

Operações de consulta por similaridade em grandes bases de dados complexos

Profa. Dra. Maria Camila Nardini Barioni
camila.barioni@ufabc.edu.br

Centro de Matemática Computação e Cognição - **UFABC**

Campinas
24 de abril de 2009



Roteiro

- 1 Introdução
- 2 Conceitos Fundamentais
- 3 Suportando Consultas por Similaridade em SQL
- 4 Algoritmo PAM-SLIM
- 5 Protótipo SIREN
- 6 Referências

Roteiro

- 1 Introdução
- 2 Conceitos Fundamentais
- 3 Suportando Consultas por Similaridade em SQL
- 4 Algoritmo PAM-SLIM
- 5 Protótipo SIREN
- 6 Referências

Introdução

Motivação

- Os SGBD foram desenvolvidos para lidar com dados numéricos e textuais (**dados tradicionais**)
 - Busca ⇒ baseada na propriedade de ordem total

Introdução

Motivação

- Os SGBD foram desenvolvidos para lidar com dados numéricos e textuais (**dados tradicionais**)
 - Busca ⇒ baseada na propriedade de ordem total

100 < 1.000 < 10.000

A < B < C

Introdução

Motivação

- Os SGBD foram desenvolvidos para lidar com dados numéricos e textuais (**dados tradicionais**)
 - Busca ⇒ baseada na propriedade de ordem total

$100 < 1.000 < 10.000$

$A < B < C$

- As aplicações de bases de dados modernas necessitam lidar com tipos de **dados complexos**
 - Busca ⇒ baseada em similaridade

Introdução

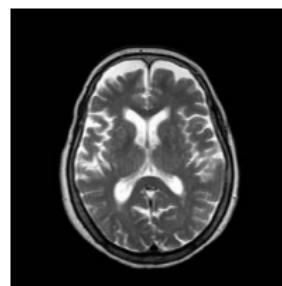
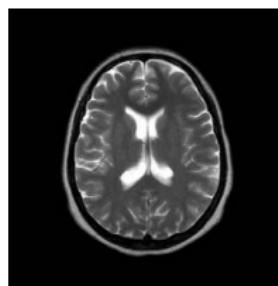
Motivação

- Os SGBD foram desenvolvidos para lidar com dados numéricos e textuais (**dados tradicionais**)
 - Busca ⇒ baseada na propriedade de ordem total

$100 < 1.000 < 10.000$

$A < B < C$

- As aplicações de bases de dados modernas necessitam lidar com tipos de **dados complexos**
 - Busca ⇒ baseada em similaridade



Introdução

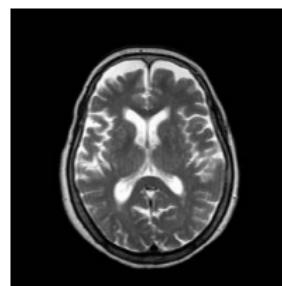
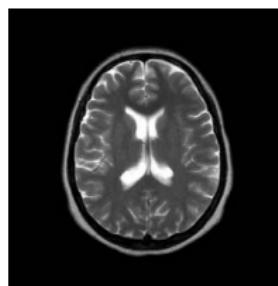
Motivação

- Os SGBD foram desenvolvidos para lidar com dados numéricos e textuais (**dados tradicionais**)
 - Busca ⇒ baseada na propriedade de ordem total

$100 < 1.000 < 10.000$

$A < B < C$

- As aplicações de bases de dados modernas necessitam lidar com tipos de **dados complexos**
 - Busca ⇒ baseada em similaridade



Introdução

Consultas por Similaridade

Contribuem para a necessidade de criar um suporte para a realização de consultas por similaridade em SGBDR:

Introdução

Consultas por Similaridade

Contribuem para a necessidade de criar um suporte para a realização de consultas por similaridade em SGBDR:

- O aumento da quantidade de dados de domínios complexos armazenados em bases de dados relacionais

Introdução

Consultas por Similaridade

Contribuem para a necessidade de criar um suporte para a realização de consultas por similaridade em SGBDR:

- O aumento da quantidade de dados de domínios complexos armazenados em bases de dados relacionais
- A integração de métodos de mineração de dados com SGBDR
 - Ponto fundamental: disponibilização de operações básicas para as técnicas de mineração de dados existentes
 - ex.: **detecção de agrupamentos de dados** ⇒ cálculo de medidas de similaridade entre os pares de objetos de um conjunto de dados

Diante disso...

Questões importantes:

- ① Como comparar dados complexos?
- ② Como indexar conjuntos de dados complexos?
- ③ Como expressar consultas sobre esses dados no SGBD?
- ④ Como otimizar os algoritmos de detecção de agrupamentos de maneira a torná-los exequíveis em SGBD?

Roteiro

- 1 Introdução
- 2 Conceitos Fundamentais
- 3 Suportando Consultas por Similaridade em SQL
- 4 Algoritmo PAM-SLIM
- 5 Protótipo SIREN
- 6 Referências

Conceitos Fundamentais

Domínios de dados complexos

Dados de domínios complexos podem ser armazenados em um banco de dados:

Conceitos Fundamentais

Domínios de dados complexos

Dados de domínios complexos podem ser armazenados em um banco de dados:

① Conjunto de atributos de domínios tradicionais

- ex.: séries temporais, informações geo-referenciadas, ...
- comparação por similaridade: aplicação de função de distância sobre os valores dos atributos que os compõem

Conceitos Fundamentais

Domínios de dados complexos

Dados de domínios complexos podem ser armazenados em um banco de dados:

① Conjunto de atributos de domínios tradicionais

- ex.: séries temporais, informações geo-referenciadas, ...
- comparação por similaridade: aplicação de função de distância sobre os valores dos atributos que os compõem

② Objeto binário BLOB

- ex.: imagens, trilhas de áudio, ...
- comparação por similaridade: aplicação de função de distância sobre um conjunto pré-definido de **características** inerentes aos dados
 - ex.: recuperação de imagens por conteúdo

Conceitos Fundamentais

Como representar dados complexos?

Representação das Imagens

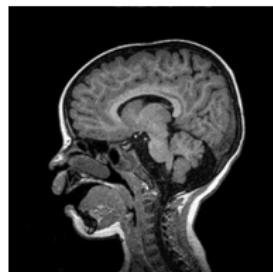


Imagen
original

Conceitos Fundamentais

Como representar dados complexos?

Representação das Imagens



Imagen
original



Extrator de
Características

Conceitos Fundamentais

Como representar dados complexos?

Representação das Imagens



Conceitos Fundamentais

Como representar dados complexos?

Representação das Imagens



Conceitos Fundamentais

Como representar dados complexos?

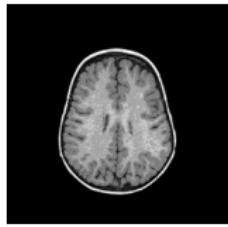
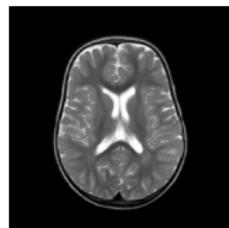
Representação das Imagens



- Exemplos de características visuais: cor, textura e forma

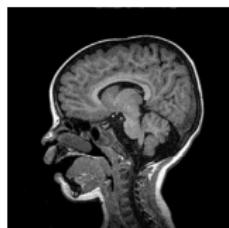
Conceitos Fundamentais

Como representar dados complexos?

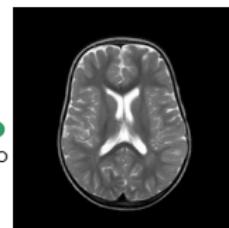


Conceitos Fundamentais

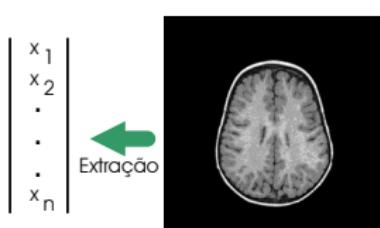
Como representar dados complexos?



$$\begin{matrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{matrix}$$

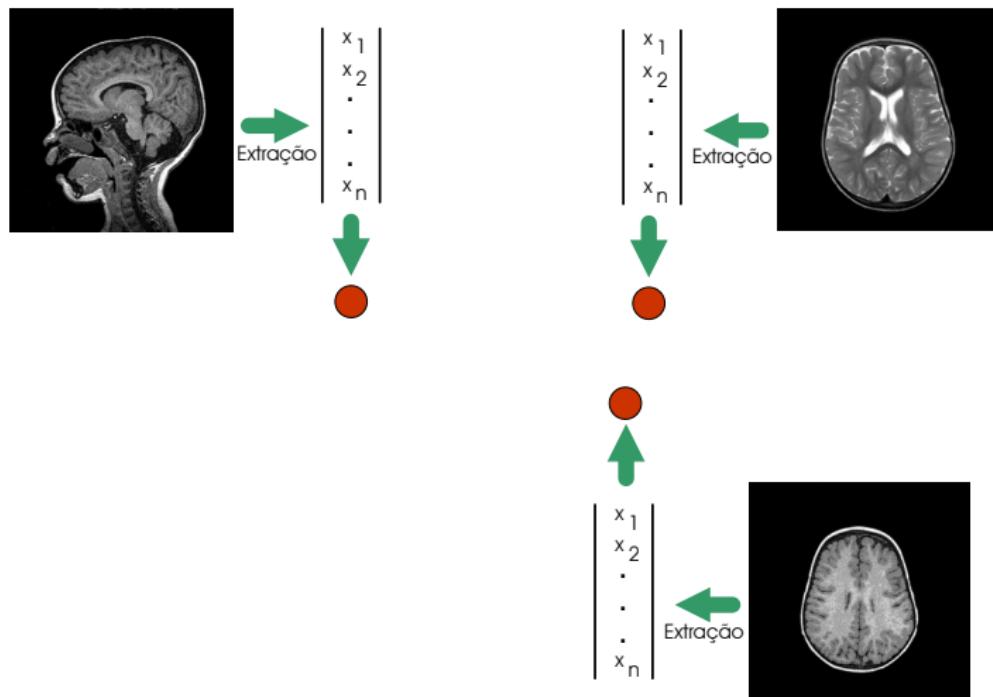


$$\begin{matrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{matrix}$$



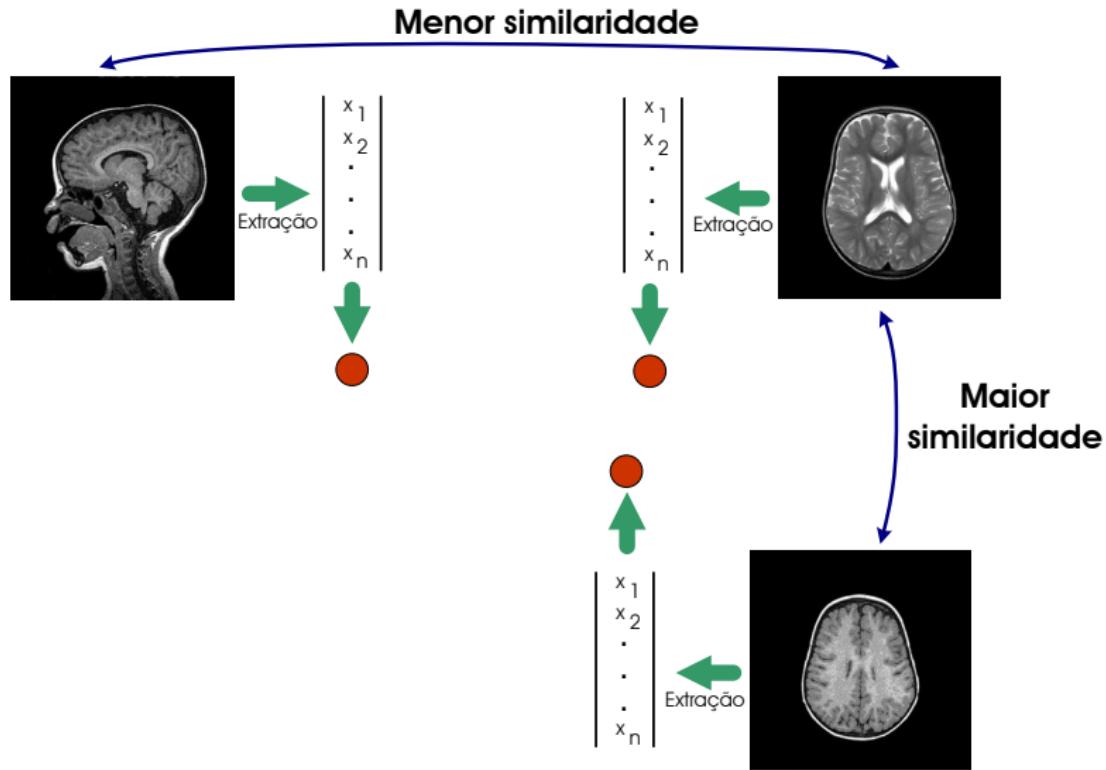
Conceitos Fundamentais

Como representar dados complexos?



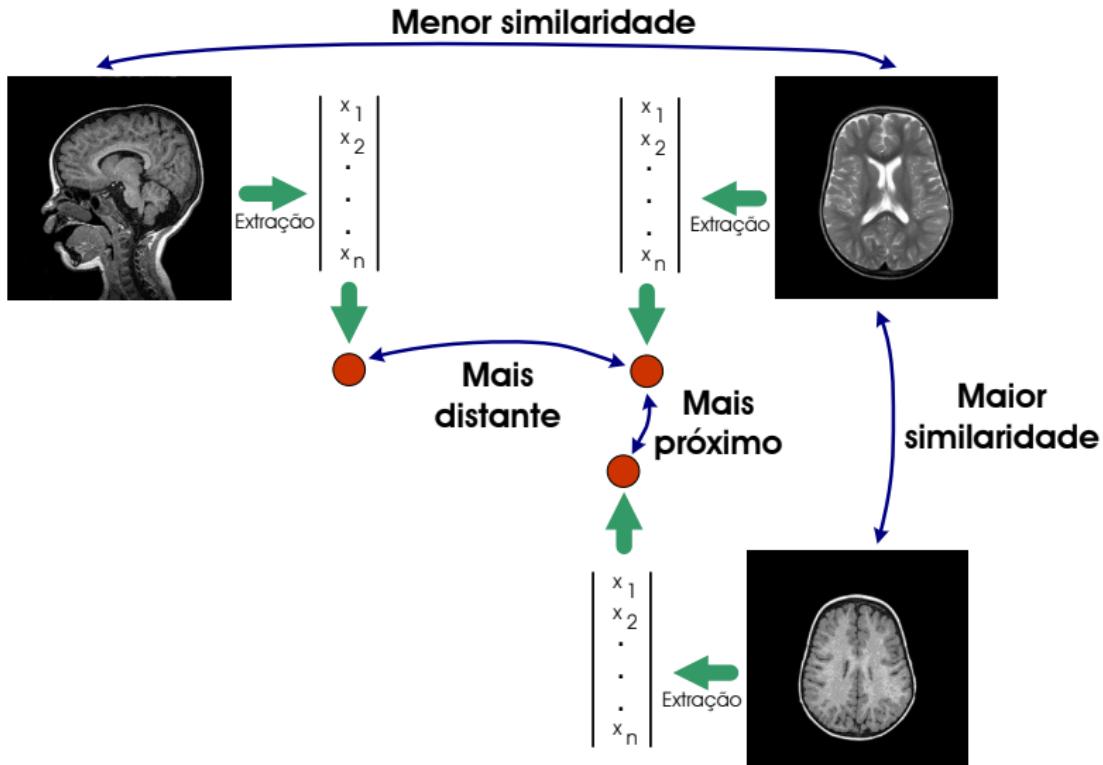
Conceitos Fundamentais

Como representar dados complexos?



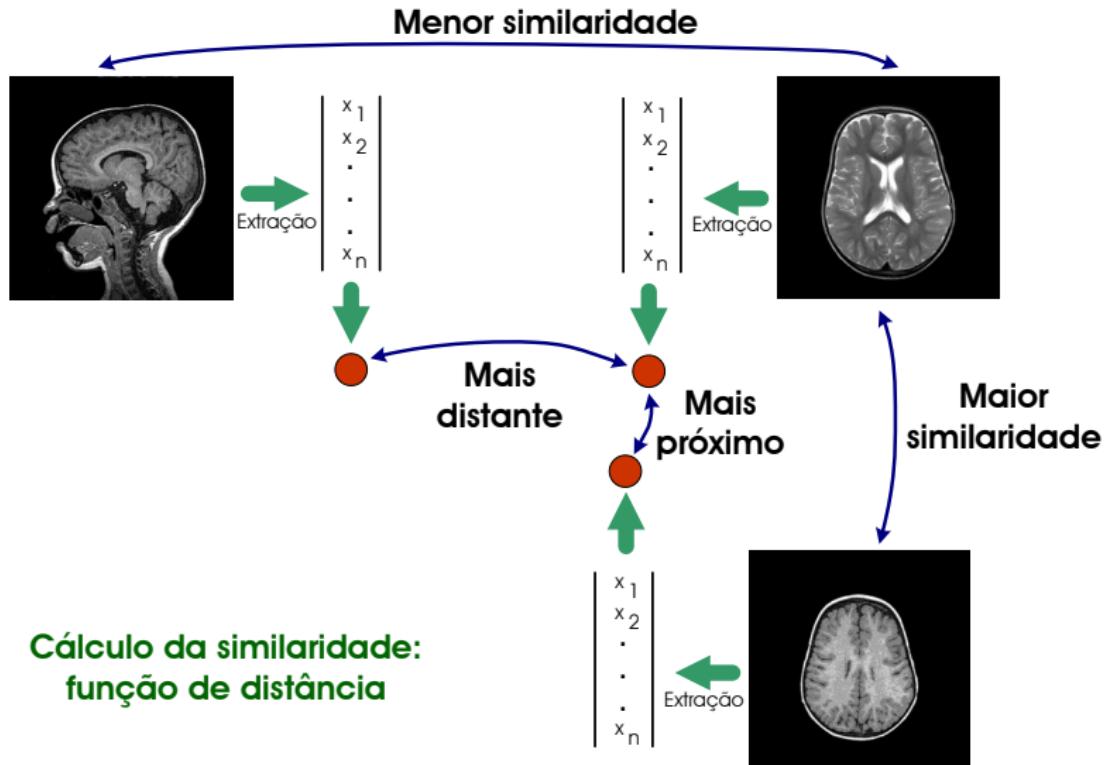
Conceitos Fundamentais

Como representar dados complexos?



Conceitos Fundamentais

Como representar dados complexos?



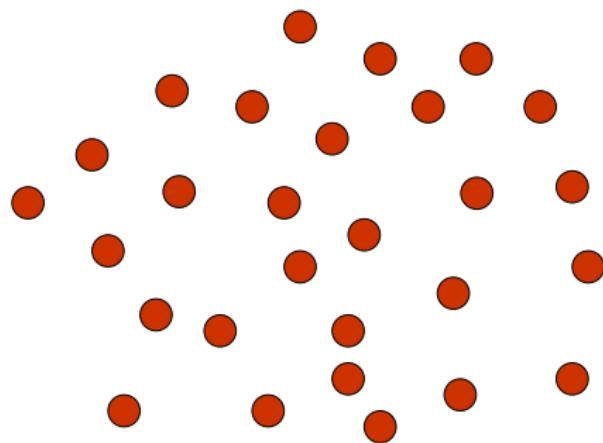
Conceitos Fundamentais

Como representar dados complexos?



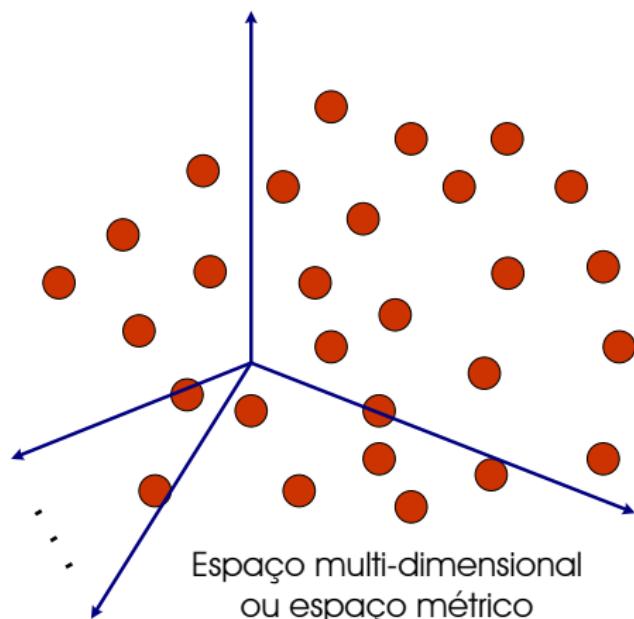
Conceitos Fundamentais

Como representar dados complexos?



Conceitos Fundamentais

Como representar dados complexos?



Conceitos Fundamentais

Como comparar dados complexos?

Espaço Métrico

Definição: Um **espaço métrico** M é definido pelo par $\{\mathbb{S}, d()\}$, onde \mathbb{S} define o domínio dos dados e $d()$ é uma **função de distância** que atende às propriedades:

- Simetria: $d(s_1, s_2) = d(s_2, s_1)$
- Não-negatividade: $0 \leq d(s_1, s_2) < \infty$
- Desigualdade triangular: $d(s_1, s_2) \leq d(s_1, s_3) + d(s_3, s_2)$

Cálculo da Similaridade \Rightarrow função de distância

Conceitos Fundamentais

Como comparar dados complexos?

- Exemplo de função de distância para domínios vetoriais:

- função de distância L_p (Minkowski)

$$d(x, y) = \sqrt[p]{\sum_{i=1}^n |x_i - y_i|^p}$$

onde n é a dimensão do espaço vetorial

Conceitos Fundamentais

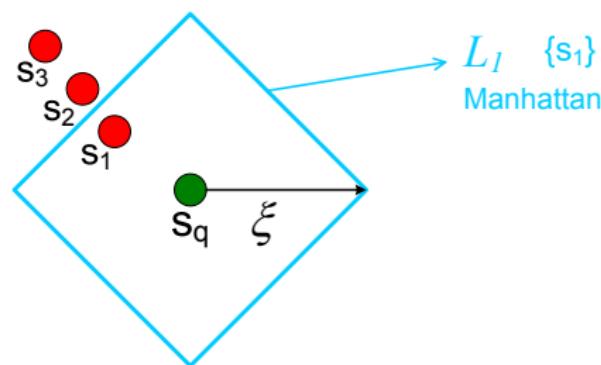
Como comparar dados complexos?

- Exemplo de função de distância para domínios vetoriais:

- função de distância L_p (Minkowski)

$$d(x, y) = \sqrt[p]{\sum_{i=1}^n |x_i - y_i|^p}$$

onde n é a dimensão do espaço vetorial



Conceitos Fundamentais

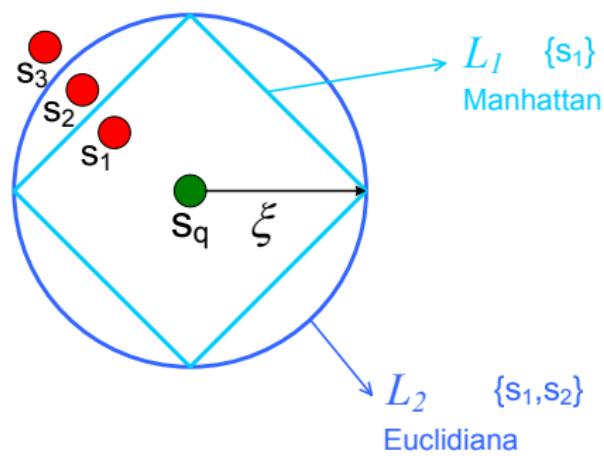
Como comparar dados complexos?

- Exemplo de função de distância para domínios vetoriais:

- função de distância L_p (Minkowski)

$$d(x, y) = \sqrt[p]{\sum_{i=1}^n |x_i - y_i|^p}$$

onde n é a dimensão do espaço vetorial



Conceitos Fundamentais

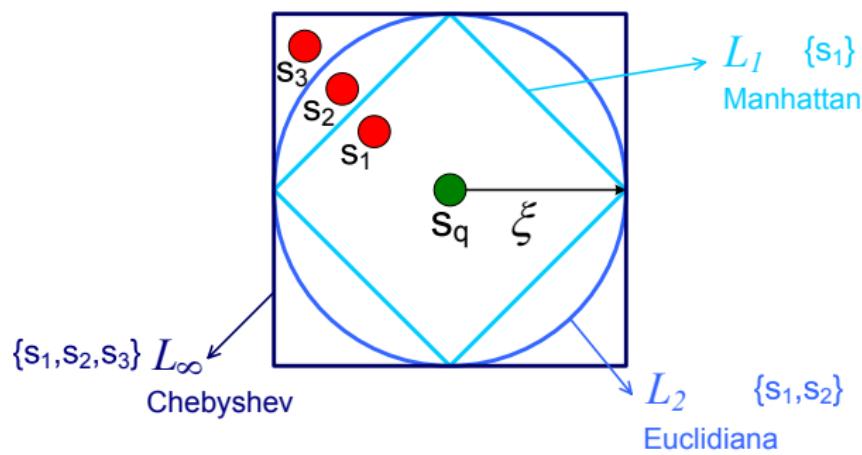
Como comparar dados complexos?

- Exemplo de função de distância para domínios vetoriais:

- função de distância L_p (Minkowski)

$$d(x, y) = \sqrt[p]{\sum_{i=1}^n |x_i - y_i|^p}$$

onde n é a dimensão do espaço vetorial



Conceitos Fundamentais

Como comparar dados complexos?

- Exemplo de função de distância para domínios não-vetoriais:
 - Função de distância $L_{EDIT}(x, y) \Rightarrow$ quantidade mínima de caracteres que precisam ser substituídos, removidos ou inseridos em x , para que ela se torne igual a y
 - Exemplos:
 - $L_{EDIT}('amora', 'aroma') = 2$ (duas substituições)
 - $L_{EDIT}('amora', 'amor') = 1$ (uma remoção)

Conceitos Fundamentais

Como consultar?

Na literatura apresentam-se:

- Duas operações que comparam um objeto de referência com aqueles armazenados em uma coleção de objetos \Rightarrow **seleção**
 - **Consulta por abrangência (Range query)**
 - **Consulta aos k -vizinhos mais próximos (k -Nearest neighbor query)**

Conceitos Fundamentais

Como consultar?

Na literatura apresentam-se:

- Duas operações que comparam um objeto de referência com aqueles armazenados em uma coleção de objetos \Rightarrow **seleção**
 - **Consulta por abrangência** (*Range query*)
 - **Consulta aos k -vizinhos mais próximos** (*k -Nearest neighbor query*)
- Três operações que comparam pares de objetos armazenados em duas coleções de objetos \Rightarrow **junção**
 - **Junção por abrangência** (*Range Join*)
 - **Junção pelos k -vizinhos mais próximos** (*k -Nearest Neighbors Join*)
 - **Junção dos k -pares de vizinhos mais próximos** (*k -Closest Neighbors Join*)

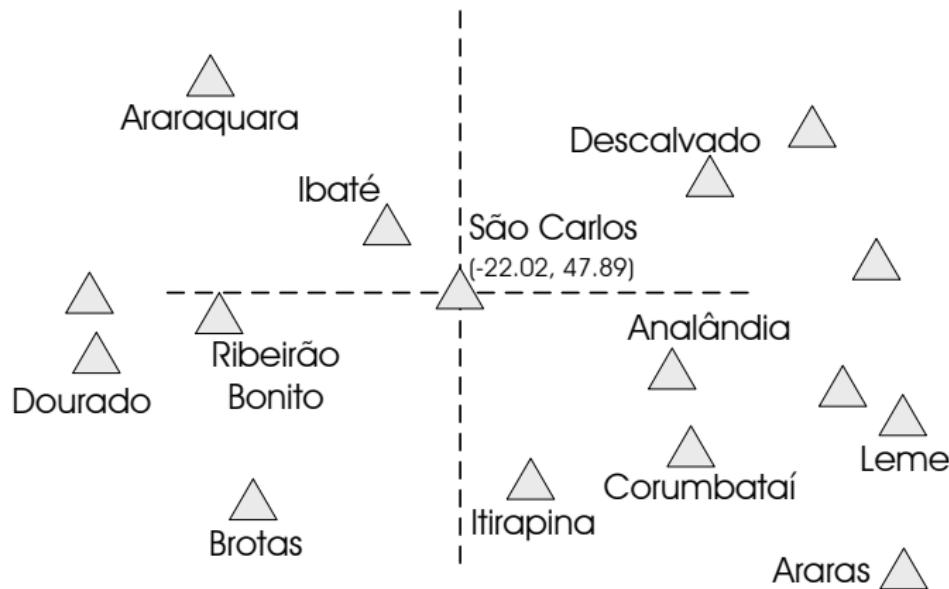
Conceitos Fundamentais

Como consultar?

- Para exemplificar:
 - Conjunto de dados CidadeBR
 - Cada tupla \Rightarrow nome da cidade + coordenadas geográficas
 - Medida de similaridade \Rightarrow distância entre as cidades computada pela aplicação da função de distância Euclidiana sobre as suas coordenadas

Conceitos Fundamentais

Como consultar? – Exemplo Seleção por Similaridade



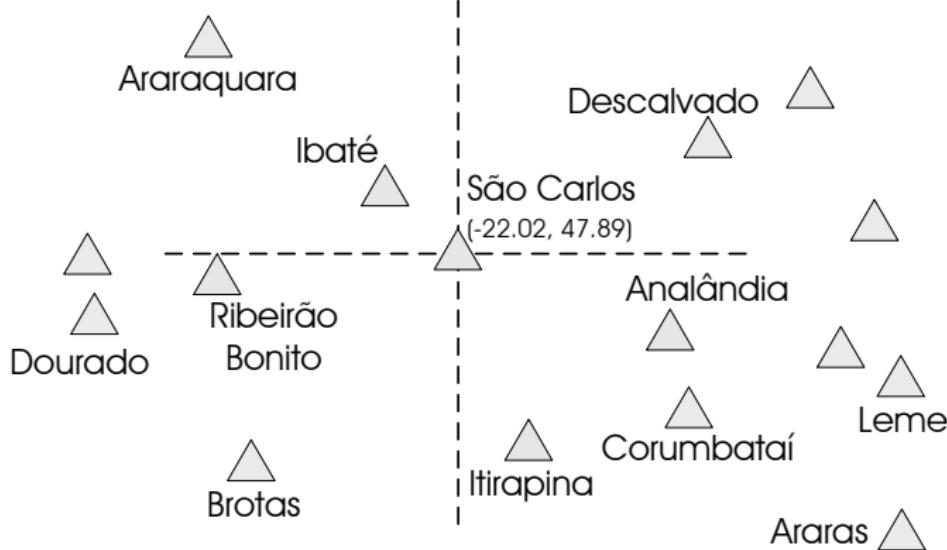
Conceitos Fundamentais

Como consultar? – Exemplo Seleção por Similaridade

Seleção dos k -vizinhos mais próximos:

$\hat{\sigma}_{(k-NNq < d(), \{obj_c\}, k)} CidadeBR$

$k = 5$



L₂

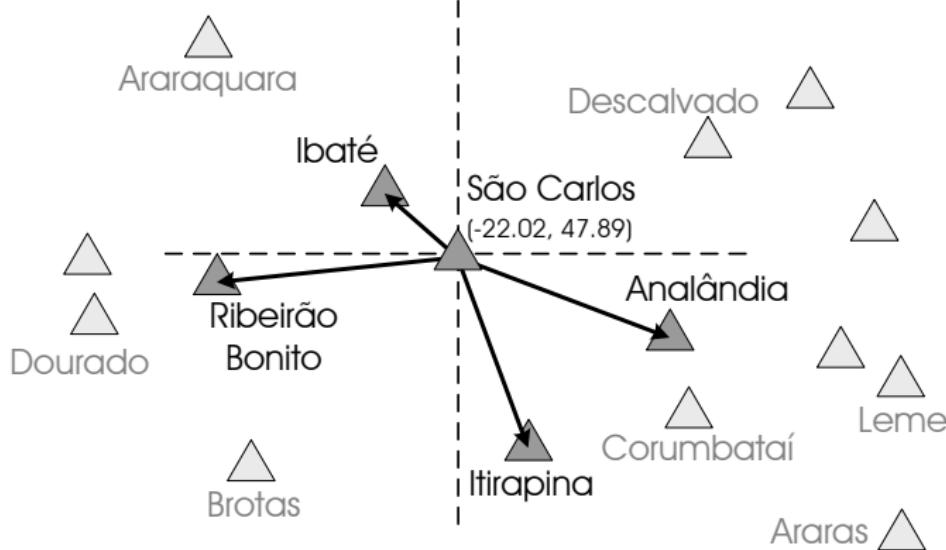
Conceitos Fundamentais

Como consultar? – Exemplo Seleção por Similaridade

Seleção dos k -vizinhos mais próximos:

$\hat{\sigma}_{k=NNq \leq d(0,\{obj_c\},k)} CidadeBR$

$$k = 5$$



L₂

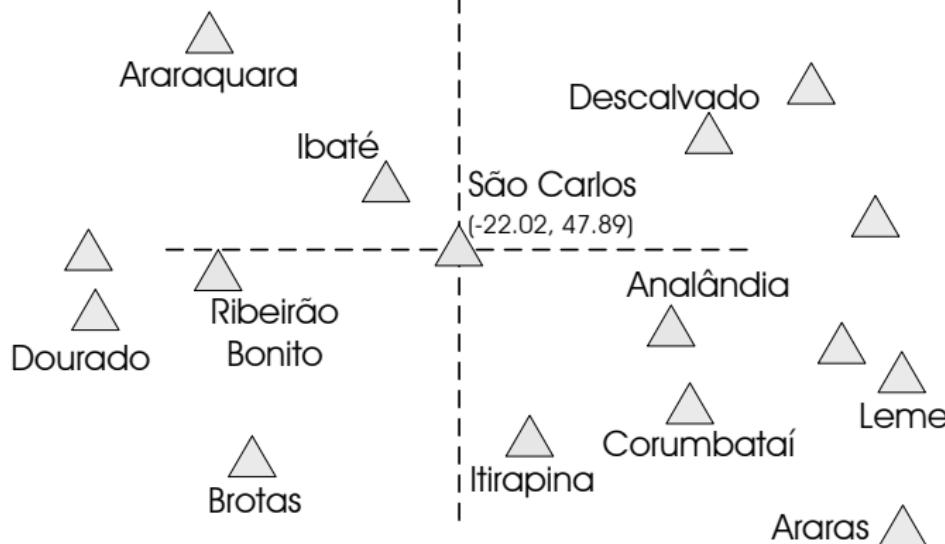
Conceitos Fundamentais

Como consultar? – Exemplo Seleção por Similaridade

Seleção por abrangência:

$\hat{\sigma}_{(Rq < d(), \{obj_c\}, \xi)} CidadeBR$

$$\xi = 0.3$$



L₂

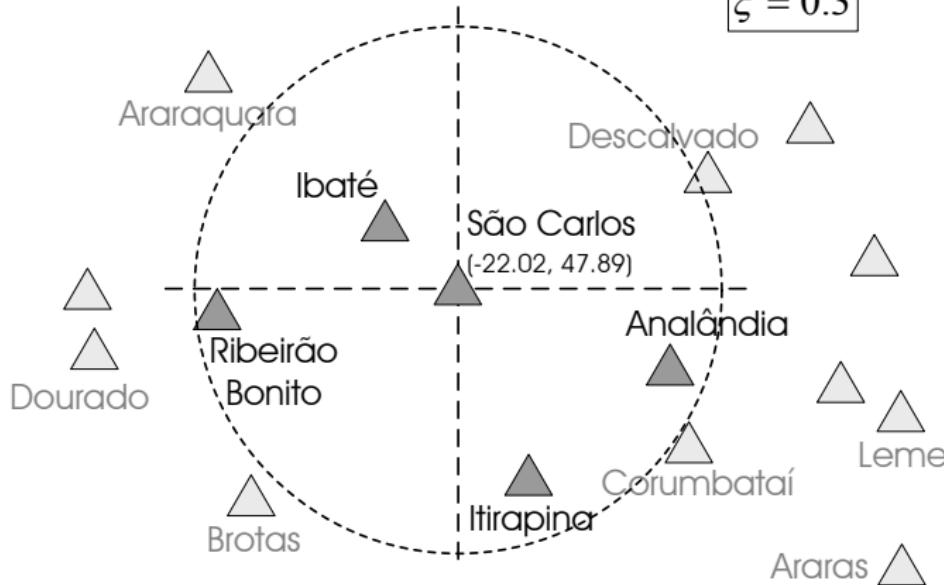
Conceitos Fundamentais

Como consultar? – Exemplo Seleção por Similaridade

Seleção por abrangência:

$\hat{\sigma}_{(Rq < d(), \{obj_c\}, \xi)} CidadeBR$

$$\xi = 0.3$$



L₂

Conceitos Fundamentais

Como consultar? – Exemplo Junção por Similaridade

Junção por abrangência:

$Rq < d(), \xi >$
 $CapitalSE \bowtie CidadeBR$

$$\xi = 0.11$$



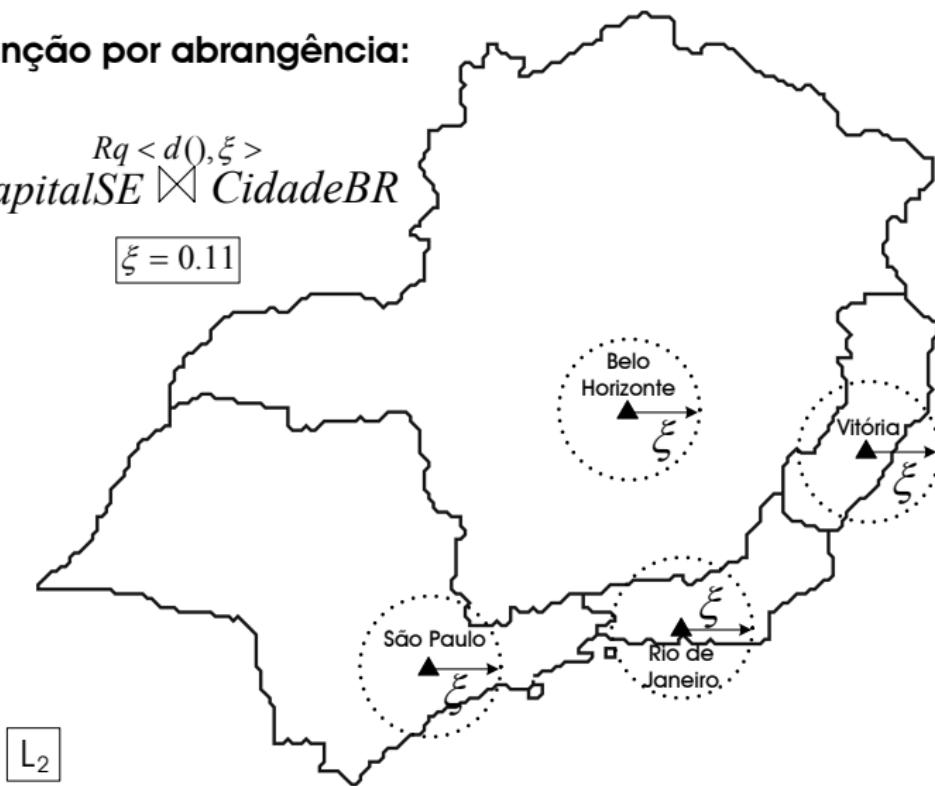
Conceitos Fundamentais

Como consultar? – Exemplo Junção por Similaridade

Junção por abrangência:

$Rq < d(), \xi >$
 $CapitalSE \bowtie CidadeBR$

$$\xi = 0.11$$



Conceitos Fundamentais

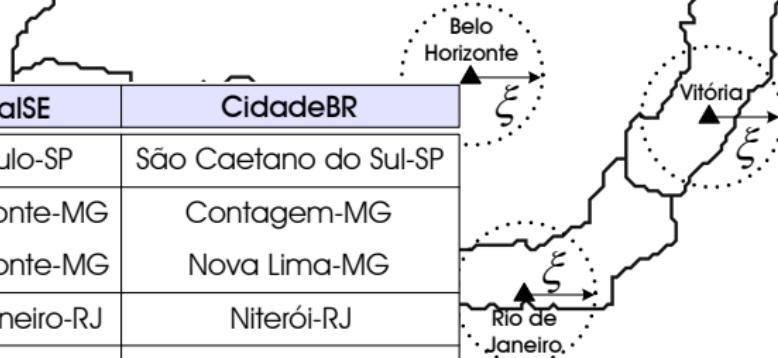
Como consultar? – Exemplo Junção por Similaridade

Junção por abrangência:

$Rq < d(), \xi >$
 $CapitalSE \bowtie CidadeBR$

$$\xi = 0.11$$

CapitalSE	CidadeBR
São Paulo-SP	São Caetano do Sul-SP
Belo Horizonte-MG	Contagem-MG
Belo Horizonte-MG	Nova Lima-MG
Rio de Janeiro-RJ	Niterói-RJ
Vitória-ES	Vila Velha-ES
Vitória-ES	Cariacica-ES



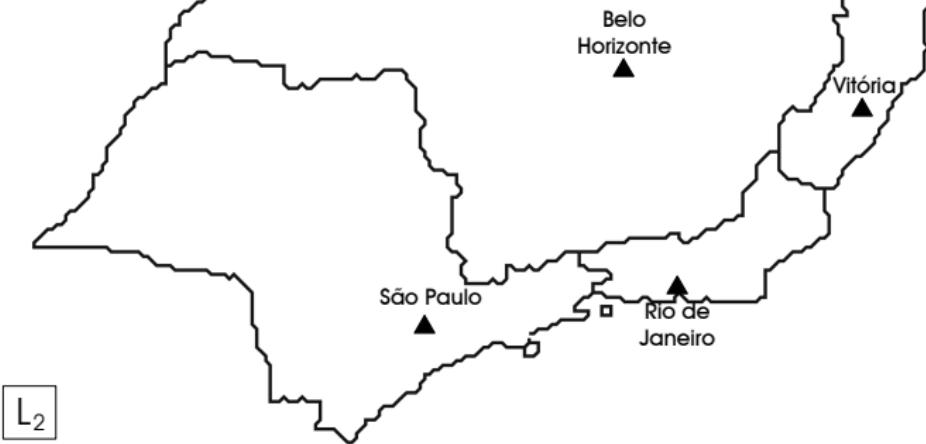
Conceitos Fundamentais

Como consultar? – Exemplo Junção por Similaridade

**Junção pelos k -vizinhos
mais próximos:**

$k - NNq < d(), k >$
CapitalSE \bowtie *CidadeBR*

$$k = 2$$



Conceitos Fundamentais

Como consultar? – Exemplo Junção por Similaridade

**Junção pelos k -vizinhos
mais próximos:**

$$k - NNq < d(), k >$$

$CapitalSE \bowtie CidadeBR$

$$k = 2$$



Conceitos Fundamentais

Como consultar? – Exemplo Junção por Similaridade

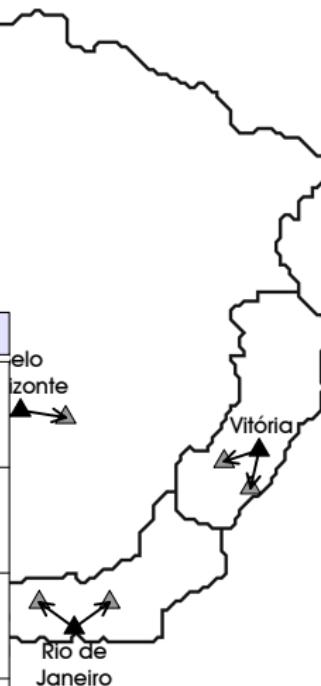
**Junção pelos k -vizinhos
mais próximos:**

$$k - NNq < d(), k >$$

CapitalSE \bowtie *CidadeBR*

$$k = 2$$

CapitalSE	CidadeBR
São Paulo-SP	São Caetano do Sul-SP
São Paulo-SP	Diadema-SP
Belo Horizonte-MG	Contagem-MG
Belo Horizonte-MG	Nova Lima-MG
Rio de Janeiro-RJ	Niterói-RJ
Rio de Janeiro-RJ	Duque de Caxias-RJ
Vitória-ES	Vila Velha-ES
Vitória-ES	Cariacica-ES



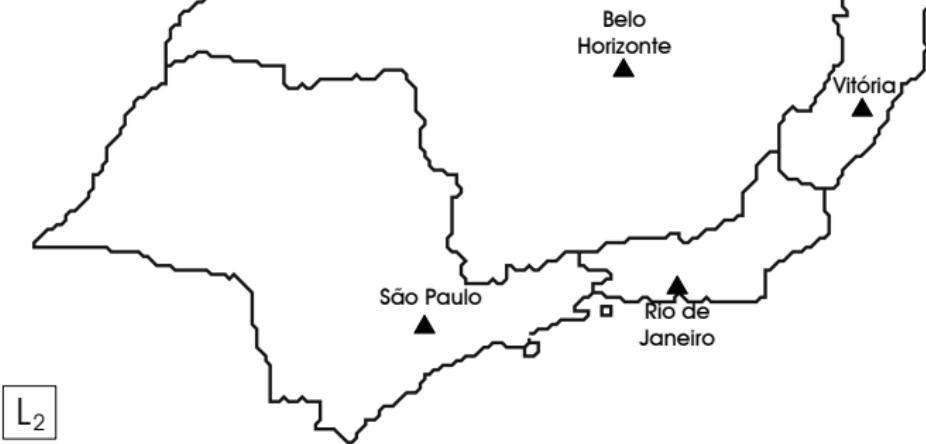
Conceitos Fundamentais

Como consultar? – Exemplo Junção por Similaridade

Junção dos k -pares de vizinhos mais próximos:

$k - CNq < d(), k >$
CapitalSE \bowtie *CidadeBR*

$$k = 2$$



Conceitos Fundamentais

Como consultar? – Exemplo Junção por Similaridade

Junção dos k -pares de vizinhos mais próximos:

$k - CNq < d(), k >$
CapitalSE \bowtie *CidadeBR*

$$k = 2$$



Conceitos Fundamentais

Como consultar? – Exemplo Junção por Similaridade

Junção dos k -pares de vizinhos mais próximos:

$k - CNq < d(), k >$
 $CapitalSE \bowtie CidadeBR$

$$k = 2$$



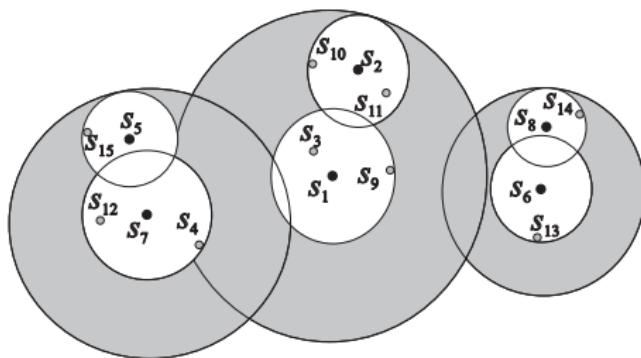
CapitalSE	CidadeBR
Vitória-ES	Vila Velha-ES
Vitória-ES	Cariacica-ES

Conceitos Fundamentais

Como indexar conjuntos de dados complexos?

Estruturas de Indexação (Métodos de Acesso)

- São utilizadas para agilizar a realização de consultas por similaridade
- Para espaços métricos: Métodos de Acesso Métrico (MAM)

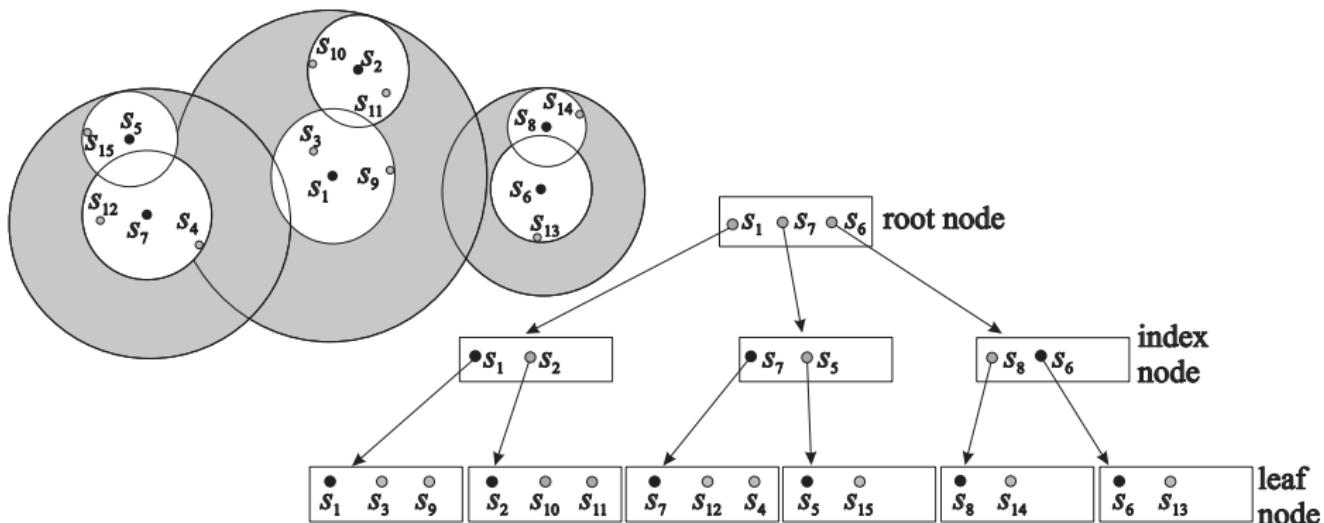


Conceitos Fundamentais

Como indexar conjuntos de dados complexos?

Estruturas de Indexação (Métodos de Acesso)

- São utilizadas para agilizar a realização de consultas por similaridade
- Para espaços métricos: Métodos de Acesso Métrico (MAM)

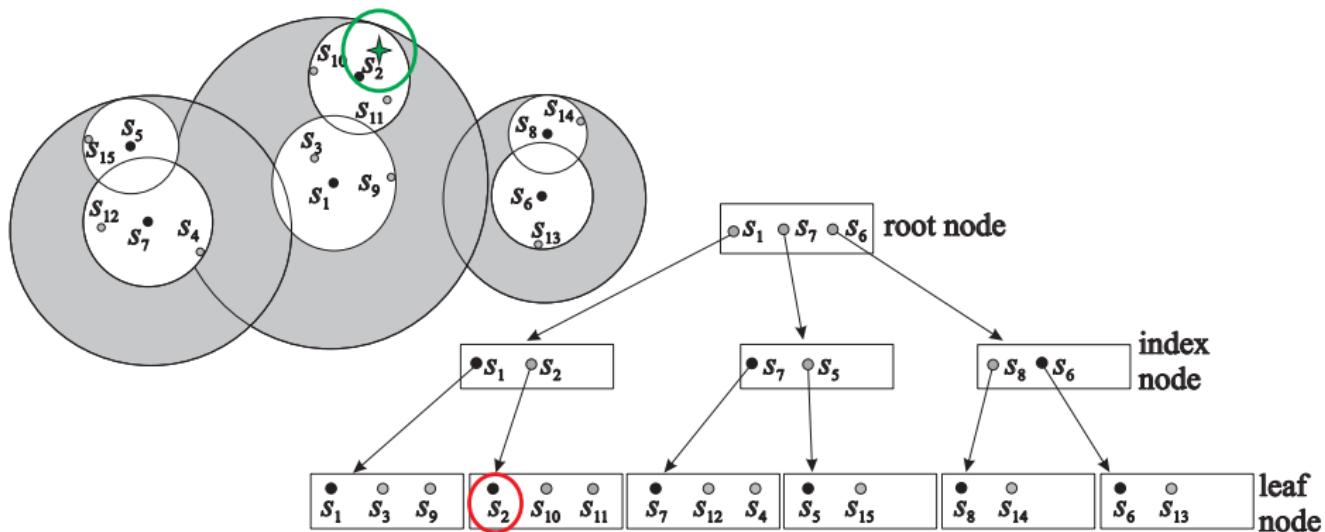


Conceitos Fundamentais

Como indexar conjuntos de dados complexos?

Estruturas de Indexação (Métodos de Acesso)

- São utilizadas para agilizar a realização de consultas por similaridade
- Para espaços métricos: Métodos de Acesso Métrico (MAM)

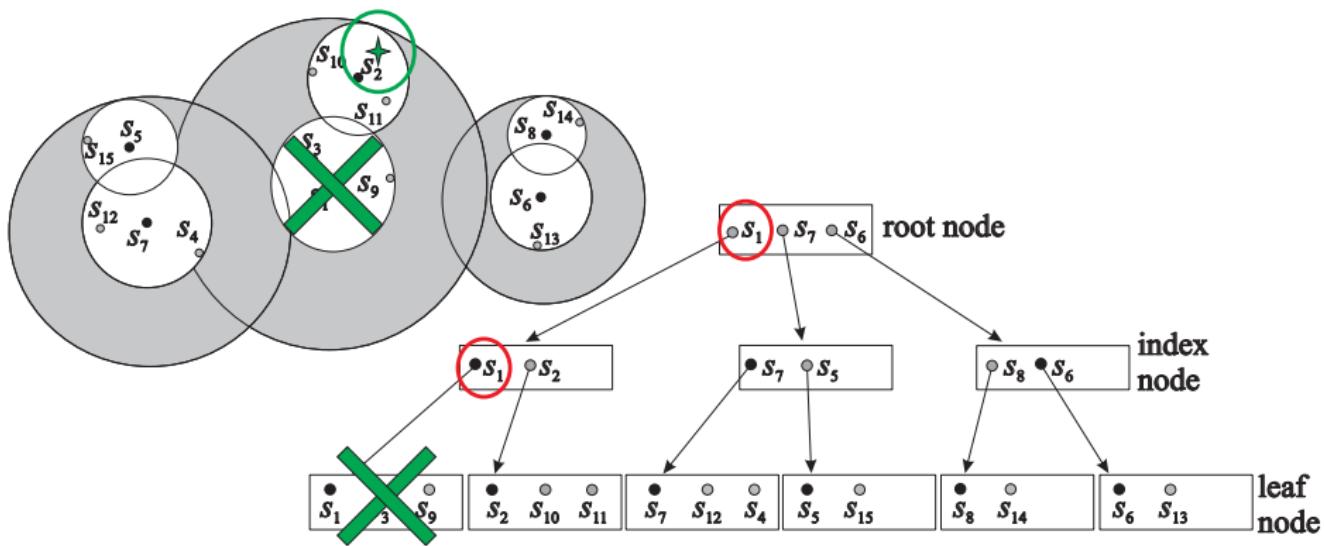


Conceitos Fundamentais

Como indexar conjuntos de dados complexos?

Estruturas de Indexação (Métodos de Acesso)

- São utilizadas para agilizar a realização de consultas por similaridade
- Para espaços métricos: Métodos de Acesso Métrico (MAM)

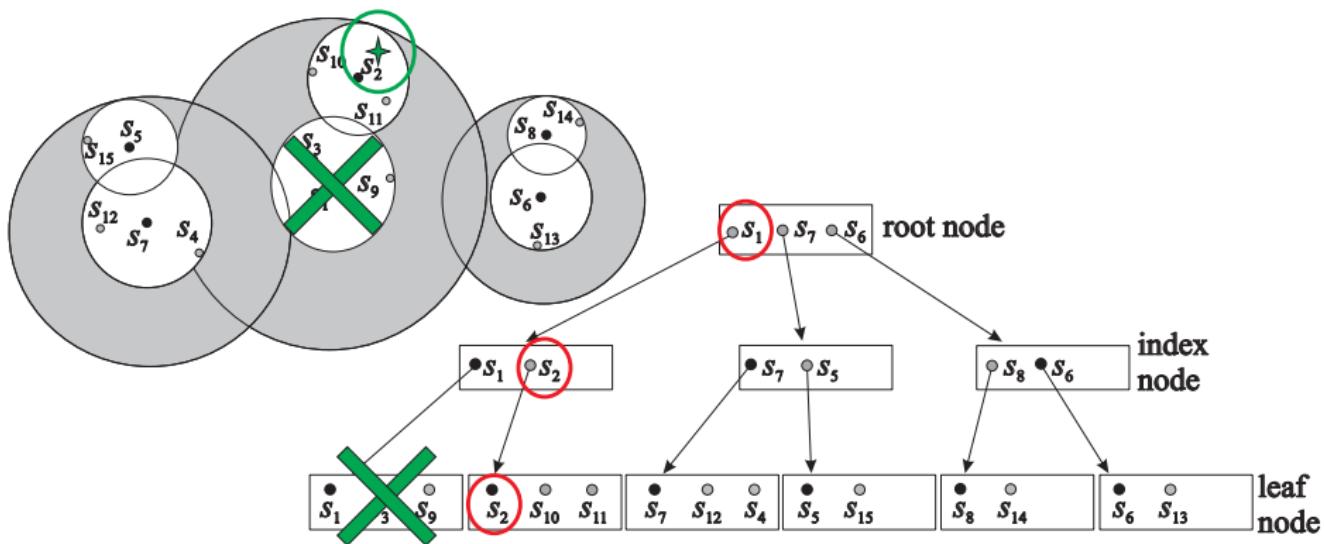


Conceitos Fundamentais

Como indexar conjuntos de dados complexos?

Estruturas de Indexação (Métodos de Acesso)

- São utilizadas para agilizar a realização de consultas por similaridade
- Para espaços métricos: Métodos de Acesso Métrico (MAM)

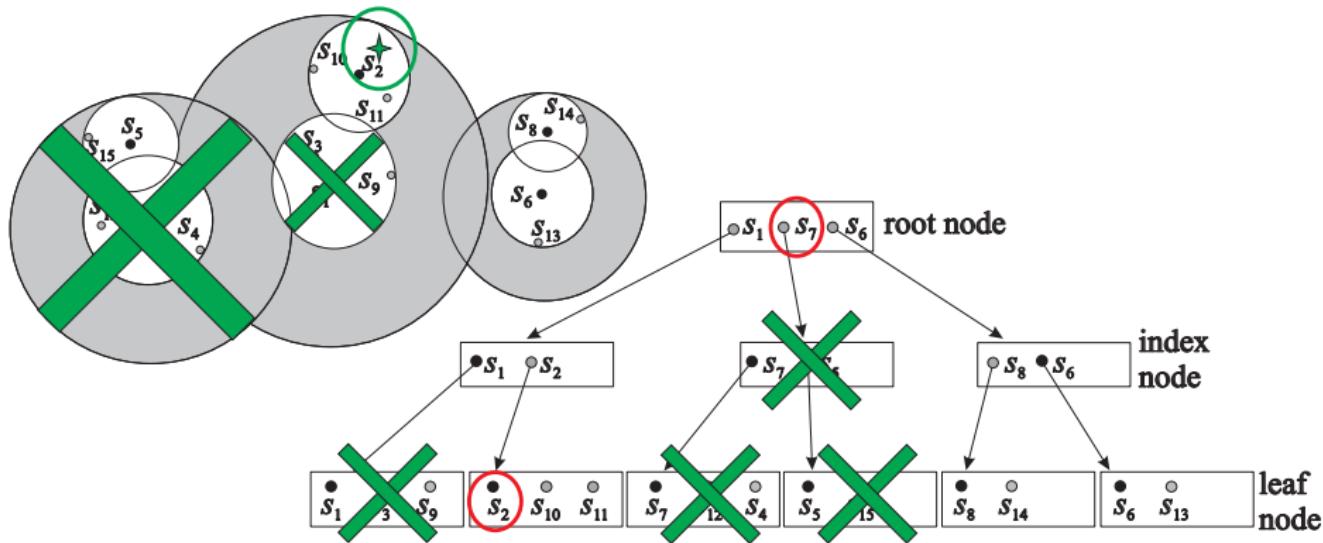


Conceitos Fundamentais

Como indexar conjuntos de dados complexos?

Estruturas de Indexação (Métodos de Acesso)

- São utilizadas para agilizar a realização de consultas por similaridade
- Para espaços métricos: Métodos de Acesso Métrico (MAM)

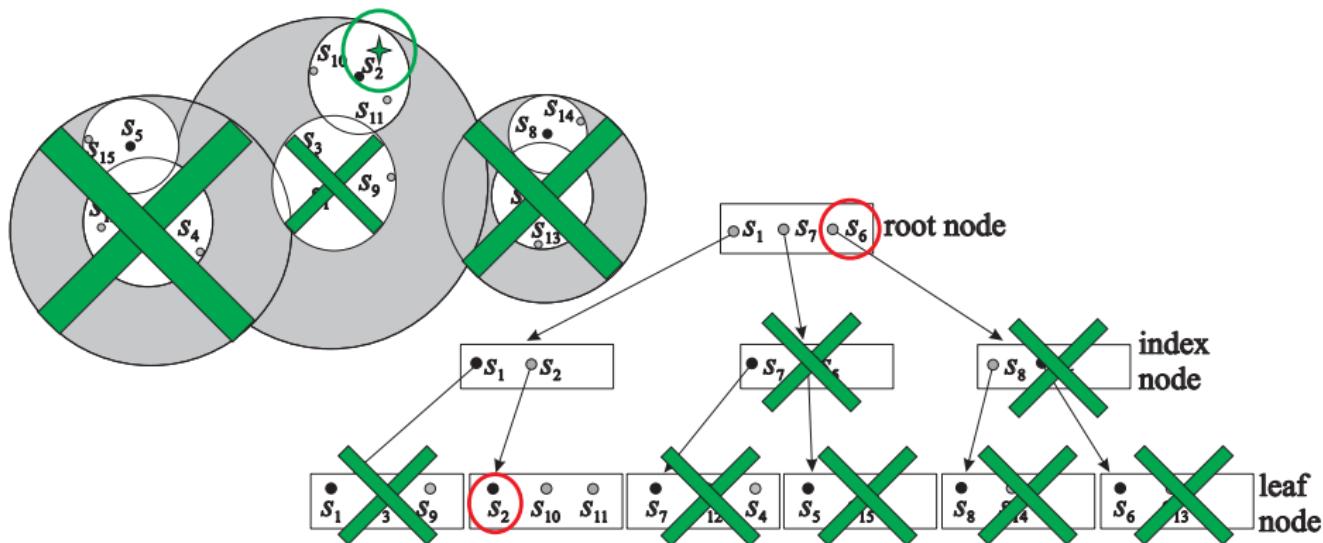


Conceitos Fundamentais

Como indexar conjuntos de dados complexos?

Estruturas de Indexação (Métodos de Acesso)

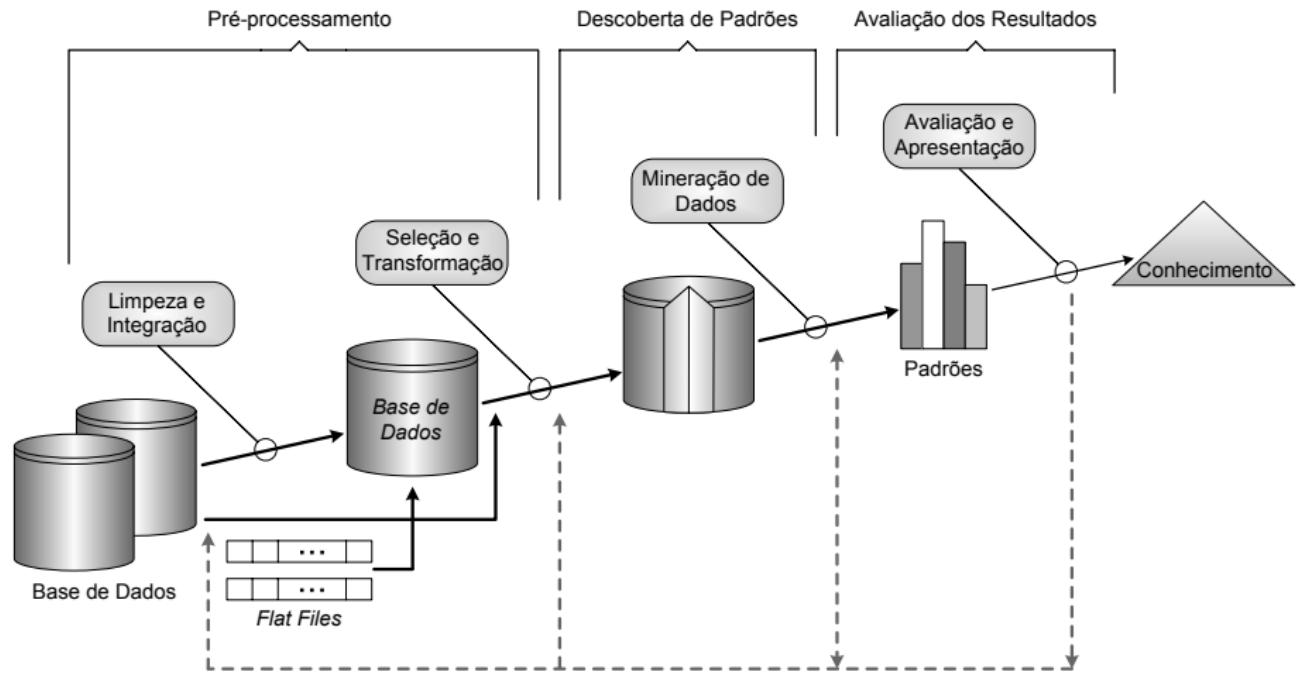
- São utilizadas para agilizar a realização de consultas por similaridade
- Para espaços métricos: Métodos de Acesso Métrico (MAM)



Conceitos Fundamentais

Descoberta de Conhecimento em Bases de Dados e Mineração de Dados

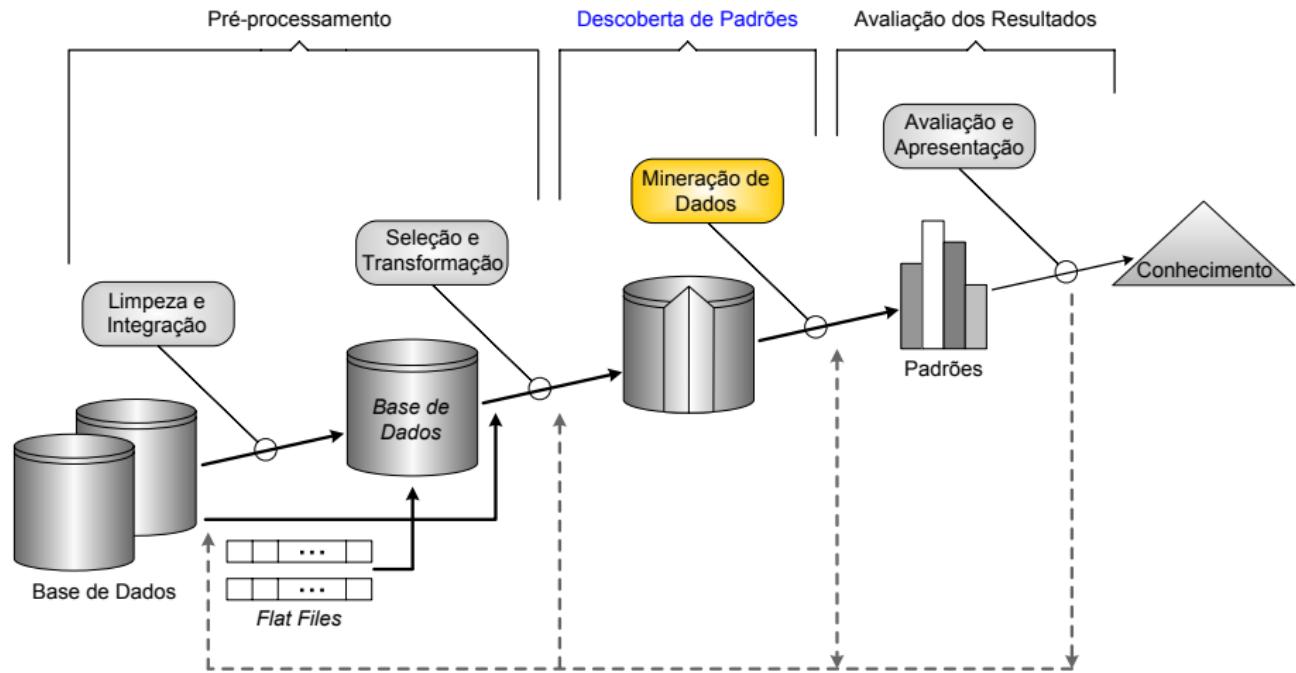
Processo interativo e iterativo que envolve três etapas:



Conceitos Fundamentais

Descoberta de Conhecimento em Bases de Dados e Mineração de Dados

Processo interativo e iterativo que envolve três etapas:



Conceitos Fundamentais

Descoberta de Conhecimento em Bases de Dados e Mineração de Dados

Demandas:

- Algoritmos de mineração de dados escaláveis

Conceitos Fundamentais

Descoberta de Conhecimento em Bases de Dados e Mineração de Dados

Demandas:

- Algoritmos de mineração de dados escaláveis

Estratégia:

- Disponibilizar operações básicas pelos SGBD
 - Exemplo
 - Técnica: Detecção de Agrupamentos
 - Operação básica: Cálculo de medidas de similaridade

Conceitos Fundamentais

Detecção de agrupamentos

Detecção de Agrupamentos

Definição: Processo de divisão de objetos em classes (ou grupos) de objetos similares de acordo com uma medida de similaridade

Conceitos Fundamentais

Detecção de agrupamentos

Detecção de Agrupamentos

Definição: Processo de divisão de objetos em classes (ou grupos) de objetos similares de acordo com uma medida de similaridade

Duas categorias de algoritmos

- Hierárquico
 - cria uma hierarquia de agrupamentos, formada por vários níveis de partições aninhadas de um conjunto de dados
- Particionamento
 - constrói um único nível de partição que divide os dados em k agrupamentos representados por
 - Centróides (k -média)
 - Medóides (k -medóide)

Conceitos Fundamentais

Detecção de agrupamentos

Detecção de Agrupamentos

Definição: Processo de divisão de objetos em classes (ou grupos) de objetos similares de acordo com uma medida de similaridade

Duas categorias de algoritmos

- Hierárquico
 - cria uma hierarquia de agrupamentos, formada por vários níveis de partições aninhadas de um conjunto de dados
- Particionamento
 - constrói um único nível de partição que divide os dados em k agrupamentos representados por
 - **Centróides (k -média)**
 - Medóides (k -medóide)

Conceitos Fundamentais

Detecção de agrupamentos

Detecção de Agrupamentos

Definição: Processo de divisão de objetos em classes (ou grupos) de objetos similares de acordo com uma medida de similaridade

Duas categorias de algoritmos

- Hierárquico
 - cria uma hierarquia de agrupamentos, formada por vários níveis de partições aninhadas de um conjunto de dados
- Particionamento
 - constrói um único nível de partição que divide os dados em k agrupamentos representados por
 - Centróides (*k-means*)
 - Medóides (*k-medóide*)

Conceitos Fundamentais

Detecção de agrupamentos

Razões para a escolha dos algoritmos baseados no k -medóide

Conceitos Fundamentais

Detecção de agrupamentos

Razões para a escolha dos algoritmos baseados no *k*-medóide

- Vantagens

- São menos sensíveis quanto à presença de *outliers*
- Não apresentam limitações quanto ao tipo de atributo
- Não dependem da ordem de entrada dos dados

Conceitos Fundamentais

Detecção de agrupamentos

Razões para a escolha dos algoritmos baseados no *k*-medóide

- Vantagens

- São menos sensíveis quanto à presença de *outliers*
- Não apresentam limitações quanto ao tipo de atributo
- Não dependem da ordem de entrada dos dados

- Desvantagens

- Apresentam um custo computacional muito elevado

Conceitos Fundamentais

Detecção de agrupamentos

Razões para a escolha dos algoritmos baseados no *k*-medóide

- Vantagens

- São menos sensíveis quanto à presença de *outliers*
- Não apresentam limitações quanto ao tipo de atributo
- Não dependem da ordem de entrada dos dados

- Desvantagens

- Apresentam um custo computacional muito elevado



Conceitos Fundamentais

Detecção de agrupamentos

Razões para a escolha dos algoritmos baseados no *k*-medóide

- Vantagens

- São menos sensíveis quanto à presença de *outliers*
- Não apresentam limitações quanto ao tipo de atributo
- Não dependem da ordem de entrada dos dados

- Desvantagens

- Apresentam um custo computacional muito elevado



- **Estratégia de otimização:** selecionar uma amostragem do conjunto de dados antes da realização do processo de agrupamento dos dados

Conceitos Fundamentais

Detecção de agrupamentos

Objetivo:

- encontrar um conjunto de agrupamentos não sobrepostos de modo que cada agrupamento possua um objeto representante (medóide)

Conceitos Fundamentais

Detecção de agrupamentos

Objetivo:

- encontrar um conjunto de agrupamentos não sobrepostos de modo que cada agrupamento possua um objeto representante (medóide)

Passos principais:

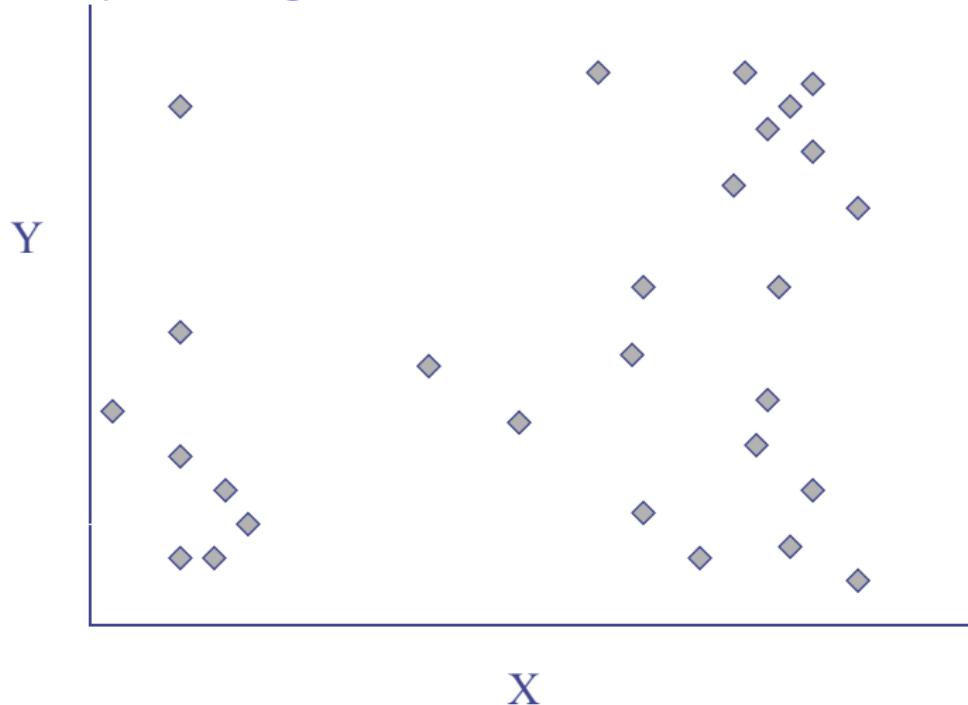
- Inicialização
 - seleciona-se um conjunto inicial de k medóides
- Avaliação
 - uma função de pontuação, baseada na soma da distância total entre os objetos não selecionados e seus medóides é minimizada

Quanto menor a soma das distâncias entre os medóides e todos os outros objetos de seus agrupamentos, melhor o resultado do agrupamento

Conceitos Fundamentais

Detecção de agrupamentos

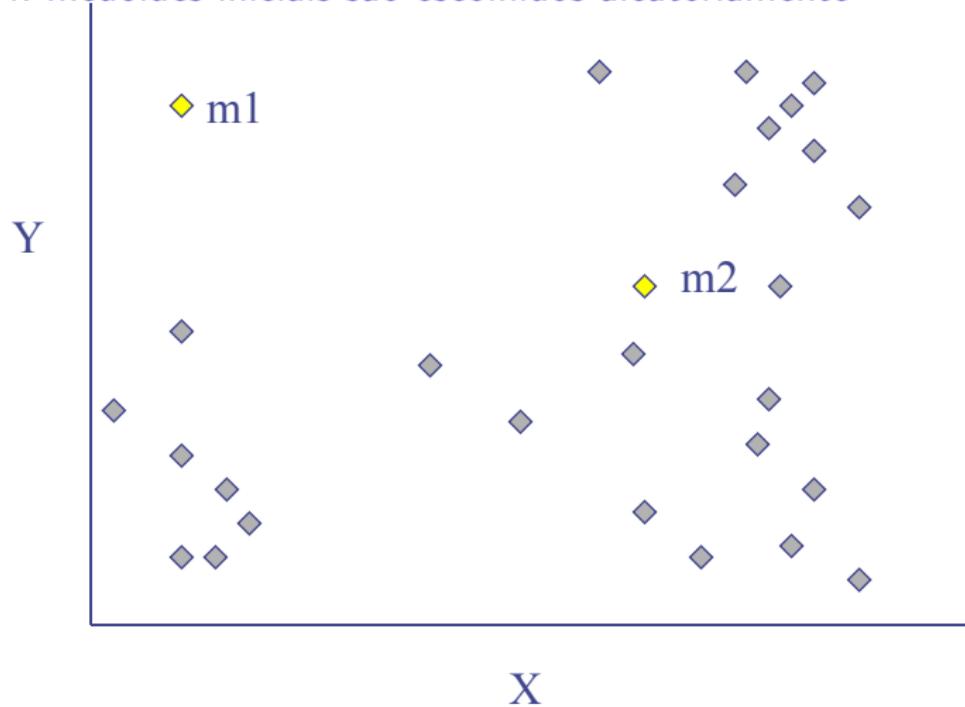
Exemplo abordagem k -medóide



Conceitos Fundamentais

Detecção de agrupamentos

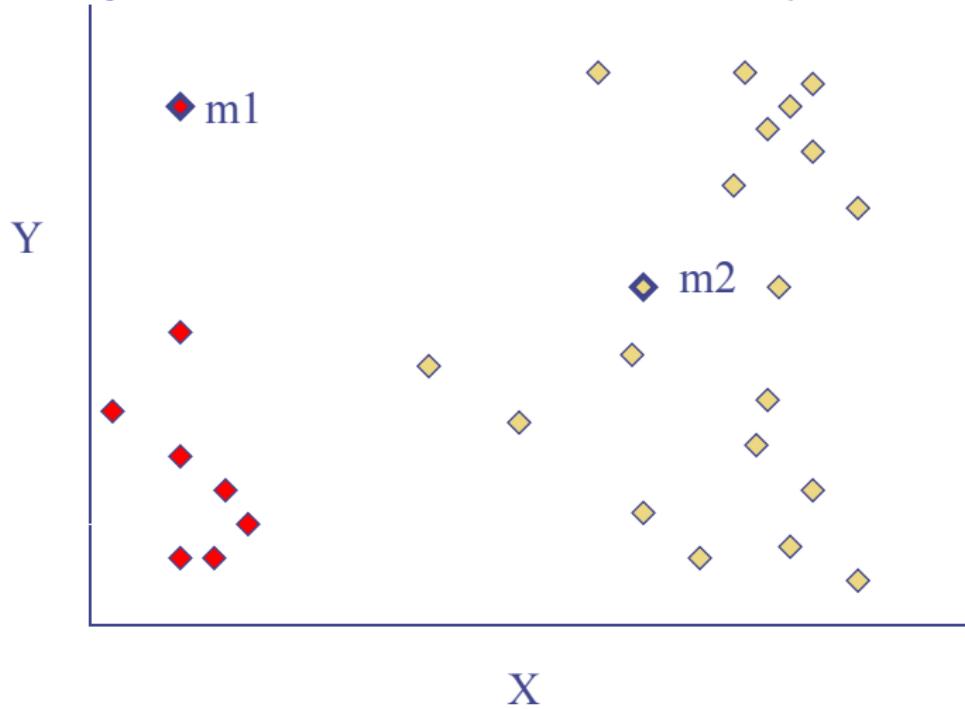
k medóides iniciais são escolhidos aleatoriamente



Conceitos Fundamentais

Detecção de agrupamentos

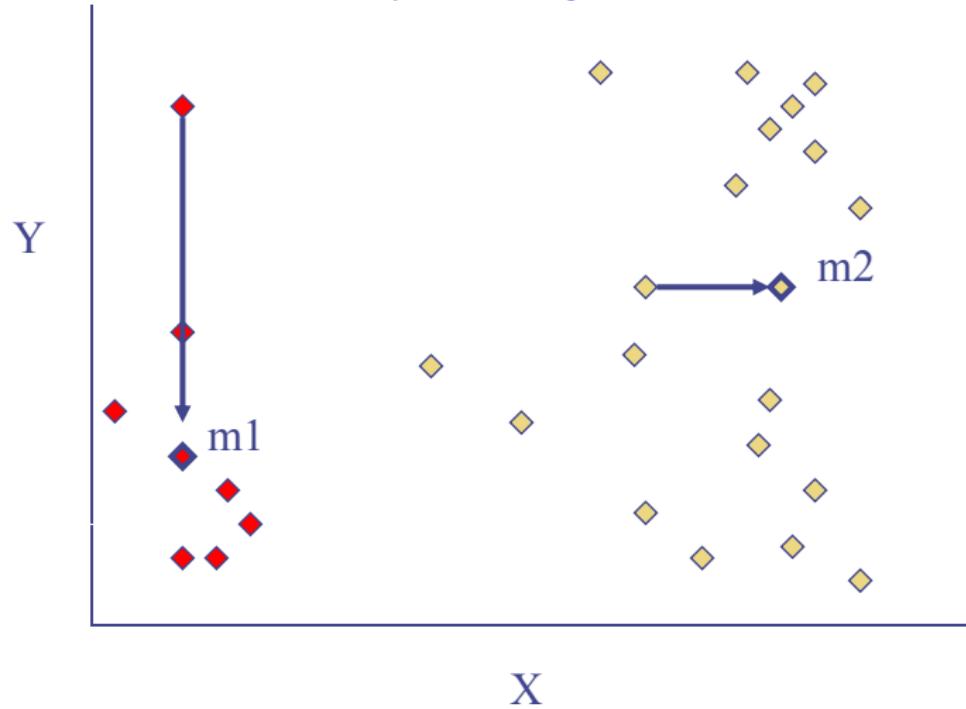
Os objetos são atribuidos aos medóides mais próximos



Conceitos Fundamentais

Detecção de agrupamentos

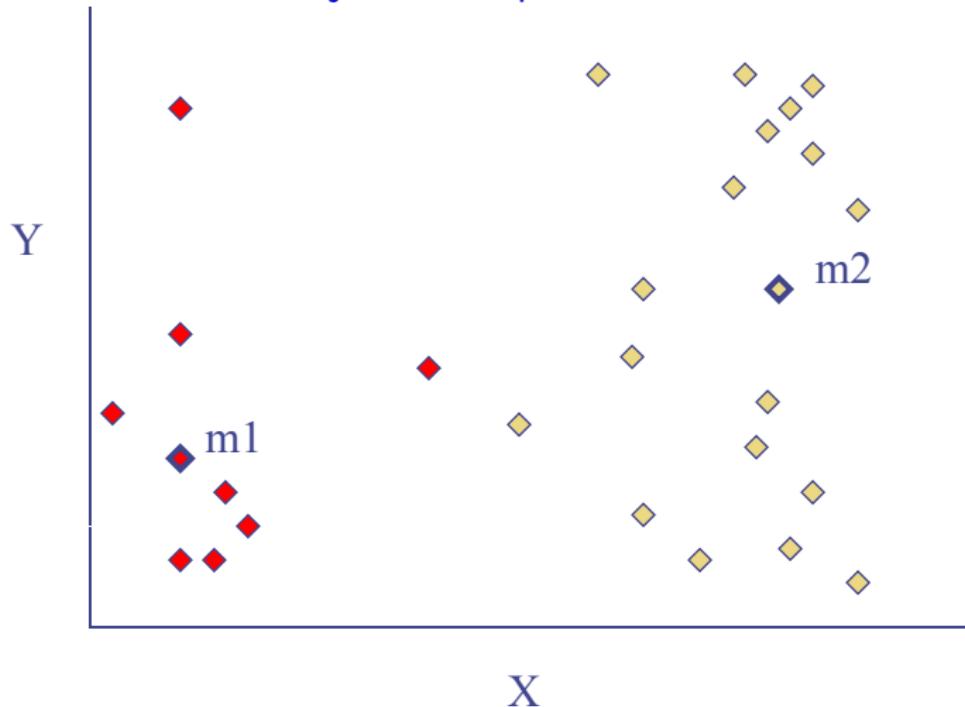
Muda-se os medóides para os objetos centrais de cada agrupamento



Conceitos Fundamentais

Detecção de agrupamentos

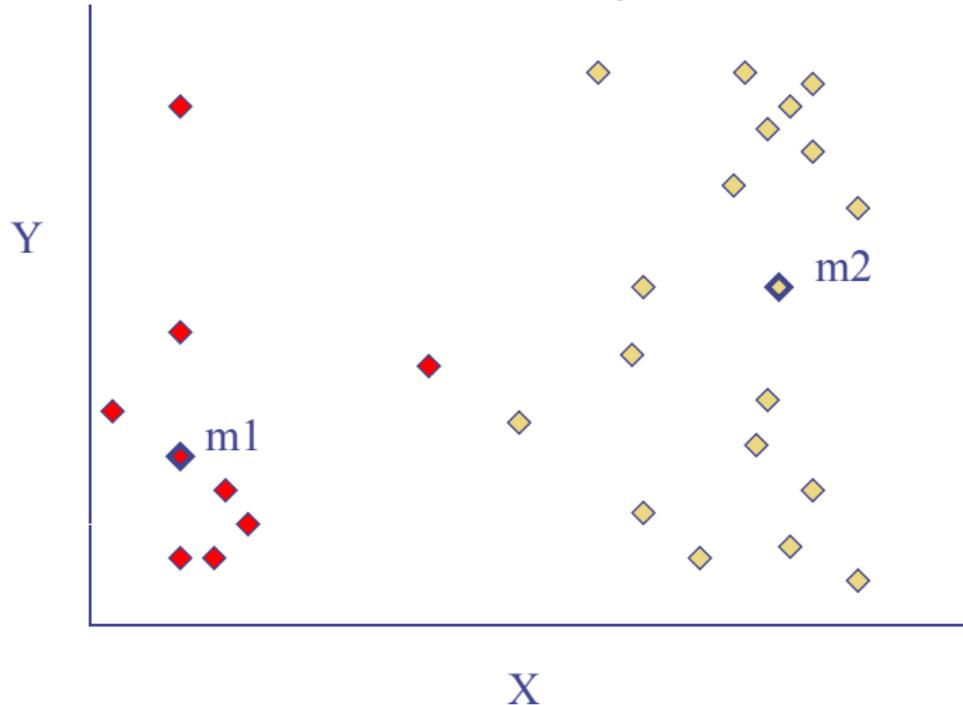
Atualizam-se os objetos mais próximos dos novos medóides



Conceitos Fundamentais

Detecção de agrupamentos

Novos medóides são calculados. Repete-se até estabilizar



Conceitos Fundamentais

Detecção de agrupamentos

Algoritmos mais conhecidos:

Conceitos Fundamentais

Detecção de agrupamentos

Algoritmos mais conhecidos:

- PAM (*Partitioning Around Medoids*)
 - Melhor qualidade
 - Maior custo computacional $\Rightarrow O(k(N - k)^2)$

Conceitos Fundamentais

Detecção de agrupamentos

Algoritmos mais conhecidos:

- PAM (*Partitioning Around Medoids*)
 - Melhor qualidade
 - Maior custo computacional $\Rightarrow O(k(N - k)^2)$
- CLARA (*Clustering LARge Applications*)
 - Baseado em amostragem $\Rightarrow O(ks^2 + k(N - k))$
 - Conjunto de dados aumenta \Rightarrow qualidade degrada

Conceitos Fundamentais

Detecção de agrupamentos

Algoritmos mais conhecidos:

- PAM (*Partitioning Around Medoids*)
 - Melhor qualidade
 - Maior custo computacional $\Rightarrow O(k(N - k)^2)$
- CLARA (*Clustering LARge Applications*)
 - Baseado em amostragem $\Rightarrow O(ks^2 + k(N - k))$
 - Conjunto de dados aumenta \Rightarrow qualidade degrada
- CLARANS (*Clustering Large Applications based upon RANdomized Search*)
 - Baseado em busca aleatória
 - Complexidade computacional $\Rightarrow O(N^2)$

Roteiro

- 1 Introdução
- 2 Conceitos Fundamentais
- 3 Suportando Consultas por Similaridade em SQL
- 4 Algoritmo PAM-SLIM
- 5 Protótipo SIREN
- 6 Referências

Suporando Consultas por Similaridade em SQL

Domínios de dados complexos

Os domínios de objetos complexos podem ser separados em:

Suporando Consultas por Similaridade em SQL

Domínios de dados complexos

Os domínios de objetos complexos podem ser separados em:

- **PARTICULATE**

- **Tipo de objeto:** composto por uma coleção de atributos tradicionais
- **Forma de comparação:** os valores armazenados nos atributos tradicionais são utilizados para calcular a distância entre cada par de objetos complexos

Suporando Consultas por Similaridade em SQL

Domínios de dados complexos

Os domínios de objetos complexos podem ser separados em:

- **PARTICULATE**

- **Tipo de objeto:** composto por uma coleção de atributos tradicionais
- **Forma de comparação:** os valores armazenados nos atributos tradicionais são utilizados para calcular a distância entre cada par de objetos complexos

- **MONOLITHIC**

- **Tipo de objeto:** armazenado como um único objeto binário BLOB (atributo indivisível)
- **Forma de comparação:** é necessário aplicar algoritmos de extração de características sobre eles

Suportando Consultas por Similaridade em SQL

Tipos de dados complexos

Para realizar consultas por similaridade em domínios complexos

- É necessário definir cada domínio onde a similaridade será medida como um novo tipo de dados
- Novos tipos de dados

PARTICULATE { **PARTICULATE**

MONOLITHIC { **STILLIMAGE**
AUDIO

Suportando Consultas por Similaridade em SQL

Passos para a realização de consultas por similaridade em SQL

- **DDL**

- ① Definir medidas de similaridade (Métricas)
- ② Especificar tipos de dados complexos na definição de tabelas
- ③ Associar atributos complexos com medidas de similaridade
- ④ Definir índices (opcional)

- **DML**

- ⑤ Popular/atualizar a base de dados
- ⑥ Especificar consultas

Suporando Consultas por Similaridade em SQL

Definir medidas de similaridade – Métricas

- Novos comandos
 - CREATE METRIC
 - ALTER METRIC
 - DROP METRIC

Suporando Consultas por Similaridade em SQL

Definir medidas de similaridade – Métricas

- Novos comandos
 - CREATE METRIC
 - ALTER METRIC
 - DROP METRIC

- Exemplos:

```
CREATE METRIC Euclidian2D USING LP2 FOR PARTICULATE  
    (Latitude FLOAT, Longitude FLOAT);
```

```
CREATE METRIC Histograma USING LP1 FOR STILLIMAGE  
    (HistogramaEXT (HistogramaC AS Histo));
```

Suporando Consultas por Similaridade em SQL

Especificar tipos de dados complexos na definição de tabelas – Exemplos

- PARTICULATE

```
CREATE TABLE CidadeBR (
    Nome CHAR(30) PRIMARY KEY,
    Lat FLOAT,
    Longit FLOAT,
    Coordenada PARTICULATE,
    ...);
```

Suporando Consultas por Similaridade em SQL

Especificar tipos de dados complexos na definição de tabelas – Exemplos

- PARTICULATE

```
CREATE TABLE CidadeBR (
    Nome CHAR(30) PRIMARY KEY,
    Lat FLOAT,
    Longit FLOAT,
    Coordenada PARTICULATE,
    ...);
```

- MONOLITHIC

```
CREATE TABLE Paisagem (
    Id INTEGER PRIMARY KEY,
    Local CHAR(20),
    Fotografo CHAR(30),
    Foto STILLIMAGE,
    ... );
```

Suporando Consultas por Similaridade em SQL

Especificar tipos de dados complexos na definição de tabelas – Exemplos

- PARTICULATE

```
CREATE TABLE CidadeBR (
    Nome CHAR(30) PRIMARY KEY,
    Lat FLOAT,
    Longit FLOAT,
    Coordenada PARTICULAR,
    ...);
```

Nome	Lat	Longit	Coordenada
São Carlos-SP	-22.02	47.89	1203

- MONOLITHIC

```
CREATE TABLE Paisagem (
    Id INTEGER PRIMARY KEY,
    Local CHAR(20),
    Fotografo CHAR(30),
    Foto STILLIMAGE,
    ... );
```

Id	Local	Fotografo	Foto
196	Cristo Redentor - RJ	Humberto	

Suporando Consultas por Similaridade em SQL

Associar atributos complexos com medidas de similaridade

A definição de como comparar pares de dados de tipos complexos é expressa como uma restrição no comando CREATE TABLE:

- Restrição de coluna
- Restrição de tabela

Suporando Consultas por Similaridade em SQL

Associar atributos complexos com medidas de similaridade – Exemplo

PARTICULATE:

```
CREATE METRIC Euclidiana2D USING LP2
    FOR PARTICULATE (Latitude FLOAT,
                      Longitude FLOAT);
```

```
CREATE TABLE CidadeBR (
    Nome CHAR(30) PRIMARY KEY,
    Lat FLOAT,
    Longit FLOAT,
    Coordenada PARTICULAR,
    METRIC (Coordenada) REFERENCES (Lat AS Latitude,
                                      Longit AS Longitude)
        USING (Euclidiana2D),
        ... );
```

Suporando Consultas por Similaridade em SQL

Associar atributos complexos com medidas de similaridade – Exemplo

MONOLITHIC:

```
CREATE METRIC Histograma USING LP1
    FOR STILLIMAGE (HistogramaEXT (HistogramaC AS Histo));
```

```
CREATE METRIC Textura USING LP1
    FOR STILLIMAGE (TexturaEXT (TexturaC AS Text));
```

```
CREATE TABLE Paisagem (
    Id INTEGER PRIMARY KEY,
    Local CHAR(20),
    Fotografo CHAR(30),
    Foto STILLIMAGE METRIC USING (Histograma DEFAULT, Textura),
    ... );
```

Suporando Consultas por Similaridade em SQL

Definir índices (opcional)

- Consultas por similaridade podem ser realizadas mais rapidamente se forem criados índices sobre os atributos complexos
 - Métodos de acesso métrico (MAM)

Suporando Consultas por Similaridade em SQL

Definir índices (opcional)

- Consultas por similaridade podem ser realizadas mais rapidamente se forem criados índices sobre os atributos complexos
 - Métodos de acesso métrico (MAM)
- Exemplos:
 - PARTICULATE:
CREATE INDEX Geografia ON CidadeBR (**Coordenada**)
 REFERENCES (**Lat** AS **Latitude**, **Longit** AS **Longitude**)
 USING **Euclidiana2D**;
 - MONOLITHIC:

```
CREATE INDEX FotoPaisagem ON Paisagem (Foto)
    USING Histograma;
```

Suporando Consultas por Similaridade em SQL

Popular/atualizar a base de dados

- Para a sintaxe do comando `INSERT` não foi necessário nenhuma alteração
- A sintaxe dos comandos `UPDATE` e `DELETE` necessitam de novas construções para expressar predicados por similaridade

Suporando Consultas por Similaridade em SQL

Especificando consultas

Comando SELECT

- Novas construções para predicados por similaridade

Suporando Consultas por Similaridade em SQL

Especificando consultas

Comando SELECT

- Novas construções para predicados por similaridade
 - Seleções e junções por similaridade \Rightarrow cláusula WHERE

Suportando Consultas por Similaridade em SQL

Especificar consultas

Comando SELECT

- Novas construções para predicados por similaridade
 - Seleções e junções por similaridade \Rightarrow cláusula WHERE
 - Junções por similaridade \Rightarrow cláusula FROM

Suporando Consultas por Similaridade em SQL

Especificando consultas

Comando SELECT

- Novas construções para predicados por similaridade
 - Seleções e junções por similaridade \Rightarrow cláusula WHERE
 - Junções por similaridade \Rightarrow cláusula FROM
 - Novas construções para suportar a análise de agrupamento por similaridade

Suporando Consultas por Similaridade em SQL

Especificando consultas

Comando SELECT

- Sintaxe básica para expressar **seleções por similaridade**:

```
<atributo> NEAR <valor>
[STOP AFTER <k>]
[RANGE <ξ>]
```

Suporando Consultas por Similaridade em SQL

Especificando consultas

Exemplos de **seleções por similaridade**:

- PARTICULATE:

```
SELECT * FROM CidadeBR
WHERE Coordenada NEAR (-22.02 AS Latitude,
                        47.89 AS Longitude) RANGE 2;
```

```
SELECT Nome FROM CidadeBR
WHERE Coordenada NEAR (SELECT Lat AS Latitude,
                        Longit AS Longitude
                     FROM CidadeBR
                     WHERE Nome = 'São Carlos-SP')
STOP AFTER 5;
```

Suporando Consultas por Similaridade em SQL

Especificar consultas

Exemplos de **seleções por similaridade**:

- MONOLITHIC:

```
SELECT * FROM Paisagem  
WHERE Foto NEAR 'c:\img09.jpg' STOP AFTER 5;
```

```
SELECT * FROM Paisagem  
WHERE Foto NEAR (SELECT Foto FROM Paisagem  
WHERE Id = 123) STOP AFTER 5;
```

Suporando Consultas por Similaridade em SQL

Especificando consultas

Sintaxe básica para expressar **junções por similaridade** na cláusula WHERE:

```
<tabela1>‘.’<atributo1> NEAR [ANY] <tabela2>‘.’<atributo2>  
[STOP AFTER <k>] [RANGE <ξ>]
```

Suporando Consultas por Similaridade em SQL

Especificando consultas

Sintaxe básica para expressar **junções por similaridade** na cláusula WHERE:

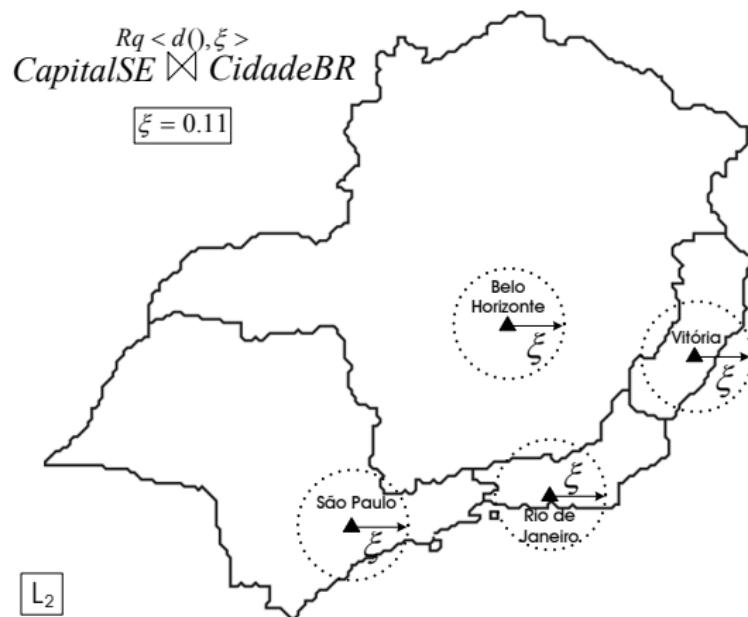
```
<tabela1>‘.’<atributo1> NEAR [ANY] <tabela2>‘.’<atributo2>  
[STOP AFTER <k>] [RANGE <ξ>]
```

- Tipos de construções:
 - $T_1.attr_1 \text{ NEAR } T_2.attr_2 \text{ RANGE } \xi \Rightarrow$ junção por abrangência
 - $T_1.attr_1 \text{ NEAR } T_2.attr_2 \text{ STOP AFTER } k \Rightarrow$ junção pelos k -vizinhos mais próximos
 - $T_1.attr_1 \text{ NEAR ANY } T_2.attr_2 \text{ STOP AFTER } k \Rightarrow$ junção dos k -pares de vizinhos mais próximos

Suportando Consultas por Similaridade em SQL

Especificar consultas

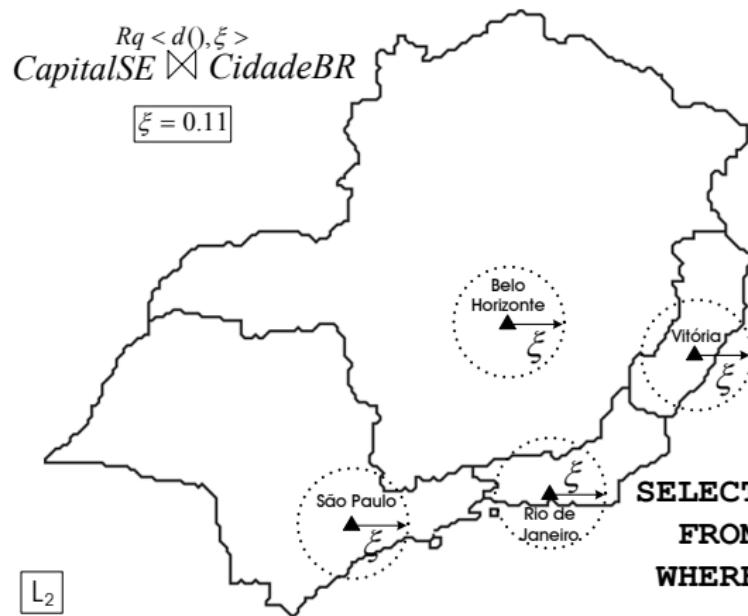
Exemplo de **junção por similaridade**:



Suportando Consultas por Similaridade em SQL

Especificar consultas

Exemplo de **junção por similaridade**:



```
SELECT *
  FROM CapitalSE, CidadeBR
 WHERE CapitalSE.Coordenada NEAR
       CidadeBR.Coordenada
      RANGE 0.11;
```

Suporando Consultas por Similaridade em SQL

Especificando consultas

- Sintaxe para expressar **junções por similaridade** na cláusula FROM:

```
<tabela1> {RANGE|NEAREST|CLOSEST} JOIN <tabela2>
    ON <nome_atr_complexo1> {NEAR|FAR}
        <nome_atr_complexo2>
    [STOP AFTER <k>] [RANGE <ξ>]
```

Suporando Consultas por Similaridade em SQL

Especificando consultas

- Sintaxe para expressar **junções por similaridade** na cláusula FROM:

```
<tabela1> {RANGE|NEAREST|CLOSEST} JOIN <tabela2>
    ON <nome_atr_complexo1> {NEAR|FAR}
        <nome_atr_complexo2>
    [STOP AFTER <k>] [RANGE <ξ>]
```

Suporando Consultas por Similaridade em SQL

Especificando consultas

Sintaxe para expressar operações de **agrupamento por similaridade**:

- Como resultado do processo de detecção de agrupamentos
 - relação de agrupamentos encontrados \Rightarrow Cluster
 - relação que associa cada objeto do conjunto de dados ao objeto central de seu agrupamento \Rightarrow Clustering

Suporando Consultas por Similaridade em SQL

Especificar consultas

Sintaxe para expressar operações de **agrupamento por similaridade**:

- Como resultado do processo de detecção de agrupamentos
 - relação de agrupamentos encontrados \Rightarrow Cluster
 - relação que associa cada objeto do conjunto de dados ao objeto central de seu agrupamento \Rightarrow Clustering
- Exemplos:
 - Para mostrar os agrupamentos de cada instância do atributo **Foto** da tabela **Paisagem** resultantes do processo **Pam_FP**

```
SELECT * FROM Clustering(Pam_FP);
```
 - Para mostrar todos os agrupamentos de um atributo **Foto** resultantes do processo **Pam_FP**

```
SELECT * FROM Cluster(Pam_FP);
```

Suporando Consultas por Similaridade em SQL

Especificando consultas

- Para parametrizar as operações de agrupamento por similaridade é preciso definir:
 - A métrica que deve ser utilizada
 - O número de agrupamentos k
 - Qual algoritmo de detecção de agrupamentos utilizar

Suporando Consultas por Similaridade em SQL

Especificando consultas

- Para parametrizar as operações de agrupamento por similaridade é preciso definir:
 - A métrica que deve ser utilizada
 - O número de agrupamentos k
 - Qual algoritmo de detecção de agrupamentos utilizar
- Sintaxe:

```
SET CLUSTERING <nome_processo>
    METHOD <nome_método>,
    METRIC <nome_métrica>,
    K <valor_inteiro>
ON <nome_tabela>.<nome_atributo>;
```

Suportando Consultas por Similaridade em SQL

A sintaxe apresentada permite

Suportando Consultas por Similaridade em SQL

A sintaxe apresentada permite

- Combinar predicados por similaridade
 - entre si
 - com predicados tradicionais

Suportando Consultas por Similaridade em SQL

A sintaxe apresentada permite

- Combinar predicados por similaridade
 - entre si
 - com predicados tradicionais
- Consultar por similaridade qualquer conjunto de objetos complexos para o qual seja possível definir uma medida de similaridade
 - domínio MONOLITHIC
 - domínio PARTICULATE

Suportando Consultas por Similaridade em SQL

A sintaxe apresentada permite

- Combinar predicados por similaridade
 - entre si
 - com predicados tradicionais
- Consultar por similaridade qualquer conjunto de objetos complexos para o qual seja possível definir uma medida de similaridade
 - domínio MONOLITHIC
 - domínio PARTICULATE
- Otimizar consultas por similaridade

Suportando Consultas por Similaridade em SQL

A sintaxe apresentada permite

- Combinar predicados por similaridade
 - entre si
 - com predicados tradicionais
- Consultar por similaridade qualquer conjunto de objetos complexos para o qual seja possível definir uma medida de similaridade
 - domínio MONOLITHIC
 - domínio PARTICULATE
- Otimizar consultas por similaridade
- Especificar consultas sobre o resultado de processos de agrupamento por similaridade

Suportando Consultas por Similaridade em SQL

A sintaxe apresentada permite

- Combinar predicados por similaridade
 - entre si
 - com predicados tradicionais
- Consultar por similaridade qualquer conjunto de objetos complexos para o qual seja possível definir uma medida de similaridade
 - domínio MONOLITHIC
 - domínio PARTICULATE
- Otimizar consultas por similaridade
- Especificar consultas sobre o resultado de processos de agrupamento por similaridade



baixo impacto na sintaxe da linguagem

Roteiro

- 1 Introdução
- 2 Conceitos Fundamentais
- 3 Suportando Consultas por Similaridade em SQL
- 4 Algoritmo PAM-SLIM
- 5 Protótipo SIREN
- 6 Referências

Algoritmo PAM-SLIM

Algoritmo desenvolvido:

- **PAM-SLIM**

Algoritmo PAM-SLIM

Algoritmo desenvolvido:

- **PAM-SLIM**

- Melhora a eficiência dos algoritmos de agrupamento baseados no k-medoid
 - Utilizando MAM para selecionar um sub-conjunto de objetos relevantes

Algoritmo PAM-SLIM

Algoritmo desenvolvido:

- **PAM-SLIM**

- Melhora a eficiência dos algoritmos de agrupamento baseados no k-medoid
 - Utilizando MAM para selecionar um sub-conjunto de objetos relevantes
- Obtém agrupamentos com qualidade comparável

Algoritmo PAM-SLIM

Estratégia:

- Realiza uma amostragem nos dados considerando certas características dos MAM (Slim-tree)

Algoritmo PAM-SLIM

Estratégia:

- Realiza uma amostragem nos dados considerando certas características dos MAM (Slim-tree)
- Divide o espaço de dados hierarquicamente em regiões atribuindo um representante para cada região

Algoritmo PAM-SLIM

Estratégia:

- Realiza uma amostragem nos dados considerando certas características dos MAM (Slim-tree)
 - Divide o espaço de dados hierarquicamente em regiões atribuindo um representante para cada região
 - Os centros dos nós podem ser vistos como centros naturais das regiões que eles representam

Algoritmo PAM-SLIM

Estratégia:

- Realiza uma amostragem nos dados considerando certas características dos MAM (Slim-tree)
 - Divide o espaço de dados hierarquicamente em regiões atribuindo um representante para cada região
 - Os centros dos nós podem ser vistos como centros naturais das regiões que eles representam
 - O centro de uma sub-árvore pode ser considerado como centro de massa dos objetos armazenados nela

Algoritmo PAM-SLIM

Estratégia:

- Realiza uma amostragem nos dados considerando certas características dos MAM (Slim-tree)
 - Divide o espaço de dados hierarquicamente em regiões atribuindo um representante para cada região
 - Os centros dos nós podem ser vistos como centros naturais das regiões que eles representam
 - O centro de uma sub-árvore pode ser considerado como centro de massa dos objetos armazenados nela



Algoritmo PAM-SLIM

Estratégia:

- Realiza uma amostragem nos dados considerando certas características dos MAM (Slim-tree)
 - Divide o espaço de dados hierarquicamente em regiões atribuindo um representante para cada região
 - Os centros dos nós podem ser vistos como centros naturais das regiões que eles representam
 - O centro de uma sub-árvore pode ser considerado como centro de massa dos objetos armazenados nela



- Apenas os objetos de um determinado nível podem ser considerados para a realização do agrupamento

Algoritmo PAM-SLIM

Estratégia:

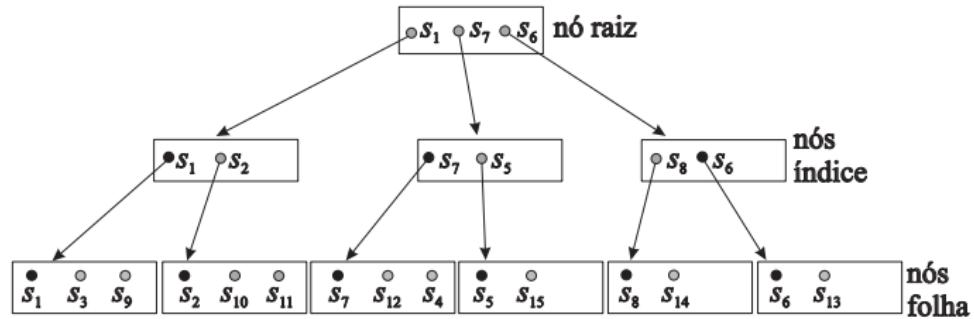
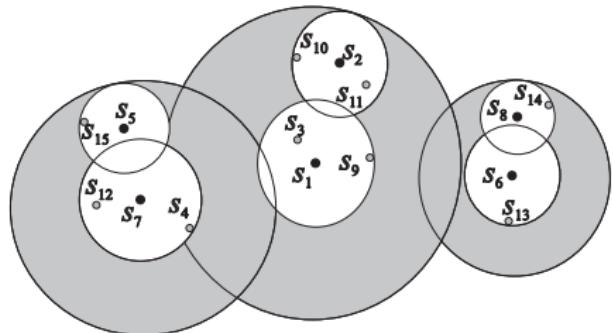
- Uma Slim-tree representa a divisão do espaço de dados com uma granularidade que cresce da raiz para as folhas

Algoritmo PAM-SLIM

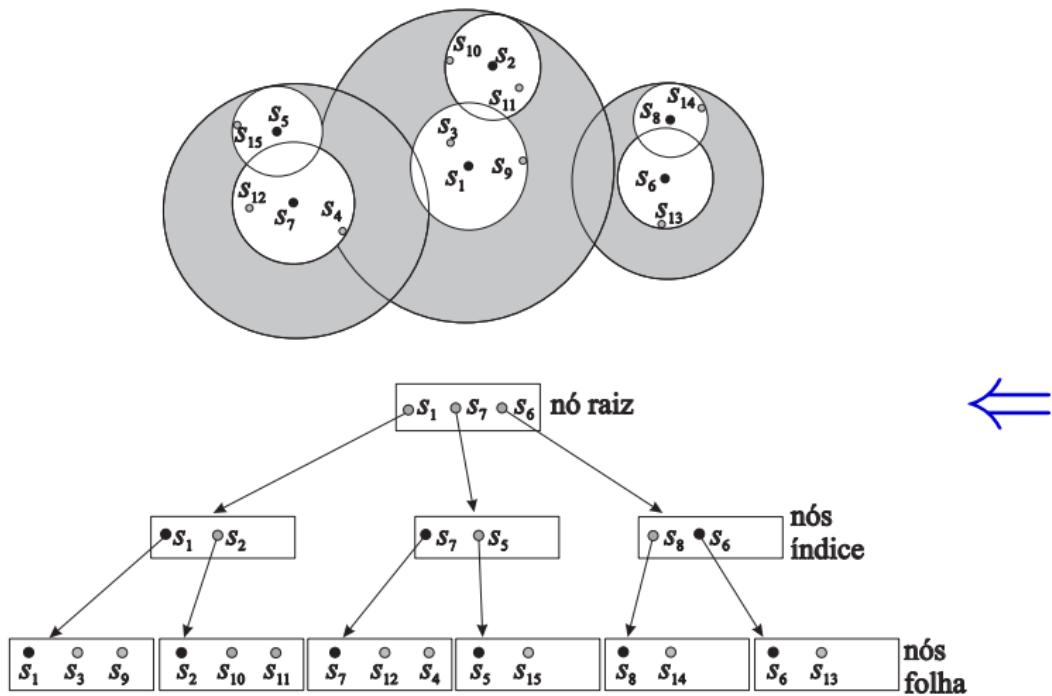
Estratégia:

- Uma Slim-tree representa a divisão do espaço de dados com uma granularidade que cresce da raiz para as folhas
- Questão:
 - Qual o nível da árvore possui informações sobre a distribuição dos dados suficientes para gerar um agrupamento mais rápido e com qualidade razoável?

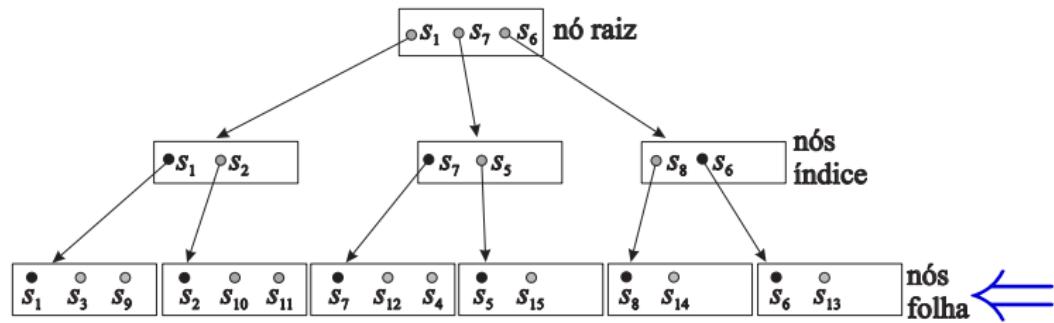
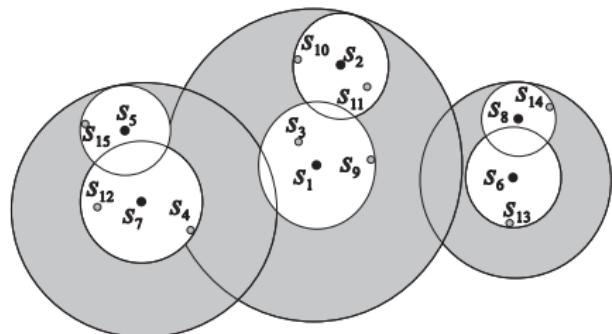
Algoritmo PAM-SLIM



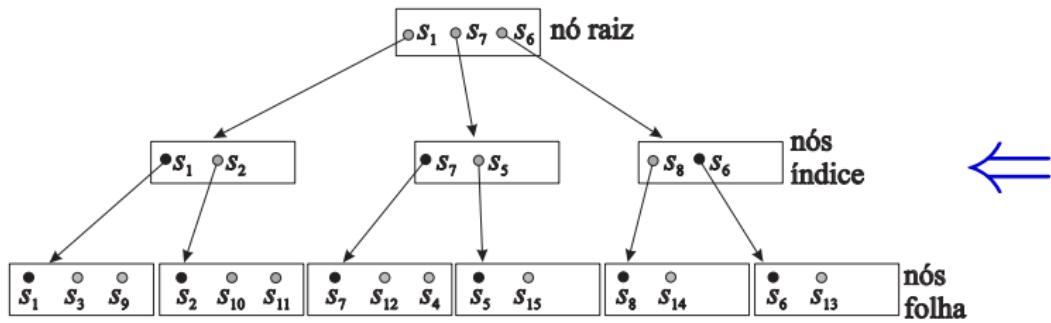
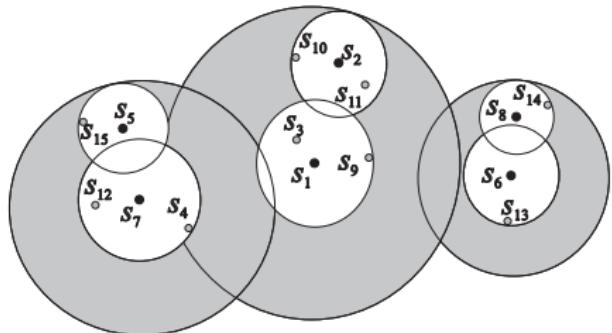
Algoritmo PAM-SLIM



Algoritmo PAM-SLIM



Algoritmo PAM-SLIM



Algoritmo PAM-SLIM

Algoritmo:

Algoritmo PAM-SLIM

Algoritmo:

- Pré-processamento
 - Escolha dos parâmetros para a construção da árvore
 - Tamanho de página
 - Política de escolha de sub-árvore (*ChooseSubtree*)
 - Construção da árvore

Algoritmo PAM-SLIM

Algoritmo:

- Pré-processamento
 - Escolha dos parâmetros para a construção da árvore
 - Tamanho de página
 - Política de escolha de sub-árvore (*ChooseSubtree*)
 - Construção da árvore
- Inicialização
 - Seleção do nível da árvore
 - Atribuição dos objetos do nível selecionado ao conjunto de objetos que serão utilizados no processo de agrupamento

Algoritmo PAM-SLIM

Algoritmo:

- Pré-processamento
 - Escolha dos parâmetros para a construção da árvore
 - Tamanho de página
 - Política de escolha de sub-árvore (*ChooseSubtree*)
 - Construção da árvore
- Inicialização
 - Seleção do nível da árvore
 - Atribuição dos objetos do nível selecionado ao conjunto de objetos que serão utilizados no processo de agrupamento
- Agrupamento
 - Aplicação do PAM sobre a amostra obtida no passo anterior
 - Atribuição de cada objeto do conjunto de dados todo aos medóides selecionados pelo PAM

Algoritmos considerados

- PAM, CLARA, CLARANS, PAM-SLIM-MD e PAM-SLIM-MO

Medidas utilizadas

- Qualidade
 - Distância média do agrupamento resultante
- Eficiência computacional
 - Número de cálculos de distância

Conjuntos de dados considerados

- Oito sintéticos e um real

Algoritmo PAM-SLIM

Resultados alcançados

- Mostra dos resultados obtidos

Tempo de execução do PAM, CLARANS, CLARA e PAM-SLIM
(horas:minutos:segundos)

Conjunto de dados	PAM	CLARANS	CLARA	PAM-SLIM-MD	PAM-SLIM-MO
Sint10_5k	00:42:48	00:01:31	00:00:01	00:00:04	00:00:02
Sint10_10k	07:49:42	00:07:18	00:00:06	00:00:10	00:00:05
Sint10_15k	21:27:41	00:23:03	00:00:20	00:00:33	00:00:15
Sint10_20k	43:44:47	00:44:10	00:00:47	00:01:14	00:00:39

Algoritmo PAM-SLIM

Resultados alcançados

- Mostra dos resultados obtidos

Tempo de execução do PAM, CLARANS, CLARA e PAM-SLIM
(horas:minutos:segundos)

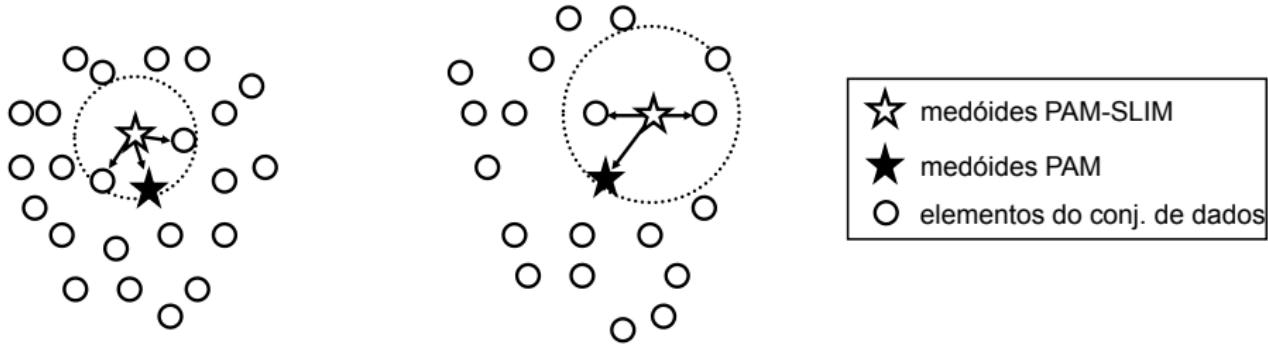
Conjunto de dados	PAM	CLARANS	CLARA	PAM-SLIM-MD	PAM-SLIM-MO
Sint10_5k	00:42:48	00:01:31	00:00:01	00:00:04	00:00:02
Sint10_10k	07:49:42	00:07:18	00:00:06	00:00:10	00:00:05
Sint10_15k	21:27:41	00:23:03	00:00:20	00:00:33	00:00:15
Sint10_20k	43:44:47	00:44:10	00:00:47	00:01:14	00:00:39

- Nota: a diminuição da qualidade do agrupamento variou apenas entre 7,4% e 11,8%

Algoritmo PAM-SLIM

Estratégia de otimização

- Leva em consideração a vizinhança dos medóides retornados no passo de agrupamento para refinar o processamento do mesmo
- Propriedade interessante ⇒ seu custo é função da quantidade de vizinhos que se solicita explorar
 - Nos experimentos realizados: agrupamentos de qualidade no máximo 7,6% inferior, mas que são obtidos até 1.500 vezes mais rapidamente



Algoritmo PAM-SLIM

Considerações Finais

A estratégia adotada pelo algoritmo

Algoritmo PAM-SLIM

Considerações Finais

A estratégia adotada pelo algoritmo

- Pode ser eficientemente aplicada para agrupar tanto conjuntos de dados multi-dimensionais quanto adimensionais (**armazenados em disco**)

Algoritmo PAM-SLIM

Considerações Finais

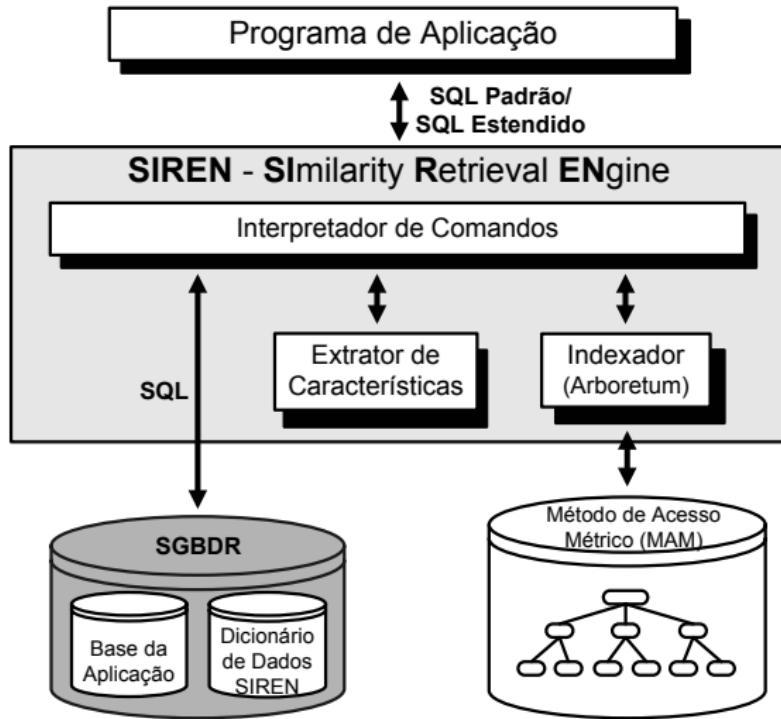
A estratégia adotada pelo algoritmo

- Pode ser eficientemente aplicada para agrupar tanto conjuntos de dados multi-dimensionais quanto adimensionais (**armazenados em disco**)
- Apresenta uma relação adequada de custo-benefício entre tempo de execução e qualidade do agrupamento obtido

Roteiro

- 1 Introdução
- 2 Conceitos Fundamentais
- 3 Suportando Consultas por Similaridade em SQL
- 4 Algoritmo PAM-SLIM
- 5 Protótipo SIREN
- 6 Referências

Arquitetura do SIREN



Além de adicionar novas construções na linguagem SQL foi necessário

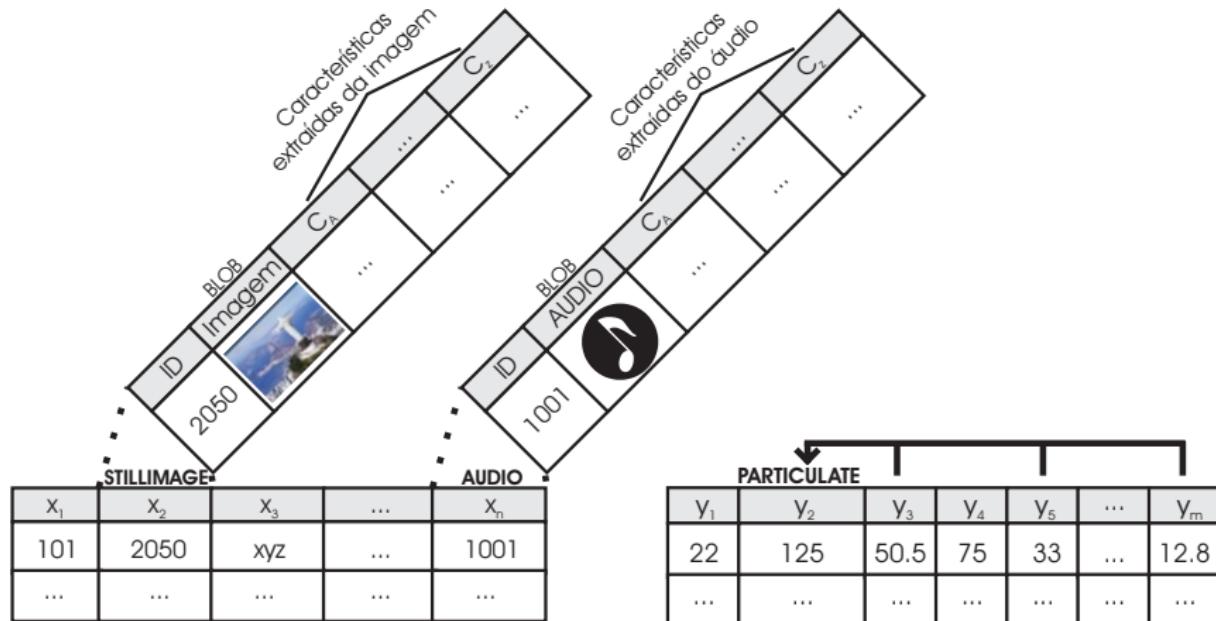
Além de adicionar novas construções na linguagem SQL foi necessário

- Estender o dicionário de dados do SGBD para armazenar informações como:
 - quais extratores de características são utilizados nas métricas
 - quais atributos tradicionais compõem cada atributo **PARTICULATE**
 - quais métricas estão disponíveis para cada domínio complexo
 - quais são os atributos de domínios complexos armazenados em cada relação
 - quais métricas estão associadas a cada atributo complexo

Além de adicionar novas construções na linguagem SQL foi necessário

- Estender o dicionário de dados do SGBD para armazenar informações como:
 - quais extratores de características são utilizados nas métricas
 - quais atributos tradicionais compõem cada atributo **PARTICULATE**
 - quais métricas estão disponíveis para cada domínio complexo
 - quais são os atributos de domínios complexos armazenados em cada relação
 - quais métricas estão associadas a cada atributo complexo
- Considerar os requisitos especiais de armazenamento dos dados dos tipos **STILLIMAGE** e **AUDIO**
 - é preciso armazenar também as características associadas a eles

Esquema de armazenamento dos novos tipos de dados complexos



Protótipo SIREN

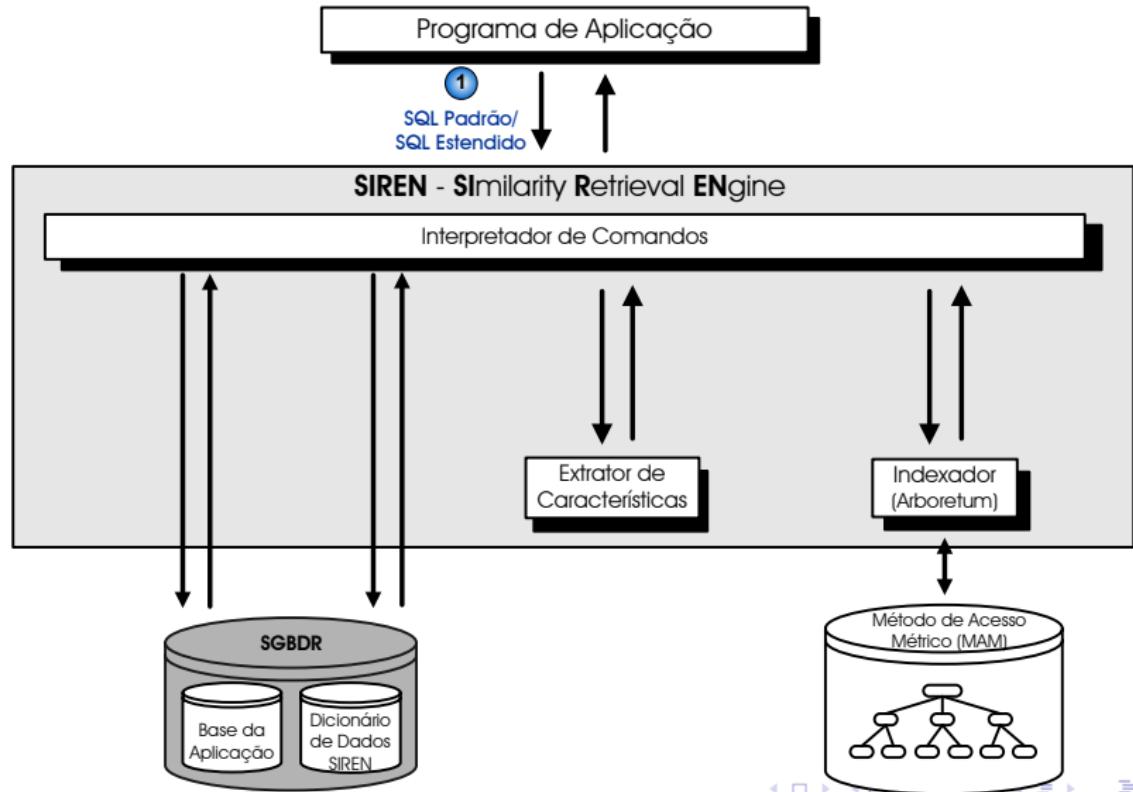
Exemplo do processamento de uma consulta

- Comando SELECT enviado pela aplicação:

```
SELECT  Fotografo, Foto  
        FROM Paisagem  
 WHERE  Foto NEAR 'D:\FotosPaisagem\img09.jpg'  
           BY Textura STOP AFTER 5;
```

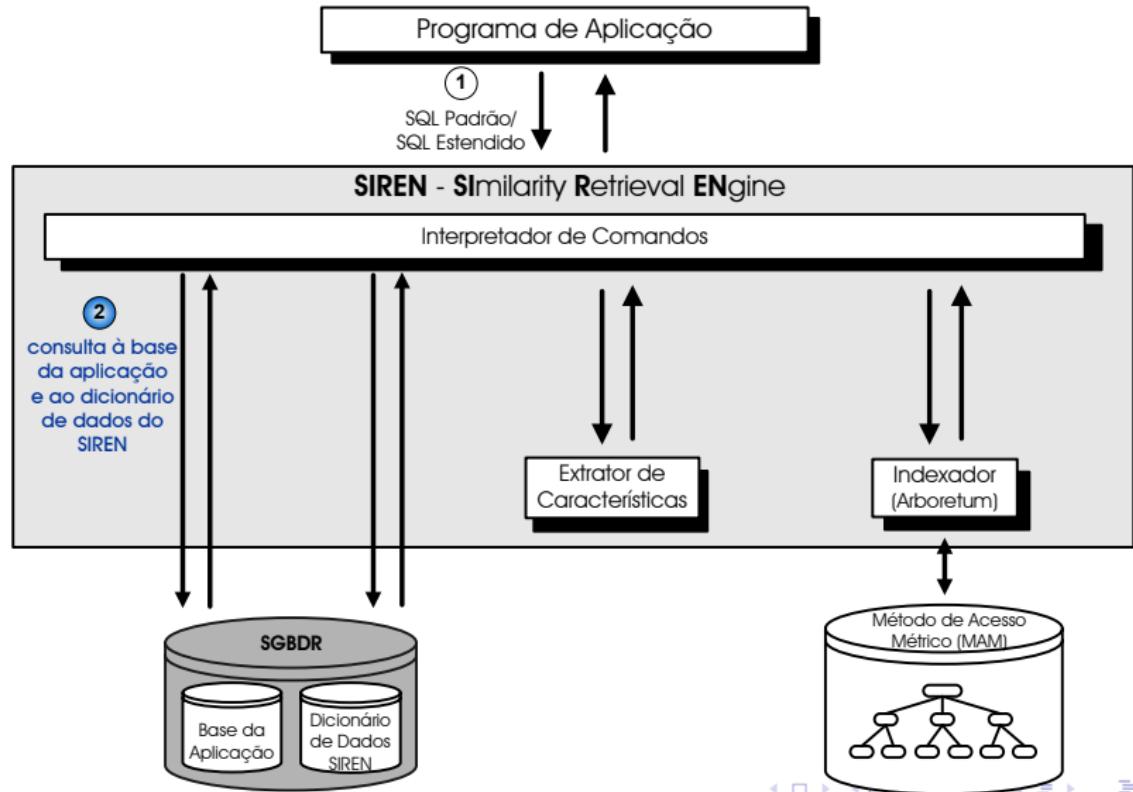
Protótipo SIREN

Exemplo do processamento de uma consulta



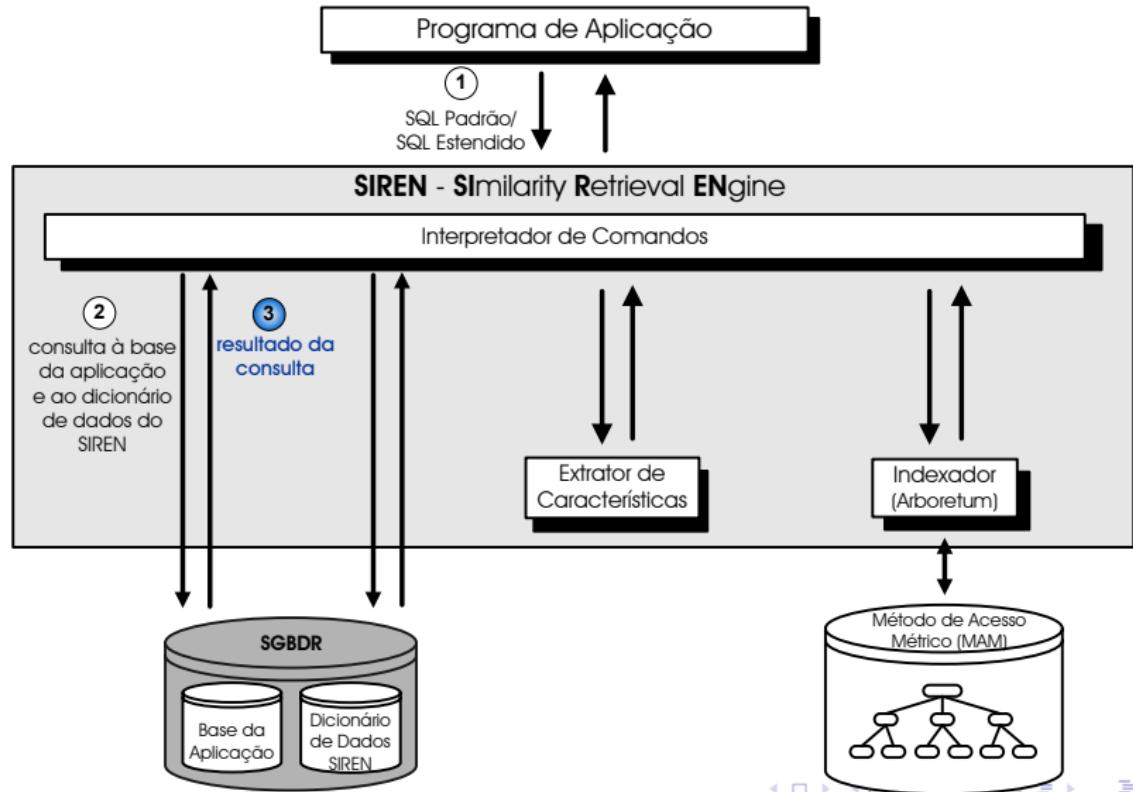
Protótipo SIREN

Exemplo do processamento de uma consulta



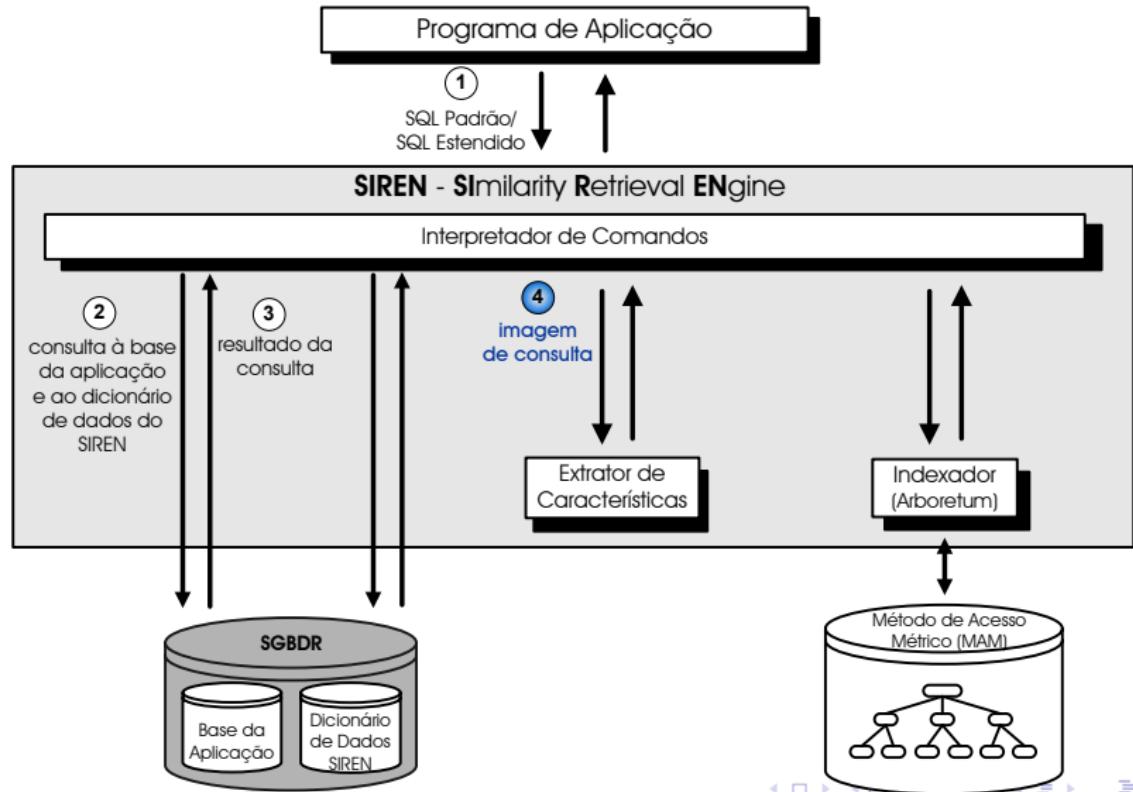
Protótipo SIREN

Exemplo do processamento de uma consulta



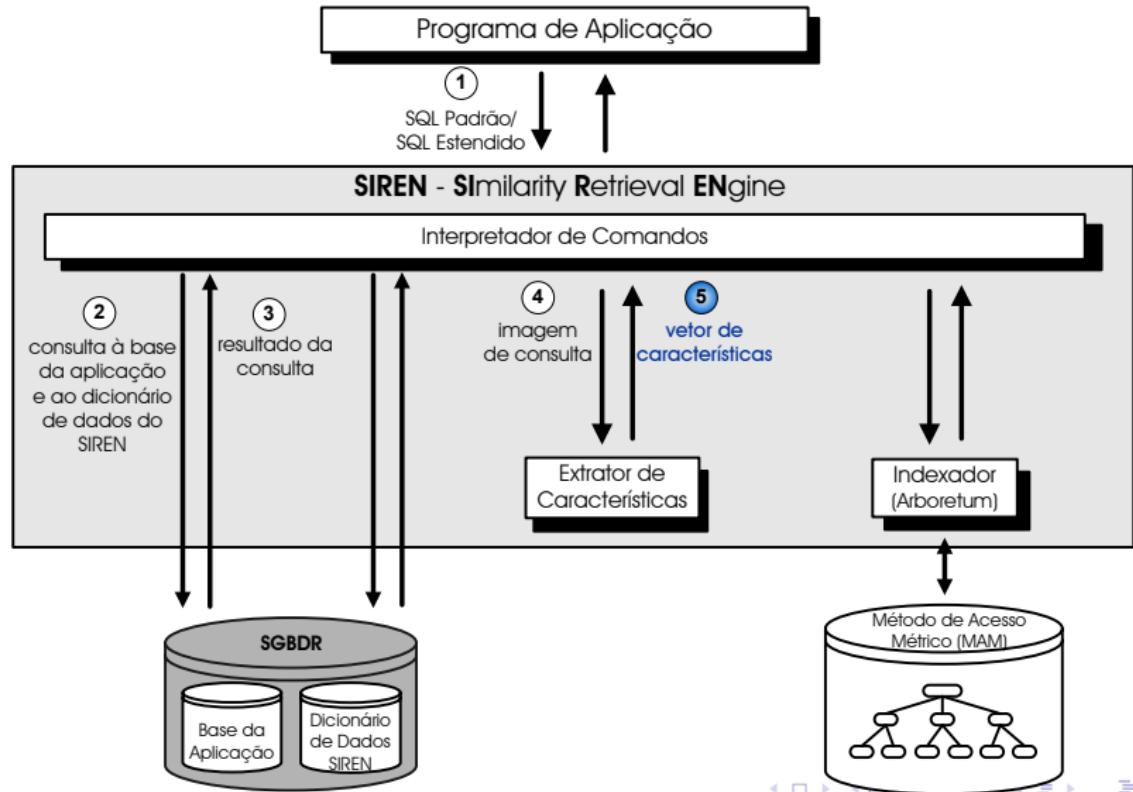
Protótipo SIREN

Exemplo do processamento de uma consulta



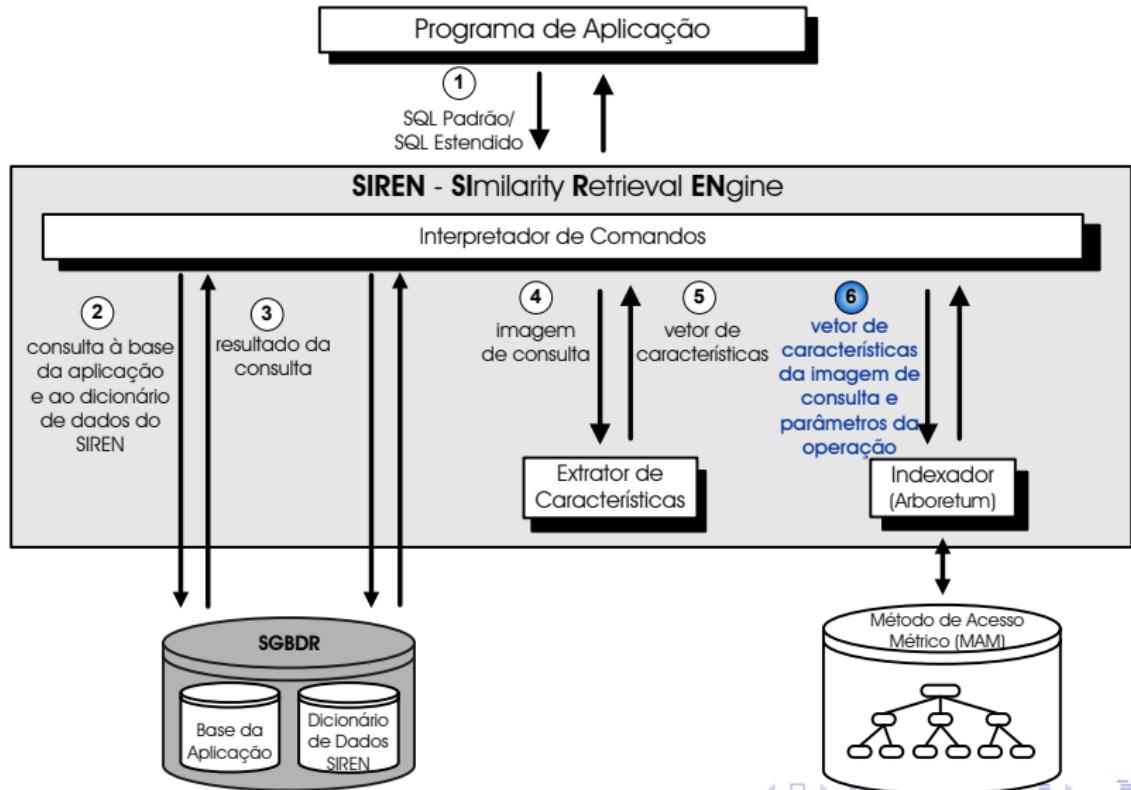
Protótipo SIREN

Exemplo do processamento de uma consulta



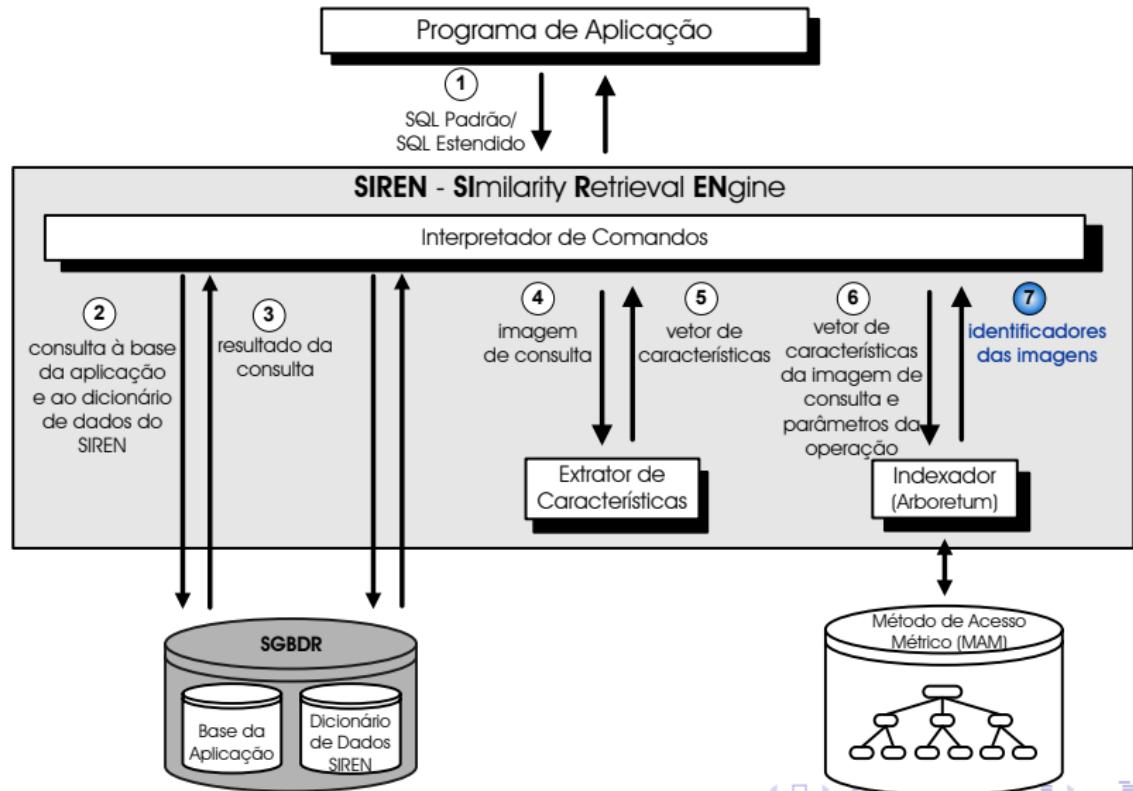
Protótipo SIREN

Exemplo do processamento de uma consulta



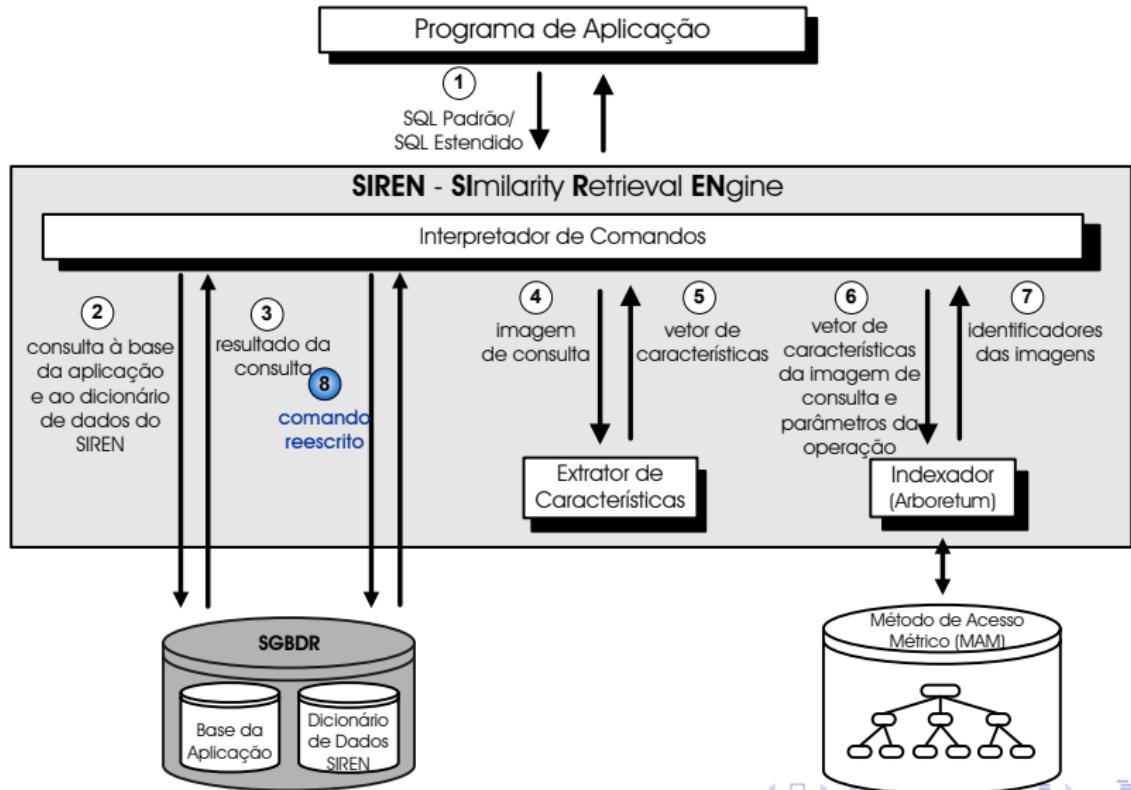
Protótipo SIREN

Exemplo do processamento de uma consulta



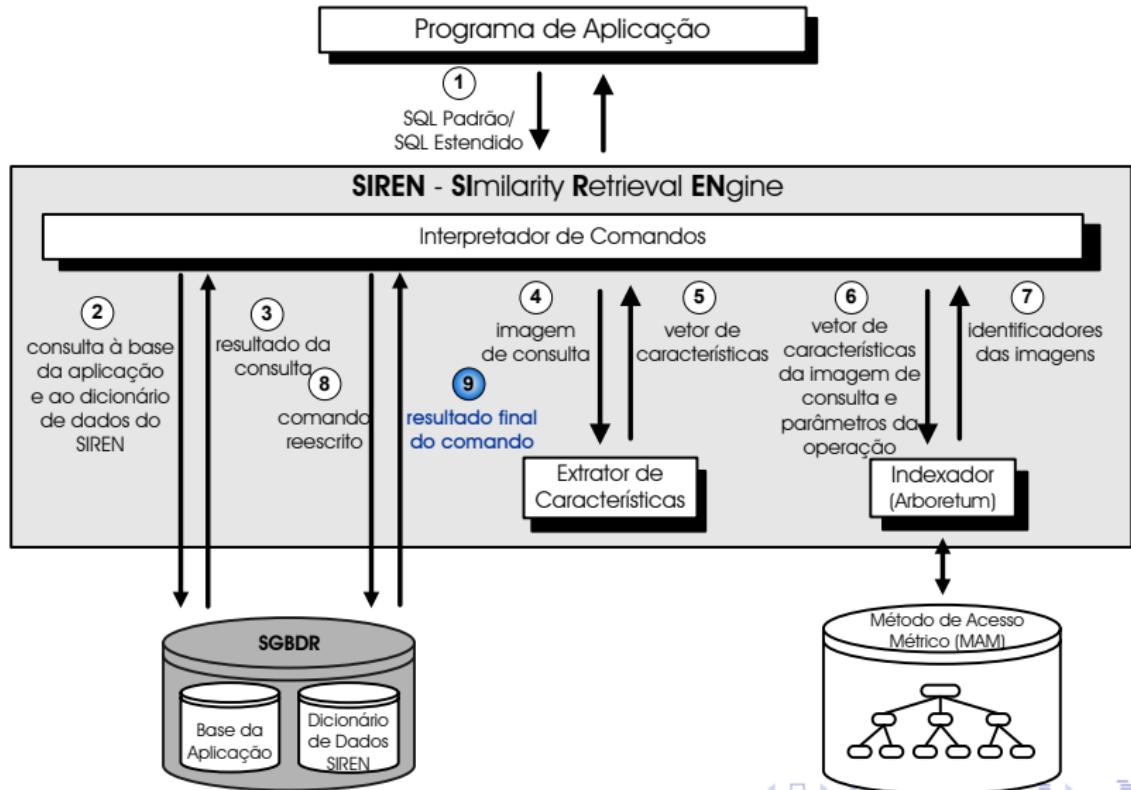
Protótipo SIREN

Exemplo do processamento de uma consulta



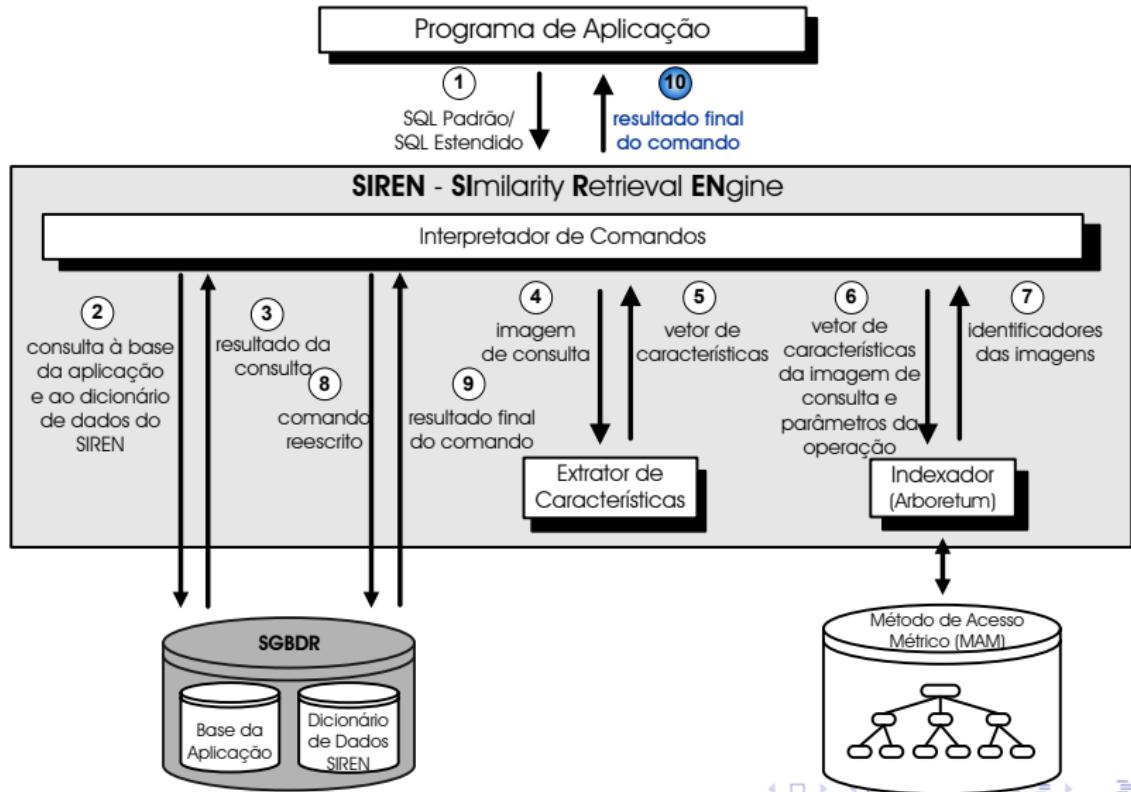
Protótipo SIREN

Exemplo do processamento de uma consulta



Protótipo SIREN

Exemplo do processamento de uma consulta



Protótipo SIREN

Exemplo do processamento de uma consulta

- Comando SELECT enviado pela aplicação:

```
SELECT Fotografo, Foto  
      FROM Paisagem  
 WHERE Foto NEAR 'D:\FotosPaisagem\img09.jpg'  
           BY Textura STOP AFTER 5;
```

- Comando SELECT reescrito pelo SIREN:

```
SELECT Fotografo, IPV$Paisagem_Foto.Image AS Foto  
      FROM Paisagem JOIN IPV$Paisagem_Foto  
        ON Paisagem.Foto = IPV$Paisagem_Foto.Image_id  
 WHERE Foto IN ( 7896, 7912, 9669, 9668, 9675 );
```

Roteiro

- 1 Introdução
- 2 Conceitos Fundamentais
- 3 Suportando Consultas por Similaridade em SQL
- 4 Algoritmo PAM-SLIM
- 5 Protótipo SIREN
- 6 Referências

Referências

- BARIONI, M. C. N.; RAZENTE, H.; TRAINA, A. J. M.; TRAINA JR, C. "SIREN: A Similarity Retrieval Engine For Complex Data" (2006). In: Demonstration Session of the 32nd International Conference on Very Large Data Bases (VLDB), v. 1., p. 1155-1158, Seul.
- BARIONI, M. C. N.; RAZENTE, H.; TRAINA, A. J. M.; TRAINA JR, C. "Accelerating k-medoid-based algorithms through metric access methods". Journal of Systems and Software, v. 81/3, p. 343-355, março de 2008. Disponível online em 05 de julho de 2007. DOI: <http://dx.doi.org/10.1016/j.jss.2007.06.019>
- BARIONI, M. C. N.; RAZENTE, H. L.; TRAINA, A. J. M.; TRAINA JR, C. "Seamlessly integrating similarity queries in SQL". Software, Practice & Experience, v. 39, p. 355-384, 2009. Disponível online em 27 de agosto de 2008. DOI: <http://dx.doi.org/10.1016/j.jss.2007.06.019>

Referências

- RAZENTE, H.; BARIONI, M. C. N.; TRAINA, A. J. M.; TRAINA JR, C. "Aggregate Similarity Queries in Relevance Feedback Methods for Content-based Image Retrieval" (2008). In: 23rd Annual ACM Symposium on Applied Computing (ACM SAC), p. 869-874, v. 2, Fortaleza (CE). ACM.
- RAZENTE, H.; BARIONI, M. C. N.; TRAINA, A. J. M.; FALOUTSOS, C. ; TRAINA JR, C. "A Novel Optimization Approach to Efficiently Process Aggregate Similarity Queries in Metric Access Methods" (2008). In: ACM 17th Conference on Information and Knowledge Management (CIKM), p. 193-202, Napa (CA). ACM.
- BARIONI, M. C. N. "Operações de consulta por similaridade em grandes bases de dados complexos" (2006). Tese de Doutorado, ICMC/USP, São Carlos, 145p. Disponível online: <http://www.teses.usp.br/>.

Operações de consulta por similaridade em grandes bases de dados complexos

Profa. Dra. Maria Camila Nardini Barioni
camila.barioni@ufabc.edu.br

Centro de Matemática Computação e Cognição - **UFABC**

Campinas
24 de abril de 2009

