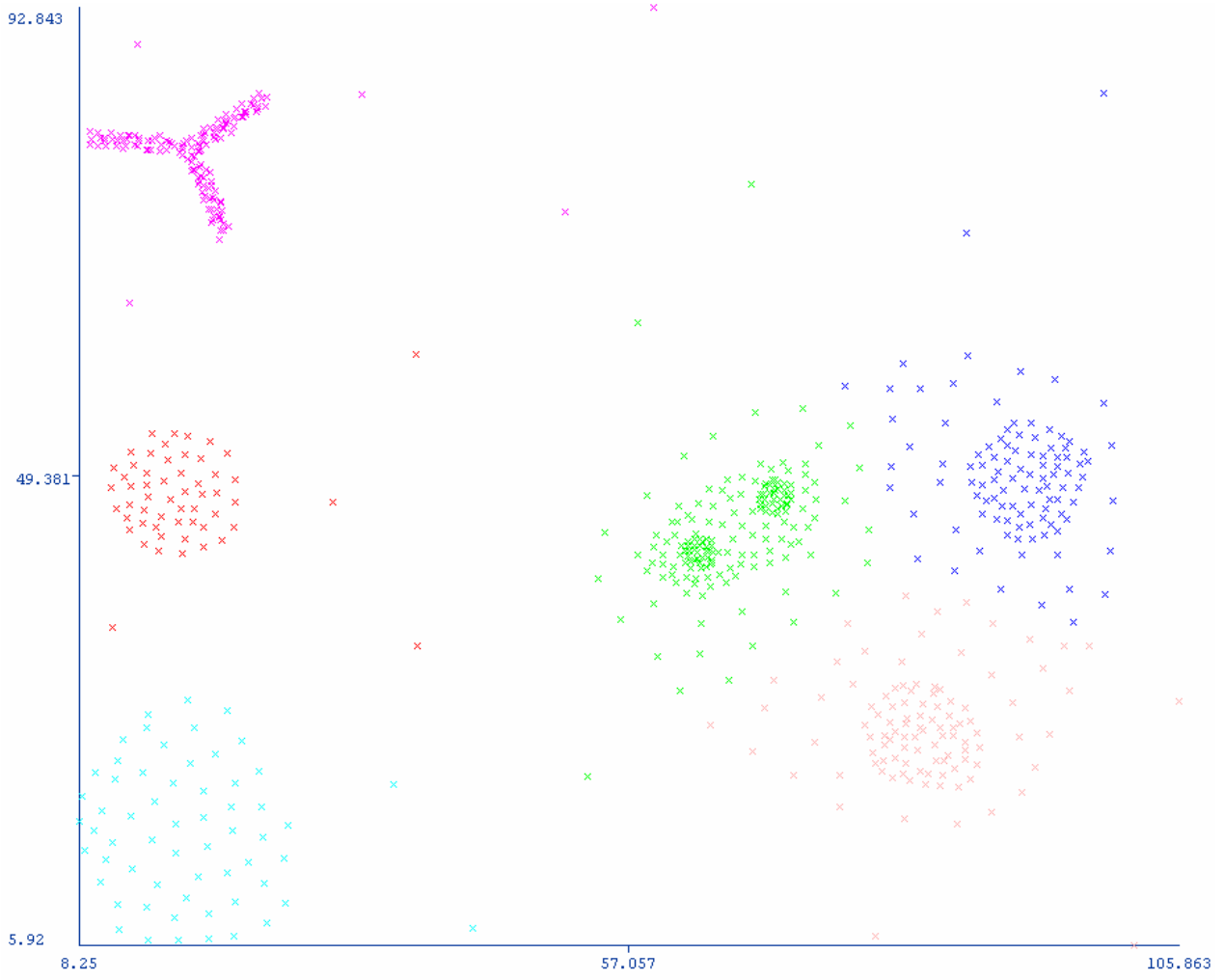
K-means: Exemplo, K=4



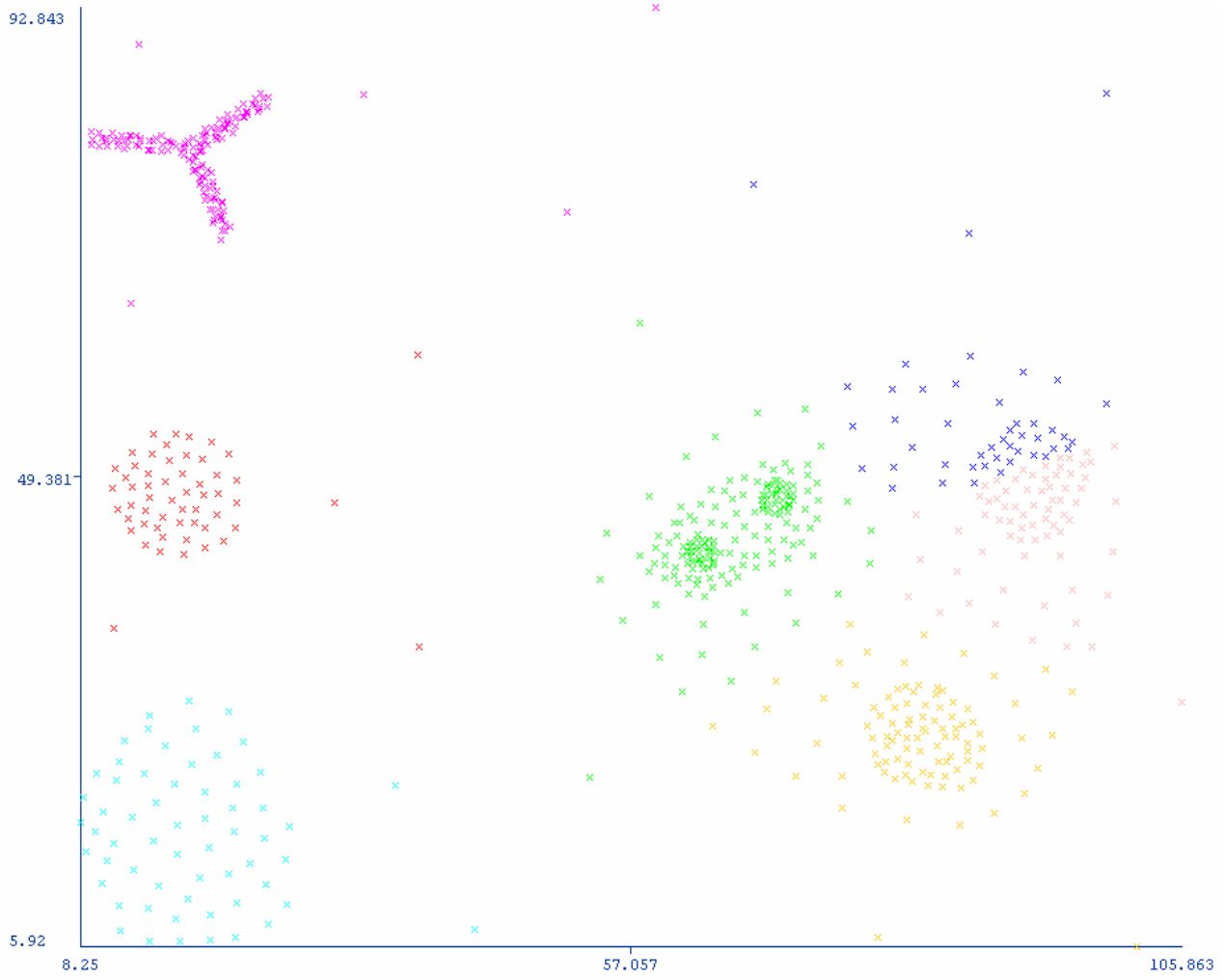
K-means: Exemplo, K=5



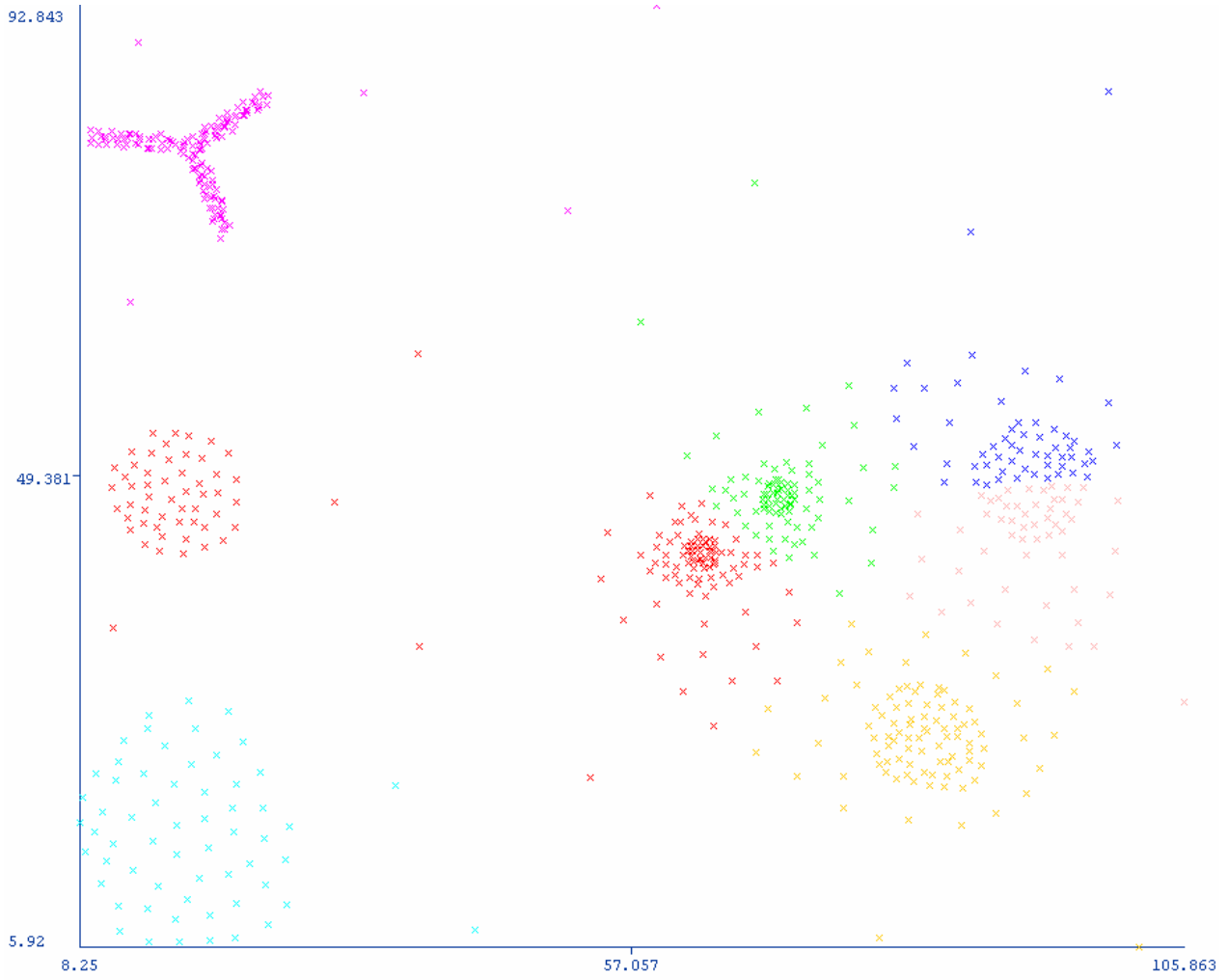
K-means: Exemplo, K=6



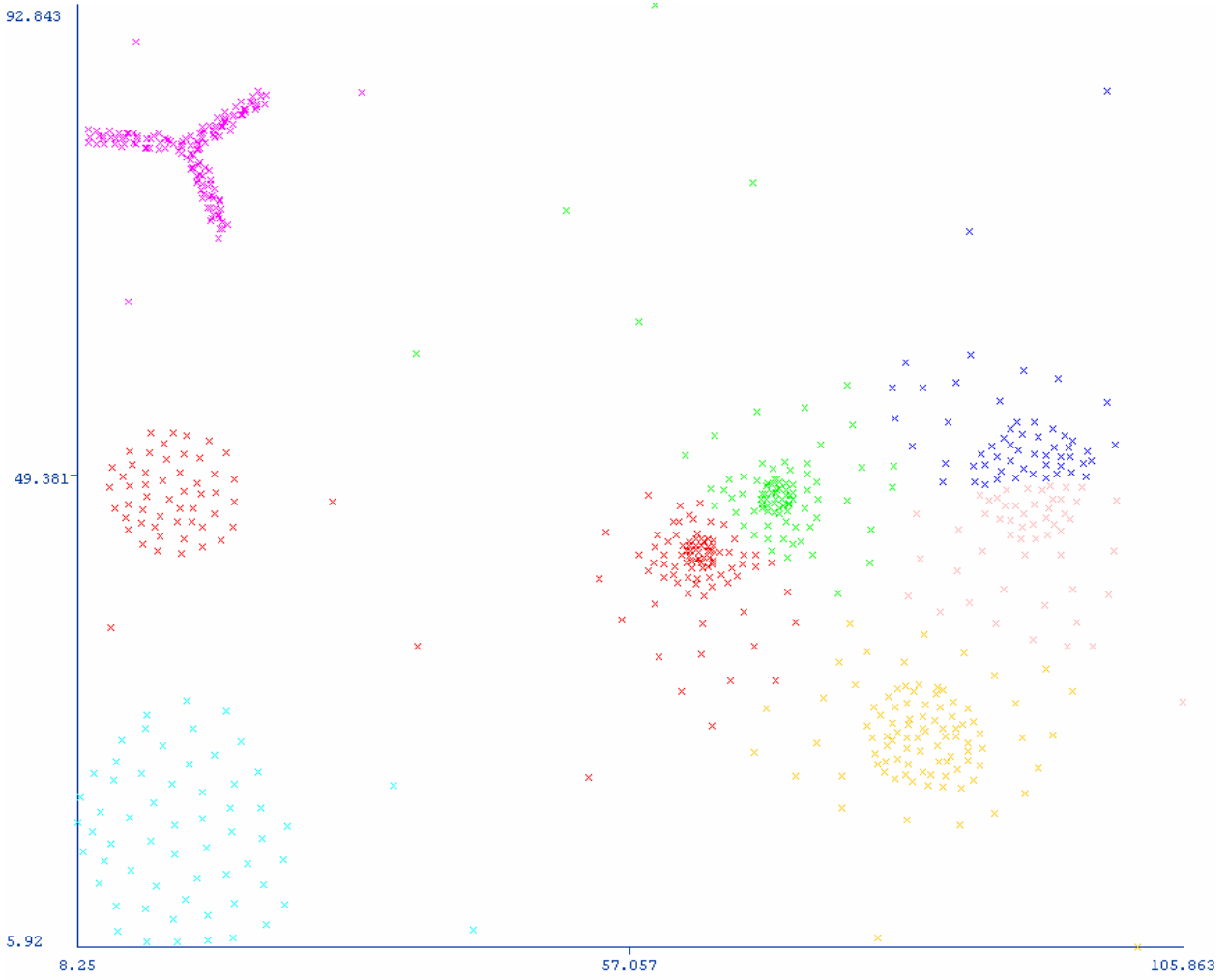
K-means: Exemplo, K=7



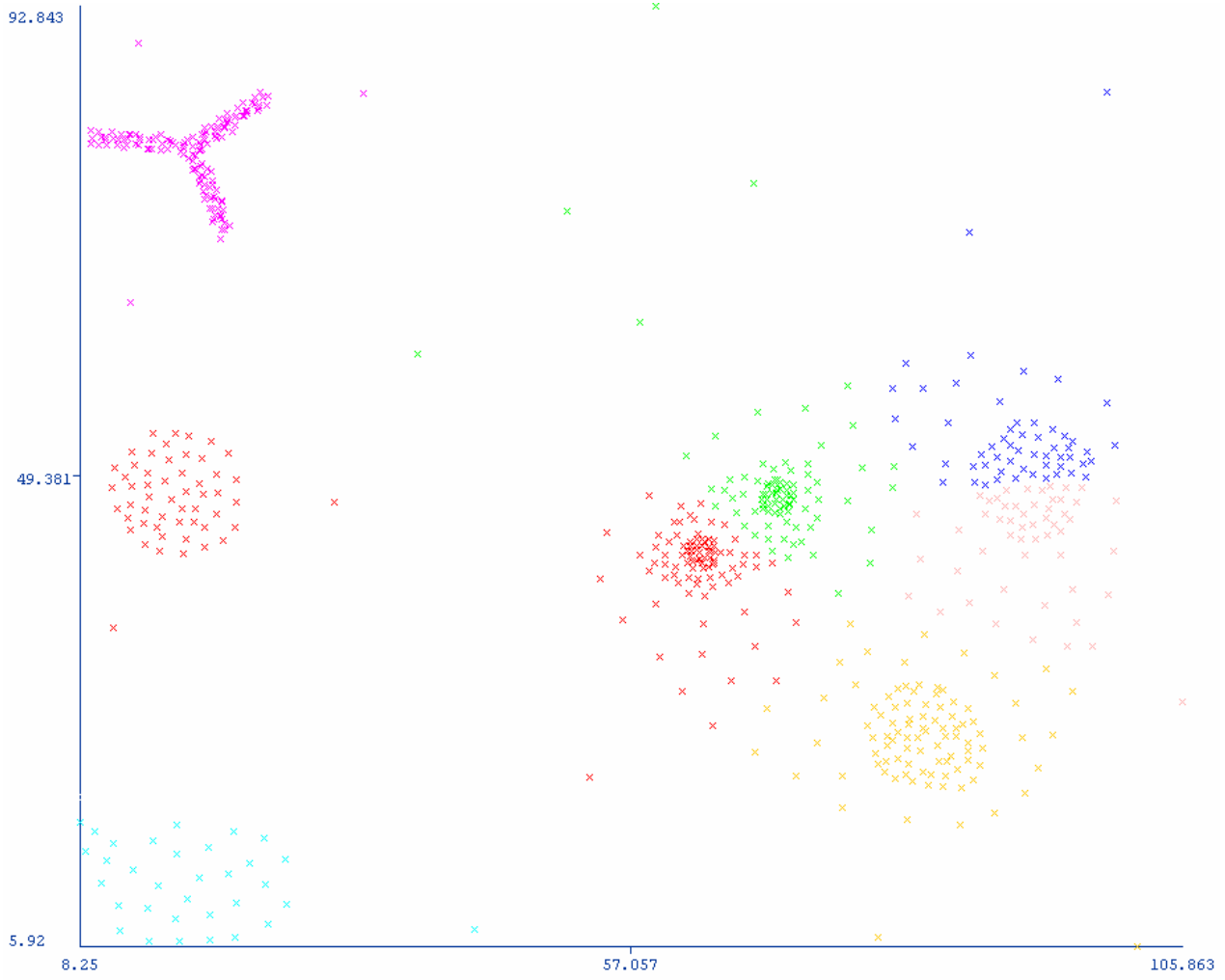
K-means: Exemplo, K=8



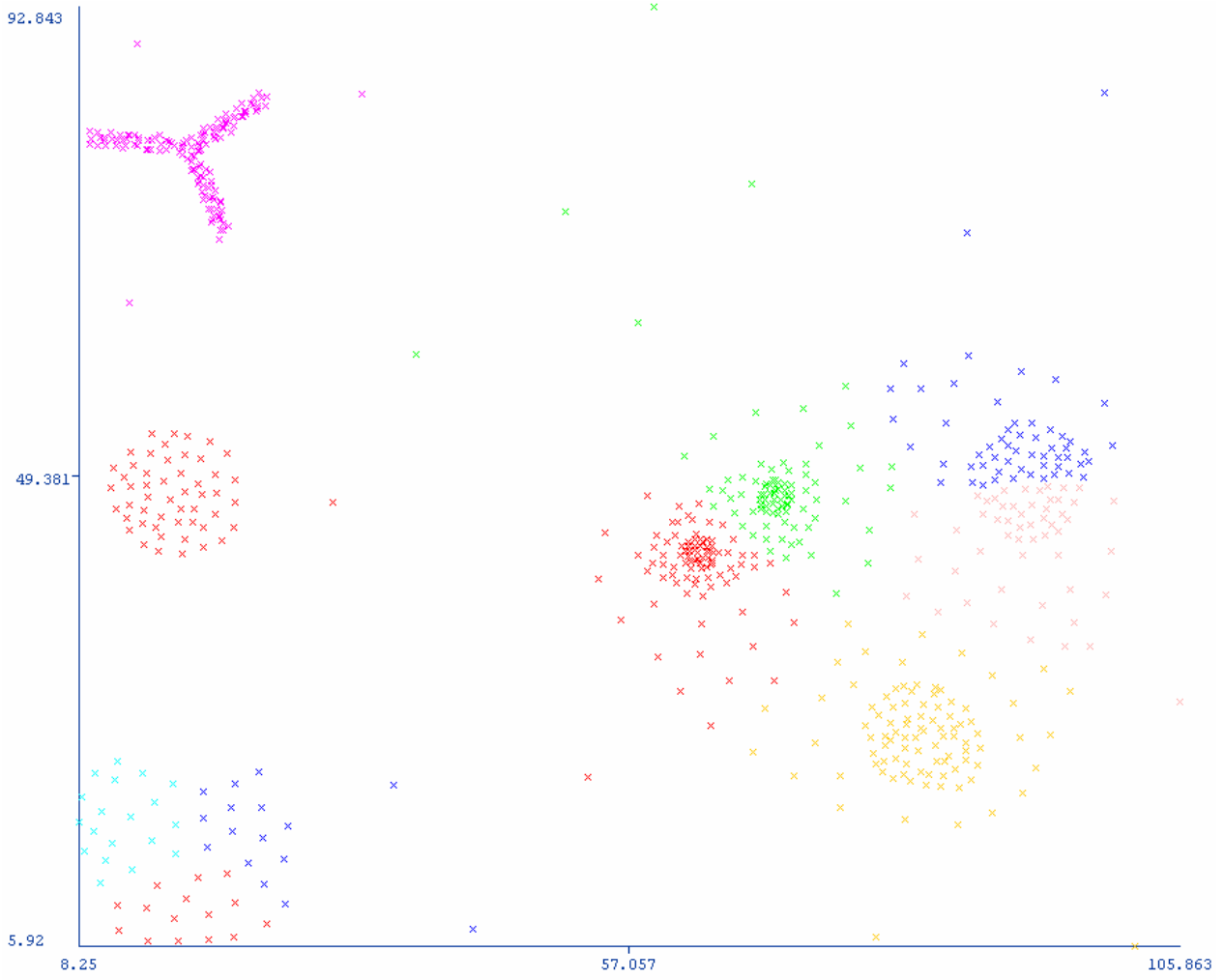
K-means: Exemplo, K=9



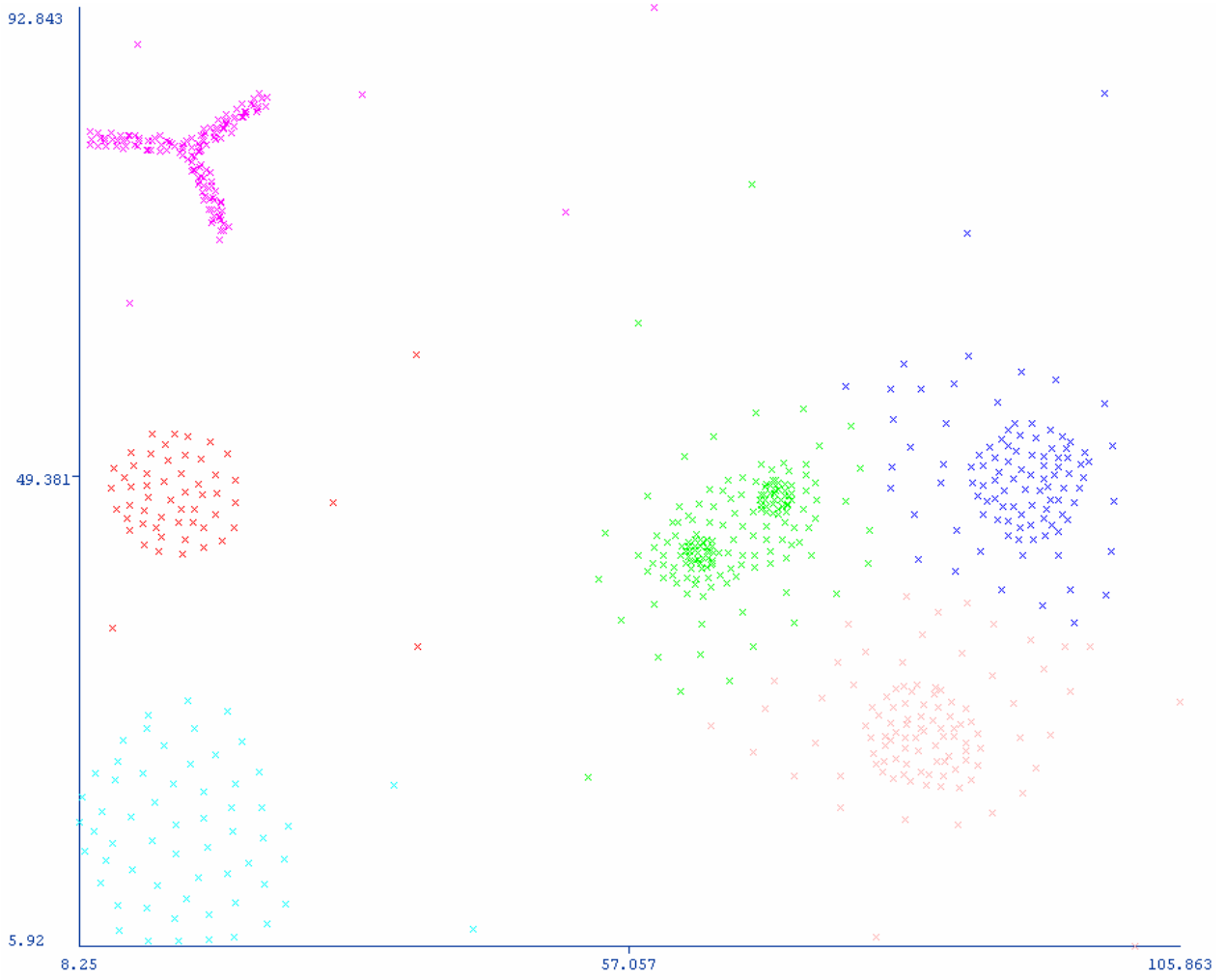
K-means: Exemplo, K=10



K-means: Exemplo, K=12

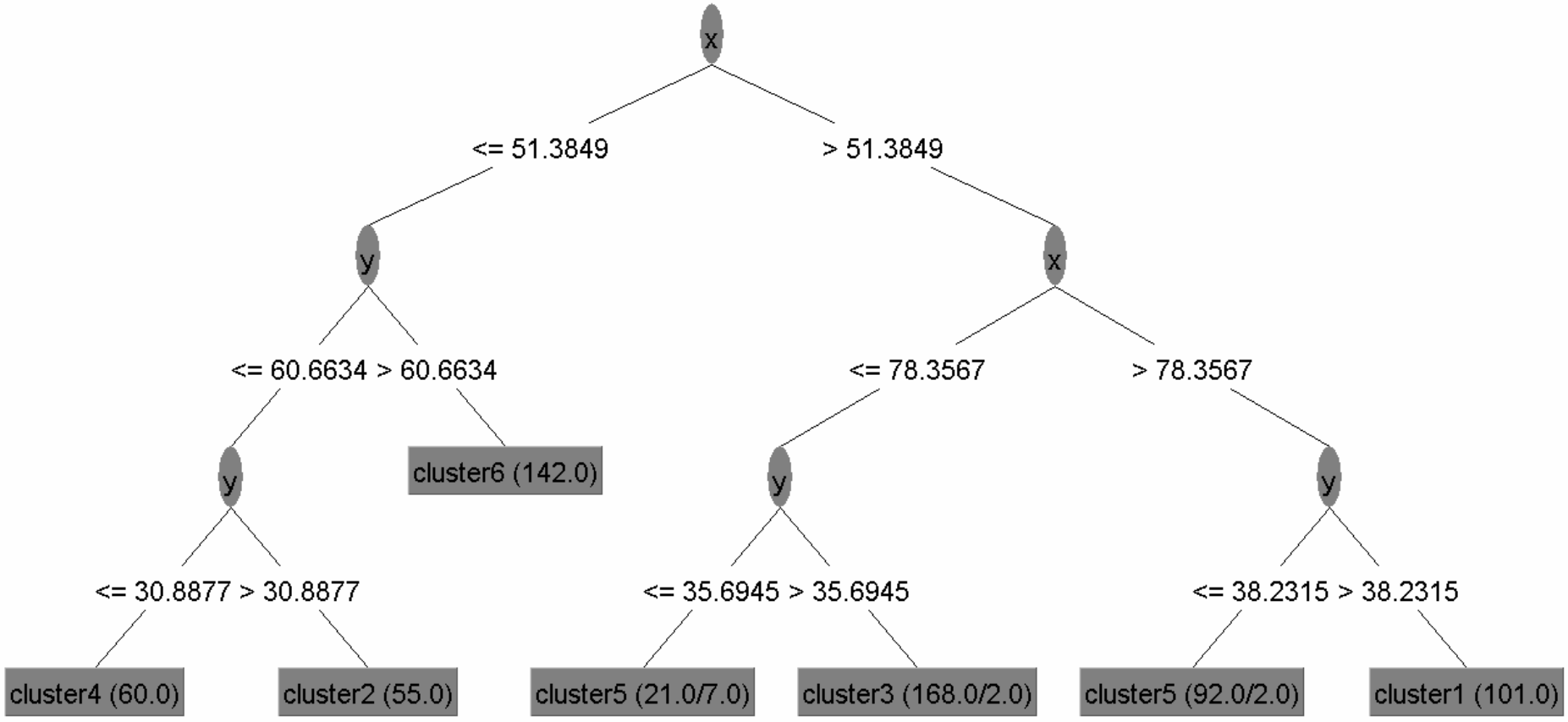


K-means: Exemplo, K=6



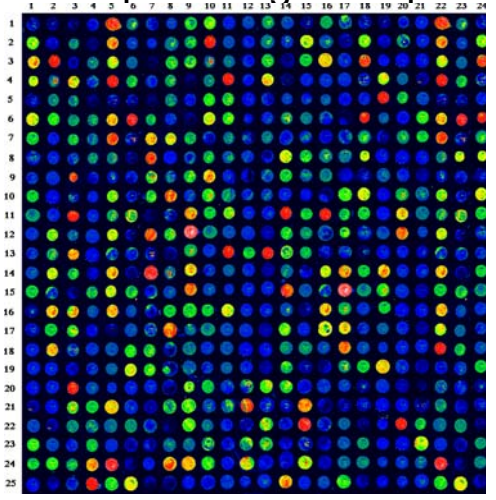
Descrição do Cluster: Exemplo, K=6

(J48 –C 0.25 –M 15)

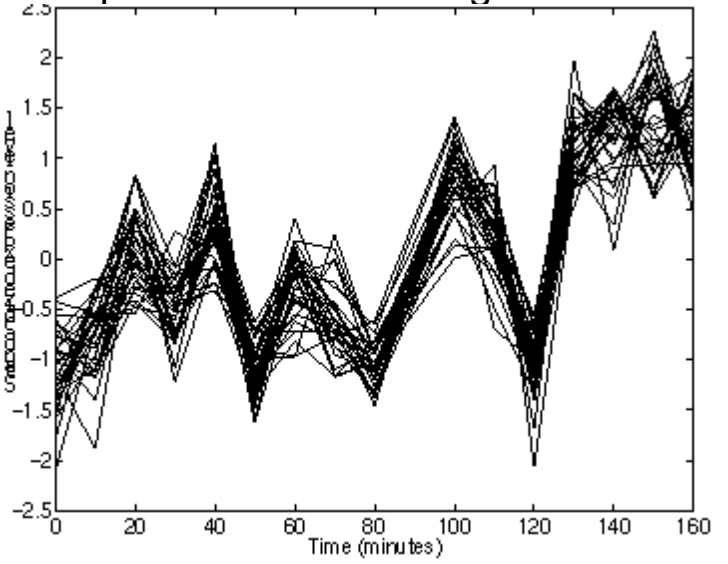


K-means: Exemplo de Aplicação

- Clustering de Genes
  - Dada uma série de experimentos de microarray medindo a expressão de um conjunto de genes a intervalos regulares de tempo numa célula
  - Normalização permite comparação entre microarrays
  - Produz clusters de genes que variam de forma similar ao longo do tempo
  - Hipótese: genes que variam da mesma forma pode ser/estar co-regulados



Amostra de um Array. Linhas são genes e colunas são pontos no tempo



Um cluster de genes co-regulados

K-means: Problemas

- Os clusters finais não representam uma otimização global mas apenas local e clusters diferentes podem surgir a partir da diferença na escolha inicial aleatória dos centróides (fig.1)
- O parâmetro K deve ser escolhido antecipadamente, ou vários valores devem ser tentados até encontrar o "melhor"
- Os dados devem ser numéricos e devem ser comparados através da distância Euclideana (há uma variante chamado algoritmo K-medians que aborda esse problema)
- O algoritmo trabalha melhor com dados que contêm clusters esféricos; clusters com outra geometria podem não ser encontrados
- O algoritmo é sensível a *outliers* (pontos que não pertencem a nenhum cluster). Esses pontos podem distorcer a posição do centróide e deteriorar o cluster

K-means: Problemas (cont.)

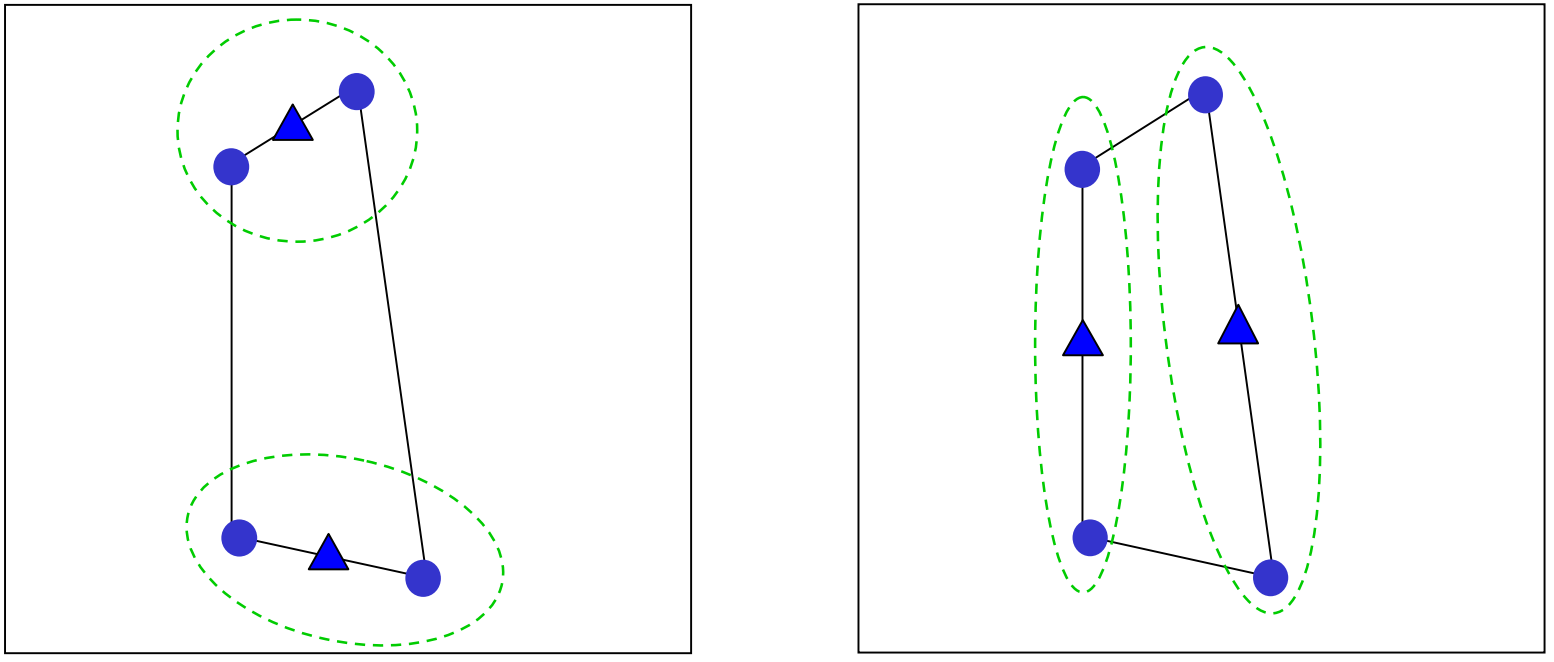


Figura 1

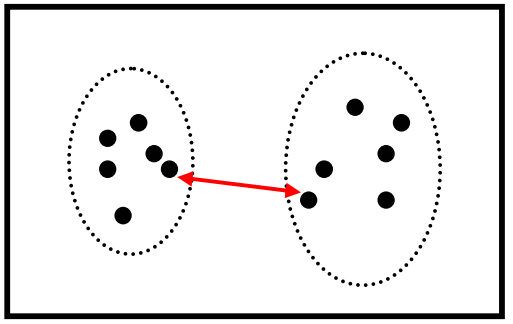
Clustering Hierárquico: Algoritmo

- Cria uma árvore na qual os objetos são as folhas e os nós internos revelam a estrutura de similaridade dos pontos
  - A árvore é frequentemente chamada “dendograma”
- O algoritmo pode ser resumido da seguinte maneira:

```
Coloque todos os pontos em seus próprios clusters
Enquanto há mais de um cluster Faça
   Agrupe o par de clusters mais próximos
Fim Enquanto
```

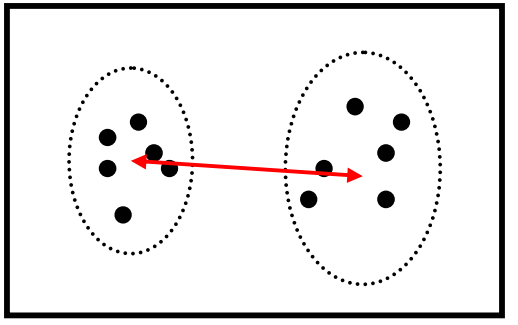
- O comportamento do algoritmo depende em como “par de clusters mais próximo” é definido

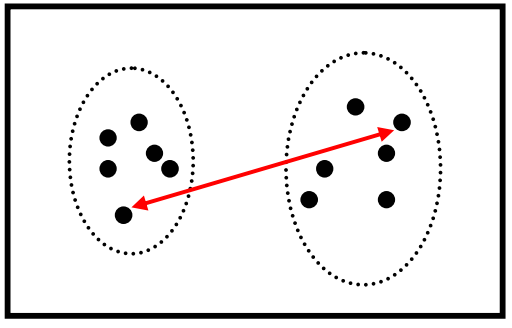
Clustering Hierárquico: Agrupando Clusters



Single Link: Distância entre dois clusters é a distância entre os pontos mais próximos. Também chamado “agrupamento de vizinhos”

Average Link: Distância entre clusters é a distância entre os centróides

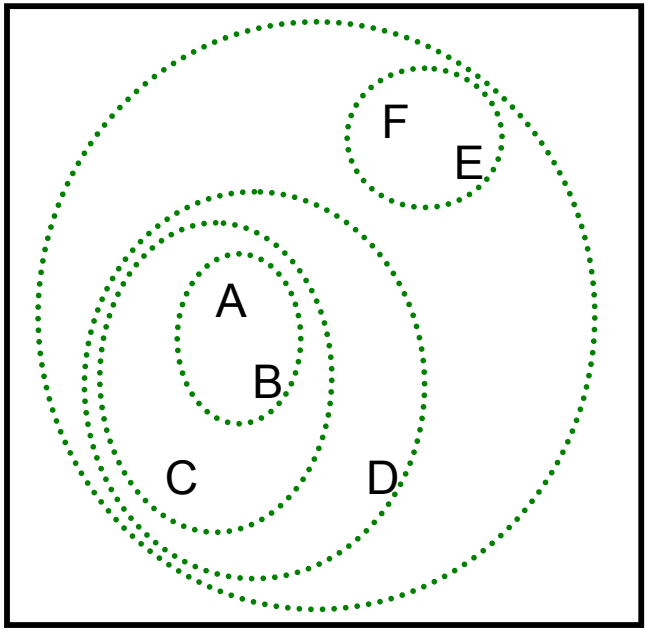


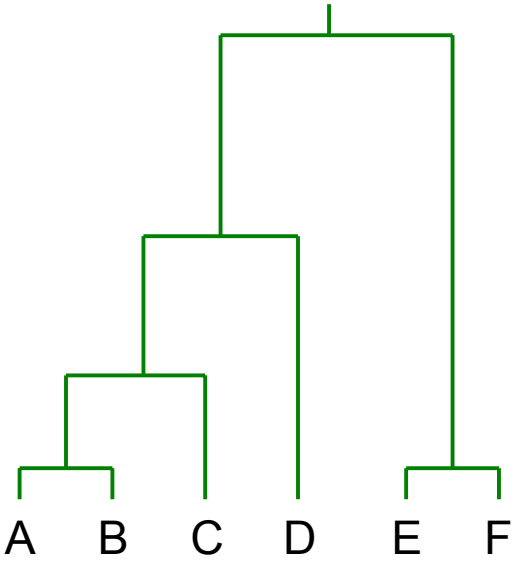


Complete Link: Distância entre clusters é a distância entre os pontos mais distantes

Clustering Hierárquico: Exemplo 1

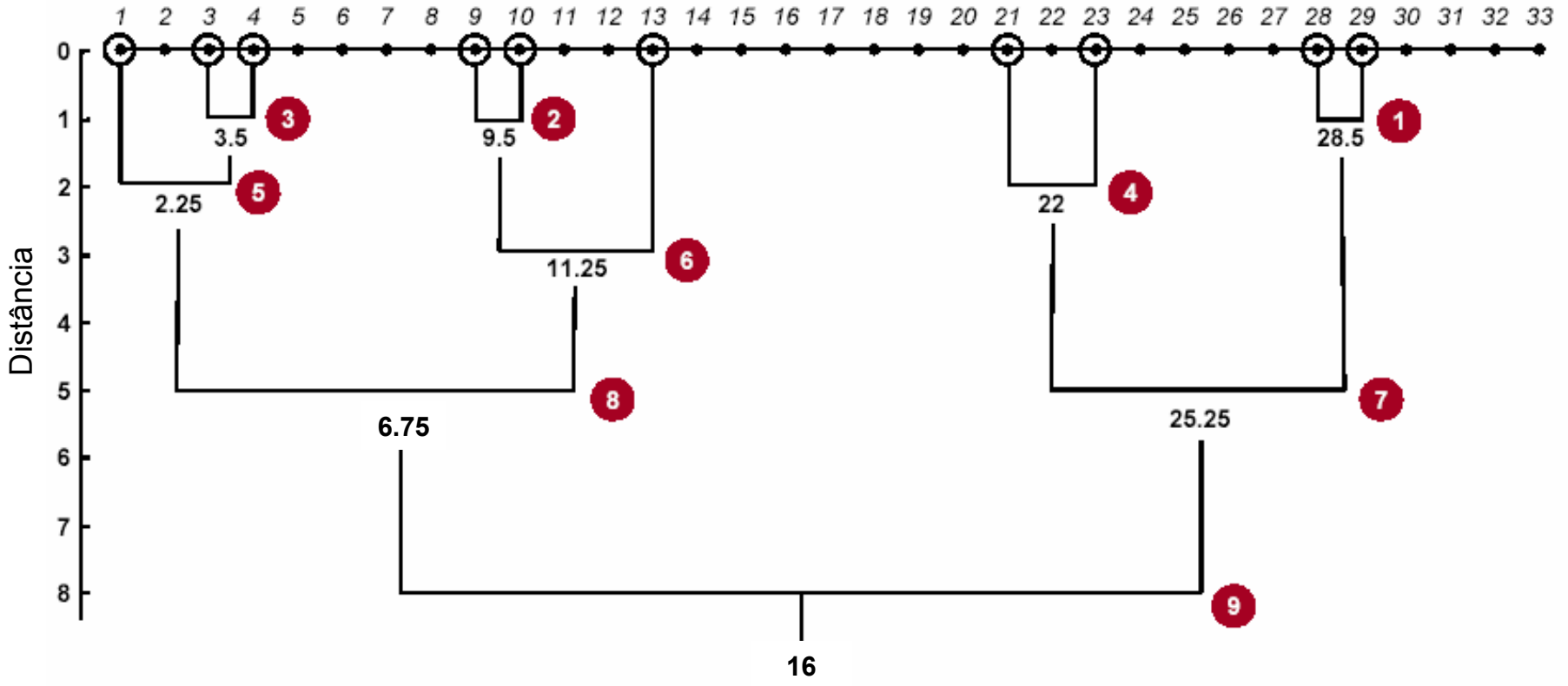
Este exemplo ilustra single-link clustering no espaço Euclideano com 6 pontos



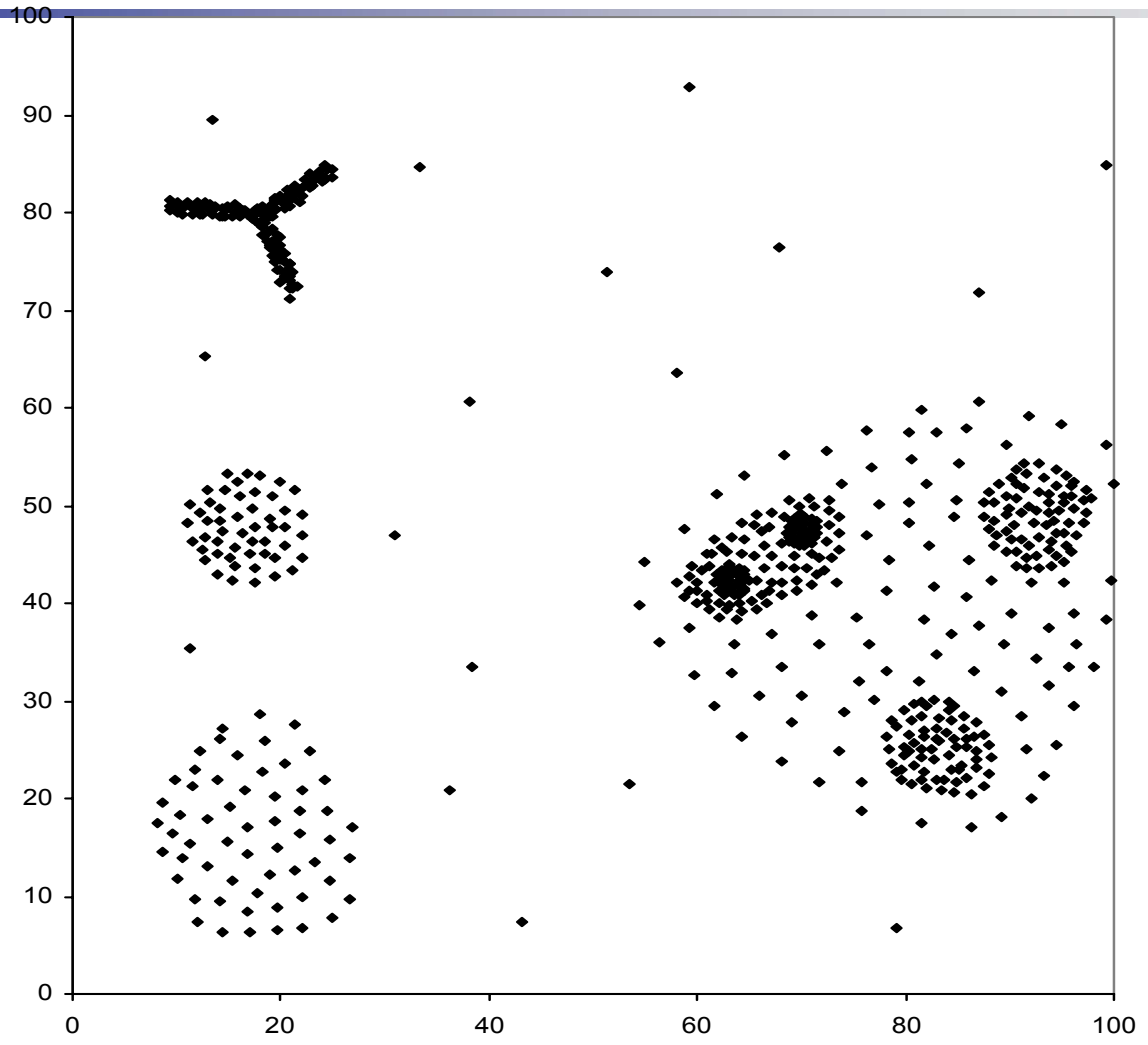


Clustering Hierárquico: Exemplo 2

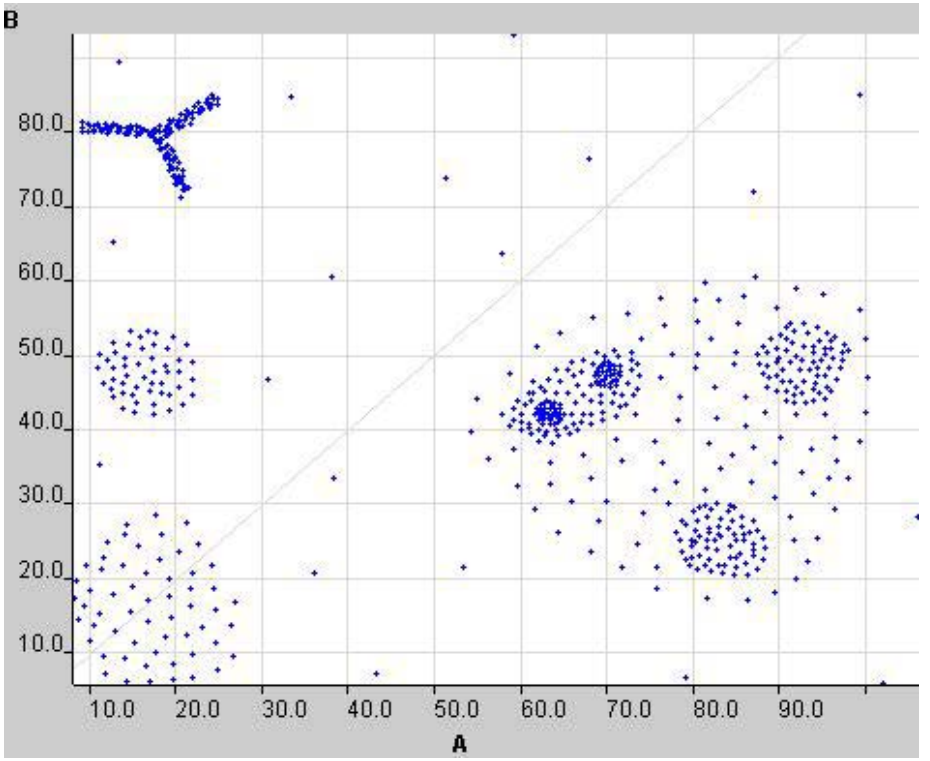
- Realizar cluster hierárquico utilizando single-link no seguinte conjunto de objetos:
  - X = {1, 3, 4, 9, 10, 13, 21, 23, 28, 29}
  - No caso de empate, sempre agrupe pares de clusters com maior média
  - Indique a ordem na qual as operações de agrupamento ocorrem

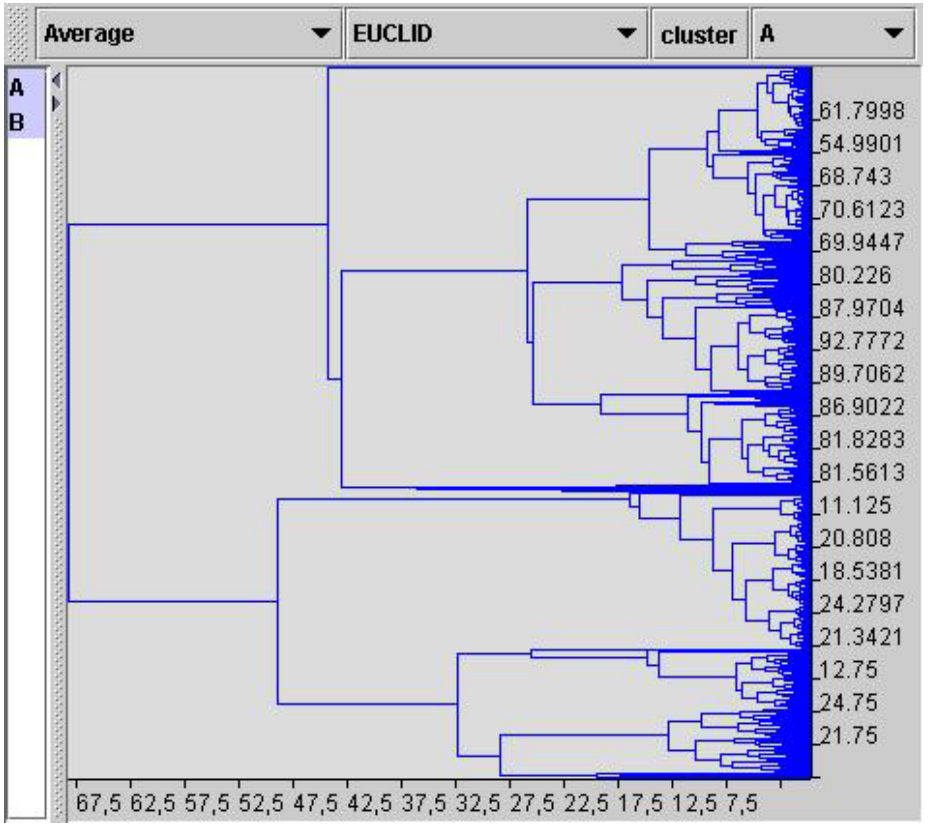


Clustering Hierárquico: Exemplo 3

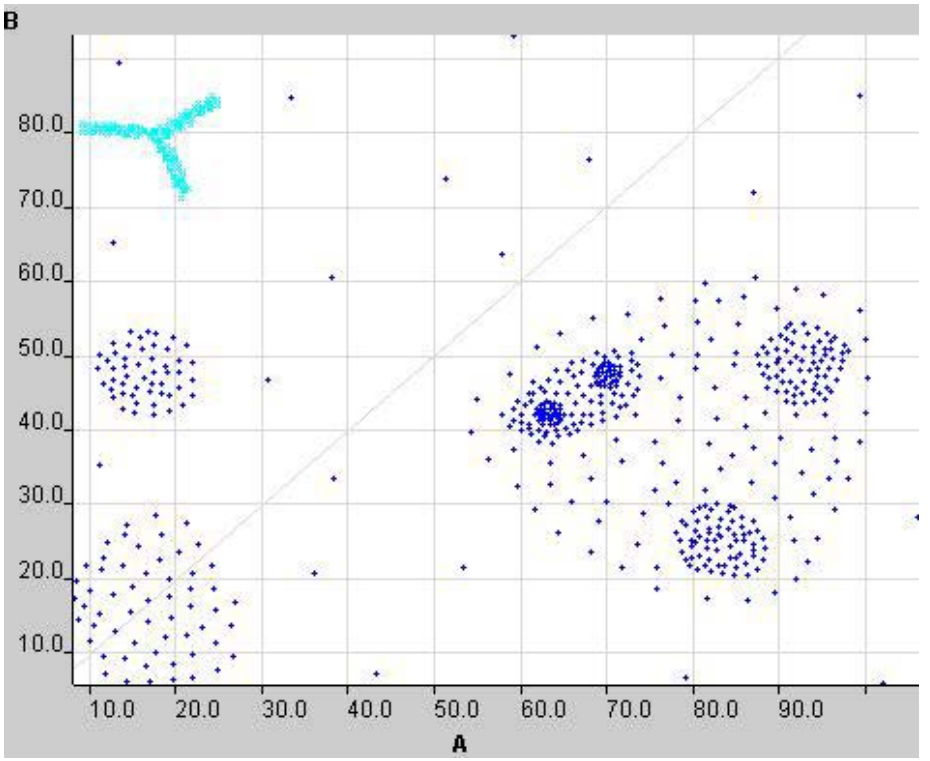


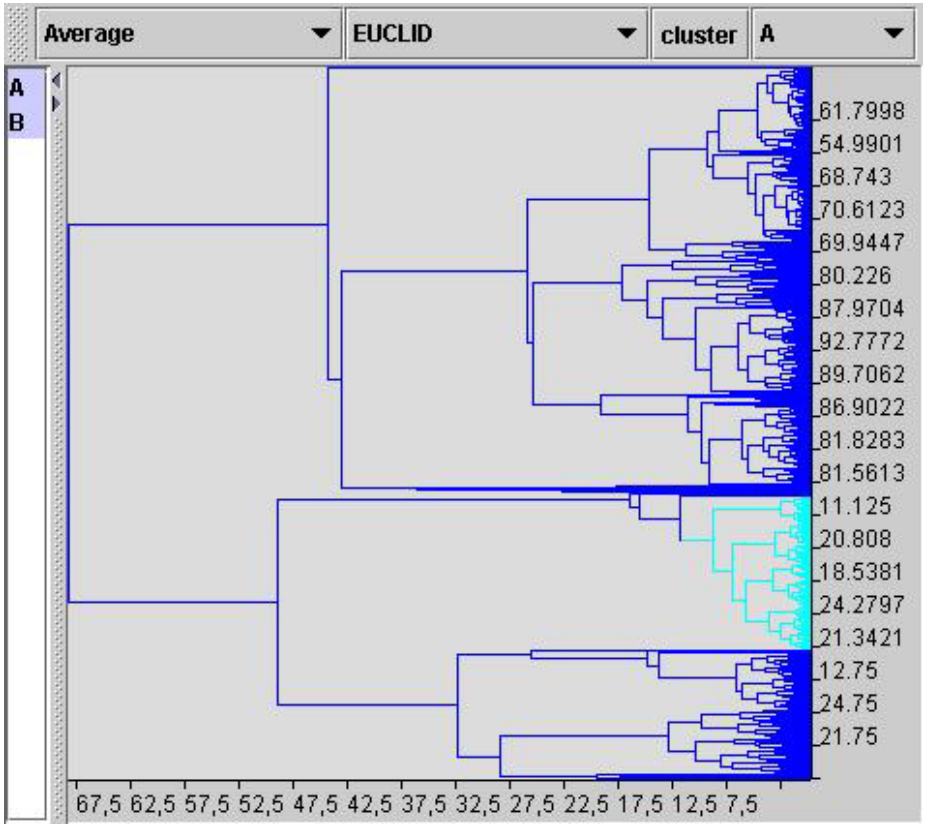
Clustering Hierárquico: Exemplo 3



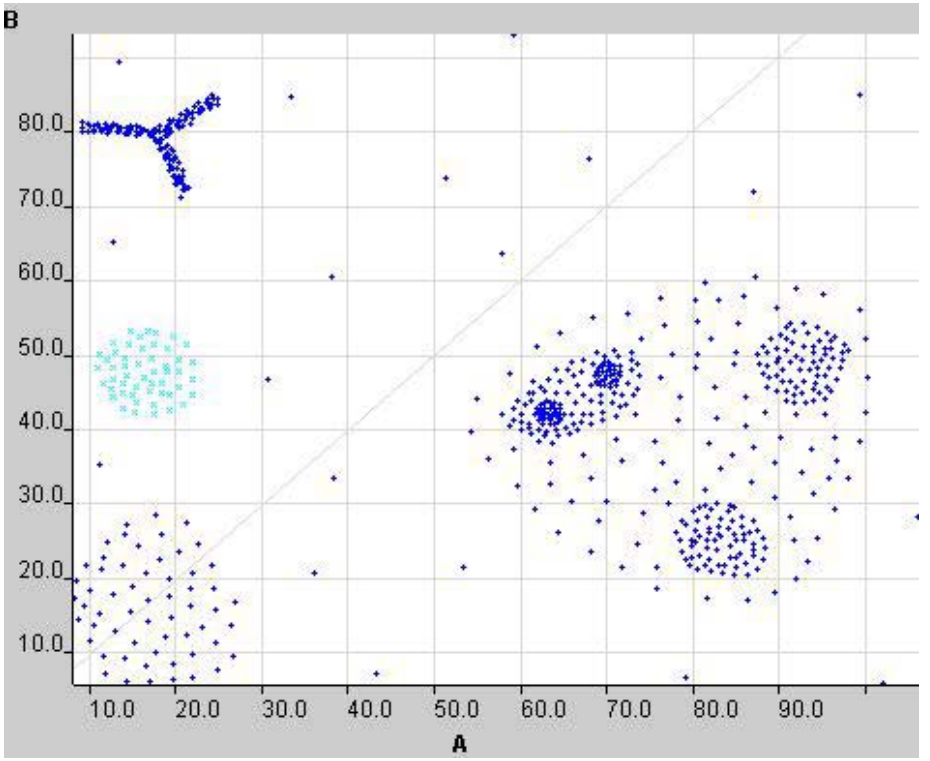


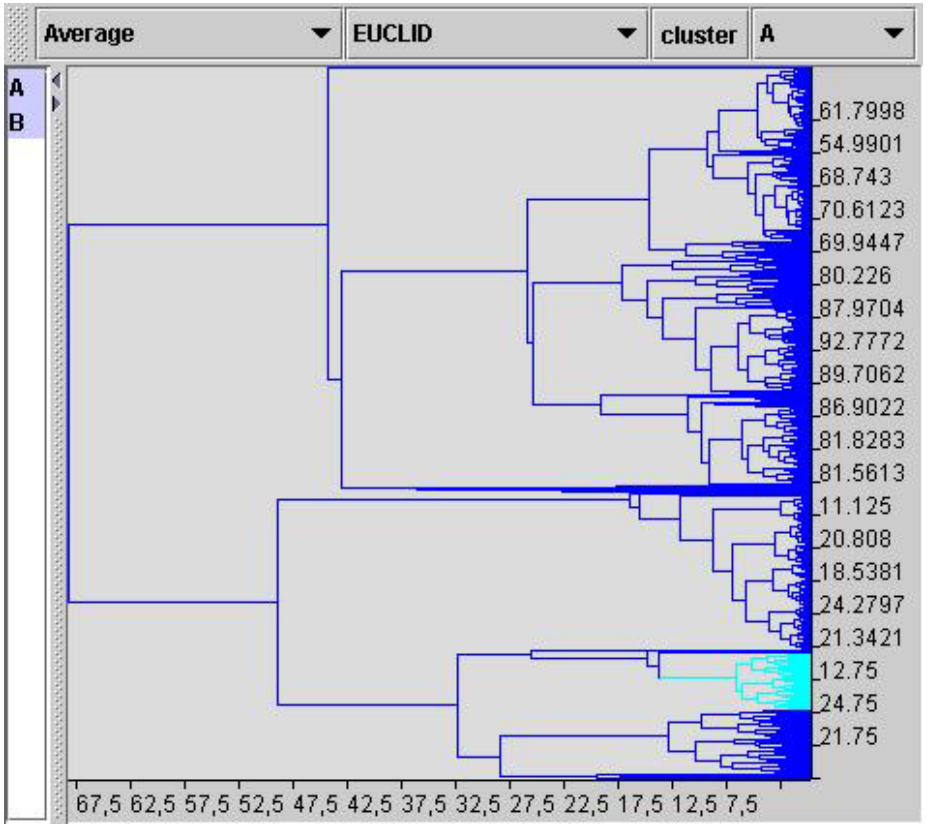
Clustering Hierárquico: Exemplo 3



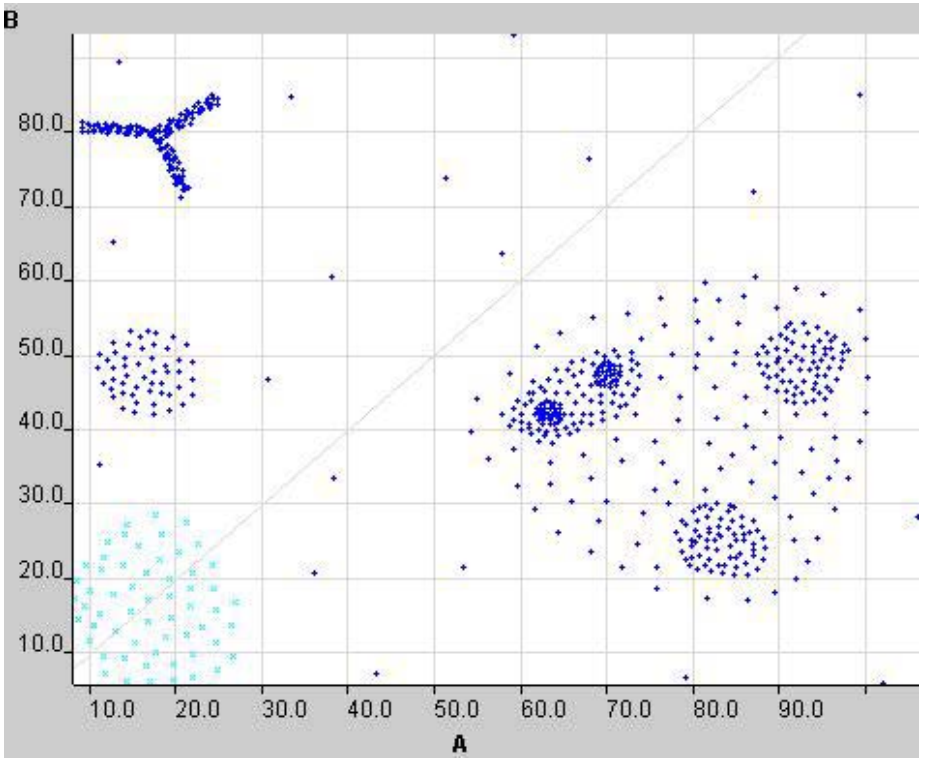


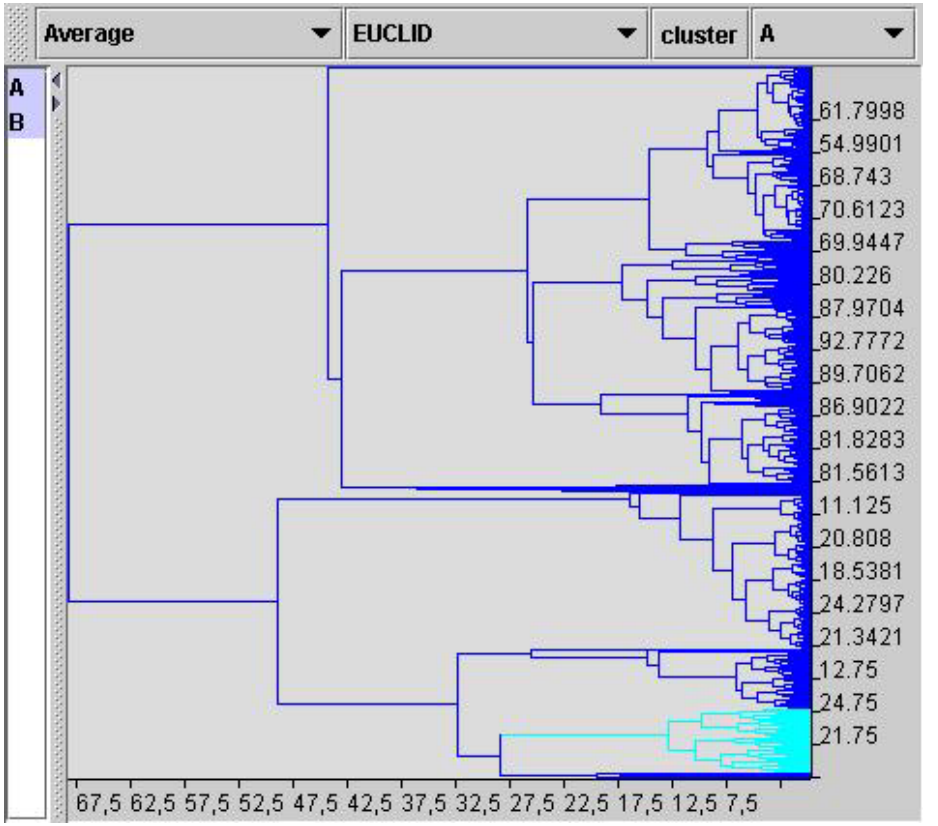
Clustering Hierárquico: Exemplo 3



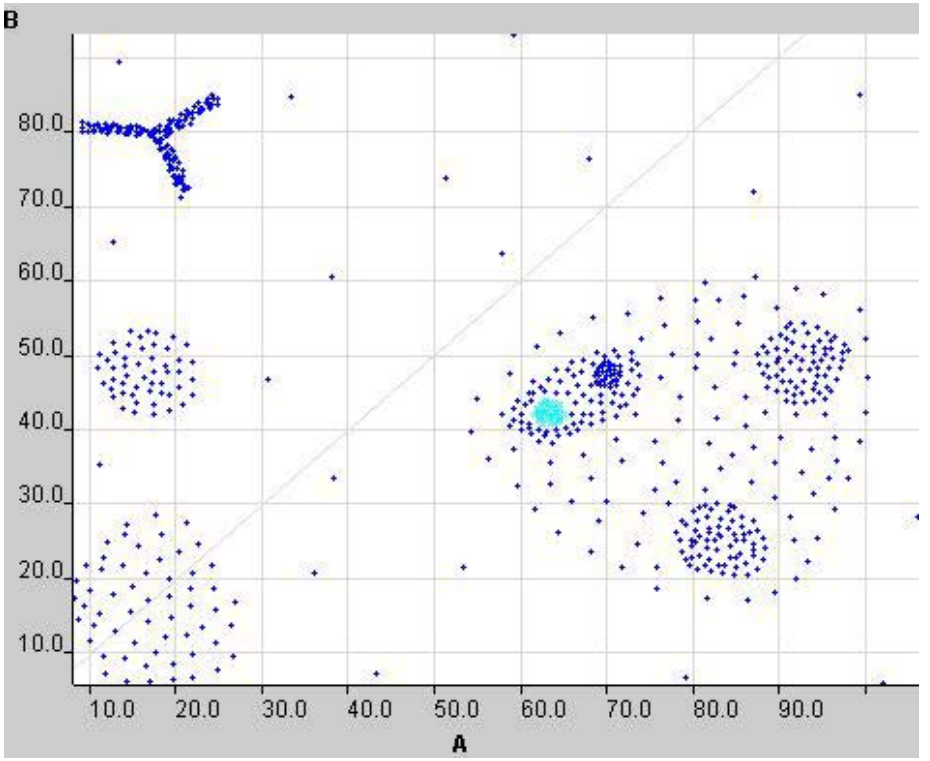


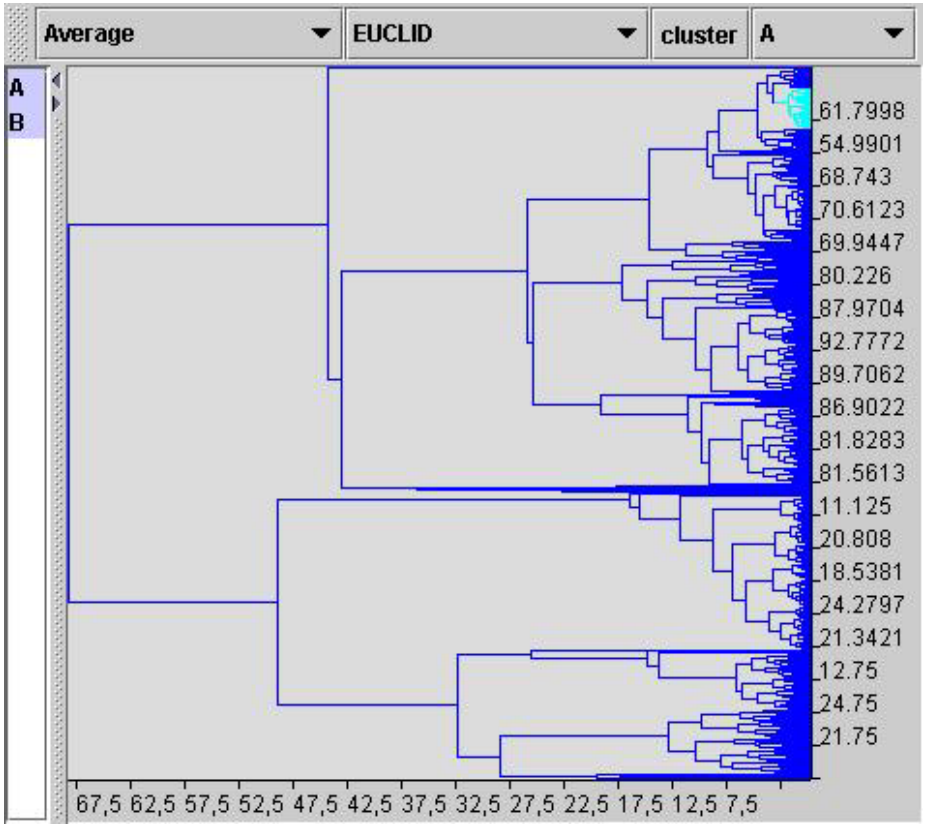
Clustering Hierárquico: Exemplo 3



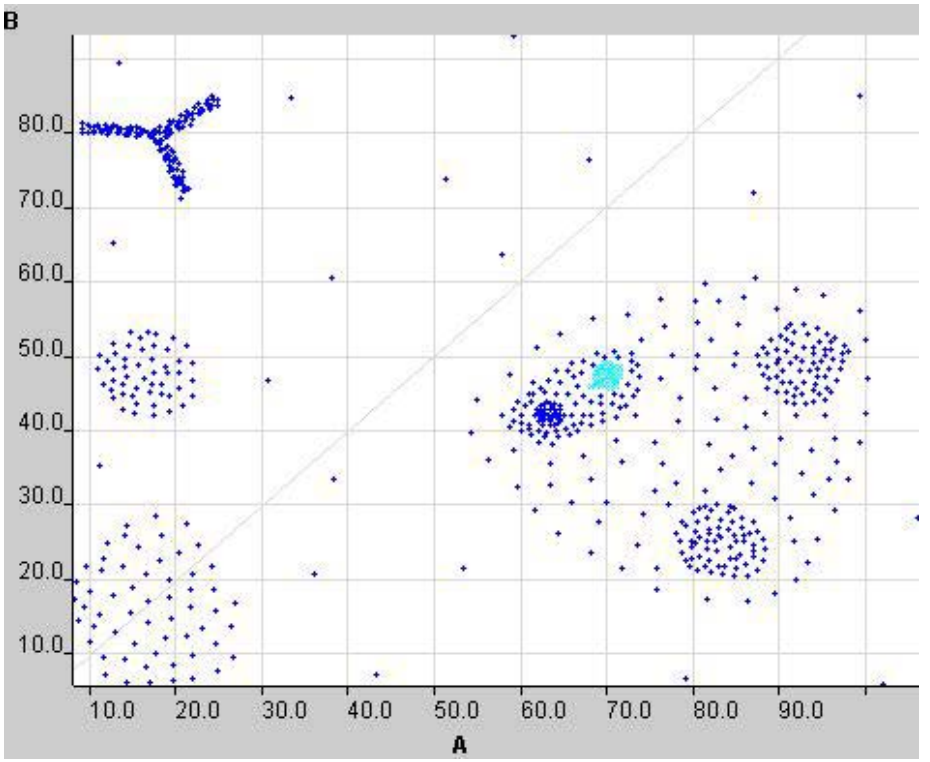


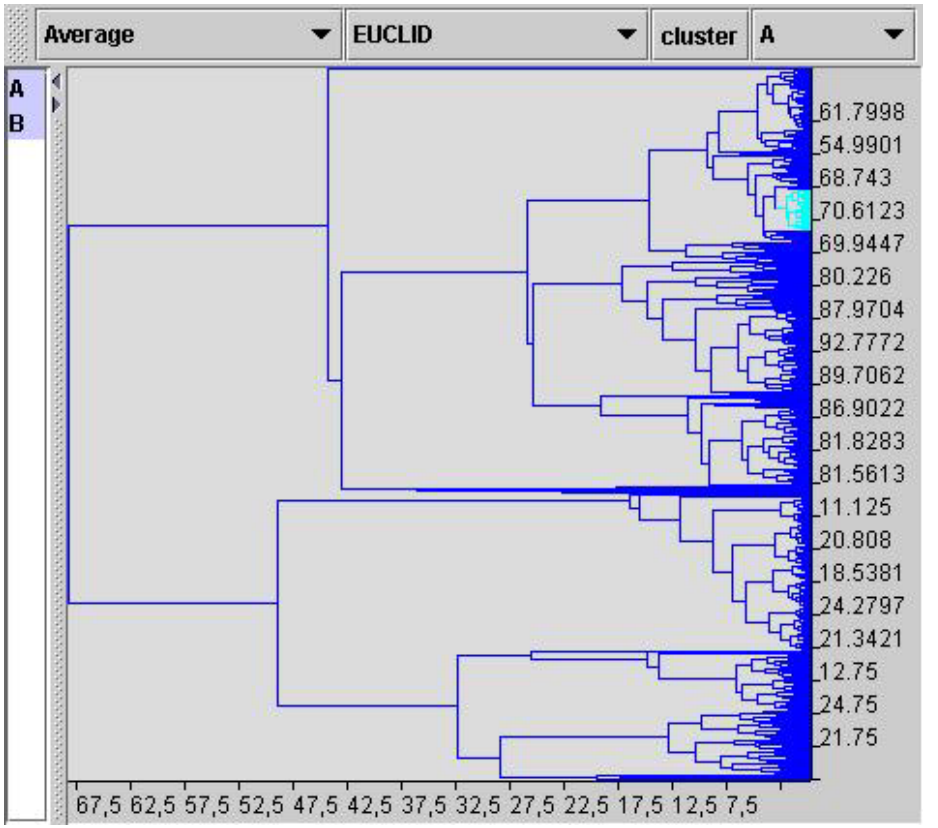
Clustering Hierárquico: Exemplo 3



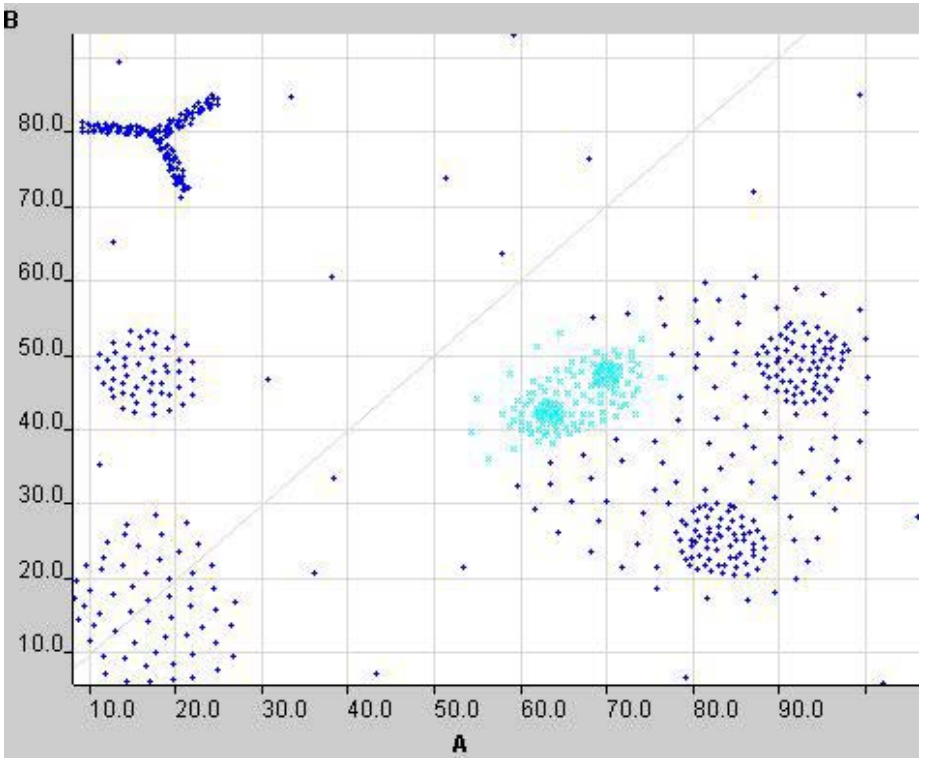


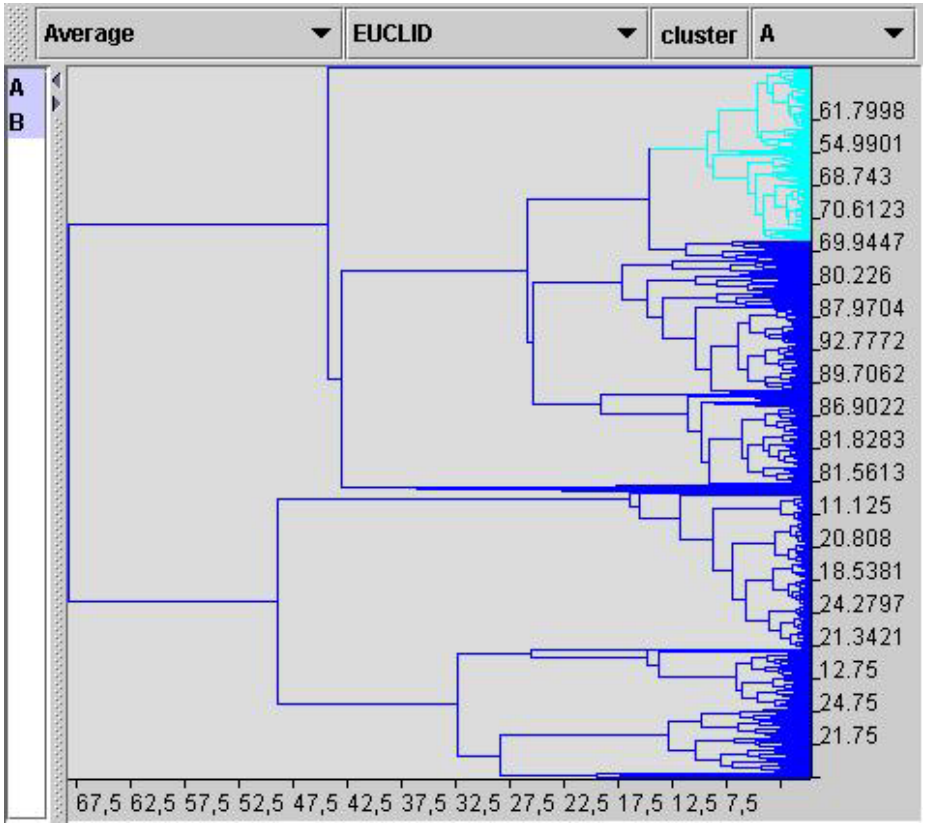
Clustering Hierárquico: Exemplo 3



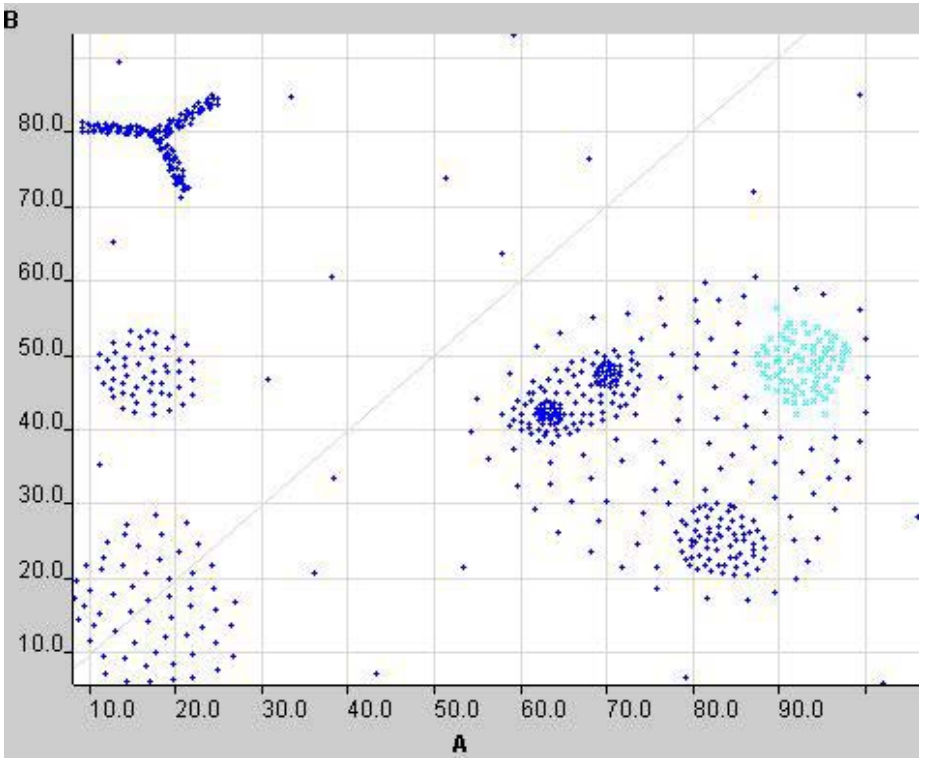


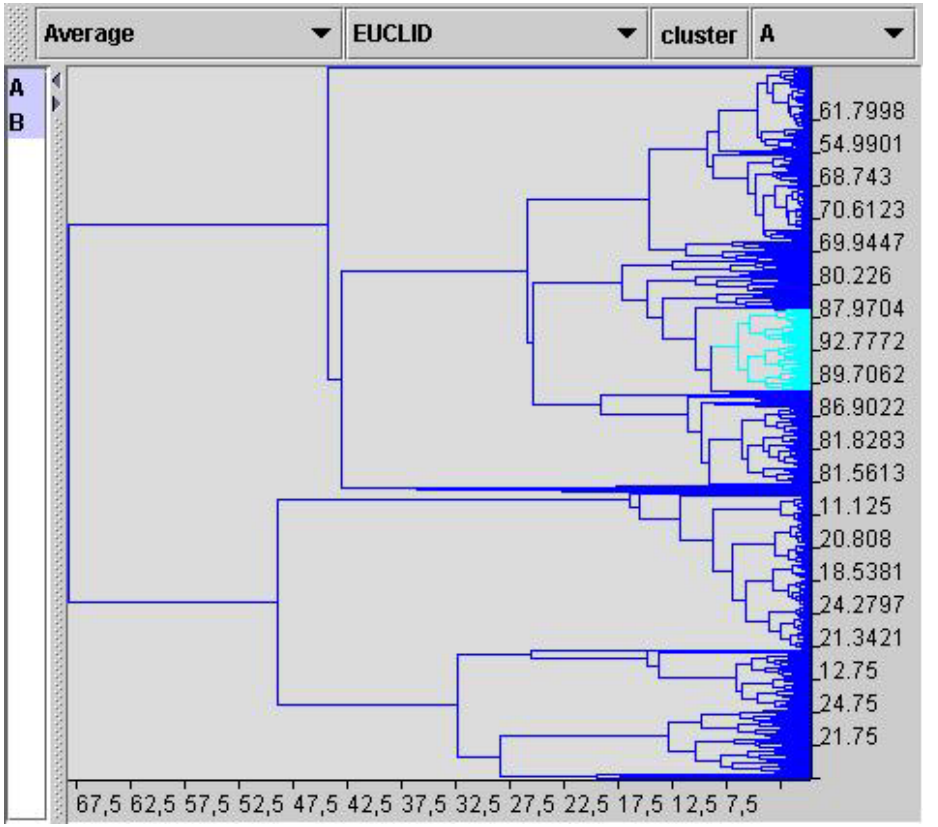
Clustering Hierárquico: Exemplo 3



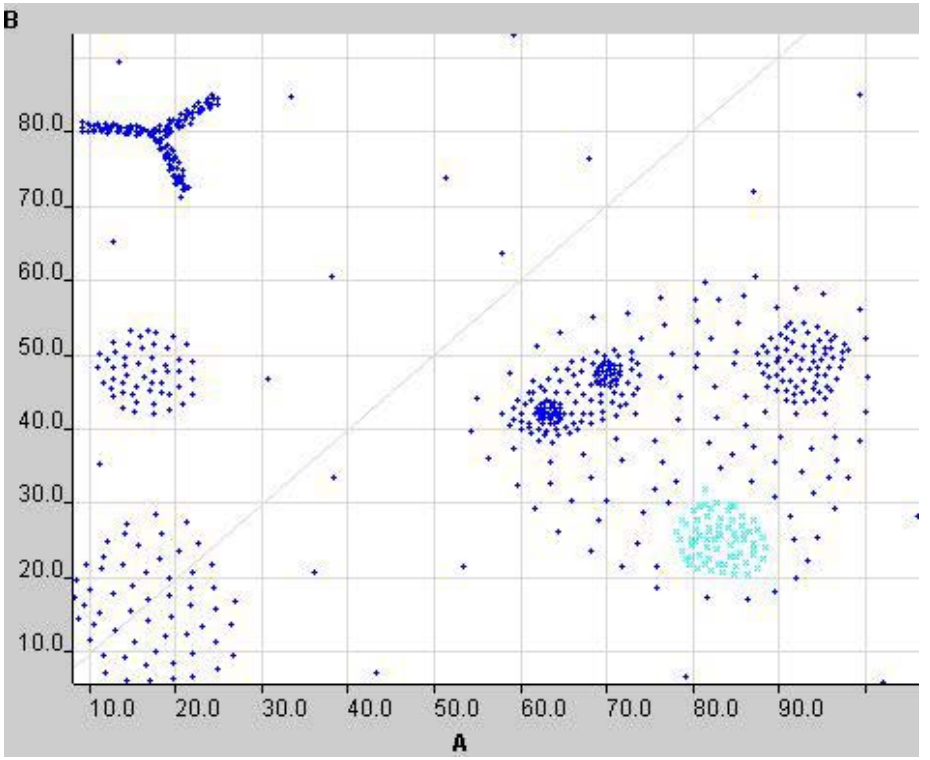


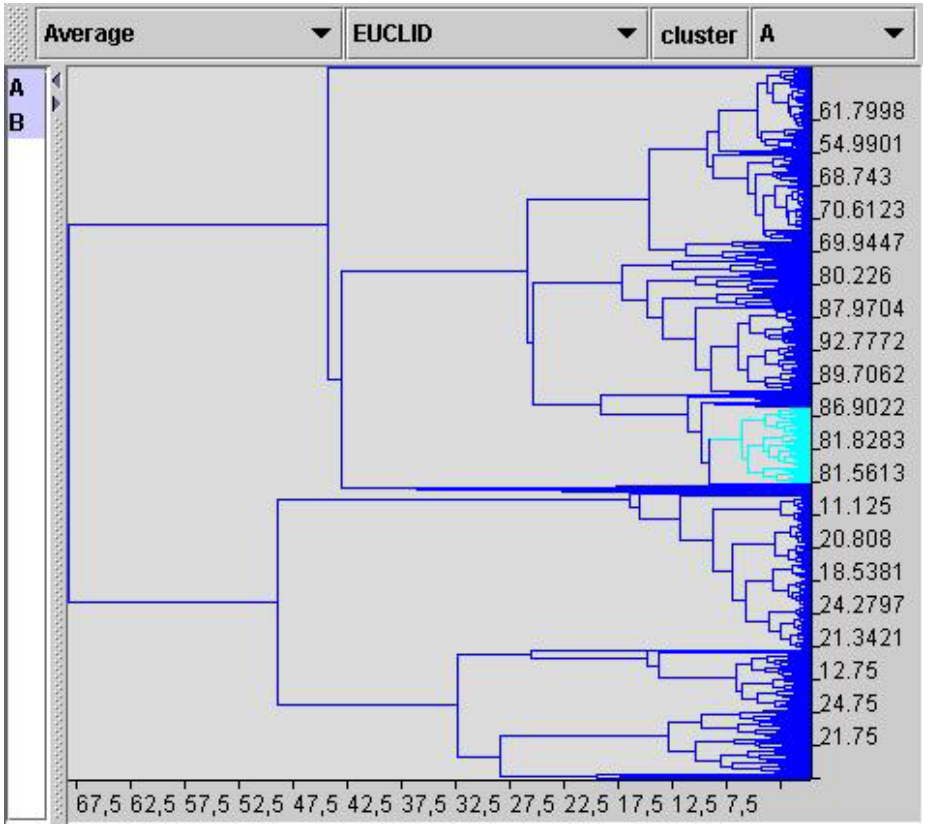
Clustering Hierárquico: Exemplo 3



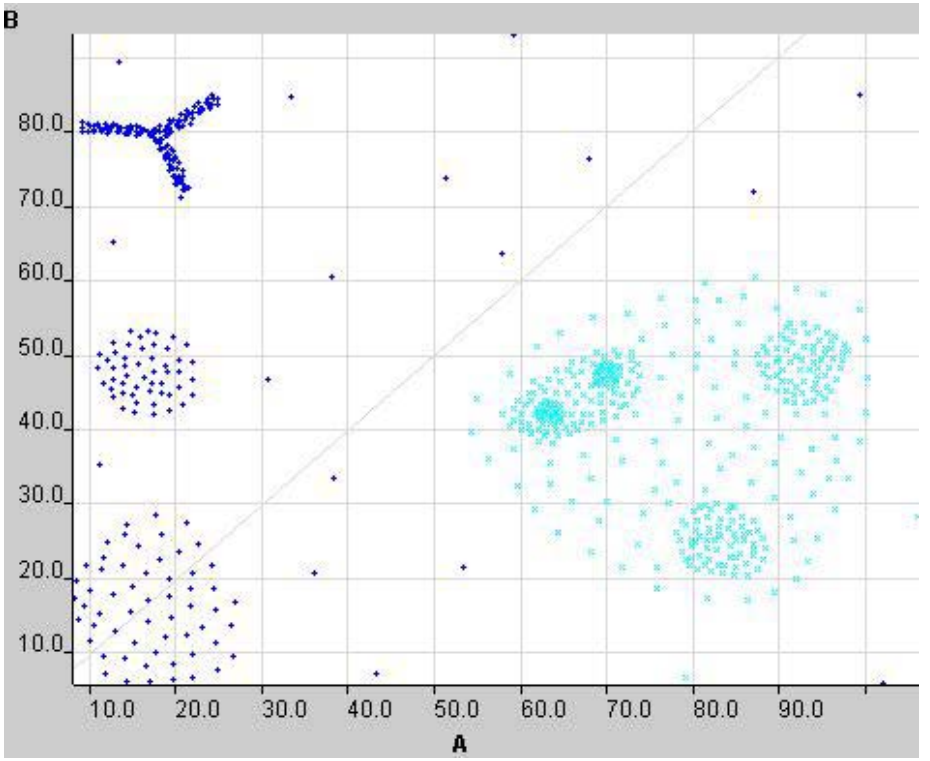


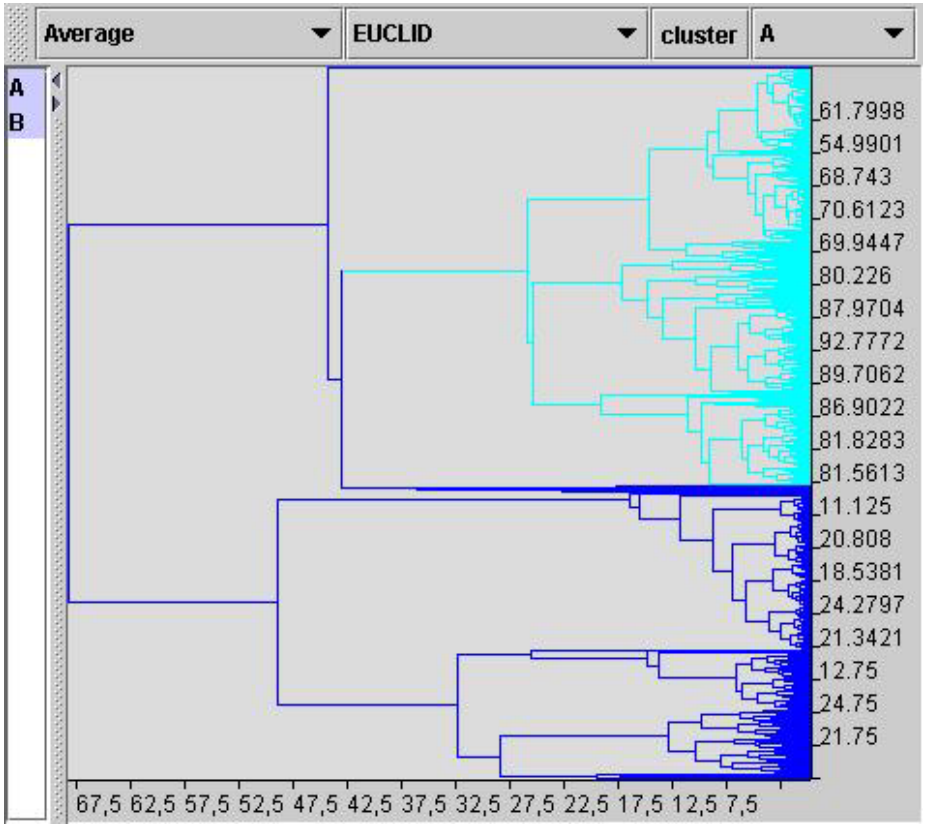
Clustering Hierárquico: Exemplo 3





Clustering Hierárquico: Exemplo 3





Clustering Hierárquico: Exemplo de Aplicação

- Alinhamento múltiplo de seqüências
  - Dado um conjunto de seqüências, produzir um alinhamento global de todas as seqüências contra todas as demais
  - NP-hard
  - Uma heurística popular é utilizar clustering hierárquico
- Estratégia
  - Cada cluster é representado por sua seqüência consenso
  - Quando os clusters são intercalados, suas seqüências consensos são alinhadas via alinhamento ótimo (optimal pairwise alignment)
  - A heurística utiliza clustering hierárquico para juntar as seqüências mais similares primeiro, sendo que o objetivo é minimizar erros potenciais no alinhamento
  - Uma versão mais sofisticada deste método encontra-se implementada no programa clustalw (http://www.ebi.ac.uk/clustalw/)

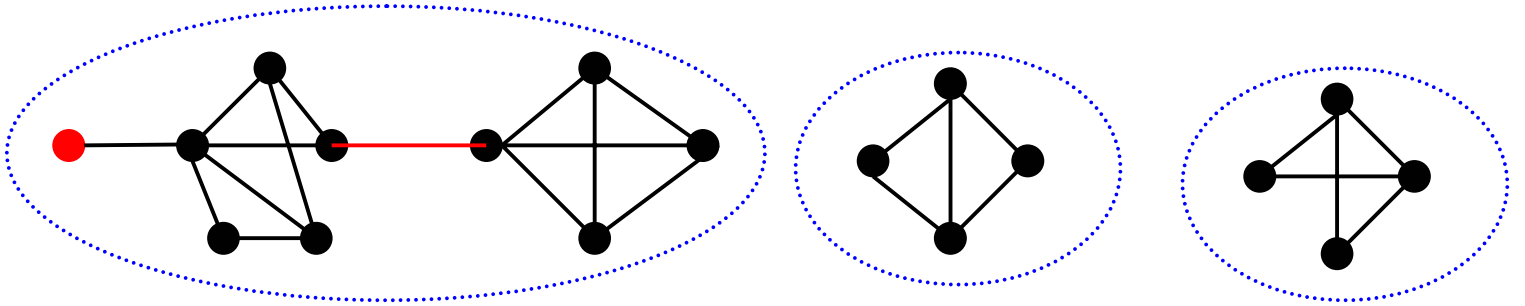
Clustering Hierárquico: Problemas

- A forma mais utilizada, single-link clustering, é particularmente *greedy*
  - Se dois pontos provenientes de clusters disjuntos encontram-se próximos entre si, a distinção entre clusters será perdida
  - Por outro lado, average- e complete-link clustering têm seus *bias* voltados para clusters esféricos da mesma maneira que K-means
- Na realidade não produz clusters; o usuário deve decidir onde “cortar” a árvore em grupos
- Como em K-means, é sensível a ruído e *outliers*

Clustering Utilizando Grafos

- Defina a similaridade de um grafo sobre um conjunto de objetos da seguinte maneira:
  - Vértices são os próprios objetos
  - Arestas interligam objetos que são considerados "similares"
    - Arestas podem ser ponderadas pelo grau de similaridade
- Um componente conexo é um conjunto maximal de objetos tal que cada objeto é alcançável através dos demais
- Um corte de peso mínimo é um conjunto de arestas de peso total mínimo que define um novo componente conexo no grafo

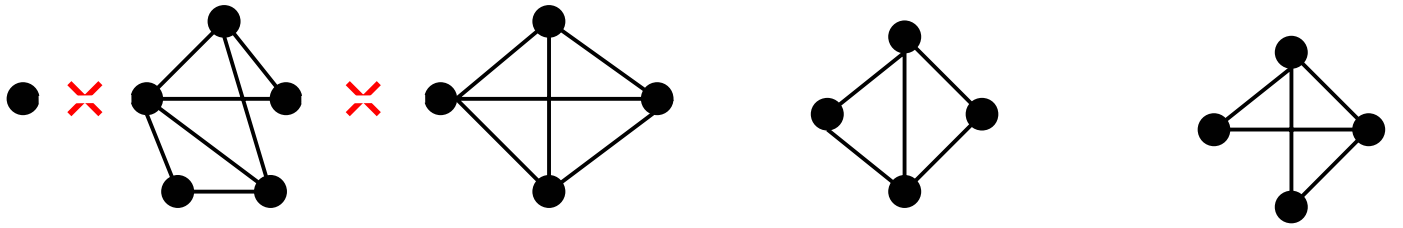
Clustering Utilizando Componentes Conexos



- O grafo acima tem 3 componentes conexos (ou clusters)
- O algoritmo para encontrá-los é muito rápido e simples
- Problemas com este método (no grafo exemplo)
  - O vértice vermelho não é similar à maioria dos objetos em seu cluster
  - A aresta vermelha conecta dois componentes que deveriam provavelmente estar separados

Corte de Peso Mínimo para Clustering

• Executar o algoritmo de corte de peso mínimo no grafo anterior para produzir um resultado melhor (assumindo o peso de cada aresta igual a 1):



• Se os objetos dentro de um cluster são muito mais similares que objetos entre outros clusters, então o método funciona bem

• Problemas

  • Algoritmo é lento e potencialmente deve ser executado várias vezes

  • Não é claro quando parar a execução do algoritmo

Clustering Utilizando Grafos: Exemplo de Aplicação

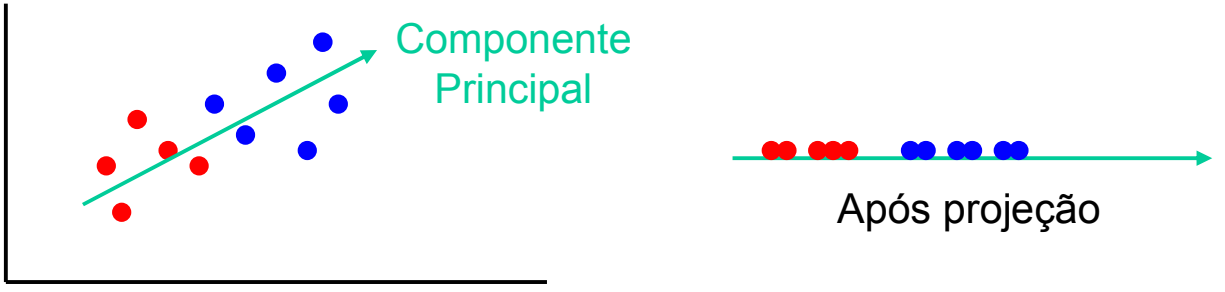
- EST Clustering
  - Dado: um conjunto de seqüências curtas de DNA que são derivadas de genes expressos no genoma
  - Produzir: um mapeamento das seqüencias para sua seqüência original no gene
  - Defina duas seqüências como “similares” se elas se sobrepõem uma certa quantidade
- Cada gene deve ter seu próprio componente conexo no grafo de similaridade
- Alguns fragmentos podem estar compartilhados entre genes, ou genes próximos podem compartilhar uma aresta
- Um algoritmo de corte de peso mínimo pode ser utilizado para solucionar discrepâncias ocasionais

Principal Component Analysis

- Problema: objetos possuem muitos atributos para serem visualizados ou manipulados convenientemente
  - Por exemplo, um simples experimento de microarray pode ter de 6.000-8.000 genes
- PCA é um método para reduzir o número de atributos de dados numéricos enquanto tenta preservar a estrutura do cluster
  - Depois da PCA, espera-se obter os mesmos clusters como se os objetos fossem “clusterizados” antes da PCA
  - Depois da PCA, gráficos dos objetos devem ainda ter clusters “caindo” nos grupos esperados
  - Utilizando PCA para reduzir os objetos para 2 ou 3 dimensões, programas convencionais de visualização podem ser utilizados

PCA: Algoritmo

- Considerar os dados como uma matriz *n* por *m* na qual as linhas são os objetos e as colunas são os atributos
- Os auto-vetores correspondente aos maiores *d* auto-valores da matriz são os "componentes principais"
- Ao projetar os objetos nesses vetores, obtém-se pontos *d*-dimensionais
- Considere o exemplo abaixo, projetando objetos 2D com 2 clusters (vermelho e azul) em 1 dimensão



Desafios em Clustering

- Cálculo de Similaridade
  - Resultados dos algoritmos dependem inteiramente da métrica de similaridade utilizada
  - Os sistemas de clustering fornecem pouco auxílio em como escolher a similidade adequada aos objetos sendo estudados
  - Calcular a correta similaridade de dados de diferentes tipos pode ser difícil
  - Similaridade é muito dependente da representação dos dados. Deve-se
    - Normalizar?
    - Representar um dado numericamente, categoricamente, etc.?
- Seleção de Parâmetros
  - Algoritmos atuais requerem muito parâmetros arbitrários, que devem ser especificados pelo usuário

Conclusão

- Clustering é uma método útil de explorar dados, mais ainda muito *ad hoc*
- Bons resultados são dependentes na escolha da correta representação dos dados e da métrica de similaridade
  - Dados: categórico, numérico, booleano
  - Similaridade: distância, correlação, etc.
- Escolha dentre muitos algoritmos, cada um com vantagens e problemas
  - *k*-means, hierárquico, praticionamento de grafos, etc.