Weka-GDPM: uma Extensão do Weka para Suportar Dados Espaciais

Aérton Dillenburg, Alisson Moscato Loy, Andrey Tietbohl Palma, Vania Bogorny, Luis Otavio Alvares

Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre – RS – Brasil

{adillenburg,amloy,andrey,vbogorny,alvares}@inf.ufrgs.br

Abstract. *One of the main challenges in spatial data mining is to automate the data preparation tasks, which consume more than 60% of the effort and time required for knowledge discovery in geographic databases. This paper describes an extension of the Weka data mining toolkit, called Weka-GDPM (Geographic Data Preprocessing Module), which provides a friendly interface for pre-processing spatial data.*

Resumo. *Um dos principais desafios em mineração de dados espaciais é automatizar a tarefa de preparação dos dados. Esta etapa consome mais de 60% do esforço e tempo necessário para a extração de conhecimento de bases de dados geográficos. Este artigo descreve uma extensão da ferramenta de mineração de dados Weka, denominada Weka-GDPM (Geographic Data Preprocessing Module), que provê uma interface amigável para realizar a tarefa de pré-processamento de dados geográficos.*

1. Introdução

O uso de dados geográficos tem crescido significativamente nos últimos anos nos mais diversos domínios de aplicação como, por exemplo, planejamento urbano, transporte, telecomunicações e saúde pública. Dados geográficos representam entidades do mundo real que estão localizados na superfície terrestre [Open Gis Consortium 1999].

Em bancos de dados geográficos, os dados (cidades, estradas, rios, etc.) são armazenados em diferentes entidades, as quais precisam ser espacialmente agrupadas para que se possa encontrar padrões interessantes no processo de descoberta de conhecimento. Esse agrupamento faz parte do pré-processamento de dados e corresponde a maior fatia de tempo/esforço necessária para executar todo o processo de descoberta de conhecimento [Addrians e Zantinge 1996].

A maioria das soluções propostas para mineração de dados espaciais não considera a etapa de preparação de dados e freqüentemente propõe estender o padrão SQL, adicionando novas funções e operações para mineração, através de uma linguagem de mineração de dados [Han et al 1997].

No entanto, a maioria dos sistemas gerenciadores de banco de dados geográficos (SGBDG) não implementa essas linguagens, o que dificulta o processo de mineração de dados geográficos em problemas e aplicações reais.

Como contribuição para a área de mineração de dados geográficos desenvolveu-se o Weka-GDPM [Bogorny et al 2006a], que é uma extensão do Weka [Witten e Frank 2005]. O Weka é uma ferramenta para mineração de dados não-espaciais gratuita, de código aberto, que provê uma interface amigável e que implementa diversos algoritmos de mineração de dados. O módulo GDPM (*Geographic Data Preprocessing Module*) é uma extensão para mineração de dados espaciais extraídos de um banco de dados geográficos. O restante do artigo está organizado da seguinte forma: na Seção 2 são apresentados os conceitos básicos de bancos de dados geográficos e relacionamentos espaciais; na Seção 3 é mostrada a estrutura da extensão GDPM para a ferramenta Weka e a interface com o usuário; finalizando, na Seção 4, são apresentadas as considerações finais.

2. Conceitos Básicos

Em banco de dados geográficos, cada tipo de objeto do mundo real é normalmente armazenado em uma tabela diferente, caracterizando uma entidade geográfica. Cada tipo de objeto tem atributos espaciais e não-espaciais, conforme ilustrado na figura 1. A figura 1 possui três tipos de objetos geográficos (*Rua, RecursoHídrico, PostoCombustivel*), os quais contêm um atributo espacial denominado *"shape"*.

(a) Rua

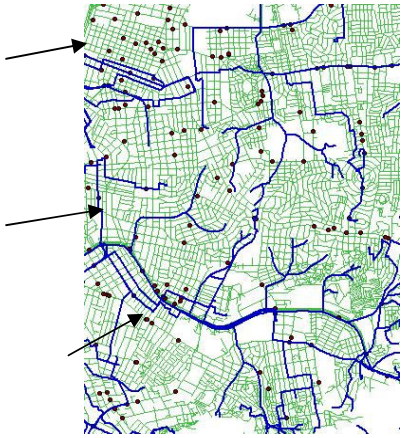
Gid	Nome	Shape
1	Erie ST	Multiline [$(x_1,y_1),(x_2,y_2),..$]
2	Oak ST	Multiline [$(x_3,y_3),(x_4,y_4),..$]

(b) RecursoHídrico

Gid	Nome	Shape
1	Jacui	Multiline [$(x_5,y_5),(x_2,y_2),..$]
2	Guaiba	Multiline [$(x_6,y_6),(x_2,y_2),..$]
3	Uruguai	Multiline [$(x_5,y_5),(x_7,y_7),..$]

(c) PostoCombustivel

Gid	Nome	VolDiesel	VolGas	Shape
1	BR	20000	85000	Point[(x_8,y_8)]
2	IPF	30000	95000	Point[(x_9,y_9)]
3	Elf	25000	120000	Point[(x_3,y_3)]



(d) GEOMETRY_COLUMNS

F_table_schema	F_table_name	F_geometry_column	Type	SRID
Public	Rua	Shape	Multiline	-1
Public	RecursoHídrico	Shape	Multiline	-1
Public	PostoCombustivel	Shape	Point	-1

Figura 1. Estrutura de um banco de dados geográfico [Bogorny et al 2007]

Os atributos espaciais dos objetos geográficos possuem relacionamentos espaciais intrínsecos (perto, longe, contém, intercepta, etc.). Devido a esses relacionamentos, as entidades do mundo real podem afetar as características de objetos vizinhos. Isso faz com que relacionamentos espaciais sejam a principal característica em dados geográficos a ser considerada para mineração de dados e descoberta de conhecimento.

Relacionamentos espaciais não são explicitamente armazenados em banco de dados geográficos, então eles devem ser calculados através de "operações espaciais". Há basicamente três tipos de relacionamentos espaciais [Gutting 1994], ilustrados na figura

2: distância, ordem e topológicos. Os relacionamentos topológicos caracterizam o tipo de interseção entre dois objetos espaciais e podem ser classificados em: igual, toca, contém/contido, cruza, sobrepõe e disjunto.

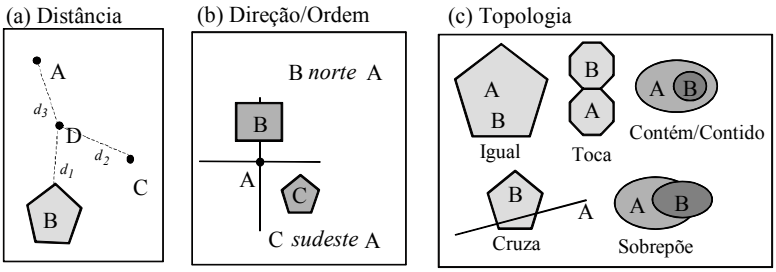


Figura 2. Relacionamentos espaciais

SGBDG e Sistemas de Informação Geográfica (SIG) implementam funções específicas para manipular dados geográficos. O OGC (*Open Gis Consortium*) é uma organização dedicada a desenvolver padrões para operações geográficas e integração de dados geográficos, com o objetivo de prover interoperabilidade para SIG. Dentre as muitas especificações criadas pelo OGC, duas são mais importantes para este trabalho: operações para calcular relacionamentos espaciais e o esquema do banco de dados.

O esquema do banco de dados é armazenado em uma tabela denominada GEOMETRY_COLUMNS, a qual é automaticamente criada e atualizada pelo SGBDG seguindo as especificações da OGC. Um exemplo é mostrado no item *d* da figura 1. Essa tabela consiste de uma linha para cada tipo de objeto em um banco de dados geográficos com atributos espaciais. Ela é populada automaticamente quando dados geográficos são carregados no banco de dados pela primeira vez e armazena todas as características do objeto geográfico, incluindo o nome do esquema (f_table_schema), o nome das tabelas geográficas (f_table_name), o nome do atributo geométrico (f_geometry_column), bem como seu tipo (type).

Para automatizar o pré-processamento de dados geográficos a tabela GEOMETRY_COLUMNS pode ser usada para buscar todos os atributos espaciais armazenados em um SGDBG, tornando o processo interoperável.

3. Weka-GDPM

A figura 3 apresenta a arquitetura do Weka-GDPM para mineração de dados espaciais, utilizando bases de dados seguindo o padrão OpenGIS.

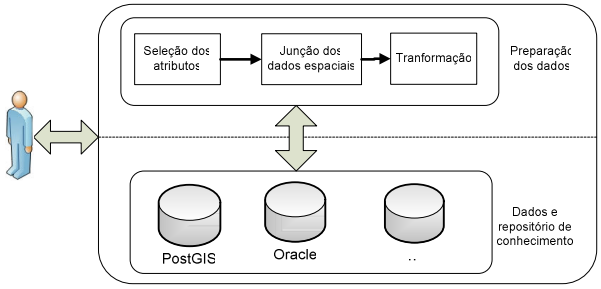


Figura 3. Arquitetura do Weka-GDPM

A ferramenta encapsula as tarefas de conexão com o banco de dados, o pré-processamento e a geração de um arquivo em formato texto com a extensão *.arff*, que é o padrão de entrada para o Weka.

A figura 4 (esquerda) ilustra o módulo denominado Weka.Explorer, o qual permite conectar a uma base de dados, abrir um *site* na *web*, ou um arquivo *.arff* (figura 5 direita). Acionando o botão *"Open DB"* é apresentada a interface ilustrada na Figura 4 (direita), que permite realizar a conexão com o banco de dados.

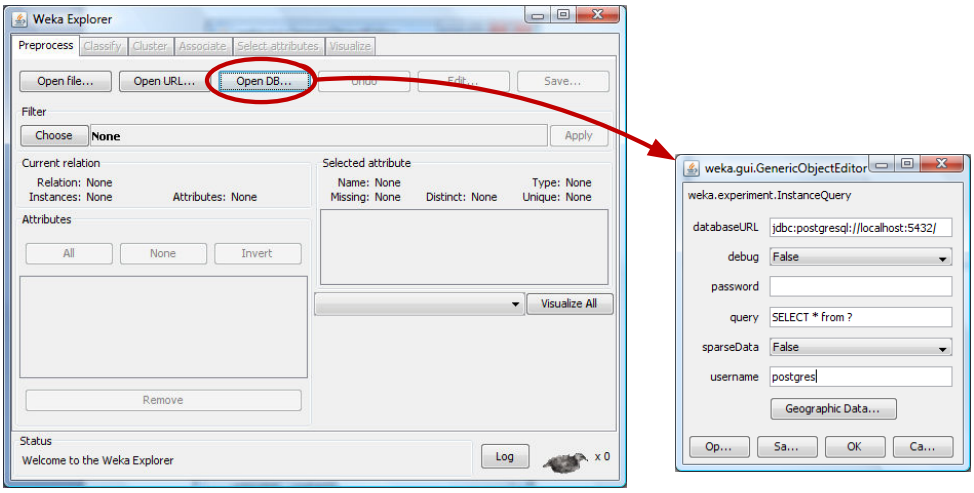


Figura 4. (esquerda) Interface do módulo denominado Weka.Explorer, (direita) Janela de conexão ao banco de dados.

Através desta interface o Weka conecta ao banco de dados utilizando JDBC e abre o módulo GDPM (Figura 5 esquerda) através do botão "Geographic Data". O GDPM é uma nova classe java completamente independente que permite automatizar o pré-processamento de dados geográficos.


```
@relation geographic_data
@attribute bairro {Restinga, Moinhos,
                   Mont_Serrat, Azenha}
@attribute dengue {null,NAO,SIM}
@attribute postos_comb_CONTAINS {yes}
@attribute hospitais_CONTAINS {yes}
@attribute arroios_CROSSES {yes}

@data
Restinga,NAO,yes,?,yes
Moinhos,NAO,yes,yes,?
Mont_Serrat,NAO,yes,?,yes
Azenha,SIM,yes,yes,yes
```

Figura 5. (esquerda) Interface da extensão GDPM, (direita) exemplo de arquivo *.arff*

A tarefa de pré-processamento de dados se inicia com o nome do esquema do banco de dados. O botão "Load" busca todos os tipos de objetos espaciais da tabela *geometry_columns* (figura 1 (*d*)). Após a carga dos objetos espaciais nos campos *"Target Feature"* e *"Relevant Feature"*, o usuário pode selecionar o objeto alvo (*target feature*) onde a tarefa de descobrimento de dados será focalizada, bem como todos os objetos relevantes (*relevant features*). Cada linha da tabela resultante que dará origem ao arquivo *arff* será uma instância do objeto alvo (corresponde a bairro na figura 5 (direita)). Cada atributo será ser um *atributo não-espacial* do objeto alvo (dengue na figura 5(direita)) ou um *atributo espacial*, representado por um relacionamento espacial do objeto alvo com os objetos relevantes (postos_comb_contains, hospitais_contains e arroios_crosses no exemplo da figura 5 (direita)).

O usuário ainda pode selecionar o nível de granularidade desejado entre "*feature type"* e "*feature instance"*[Bogorny et al 2006b]. No nível de granularidade *"feature instance"* (Tabela 1 esquerda), quando pré-processado e formatado para o padrão de entrada do Weka (Tabela 1 direita), o tipo do atributo com a respectiva instância é transformado no nome do atributo. Para atributos que não possuem relacionamentos é usado o símbolo "?", que é o padrão para ausência de dados no Weka.

TargetF_id (city)	RelevantF Instance	Relationship
1	River_1	Contains
1	River_2	Crosses
2	River_3	Contains
2	River_4	Crosses
3	River_2	Crosses
1	Slum_1	Contains
2	Slum_2	Contains

TargetF_id (city)	River_1	River_2	River_3	River_4	Slum_1	…
1	Contains	Crosses	?	?	Contains	
2	?	?	Contains	Crosses	?	
3	?	Crosses	?	?	?	

Tabela 1. (esquerda) Resultado da consulta no banco de dados e (direita) transformação para o formato do Weka utilizando a granularidade *"feature instance"* [Bogorny et al 2006a]

No nível de granularidade de *"feature type"*, como mostrado na Tabela 2 (esquerda), as instâncias não são passadas para a tabela de saída. Por exemplo, a cidade 1 possui dois relacionamentos topológicos com o tipo *River: contains (river 1)* e *crosses (river 2)*. Neste nível de granularidade, para preservar os diferentes tipos de relacionamento topológico com o mesmo objeto relevante, é necessário criar diferentes nomes de atributos (contains_river, crosses_river). Neste caso, o atributo recebe a string "yes" quando há relacionamento e o caracter "?" na ausência deste, conforme ilustra a Tabela 2 (direita).

TargetF_id (city)	RelevantF Type	Relationship
1	River	Contains
1	River	Crosses
2	River	Contains
2	River	Crosses
3	River	Crosses
1	Slum	Contains
2	Slum	Contains

TargetF_id (city)	Contains_River	Crosses_River	Contains_Slum	…
1	Yes	Yes	Yes	
2	Yes	Yes	Yes	
3	?	Yes	?	

Tabela 2. (esquerda) Resultado da consulta no banco de dados e (direita) transformação para o formato do Weka utilizando a granularidade de *"feature type"* [Bogorny et al 2006a]

Uma vez transformados os dados para o formato ilustrado nas Tabelas 1 e 2 (direita), é então gerado o arquivo *.arff* , cujo exemplo é ilustrado na figura 5 (direta). Após gerado o arquivo *.arff,* qualquer algoritmo de mineração implementado no Weka poderá ser utilizado.

4. Considerações finais

O Weka-GDPM tem sido utilizado em diversos trabalhos acadêmicos internacionais, e o objetivo desta demonstraçao é apresentar e disponibilizar a ferramenta para pesquisadores brasileiros. Esta ferramenta já foi oficialmente reconhecida pelos desenvolvedores do Weka, na Universidade de Waikato na Nova Zelandia, conforme pode ser observado no *site* http://www.cs.waikato.ac.nz/ml/weka/index_related.html.

Esta extensão foi originalmente desenvolvida para pré-processar dados geográficos armazenados no SGBD PostgreSQL [Bogorny et al 2006a]. No Workshop de Software Livre [Bogorny et al 2007] foi apresentada uma versão que permite eliminar padrões óbvios passíveis de serem detectados no pré-processamento. Esta versão demo, além de estar adaptada à última versão estável do WEKA (3.4), permite também conectar ao SGBD Oracle, que não segue o padrão OGC.

Como trabalho futuro está sendo desenvolvida uma extensão da ferramenta para suportar dados espaço-temporais gerados a partir de dispositivos móveis.

Referências

Addrians, P. e Zantinge, D. (1996) “Data mining”. Addison Wesley Longman, Harlow, England.

Bogorny, V.; Palma, A.; Engel, P.; Alvares, L.O. (2006a): “Weka-GDPM: Integrating Classical Data Mining Toolkit to Geographic Information Systems.” In: SBBD Workshop WAAMD'06, Florianópolis, Brasil, 16-20 outubro, p. 9-16.

Bogorny, V.; Engel, P.; Alvares, L.O. (2006b): “GeoARM: an Interoperable Framework to Improve Geographic Data Pre-processing and Spatial Association Rule Mining.” In: Proc. of the 18th International Conference on Software Engineering and Knowledge Engineering (SEKE'06), San Francisco, USA, p. 79-84.

Bogorny, V.; Palma, A.; Kuijpers, B.; Alvares, L.O. (2007) “Spatial Data Mining: From Theory to Practice with Free Software.” In: Proc. of International Workshop on Free Software (WSL'07). Porto Alegre, Brasil.

Gutting, R.H. (1994) “An Introduction to Spatial Database Systems”. International Journal on Very Large Data Bases, v3 (4), p. 357-556.

Han, J.; Koperski, K. e Stefanvic, N. (1997) “GeoMiner: a system prototype for geographic data mining” In ACM-SIGMOD, ACM Press, Arizona, p. 553-556

Open Gis Consortium. (1999) Topic 5, the OpenGIS abstract specification – OpenGis features – Versão 4. Disponível em : *http://www.opengeospatial.org/standards* Acessado em: junho/2008

Witten, I. e Frank, E. (2005) “Data Mining: Practical machine learning tools and techniques”, 2ª Edição, Morgan Kaufmann, San Francisco.