SIREN: Um protótipo para busca por similaridade em dados complexos com suporte a detecção de agrupamentos*

Gabriel de Souza Fedel[1], Humberto Luiz Razente[1], Caetano Traina Júnior[1], Maria Camila Nardini Barioni[2]

[1]Instituto de Ciências Matemáticas e de Computação – ICMC/USP
Caixa Postal 668, CEP 13.560-970, São Carlos - SP

[2]Centro de Matemática, Computação e Cognição – CMCC/UFABC
Rua Santa Adélia, 166 - Bairro Bangu, CEP 09.210-170, Santo André - SP

{fedel@grad., hlr@, caetano@}icmc.usp.br, camila.barioni@ufabc.edu.br

Resumo. *Este artigo apresenta uma nova versão do protótipo SIREN na qual foi adicionada à sua linguagem de consulta SQL estendida a possibilidade de especificar consultas sobre o resultado de processos de detecção de agrupamentos de dados, que podem ser gerados por vários algoritmos baseados no método k-medóide. Além das características principais dessa nova versão, este artigo também apresenta uma descrição de parte significativa dos exemplos que deverão ser usados na demonstração da ferramenta.*

1. Introdução

Com o aumento explosivo do volume de dados acumulado nas últimas décadas e da necessidade de analisar estes dados, uma grande variedade de técnicas de mineração de dados foi desenvolvida para atender aos objetivos de diversas aplicações. Inicialmente, o foco principal das pesquisas nessa área estava voltado para a definição de novas operações e o desenvolvimento de novos algoritmos de mineração. Entretanto, com o objetivo de tornar o processo de mineração de dados uma ferramenta para aplicações reais que armazenam seus dados em Sistemas de Gerenciamento de Banco de Dados Relacionais (SGBD-R), vários pesquisadores começaram a abordar questões relacionadas à integração de técnicas de mineração de dados com SGBD-R [Netz et al. 2001].

É importante notar que um ponto fundamental para essa integração está relacionado ao fornecimento de recursos para a realização de operações básicas para as diversas técnicas de mineração existentes. Uma operação básica para várias dessas técnicas, tais como recuperação por conteúdo e detecção de agrupamentos, é o cálculo de medidas de similaridade entre os pares de objetos de um conjunto de dados. A disponibilização de recursos para a realização dessa operação já havia sido explorada na primeira versão do protótipo SIREN [Barioni et al. 2006]. Assim, o trabalho apresentado aqui teve como objetivo dar continuidade ao desenvolvimento desse protótipo explorando: a integração de uma classe de operadores de consulta por similaridade para contemplar o tratamento de operações de detecção de agrupamentos de dados baseados no método k-medóide em SGBD-R; e a possibilidade de especificar consultas sobre o resultado dessas operações, considerando tanto conjuntos de dados tradicionais quanto conjuntos de dados complexos, como imagens e áudio.

*Projeto financiado pela Fapesp.

O restante deste artigo é apresentado da seguinte maneira. A Seção 2 aborda os algoritmos de detecção de agrupamentos considerados neste trabalho. Na Seção 3 é apresentada uma breve descrição do protótipo SIREN. A Seção 4 descreve a extensão do SIREN para suportar consultas ao processo de detecção de agrupamentos. Na Seção 5 são exibidos alguns comandos-exemplo. A Seção 6 apresenta as conclusões deste trabalho.

2. Detecção de Agrupamentos de Dados

O processo de detecção de agrupamentos visa dividir os objetos de um conjunto de dados em grupos que sejam significativos, com base na similaridade existente entre eles. A medida de similaridade é obtida a partir dos valores dos atributos que descrevem os objetos de um conjunto de dados, normalmente, por meio da aplicação de uma função de distância específica para o domínio de dados considerado. Quanto maior a similaridade (ou homogeneidade) dos objetos dentro de um grupo e quanto maior a diferença entre os grupos, melhor ou mais distinto é o agrupamento resultante. Nas últimas décadas, vários algoritmos de detecção de agrupamentos foram desenvolvidos, dentre eles os baseados no método k-medóide. De maneira geral, o objetivo desses algoritmos consiste em encontrar um conjunto de k agrupamentos não sobrepostos, de modo que cada agrupamento possua um objeto representante (denominado medóide), ou seja, um objeto que esteja o mais próximo possível do centro do agrupamento.

Os três algoritmos baseados em k-medóides mais conhecidos são: PAM (*Partitioning Around Medoids*) que identifica agrupamentos com alta qualidade; CLARA (*Clustering LARge Applications*) que é um algoritmo baseado em amostragem; e CLARANS (*Clustering Large Applications based upon RANdomized Search*) que utiliza uma estratégia baseada em busca aleatória superando PAM e CLARA em relação à complexidade computacional e à qualidade do agrupamento resultante [Kaufman and Rousseeuw 2005]. Recentemente foi proposto o algoritmo PAM-SLIM que utiliza uma abordagem que emprega métodos de acesso métrico para tornar o algoritmo PAM escalável para grandes conjuntos de dados [Barioni et al. 2008]. Todos esses algoritmos foram utilizados no trabalho apresentado aqui.

3. O Protótipo SIREN

A primeira versão do SIREN (*SImilarity Retrieval ENgine*) foi desenvolvida para validar a proposta de extensão da linguagem SQL padrão para permitir a realização de consultas por similaridade sobre conjuntos de dados complexos [Barioni et al. 2006]. Esse protótipo implementa um serviço entre um SGBD convencional e os programas de aplicação, que intercepta todos os comandos SQL enviados por uma aplicação, tratando as construções sintáticas relacionadas às operações por similaridade.

O protótipo SIREN é constituído por três componentes principais: **o interpretador de comandos**, responsável pela interpretação da extensão da sintaxe SQL para a definição e manipulação de objetos complexos; **o extrator de características**, responsável pela extração de características que são utilizadas para a representação e indexação de objetos complexos (como imagens e áudio); e **o indexador**, responsável pela utilização de estruturas de indexação apropriadas para responder às consultas por similaridade. A extensão da linguagem SQL suportada por esse protótipo fornece construções sintáticas tanto para a definição de medidas de similaridade (métricas que

representam a associação entre uma função de distância e a forma de representação dos objetos) quanto para a especificação de consultas por similaridade. O protótipo SIREN foi desenvolvido em C++ para ambiente Windows e utiliza o protocolo ODBC para conexão com o SGBD PostgreSQL 8.3 ou Oracle 11g e atualmente suporta três tipos de dados complexos: `PARTICULATE` para objetos que são representados como uma coleção de atributos tradicionais; `STILLIMAGE` (para imagens) e `AUDIO` (para trilhas de áudio), sendo os dois últimos para objetos que são representados por um conjunto de características automaticamente extraídas.

4. Suporte para Processos de Detecção de Agrupamentos no SIREN

Para permitir a inclusão do suporte à especificação de operações de detecção de agrupamentos de dados na sintaxe da linguagem utilizada pelo SIREN, a abordagem utilizada considerou os seguintes recursos presentes no padrão SQL:2003 – a possibilidade de especificar funções na cláusula `FROM`, as quais retornam os seus resultados no formato de tabelas (*table functions*); e a utilização de construções sintáticas como as empregadas em comandos `SET`, utilizados na inicialização dos parâmetros de uma sessão, para a inicialização dos parâmetros necessários à realização de uma sessão de detecção de agrupamentos sobre um determinado atributo complexo.

Para a realização de um processo de detecção de agrupamentos de dados sobre um atributo A_i de uma relação R, é necessário que A_i seja um atributo de um dos tipos complexos suportados pelo SIREN e que admita busca por similaridade segundo uma métrica previamente associada a ele. Então, para se definir um processo de detecção de agrupamentos, deve-se utilizar o comando `SET CLUSTERING` antes da realização de comandos SQL que façam referência ao resultado do processo de detecção de agrupamentos definido sobre um atributo complexo. A sintaxe desse comando é apresentada abaixo:

```
<especificação_agrupamento>::=
     SET CLUSTERING <nome_processo_detecção_agrupamentos>
         METHOD <nome_método> K <valor_k>
         METRIC <nome_métrica> ON <nome_tabela>'.'<nome_atributo>
```

A definição de um processo de detecção de agrupamentos compreende a especificação dos seguintes parâmetros: o método a ser utilizado (`<nome_método>`), o número de agrupamentos que se deseja encontrar (`<valor_k>`), a medida de similaridade a ser considerada no processo (`<nome_métrica>`), e sobre qual atributo complexo o processo deve ser realizado (`<nome_tabela>.<nome_atributo>`).

Como resultado de cada processo de detecção de agrupamentos, são obtidas duas relações: a relação de agrupamentos encontrados $C(A_i)$ e uma relação $CI(A_i)$ que associa cada valor de A_i a um agrupamento em $C(A_i)$. Portanto, sempre que o usuário quiser se referir a um processo de detecção de agrupamentos, ele deve se referir a pelo menos uma dessas relações em uma cláusula `FROM` de um comando `SELECT`, por meio das funções `Cluster` e `Clustering` seguindo a seguinte sintaxe:

```
CLUSTER '('<nome_processo_detecção_agrupamentos>')'
CLUSTERING '('<nome_processo_detecção_agrupamentos>')'
```

Cada tabela resultante da aplicação das funções `Cluster` e `Clustering` deve se referir a um processo de detecção de agrupamentos (identificado por um nome único) sobre um atributo A_i de uma relação R, a qual, sendo uma tabela definida pelo sistema, tem

uma estrutura pré-definida. A tabela resultante da aplicação da função `Cluster` contém todas as informações sobre cada agrupamento, como: o número de objetos pertencentes ao agrupamento; a dissimilaridade média dos objetos ao objeto central; a dissimilaridade máxima de qualquer objeto ao objeto central; e o diâmetro do agrupamento. Já a tabela resultante da aplicação da função `Clustering` armazena as informações relativas a associação de cada objeto do conjunto de dados ao agrupamento a que ele pertence.

Para permitir a realização de consultas sobre o resultado de processos de detecção de agrupamentos foi necessário: estender o dicionário de dados do SIREN para suportar o armazenamento dos dados provenientes de cada processamento; e estender o seu módulo de interpretação de comandos de modo que ele pudesse reconhecer a nova sintaxe e realizar alguns processamentos necessários, como consultar/armazenar informações nas tabelas apropriadas do dicionário de dados e realizar chamadas às operações de detecção de agrupamentos especificadas nos comandos.

5. Exemplos de Utilização

Nessa seção são apresentados diversos exemplos de comandos que foram usados para definir e consultar o resultado de processos de detecção de agrupamentos de dados no SIREN. A descrição dos conjuntos de dados utilizados pelos comandos-exemplo é apresentada a seguir juntamente com os respectivos comandos-exemplo.

5.1. Conjunto de Dados: `Música`

O primeiro conjunto de exemplos de consulta utiliza o conjunto de dados `Música`. Esse conjunto é composto por 450 músicas em formato MP3 divididas em três gêneros musicais (cada um com 150 tuplas): Rock, MPB e Música Clássica. Para esse conjunto de exemplos foi empregado o algoritmo PAM, e foram utilizadas duas métricas, uma baseada no extrator STFT (*Short Time Fourier Transform*) e outra baseada no extrator MFCC (*Mel Frequency Cepstral Coefficients*) [Tzanetakis 2001], ambas disponíveis no SIREN para o tipo de dados `AUDIO`. Esses tipos de extratores de características podem ser utilizados para capturar aspectos relacionados ao timbre de um sinal de áudio. Os comandos que definem as métricas utilizadas nas consultas, a tabela que armazena esse conjunto de dados e os processos de detecção de agrupamentos são apresentados a seguir.

```
CREATE METRIC AudioSTFT FOR AUDIO (SoundTextureExt (STFT AS STFT));
CREATE METRIC AudioMFCC FOR AUDIO (SoundTextureExt (MFCC AS MFCC));

CREATE TABLE Musica (ID integer primary key, Titulo varchar(70),
                     Genero varchar(25), ArquivoMP3 AUDIO METRIC
                     USING (AudioSTFT DEFAULT, AudioMFCC));

SET CLUSTERING Musica_Pam_Stft METHOD PAM K 3
               METRIC AudioSTFT ON Musica.ArquivoMP3;
SET CLUSTERING Musica_Pam_Mfcc METHOD PAM K 3
               METRIC AudioMFCC ON Musica.ArquivoMP3;
```

Considerando os processos de detecção de agrupamentos definidos acima é possível fazer consultas do tipo – "(1) Quais músicas pertencem a cada um dos 3 agrupamentos de músicas, segundo a métrica AudioSTFT?"; "(2) Quais são as músicas mais representativas de cada um dos 3 agrupamentos obtidos?"; ou ainda: "(3) Para cada música, indicar a qual de cada um dos 3 agrupamentos ela pertence, qual a música mais representativa de seu agrupamento, e a distância entre ambas." Os comandos utilizados para responder a cada uma dessas consultas são apresentados abaixo.

```
SELECT Musica.Titulo, Agrupamento.ClusterLabel, Musica.Genero
  FROM Musica, Clustering(Musica_Pam_Stft) Agrupamento
 WHERE Musica.ArquivoMP3 = Agrupamento.Object; (1)

SELECT Musica.Titulo, Grupo.ClusterLabel
  FROM Musica, Cluster(Musica_Pam_Stft) Grupo
 WHERE Musica.ArquivoMP3 = Grupo.CenterObject; (2)

SELECT M1.Titulo, Agrupamento.ClusterLabel,
       M2.Titulo, Agrupamento.Distance
  FROM Musica M1, Musica M2, Clustering(Musica_Pam_Stft) Agrupamento
 WHERE M1.ArquivoMP3 = Agrupamento.Object AND
       M2.ArquivoMP3 = Agrupamento.CenterObject; (3)
```

Comandos como esses poderiam ser empregados, por exemplo, para explorar a adequação de características de áudio para a realização de classificação automática de gênero musical. Para permitir a realização dessa análise é exibido na Tabela 1 o resultado da consulta (4) para ambas as métricas definidas para o conjunto de dados `Música`.

```
SELECT Musica.Genero, Agrupamento.ClusterLabel, Count(*)
  FROM Clustering(Musica_Pam_Stft) Agrupamento, Musica
 WHERE Agrupamento.Object = Musica.ArquivoMP3
 GROUP BY Musica.Genero, Agrupamento.ClusterLabel (4)
```

Tabela 1. Resultado do algoritmo PAM para os extratores STFT e MFCC.

STFT

	Rock	MPB	Clássica
Agrupamento 1	31	31	39
Agrupamento 2	84	91	66
Agrupamento 3	35	28	45

MFCC

	Rock	MPB	Clássica
Agrupamento 1	27	27	86
Agrupamento 2	64	85	38
Agrupamento 3	59	38	26

Analisando a Tabela 1 é possível perceber que não houve uma predominância significativa de um gênero musical em nenhum dos agrupamentos obtidos. E que esse comportamento ocorreu independentemente da métrica utilizada no processo de detecção de agrupamentos. Assim, é possível concluir que características intrínsecas relativas ao timbre de um sinal de áudio não são suficientes para permitir a separação dessas músicas nos gêneros musicais considerados. Outros extratores que representam sinais de áudio considerando diferentes características, baseadas por exemplo em ritmo e/ou harmonia, precisariam ser analisados.

5.2. Conjunto de Dados: `Pen-Based Recognition of Handwritten Digits`

Para o segundo conjunto de exemplos foi utilizado o conjunto de dados `Pen-Based Recognition of Handwritten Digits` do repositório UCI Machine Learning (http://archive.ics.uci.edu/ml/). Ele contém 10.992 tuplas com 16 atributos representando características extraídas de números escritos à mão, divididas em 10 classes (0 a 9). Para esse conjunto de dados foram definidos processos de detecção de agrupamentos considerando os algoritmos `CLARA`, `CLARANS` e `PAM-SLIM`. Os comandos que definem a métrica utilizada nas consultas, a tabela que armazena esse conjunto de dados e o processo de detecção de agrupamentos para o algoritmo CLARA são apresentados a seguir.

```
CREATE METRIC MetricaPenDigits USING LP2 FOR PARTICULATE (
       a1 INT, b1 INT, c1 INT, d1 INT, e1 INT, f1 INT, g1 INT, h1 INT,
       i1 INT, j1 INT, k1 INT, m1 INT, n1 INT, o1 INT, p1 INT, q1 INT);
```

```
CREATE TABLE Pendigits (
       a INT, b INT, c INT, d INT, e INT, f INT, g INT, h INT,
       i INT, j INT, k INT, m INT, n INT, o INT, p INT, q INT,
       pd PARTICULATE METRIC REFERENCES (a AS a1, b AS b1, c AS c1,
       d AS d1, e AS e1, f AS f1, g AS g1, h AS h1, i AS i1, j AS j1,
       k AS k1, m AS m1, n AS n1, o AS o1, p AS p1, q AS q1) USING
       (MetricaPenDigits DEFAULT), classe INT);

SET CLUSTERING PendigitsClara METHOD Clara K 10
               METRIC MetricaPenDigits ON Pendigits.pd;
```

Um resumo dos resultados obtidos com todos os processos de detecção de agrupamentos definidos é exibido na Tabela 2. Tais resultados poderiam ser utilizados para comparar, por exemplo, a adequação de cada algoritmo de acordo com o tempo disponível para o seu processamento e a qualidade desejada dos agrupamentos resultantes.

Tabela 2. Execução dos algoritmos CLARA, CLARANS e PAM-SLIM.

	Dissimilaridade média	Tempo de execução
CLARA	73,5440	16 segundos
CLARANS	67,1720	29 minutos e 16 segundos
PAM-SLIM	68,7879	6 minutos e 7 segundos

6. Conclusão

Este artigo abordou a questão da integração de mineração de dados com SGBD, apresentando um protótipo no qual foram incorporados vários algoritmos de detecção de agrupamentos que podem ser aplicados em conjuntos de dados de vários tipos (tradicionais e complexos) e tamanhos. Essa integração facilita o processo de descoberta de conhecimento, pois elimina a necessidade de utilizar vários ambientes para as diferentes fases do processo. Além disso, a integração com uma linguagem de consulta flexível, como a SQL, permite que consultas diversas possam ser executadas sobre o resultado desses algoritmos, tornando possível a persistência e o compartilhamento das novas informações geradas.

Referências

Barioni, M. C. N., Razente, H., Traina, A. J. M., and Traina-Jr., C. (2006). SIREN: A similarity retrieval engine for complex data. In *Demo Session, Int'l Conf. on Very Large Data Bases (VLDB)*, pages 1155–1158. VLDB Endowment.

Barioni, M. C. N., Razente, H., Traina, A. J. M., and Traina-Jr., C. (2008). Accelerating k-medoid-based algorithms through metric access methods. *Journal of Systems and Software*, 81(3):343–355.

Kaufman, L. and Rousseeuw, P. J. (2005). *Finding groups in data: An introduction to cluster analysis*. John Wiley and Sons.

Netz, A., Chaudhuri, S., Fayyad, U. M., and Bernhardt, J. (2001). Integrating data mining with SQL databases: Ole db for data mining. In *Int'l Conf. on Data Engineering (ICDE)*, pages 379–387, Heidelberg, Germany. IEEE.

Tzanetakis, G. (2001). Automatic musical genre classification of audio signals. In *Int'l Symp. on Music Information Retrieval (ISMIR)*, pages 205–210, Bloomington.