An XML-based Approach for Data Preprocessing of Multi-Label Classification Problems

Eduardo Corrêa Gonçalves, Vanessa Braganholo
Universidade Federal Fluminense (UFF) – Brazil

XML London 2014, July 7-8, University College London
Outline

- Introduction
- Multi-Label Classification
- ARFF versus XML
- XML-based Preprocessing of the IMDb Dataset
  - The IMDb dataset
  - A Study on the Words
  - Data Transformation
- Conclusions and Future Work
**Classification**

- Active topic of research in the fields of A.I. and Data Mining.
- Task of automatically assigning objects to **discrete classes** (known as “labels”) based on the features of the objects.
  - I.e.: predicting the category(ies) to which an object belongs.

**Example**: Spam detection

*object*: message

*label*: spam
Introduction (2/4)

**Single-Label Classification (SLC)**

- Object must be associated to **one and only one** class label.

- **Spam detection** – an incoming e-mail either belongs to the class “**spam**” or to the class “**normal**”.

- **Loan risk prediction** – a loan applicant can be classified as “**low**”, “**medium**” or “**high**” credit risk.

**Multi-Label Classification (MLC)**

- Objects can be assigned to **various labels**.

- **Text categorization** – A news article about the 2014 Football World Cup can be classified as “**Sports**”, “**Politics**” and “**Brazil**”.
Problem Statement

- It is well-known that a large (perhaps the largest) part of the available data in the world takes the form of free text on the Web.

**User Reviews**

**Meryl Streep Italian Style**
29 August 2005 | by [REVIEWER](Rome, Italy) – See all my reviews

Meryl Streep is absolutely astonishing. I forgot it was her ten seconds into the film. That opening breakfast scene where all of her story is written in her magnificent face. As an Italian I know there is no acting involved here. She IS Italian. She reminded me of Anna Magnani in "Bellissima" there is not a single false note. Clint Eastwood, clearly, dedicates the film to her and the results are pure magic. The film is based on an unreadable book- at least I couldn't get through it, in spite of the brevity of the volume - the film however, is bound to become a classic thanks to the powerful chemistry of the stars.

- There has been a increasing interest in the application of classification techniques to these data!
  
  - E.g.: **sentiment analysis**.

- **PROBLEM**: text data are tend to be more difficult to clean and **transform** (highly susceptible to **noisy**)

- **CONSEQUENCE**: low quality data → low quality classification.

Our proposal:

- The use of an XML-based approach for **data preprocessing** in **multi-label classification** of **text documents**.
**Goal:** demonstrate that XML facilitates the major steps involved in preprocessing.

**Classification task:** associate movie summaries to genres.

**Data:** IMDb (Internet Movie Database - www.imdb.com)
Multi-label Classification (1/5)

- Recently, several modern applications of MLC have emerged:
  - **Scene Classification:**
  - **Music into Emotions:**
  - **Functional Genomics:** predicting functional classes of genes and proteins
  - **Text Classification:** documents into topics (ex: sports, ecology, religion, ...)

- **Scene Classification:** mountains + trees

- **Music into Emotions:**

- **Functional Genomics:** predicting functional classes of genes and proteins

- **Text Classification:** documents into topics (ex: sports, ecology, religion, ...)

Multi-label Classification (2/5)

- **How to build a multi-label classifier (1/2)?**
  - MLC algorithms need to **learn** from a set objects whose classes are known:
    - The **training dataset**.

- **Example:**
  - **MLC task**: associating movies to genres according to their summaries.
    - Four possible genres: “drama”, “romance”, “horror”, “action”.

- Training dataset

<table>
<thead>
<tr>
<th>Text Id</th>
<th>Feature Vector (words of the movie summary)</th>
<th>Drama</th>
<th>Romance</th>
<th>Horror</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$x_1$</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$x_2$</td>
<td></td>
<td></td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>3</td>
<td>$x_3$</td>
<td></td>
<td></td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$x_4$</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>$x_5$</td>
<td>●</td>
<td>●</td>
<td></td>
<td>●</td>
</tr>
</tbody>
</table>
Multi-label Classification (3/5)

- **How to build a multi-label classifier (1/2)?**
  - From the training set, the MLC algorithm learns a **classifier**.

- **Classifier**: function that receives the features of a new object as input and outputs its predicted label set

\[ h : X \rightarrow \{0,1\}^q \]

where \( q \) = number of labels
Several distinct techniques have been developed for building classifiers:

- k-Nearest Neighbours (k-NN).
- Decision trees.
- Probabilistic classifiers.
- Neural networks.
- Support vector machines.

They are based on different mathematical principles for addressing the classification task.

In the next slide we give an example of classification with the k-NN technique.
Example: k-Nearest Neighbours.

- A new object \( x \) is classified based on the \( k \) objects in the training set which are more similar to it.

Example: new object = “The Lunchbox” \( k=3 \)

- Neighbour\(_1\) – Slumdog Millionaire (class labels = Action, Romance, Drama)
- Neighbour\(_2\) – Midnight in Paris (class labels = Romance, Fantasy, Comedy)
- Neighbour\(_3\) – The Bridges of Madison County (class labels = Romance, Drama)
  - **The Lunchbox** is assigned the labels Romance and Drama
ARFF versus XML (1/7)

- Most classification tools work with training data either structured in:
  - Relational tables; or
  - Flat-files (one record per line).

---

Re: Text-classification: Data from XML and multiple keywords

« Reply #1 on: November 29, 2010, 01:01:29 PM »

Hi,

I just installed [redacted] and wanted to start working on it with some test data that is also in XML. I also haven't found a way to import my files to it. Is it really not possible to do so?

---

Re: Text-classification: Data from XML and multiple keywords

« Reply #2 on: November 29, 2010, 09:31:35 PM »

Generally speaking, [redacted] works with flat-file data. The same as almost all other statistical software.

XML is hierarchical by nature, so it is hard to say how this would work.

You could try reading in the file as HTML and using XPATH to get the attribute values, but it is probably easiest to convert to CSV or Excel first.
The ARFF format

- Flat-file format
- Popularly used in the data mining field

```
@relation loan_risk_prediction

@attribute age numeric
@attribute gender {F, M}
@attribute marital_status {SINGLE, MARRIED, DIVORCED, WIDOWED}
@attribute monthly_income numeric
@attribute risk {LOW, MEDIUM, HIGH}

@data
18,M,SINGLE,550.00,HIGH
38,F,MARRIED,1700.00,LOW
23,M,MARRIED,1300.00,MEDIUM
32,M,DIVORCED,2500.00,LOW
19,M,SINGLE,900.00,HIGH
68,F,WIDOWED,2200.00,MEDIUM
34,M,MARRIED,1350.00,MEDIUM
32,F,MARRIED,1400.00,LOW
20,F,MARRIED,1100.00,HIGH
20,M,DIVORCED,2100.00,LOW
```
### ARFF versus XML (3/7)

**The ARFF format**

- Flat-file format
- Popularly used in the data mining field

#### Header section

```xml
@relation loan_risk_prediction

@attribute age numeric
@attribute gender {F, M}
@attribute marital_status {SINGLE, MARRIED, DIVORCED, WIDOWED}
@attribute monthly_income numeric
@attribute risk {LOW, MEDIUM, HIGH}
```

#### Data section

```
@data
18,M,SINGLE,550.00,HIGH
38,F,MARRIED,1700.00,LOW
23,M,MARRIED,1300.00,MEDIUM
32,M,DIVORCED,2500.00,LOW
19,M,SINGLE,900.00,HIGH
68,F,WIDOWED,2200.00,MEDIUM
34,M,MARRIED,1350.00,MEDIUM
32,F,MARRIED,1400.00,LOW
20,F,MARRIED,1100.00,HIGH
20,M,DIVORCED,2100.00,LOW
```
ARFF versus XML (4/7)

- **The ARFF format**
  - Simple and intuitive.
  - Sufficient for several classification tasks... as long as they involve:
    - Relational data ("one record per line").
    - Conventional attributes ("age", "salary", "marital status", ...).
  - However ARFF is **not suitable** for **text classification**... this is because:
    - We normally have to deal with multiple labels.
    - We need to deal with a "less conventional" attribute:
      - The **words** that appear documents!
Remembering our classification task:
- Prediction of movie genres in function of their summaries.

The Bridges of Madison County (1995)
- Drama | Romance
- 135 min
- 15 September 1995 (UK)
- Ratings: 7.5/10 from 44,390 users
- Metascore: 66/100
- Reviews: 182 user | 68 critic | 22 from Metacritic.com

Storyline
The path of Francesca Johnson's future seems destined when an unexpected fork in the road causes her to question everything she had come to expect from life. While her husband and children are away at the Illinois state fair in the summer of 1965, Robert Kincaid happens turn into the Johnson farm and asks Francesca for directions to Roseman Bridge. Francesca later learns that he was in Iowa on assignment from National Geographic magazine. She is reluctant seeing that he's a complete stranger and then she agrees to show him to the bridges and gradually she talks about her life from being a war-bride from Italy which sets the pace for this bittersweet and all-too-brief romance of her life. Through the pain of separation from her secret love and the stark isolation she feels as the details of her life consume her, she writes her thoughts of the four-day love affair which took up three journals. The journals are found by her children after the lawyer was going over Francesca's will and ... Written by Mark
Example of ARFF file for movie genres classification.

```arff
@relation movies

@attribute a {0,1}
@attribute abandon {0,1}
@attribute about {0,1}
...
@attribute zero {0,1}
@attribute zoology {0,1}
@attribute genre_action {0,1}
@attribute genre_comedy {0,1}
@attribute genre_drama {0,1}
...
@attribute genre_romance {0,1}

@data
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,...
1,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,...
0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,...
...
```

Problems:

- Each word must be declared as a binary attribute in the header (bag of words)
  - IMDb: \( \approx 190,000 \) words
    \( \approx 154,000 \) movies
  
- Cumbersome to query, explore and transform.

- Highly sparse.

- Does not support the specification of multi-valued attributes:
  - Movies with multiple genres or plots.
So… Why not to use XML?

- Text represented in a natural way.
- Easy to query, explore and transform:
  - SAX
  - XQuery
  - XSLT
- Definition of multi-valued attributes is straightforward (movies with multiple plots and genres).
**Experiment (1/10)**

- **Goal:**
  - Transform the original IMDb data* (plain text files) into a XML database.
  - Study and preprocess this database.
    - As a result, we will obtain a dataset, ready to be mined.
      - high quality data → high quality classification.

*The IMDb plain text files can be download: [www.imdb.com/interfaces](http://www.imdb.com/interfaces)*
Experiment (2/10)

- **Step 1** – Generation of the “raw” XML dataset

  - **plot.list**: 256,486 movies, 3.88M lines
  - **genres.list**: 778,676 movies, 1.33M lines

  - Merging of the two plain IMDb files into a single XML dataset.
  - **Result**: XML file containing 153,499 movies.
Step 1 – Generation of the “raw” XML dataset

- Nice file!!!
  - But **not yet ready** to be mined!

- The reasons are presented in the next slides
  - *Let’s go to the Step 2 of the experiment.*
Step 2 – Preprocessing

Two sub-steps:

1. STUDY:
   - The XQuery Language and the SAX API were used to querying and exploring the XML dataset.

2. TRANSFORMATION:
   - According to the results of the study, we clean and transform the XML dataset.
**Step 2.1 – Preprocessing / Study**

- XQuery was used to generate frequency tables

```xml
(freq_genres
 for $u in distinct-values(doc("imdb.xml")//movie/class)
 let $b := doc("imdb.xml")//movie[class=$u]
 return
 <row>
   <genre>{$u}</genre>
   <count>{count($b)}</count>
 </row>
)
```

```xml
(freq_genres
 <row>
   <genre>Drama</genre>
   <count>59177</count>
 </row>
 <row>
   <genre>Action</genre>
   <count>14416</count>
 </row>
 <row>
   <genre>Comedy</genre>
   <count>38373</count>
 </row>
 <row>
   <genre>Crime</genre>
   <count>10875</count>
 </row>
 <row>
   <genre>Adult</genre>
   <count>1625</count>
 </row>
 <row>
   <genre>Adventure</genre>
   <count>9596</count>
 </row>
 ... 
)
```
## Experiment (6/10)

- **Step 2.1 – Preprocessing / Study**
  - SAX was used to perform a study on the words.
  - Some results:

<table>
<thead>
<tr>
<th>Description</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of words</td>
<td>16,305,677</td>
</tr>
<tr>
<td>Number of distinct words</td>
<td>187.718</td>
</tr>
<tr>
<td>About half of the words occur only once</td>
<td>“agnosticism”, “polyvision”</td>
</tr>
<tr>
<td>Several misspelled words and typos</td>
<td>“marjuana”, “caracters”, “theforce”, ...</td>
</tr>
<tr>
<td>Synonyms, multiple languages</td>
<td>“Brazil” (741), “Brasil” (49), ...</td>
</tr>
</tbody>
</table>
Experiment (7/10)

- **Step 2.2 – Preprocessing / Transformations**
  - From the results of our study we could do:
    - **Data reduction**:
      - Words that appeared only once were removed.
      - Removal of stop words (*details soon*)
      - Stemming (*details soon*)
    - It would also be possible to perform **data cleaning**
      - E.g: correction of typos.
Step 2.2 – Preprocessing – Transformations

Stop Words.

- Words that tend to be very frequent, but do not help on discriminating the movie genres.
  - articles, prepositions, adverbs, ...
  - E.g.: “the” occurs in 100% of the movies...
  - On the IMDb domain, there are also specific words that can be regarded as useless: “movie”, “film”, the proper names.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<stopwords>
  <stopword>the</stopword>
  <stopword>and</stopword>
  <stopword>to</stopword>
  <stopword>mr</stopword>
  <stopword>that</stopword>
  <stopword>from</stopword>
  <stopword>movie</stopword>
  ...
</stopwords>
```
Step 2.2 – Preprocessing – Transformations

Stemming

- The process of conflating the variant forms of a word into a compact representation: the stem.

- Intuition: morphological variants of words typically have similar interpretations and can be considered as equivalent for the purpose of data mining analysis.

- Example:
  - The words “educate”, “educational”, “education” and “educating” could all be reduced to the stem “educ”.

- In this work we used the Porter Algorithm* (JAVA implementation).

*The specification of the Porter Algorithm can be found at: http://tartarus.org/martin/PorterStemmer/
Experiment (10/10)

- **Summary**

Original XML Dataset

Raw Data

```
XML
```

187,817 words

Transformed XML Dataset (prepared to be mined)

Preprocessed Data

```
XML
```

79,753 stems
Conclusions

- XML facilitates the major steps involved in data preprocessing of text data.
- With the use of the SAX and XQuery, we could easily:
  - Querying, exploring and transforming the IMDb dataset.
Future Work (1/2)

- Define the final format of the preprocessed XML dataset.
- Develop an algorithm to directly mining this dataset.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<imdb>
  <movie id=1>
    <term>
      <stem>comput</stem>
      <weigth>0.8730</weigth>
    </term>
    <term>
      <stem>hyper</stem>
      <weigth>0.3020</weigth>
    </term>
    ...
    <class>drama</class>
    <class>suspense</class>
  </movie>
  ...
</imdb>
```
Evaluating the feasibility of developing an XSLT version of the Porter Stemming Algorithm.

- This algorithm relies on the idea that the suffixes in English language are mostly made up of a combination of smaller and simpler suffixes.

- It works in 5 steps:
  - Within each step the word is tested against a few set of suffix transformation rules.
    - If a test results in TRUE, the word suffix is removed or transformed; The control moves to the next step.
    - Otherwise, the next rule in the step is tested.