# MÉTODOS DE CLASSIFICAÇÃO PARA DOCUMENTOS XML - UMA COMPARAÇÃO QUALITATIVA

## METHODS OF CLASSIFICATION FOR XML DOCUMENTS - A QUALITATIVE COMPARISON

Williams Diaz, Paulo Caetano da Silva

Universidade Salvador (UNIFACS), Bahia, Brasil
williams@interban.com.br , paulo.caetano@prof.unifacs.br

**Resumo**
A classificação de documentos semi-estruturados, a exemplo dos documentos XML, se torna necessária devido a grandes bases de dados que estão sendo criadas neste formato, em todas as áreas do conhecimento. A extração de informações úteis nessas bases é um grande desafio e passa necessariamente pela classificação das informações. Um dos principais desafios não é o de como classificar, mas sim o que classificar, em razão da possível heterogeneidade das bases de dados. Neste artigo descrevem-se e comparam-se alguns dos métodos e algoritmos utilizados para realizar esta tarefa.

**Palavras-chave:** Documentos XML; Classificação XML; Algoritmos de Classificação; Métodos de Classificação; Mineração de dados.

**Abstract**
The classification of semi-structured documents, like XML documents becomes increasingly necessary due to large databases that are being created in all areas of knowledge in this format. The extraction of useful information from these databases becomes a major challenge and necessarily involves the classification of information. One such challenge is not how to classify, but what sort, because of the possible heterogeneity of databases. In this article we describe and compare some of the methods and algorithms used to accomplish this task.

**Keywords:** XML Documents; XML Classification; XML Algorithms; Classfiication Algorithms; Classification Methods; Data Mining.

## 1 INTRODUÇÃO

Esta avaliação se refere basicamente aos dois anos de pesquisa do XML Mining Track no Inex 2005 e 2006. Este trabalho focou exclusivamente a classificação de documentos XML, de formar a vislumbrar desafios para futuras pesquisas nesta área.

As pessoas produzem mais informações do que são capazes de ler e analisar [05]. O volume de informações disponíveis na Web, por exemplo, duplica a cada três anos [07]. Grande parte destas informações se encontra no formato XML. Um documento XML é um texto com uma estrutura definida pelo Consórcio W3C [08]. XML é uma linguagem de marcação, assim como HTML. Os objetivos das duas são diferentes. Enquanto o HTML foi criado para mostrar informações, o XML foi criado para guardar dados. O documento XML não possui tags pré-definidas, o próprio usuário pode defini-las, seguindo as regras estabelecidas pelo W3C. Documentos XML são documentos definidos pela sua estrutura e pelo seu conteúdo.

Frequentemente é necessário explorar grandes quantidades de dados a procura de padrões consistentes, de forma a detectar relacionamentos entre variáveis, criando assim um novo conjunto de dados que possa ter alguma utilidade. Esse processo é chamado de mineração de dados. Os dados devem ser acessados e ordenados, ou seja, classificados de uma forma lógica a fim de obter uma efetiva mineração de dados. A maioria dos métodos existentes para classificação de dados se concentra ou na estrutura ou nos dados propriamente ditos. No entanto, dependendo da origem das informações e da aplicação, se faz necessário processar a classificação das duas formas.

A classificação consiste em descobrir uma função que mapeia um conjunto de registros em um conjunto de rótulos pré-definidos, denominados classes [05]. Uma vez descoberta, esta função é aplicada a novos registros de forma a prever a classe em que tais registros se enquadram. Por exemplo, na área médica podem-se armazenar os registros médicos de pacientes que sofrem de determinadas doenças e permitir com isso identificar a doença em outros pacientes que possuem os mesmos valores clínicos ou laboratoriais.

Em 2005 foi criado o grupo do Mining Track [03] com o objetivo de identificar problemas no processo de mineração de dados em documentos semi-estruturados, e.g. documentos XML, e identificar o potencial das técnicas de inteligência artificial para clusterização e classificação nestes documentos. Em 2006 foi proposta para o grupo uma nova tarefa: o mapeamento estrutural. Lidar

com estruturas é útil para extrair classes de diferentes fontes de documentos. Gerenciar classes e conteúdos ao mesmo tempo requer maior complexidade nos algoritmos das regras de classificação. O resultado obtido pelo Mining Track foi à criação de diferentes modelos de clusterização, i.e. agrupamento, e classificação. Neste artigo somente são descritos os modelos de classificação.

Este texto se organiza da seguinte forma: na seção 2 são discutidos os algoritmos e métodos de classificação e na seção 3 são descritas as conclusões.

## 2 ALGORITMOS E MÉTODOS DE CLASSIFICAÇÃO

Nesta seção serão discutidos e comparados os trabalhos apresentados no Mining Track de 2005 e 2006. Freqüentemente, um método de classificação é composto por dois ou mais algoritmos de computação.

### 2.1 Garboni et Al.

Este método de classificação [01], é baseado na informação estrutural do documento XML. Cada documento é visualizado como uma estrutura em forma de árvore, representados pelas suas tags. Desta forma, o algoritmo utiliza a informação contida na estrutura para detectar famílias de documentos com estruturas similares. O trabalho de Garboni consiste em descobrir padrões estruturais e depois classificá-los adequadamente.

Para realizar a classificação, em primeiro lugar são definidos dois parâmetros especificados pelo usuário: mínimo de ocorrências e mínimo de confiabilidade. Em seguida, o algoritmo procura por padrões freqüentes na estrutura do documento e identificando-os, um novo schema é criado com as árvores mais freqüentes. A Figura 1 ilustra o exemplo do resultado do processamento desse algoritmo.

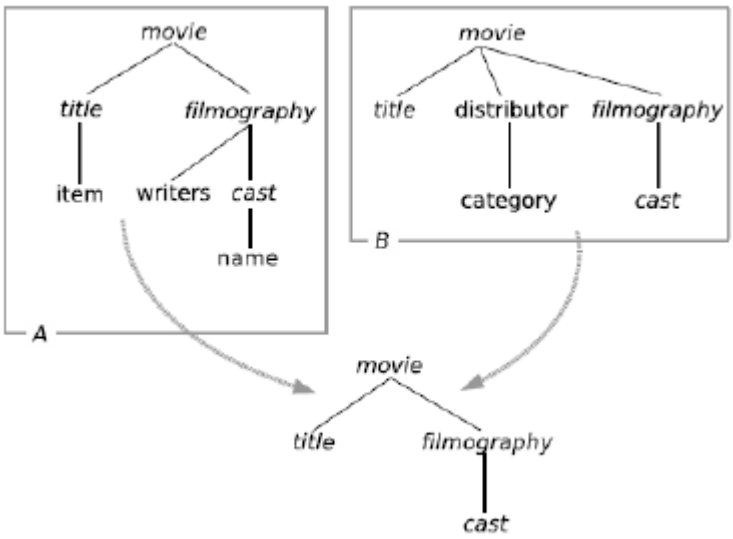


Figura 1 – Exemplo de criação de uma nova árvore [01].

Esse método é processado da seguinte forma: transforma o documento XML em uma seqüência de classes. No exemplo acima, fica evidente que a seqüência ***movie, title, filmography*** e ***cast*** é um padrão presente nas duas árvores A e B. Utilizando então aqueles parâmetros definidos anteriormente (ocorrências e confiabilidade), é utilizado um algoritmo padrão de extração de seqüência e obtemos um novo schema de estrutura. Os autores denominam este passo de "redução".

Padrões muito freqüentes são excluídos pelo algoritmo quando são considerados irrelevantes. Deve-se lembrar que a idéia central é obter similaridade de estruturas, e não de identificadores (, i.e. tags.

A distância de edição [09] é uma função de similaridade muito utilizada para categorizar árvores hierárquicas. Ela serve para determinar o quão parecidas são duas strings. Esse valor é calculado em uma métrica de distância a ser escolhida, a exemplo de número de caracteres diferentes na string, ou o número de intervenções necessárias para converter uma string em outra.

Os autores propõem uma métrica de edição (ou função de similaridade), baseada na seguinte expressão:

$$score2(D_i, C_j) = \frac{\sum_{k=1}^{|C_j|} \frac{i \times |LCS(D_i, sp_{C_j}(k))|}{|sp_{C_j} k|}}{|C_j|}$$

Na qual D é o documento dentro do conjunto C de documentos. LCS é a função **Longest Common Subsequence**, (subsequência de estrutura mais característica). E **i** é um fator experimental dependendo do tamanho das sequências coincidentes.

**Teste do método:** o algoritmo foi escrito em C++, usando um Pentium PC rodando em plataforma Red Hat. A base de dados utilizada foi a MovieDB, uma base que possui 9.643 títulos de filmes em formato XML.

**Vantagens deste método:** este método é muito eficiente quando utilizados em documentos extensos e com similaridade de estruturas entre si.

**Desvantagens:** Este método é "estrutura-dependente". Isto implica que os índices de exatidão alcançados dependem muito do padrão de estrutura do documento, dos seus níveis e da repetição destes padrões.

### 2.2 Método de Candilier et Al.

O método de Candilier [02] propõe que ao invés de trabalhar diretamente com as árvores hierárquicas de um documento XML, estas sejam transformadas num conjunto de valores-atributos, para então serem classificadas utilizando um algoritmo adequado.

Os autores utilizam como métrica de distância de edição as características de relacionamento pai-filho, irmão-irmão e caminhos distintos. Eles sugerem criar novos atributos a medida que estes vão sendo conhecidos. Estes atributos sempre têm um valor inteiro. Por exemplo, o nó principal poderia ser 1, os filhos 1.1 e 1.2 - sendo sempre valores inteiros. Ao comparar estes valores de dois documentos diferentes pode-se descobrir uma similaridade ou diferença entre eles, simplesmente comparando os seus atributos. Da mesma forma, seria possível perceber se é permitida a relação pai-filho em um determinado documento, quantificando o número de filhos existentes entre um documento e outro.

O algoritmo requer que algumas regras sejam definidas a priori. Isso permite que alguns dados sejam descartados por não pertencerem a essas regras, ou seja, um processo de "limpeza" do resultado obtido.

O algoritmo é executado diversas vezes, de forma iterativa, até que o processo de aprendizagem esteja finalizado, e a "limpeza" efetuada. O processo de aprendizagem basicamente consiste em separar as classes encontradas em diferentes partes de acordo com as regras pré-

definidas. Ao resultado é aplicado um modelo probabilístico, de forma a tomar decisões de classificação.

O algoritmo proposto pelos autores, e originário do SSC (Subspace Clustering Algorithm), o qual usa um modelo probabilístico para determinar uma nova hierarquia de classes de um texto XML. A representação do algoritmo é mostrada a seguir:

***Input:*** *the dataset D of XML documents*

*- initialize the unique cluster C1 with all the documents of the dataset D*

*- create a new empty hierarchy H*

- set CUT = 1

while CUT = 1 do

- set CUT = 0

for all k 2 [1::K] do

- set CUTk = 0 and i = 1

while CUTk = 0 and i <= SA do

if in Cut(Ck;Ai), no class is splitted into diferent parts then

- perform the partitionning

- compute the associated rules and update the hierarchy

- set CUTk = 1 and CUT = 1

else

- i = i + 1

end if

end while

end for

end while

***Output:*** *the hierarchy H, and the current partition*

O resultado da execução deste algoritmo é utilizado para diferenciar uma classe de outras, e é criada uma nova regra, que será utilizada no próximo passo, que é a classificação propriamente dita.

Na Tabela 1 podem ser visualizados os resultados obtidos pelo autor nas bases de dados definidas pelo INEX.

**Tabela 1 - Número de atributos gerados após rodar o algoritmo [02].**

| dataset | number of tags | number of parent-child relations | number of next-sibling relations | number of node positions | number of paths | total |
|---|---|---|---|---|---|---|
| inex-s | 150 | 1038 | 827 | 2475 | 3674 | 8164 |
| m-db-s-0 | 197 | 2172 | 419 | 6575 | 320 | 9683 |
| m-db-s-1 | 197 | 6477 | 5617 | 9159 | 16772 | 38222 |
| m-db-s-2 | 196 | 8953 | 7455 | 9183 | 25628 | 51415 |
| m-db-s-3 | 199 | 10639 | 9557 | 8537 | 37576 | 66508 |

Percebe-se claramente que o número de atributos descobertos supera largamente o número de tags (primeira coluna). As colunas subseqüentes indicam o número de relações pais-filho, relações de irmão, número de nós e número de caminhos das árvores criadas.

**Vantagens deste método:** a descoberta de novos atributos é o ponto positivo desse método, já que é possível aplicar este algoritmo a qualquer base de dados para obter novos parâmetros e criar novas regras a partir deles.

**Desvantagens:** Existe um limite muito tênue entre o número de atributos descobertos e a sua real utilidade, ou seja, a importância da informação que os atributos possuem. Neste método não é determinado como se identifica a relevância de determinados atributos, como aqueles obtidos na base m-db-s-3 da Tabela 1.

### 2.3 Método de Doucet e colaboradores

O método proposto por Doucet [04] assume que os documentos têm que ser clusterizados antes de serem classificados. Por hipótese, documentos relevantes tendem a aparecer no mesmo cluster. Os autores propõem uma mistura de uso de elementos estruturais e textuais para classificar um documento.

O método representa o documento transformando-o em vetores multidimensionais, e depois aplica o algoritmo k-means [10], no qual k identifica o número desejado de classes para classificação. Este número pode ser fornecido pelo usuário, como uma variável de entrada, ou determinado durante o processamento. Este algoritmo (também chamado de k-Médias) fornece

uma classificação de informações de acordo com os próprios dados. Esta classificação é baseada em análise e comparações entre os valores numéricos dos dados, sem a supervisão humana. O algoritmo não utiliza de pré-classificação.

Esse método utiliza uma análise estrutural em primeiro lugar, para depois analisar o documento textualmente. Devido ao fato do método não utilizar os documentos de definição de estruturas (DTD), ele é adequado para o uso de bases heterogêneas, tais como as usadas na proposta do INEX: Jornais do IEEE, e Wikipédia.

O algoritmo k-means básico está estruturado pelos seguintes passos:

- inicialização, onde são escolhidos os parâmetros iniciais;
- iteração, onde o arquivo textual é lido de forma sequencial e direcionado a um cluster. Esta tarefa é executada diversas vezes;
- finalização, quando a fase anterior permanece estável.

O algoritmo descarta as *stop-words*, ou palavras de parada, i.e. palavras consideradas irrelevantes, como as preposições “a”, “os”, etc. Após esse descarte, o algoritmo organiza as palavras remanescentes num vetor, combinando o texto não estruturado com o texto estruturado.

Os autores propõem uma modificação no algoritmo k-menas, misturando elementos estruturais e textuais, da seguinte forma:

- inicio, com a entrada do documento, do número de clusters desejados para classificação e um indicativo limite de similaridade, por exemplo, distância de edição;
- leitura das tags e descoberta de similaridade;
- leitura do textos e descoberta de atributos;
- combinação dos elementos textuais com estruturais;
- classificação

Devido ao fato de que cada documento é categorizado dependendo da sua relevância, os autores definiram um *ranking* de precisão para os documentos classificados erroneamente na etapa de teste. Este *ranking* é baseado nas seguintes expressões:

$$Precision = \frac{TP}{TP + FN}, \; Recall = \frac{TP}{TP + TN}.$$

No qual TP (true positive) indica os documentos colocados na categoria certa, FP (false positive) indica os documentos classificados erroneamente e FN (false negative) indica os documentos excluídos erroneamente de uma determinada categoria.

Usando esses dois parâmetros os autores definem a métrica de comparação na classificação com base na seguinte expressão:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}.$$

Os resultados obtidos utilizando este método de classificação nas bases do IEEE e da Wikipédia estão descritos nas Tabelas 2 e 3, respectivamente.

**Tabela 2 - Resultados para a base IEEE - [04].**

| Features | Micro F1 | Macro F1 | Overall rank (out of 13) |
|---|---|---|---|
| Text | .348789 | .290407 | 3rd |
| Tags | .132453 | .081541 | 7th |
| Text+Tags | .253369 | .203533 | 5th |
| Tags→Text, 0.8 | .270350 | .222365 | 4th |
| Text + T/E | .348828 | .290379 | 2nd |

**Tabela 3 - Resultados para a base Wikipédia [04].**

| Features | Micro F1 | Macro F1 | Overall rank (out of 7) |
|---|---|---|---|
| Text | .444455 | .210621 | 1st |
| Tags | .221829 | .072834 | 6th |
| Text+Tags | .372376 | .128239 | 5th |
| Tags→Text, 0.8 | .406129 | .155034 | 4th |
| Tags→Text, 0.9 | .413439 | .159473 | 3rd |
| Text + T/E | .427438 | .183567 | 2nd |

Na primeira coluna temos os atributos usados, se texto, tags, texto + tags ou texto + T/E. O T/E é um novo atributo definido pelos autores que indica a relação Texto / Estrutura. Na última coluna estão descritos o *ranking* global no INEX. A diferença entre o Micro F1 e o Macro F2 é que no Micro é utilizado um fator de peso referente a quantos documentos existem em cada classe e no Macro este fator não é utilizado.

**Vantagens deste método:** a possibilidade de não depender de elementos estruturais, tais como DTD (documento de estrutura de um arquivo XML), permite aplicar o algoritmo a bases muito heterogêneas, fato comum nos conjuntos atuais de documentos disponíveis na Web.

**Desvantagens:** A utilização de parâmetros textuais traz, no entanto, outras dificuldades, tais como menor desempenho na execução do algoritmo, assim como um índice de confiabilidade mais restrito. Deve-se lembrar que textos diferentes em bases distintas podem ter uma relação semântica muito estreita. Sem uma avaliação estrutural profunda, muitos erros podem ser cometidos.

Os autores declaram ter obtido os melhores resultados quando comparados com outras equipes. No entanto, verifica-se que fugiram bastante da regra inicial, que era dar uma maior atenção aos elementos estruturais e não textuais. Ou seja, o método praticamente ignorou a informação estrutural hierárquica de um documento XML. Outro item a ser levado em consideração é que este trabalho não apresentou o resultado final da classificação e sim da clusterização.

### 2.4 Método XRules

Este modelo [05] e [06], é baseado em regras pré-determinadas, aplicadas a um conjunto de dados conhecidos. Estas regras são usadas para fazer deduções ou determinar escolhas. Basicamente, um sistema baseado neste modelo possui quatro componentes básicos:

- Uma lista de regras, provenientes de uma base de conhecimento;
- Um executor que toma uma determinada ação dependendo do dado de entrada e da base de regras pré-estabelecidas;
- Uma memória temporária de trabalho;
- Uma interface com o meio exterior.

De acordo com [06], este método pode ser aplicado não somente para documentos XML, mas para quaisquer tipos de dados estruturados,

A maneira tradicional, ou seja, a classificação de texto, também é utilizada neste algoritmo, mas com a diferença de que em primeiro lugar é aplicada a regra estrutural. Este algoritmo identifica os padrões das estruturas internas da base de dados, criando classes que serão utilizadas depois para criar as regras de classificação.

O método XRules considera que todos os documentos XML são formados por árvores e sub-árvores, e mesmo aqueles que não o são, podem ser convertidos utilizando uma metodologia chamada de "*Node Splitting*."

O algoritmo tenta encontrar similaridades dentro da árvore, como a ilustrada na Figura2. Percebe-se claramente que existe uma subárvore frequente nas três estruturas (T1, T2 e T3).

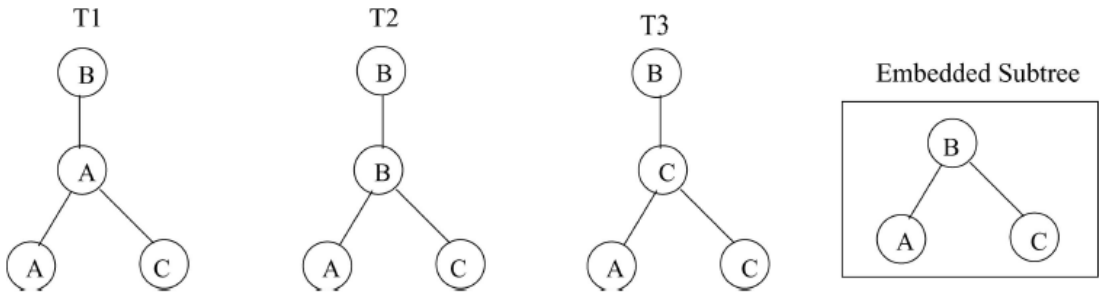


Figura 2 – árvores freqüentes [06].

O objetivo deste método está em encontrar ocorrências de subárvores dentro da árvore principal, e está definido pela expressão matemática que identifica a frequência de T em D, na qual D é o *dataset,* i.e. o conjunto de dados, e $\pi^{A}(T, D)$ é chamado de fator absoluto de

suporte e indica o número de árvores em D que contém ao menos uma ocorrência em T, i.e. a frequência de ocorrências.

Com isso, consegue-se calcular o fator de suporte relativo de ocorrências por meio da expressão:

$$\pi(T, D) = \frac{\pi^A(T, D)}{|D|}$$

Quaisquer dos dois fatores podem ser utilizados na implementação do algoritmo, a depender da heterogeneidade do domínio a ser classificado.

A base de dados a ser usada para treinamento consiste em um conjunto de estruturas, cada uma associada a uma classe em particular. Essencialmente, esta base é um documento XML em forma de árvore com N componentes, onde cada uma dessas árvores na floresta representa uma variável também denominada classe.

O objetivo desta classificação é obter um modelo de aprendizado que possa predizer com certo grau de segurança essas mesmas ocorrências em outros documentos com estrutura desconhecida.

O que importa em termos de custo é a relação entre as predições corretas e o número de predições encontradas pelo algoritmo. Esta relação está dada pelo fator de exatidão representado a seguir:

$$\alpha(\mathcal{R}, \mathcal{D}) = \frac{\eta(\mathcal{D})}{|\mathcal{D}|}.$$

Na qual R é o modelo, D é a base de dados e n é o número de predições corretas feitas na base de teste D. Este fator às vezes não é adequado já que a probabilidade da ocorrência de determinadas classes é superior a outras, isto levou os autores a usar a média ponderada dos fatores de exatidão, dada por:

$$\alpha^{cs}(\mathcal{R}, \mathcal{D}) = \sum_{i=1}^{k} (w_i \times \alpha(\mathcal{R}, \mathcal{D}_i))$$

Os autores basearam o seu método de classificação em quatro formas diferentes de custo e exatidão, que irão definir o *wi* :

- Modelo Proporcional: no qual o peso utilizado para aferir o custo de classificação é proporcional à probabilidade de aparição das classes no arquivo. Normalmente utilizado em bases heterogêneas, nas quais a possibilidade de aparição das classes no arquivo pode ser muito diferente de uma base para outra.
- Modelo Igualitário: todas as classes tem a mesma probabilidade de aparecer. É adequado para usar numa base de dados homogênea.
- Modelo Inverso: os pesos são inversamente proporcionais à sua probabilidade dentro da base de dados. Funciona bem para classificações binárias naquelas bases de dados que tem muitas classes com poucas possibilidades de aparecer.
- Modelo Customizado, no qual o peso é escolhido previamente de acordo com as necessidades do usuário. Deve ser utilizado em base de dados cujo conteúdo seja bastante conhecido, e, portanto, este parâmetro é escolhido com base em informações existentes.

**Visão estatística imposta ao modelo**

Além da frequência e da exatidão discutidas anteriormente, o algoritmo impõe uma visão probabilística para cada modelo. Cada classe é classificada num intervalo de confiança, inclusive comparando cada uma com o grupo de classes ainda não classificado. Desta forma, tenta-se descobrir uma nova regra de similaridade. O algoritmo rejeita as regras que não se aplicam a determinado intervalo de confiança pré-definido.

Os pesquisadores propõem a Inferência Bayesiana [11] para tratar as incertezas. A Inferência Bayesiana é um modelo baseado no teorema de Bayes, que descreve as incertezas sobre quantidades invisíveis de forma probabilísticas. Estas incertezas são modificadas periodicamente após observações de novos dados ou resultados. Isso é chamado de operação Bayesiana. O teorema mostra a relação entre uma probabilidade condicional e a sua inversa. Por exemplo, a probabilidade de uma hipótese dada a observação de uma evidencia e a probabilidade da evidência dada pela hipótese. A seguir é mostrada a expressão proveniente do teorema de Bayes:

$$\Pr(A|B) = \frac{\Pr(B|A)\,\Pr(A)}{\Pr(B)}$$

Pr(A) e Pr(B) são as probabilidades *a priori* de A e B. PR(B\A) e Pr(A\B) são as possibilidades *a posteriori* de B condicional de A e A condicional de B, respectivamente. Ou seja, este teorema mostra como alterar as probabilidades *a priori* utilizando as novas evidencias de forma a obter probabilidades *a posteriori*.

A implementação do método XRules consiste em duas fases: a fase de treinamento e a fase de testes:

(i) Na fase de treinamento mineram-se as estruturas de cada classe até adquirir os parâmetros que satisfaçam os níveis definidos pelo usuário, de acordo com os parâmetros estatísticos discutidos previamente. Isto é feito utilizando uma base de estruturas com classes conhecidas a partir das quais é construído o modelo de classificação, chamado de conjunto de regras ("*rule set*"). O objetivo é usar este modelo de classificação para inferir as classes. As regras são classificadas de acordo com a sua força e precedência descartando aquelas que não atingem certo nível de predição. Depois da fase de treinamento, tem-se um conjunto de regras validadas e parametrizadas, não necessariamente atingindo todas as situações, mas em número suficiente para que outras regras possam ser inferidas.

(ii) Em seguida, ocorre a fase de testes, na qual cada regra é testada e validada de acordo com a força preditiva, combinando a probabilidade estatística para inferir novas regras. Na fase de testes, é utilizada uma base de dados desconhecida ou mesmo conhecida, mas que não tenha sido usada na fase do treinamento.

**Vantagens deste método:** este método depende unicamente de elementos estruturais, ignorando a parte textual. Portanto, ele pode ser aplicado com eficiência a uma base de dados de grande tamanho, sem degradação no tempo de processamento.

**Desvantagens:** a utilização de parâmetros estruturais unicamente pode criar regras incoerentes de classificação, que poderiam ser validadas ou testadas relacionando a estrutura com o aparecimento de padrões freqüentes nos elementos textuais.

## 3 CONCLUSÃO

Todos os métodos e algoritmos discutidos representam o estado da arte em matéria de classificação de documentos semi-estruturados. A iniciativa do INEX 2005-2006 conseguiu aglutinar diversas e consagradas instituições acadêmicas ao redor do mundo, que se traduziu em trabalhos que ainda hoje são referencias para novas pesquisas.

Os resultados de cada método utilizado são avaliados fundamentalmente em função da exatidão apresentada, e não no tempo de processamento de cada tarefa. Ou seja, não foi avaliado o desempenho dos algoritmos e/ou métodos. O objetivo do Mining Track não era realizar um "*benchmark*", e sim criar um fórum investigativo para classificação e clusterização de documentos XML. Os participantes tiveram acesso às bases de dados antes de apresentar o trabalho.

Sugere-se que trabalhos futuros devem ser realizados com critérios de desempenho e exatidão pré-definidos. As bases de dados devem ser desconhecidas dos participantes. Desta forma poderá ser avaliada a real vantagem de um método sobre o outro. Na Tabela 4 pode-se observar um quadro comparativo entre as diferentes metodologias utilizadas.

| Método | Estrutura Dependente | Bases Grandes | Bases Heterogêneas | Base Usada | Máxima Exatidão alcançada |
|---|---|---|---|---|---|
| Garboni | SIM | SIM | SIM | IEEE | 95% |
| Candilier | SIM | SIM | NÃO | MovieDB | 96.8% |
| Ducet | NÃO | SIM | SIM | IEEE | - |
| Xrules | SIM | SIM | SIM | IEEE | 96% |

Tabela 4 - Comparação entre os métodos.

O método XRules é o que desponta como o mais adequado para classificação de documentos XML em qualquer base de dados. O método de Candilier, cujo resultado está na Tabela 4, alcançou uma exatidão maior, mas deve-se ter em conta que ele foi utilizado na base MovieDB, uma base mais conhecida, homogênea e estruturada. Portanto, o maior valor percentual alcançado não é indicativo de uma melhor eficiência na classificação.

O XRules combina classificação estrutural com textual, e, o que é mais importante, utiliza um método de auto-aprendizagem baseado em regras de inteligência artificial e modelos probabilísticos.

Trabalhos futuros poderão ser realizados no sentido de aprofundar esta comparação. Uma das tarefas propostas é estudar com maiores detalhes quais os elementos estruturais mais relevantes para classificação numa base de dados heterogênea. O tratamento estatístico usado pelo XRules poderia ser aplicado a outros métodos no sentido de melhorar a exatidão das pesquisas.

## REFERÊNCIAS

[01] Garboni, Calin, Masseglia, Florent. **Sequential Pattern Mining for Structure-Based XML Document Classification.** West University of Timisoara, Romania. (2006).

[02] Candillier, L, Tellier, I. **Transforming XML trees for efficient classification and Clustering.** INEX (2005).

[03] Denoyer, Ludovic, Gallinari, Patrick. **Report on the XML Mining Track at INEX 2005 and INEX 2006 - Categorization and Clustering of XML** – University of Paris 6.

[04] Doucet, A., Lehtonen, M.: **Unsupervised classification of text-centric XML document collections**. INEX Retrieval (2006).

[05] Rios, Eneida. Britto, Roberta. Diaz, Williams. **Métodos de Classificação para Mineração de Dados – Uma visão do XRules**. Unifacs (2011).

[06] Zaki, Mohammed J., Aggarwal, Charu C. **XRules: An Effective Algorithm for Structural Classifier for XML Data.** Mach Learn (2006) Pág. 137-170.

[07] http://www.netcraft.com – **Pesquisas na Web** – disponível em 01/08/2011 as 20:30 hs.

[08] http://www.w3schools.com/xml – **Consórcio W3C** – disponível em 04/08/2011 as 22:45 hs.

[09] Da Silva, Maria Estela Vieira da, **XSimilarity: Uma ferramenta para consultas por similaridade embutidas na linguagem XQuery** Trabalho de graduação, UFRGS (2007)

[10] http://imasters.com.br/artigo/4709/sql_server/data_mining_na_pratica_algoritmo_k-means/ : **algoritmo k-means na prática** – disponível em 05/08/2011 as :17:15

[11] Neville, J. Jensen, **Simple estimators for relational bayesian classifers.** International Conference on Data Mining (2003)