Fuzzy Clustering for TV Program Classification

Yu Zhiwen
College of Computer
Northwestern Polytechnical University
Xi'an, P.R.China, 710072
yuzhiwen77@yahoo.com.cn

Zhou Xingshe
College of Computer
Northwestern Polytechnical University
Xi'an, P.R.China, 710072
zhouxs@nwpu.edu.cn

Gu Jianhua
College of Computer
Northwestern Polytechnical University
Xi'an, P.R.China, 710072
gujh@nwpu.edu.cn

Yang Zhiyi
College of Computer
Northwestern Polytechnical University
Xi'an, P.R.China, 710072
yangzy@nwpu.edu.cn

Abstract*

In order to achieve TV program group recommendation, an approach based on fuzzy clustering is proposed for program classification in this paper. This paper firstly describes the XML based program description metadata representation, in which both textual and symbolic information is included. Secondly it presents the program feature extraction and presentation method. A program is defined as two vectors, one is based on term statistics implying what the program is about, and the other reflects broadcasting characteristics of the program. Then the classifying approach based on fuzzy clustering is proposed. The approach goes: normalizing original data, building fuzzy similarity matrix, and then clustering. The final fuzzy similarity matrix is constructed by combining two fuzzy similarity matrices calculated from two different aspects.

1. Introduction

With the rapid growth of DTV (Digital Television) technologies, there exists an overabundance of programs available from which each consumer can choose. This precipitates a need for new technologies to provide consumers access to what they want, when they want it, and how they want it. To meet this new requirement, the TV-Anytime Forum has defined specifications that will enable applications to exploit local persistent storage in consumer electronics platforms [1].

Although there are many existing TV-Anytime applications, such as Virtual Channel and EPG (Electronic Program Guide), the group recommendation is increasingly required by the consumer. Group recommendation means recommending a series of TV programs with something in common to TV viewer. For example, a user likes the program he is currently viewing and wants to see more programs like this one. TV-Anytime provides MemberOf elements for this usage scenario [2]. MemberOf means a group of which the program is a member, such as <MemberOf xsi:type="MemberOfType" crid="groupcrid"/>. With MemberOf and group CRID (Content Reference Identifier) [3], when a program is enjoyed by the user, the broadcaster can push other programs belonging to the group, of which the currently displaying program is a member, to the user. Further more, a group for an entire series would allow the PDR (Personal Digital Recorder) to acquire an entire series of programs by just selecting one CRID to acquire.

In order to achieve group recommendation, we should firstly classify the programs into groups, and assign a unique group CRID for it, and then add MemberOf element into the program metadata.

Clustering, which has been widely studied in data mining community, is used to partition a data set into clusters so that intra-cluster data are similar and inter-cluster data are dissimilar [4]. But in real applications there is very often no sharp boundary between clusters so that fuzzy clustering is often better suited for the data. Membership degrees between zero and one are used in fuzzy clustering instead of crisp assignments of the data to clusters [5]. Fuzzy clustering is increasingly been applied to different technological fields, such as data analysis, unsupervised learning, and image recognition, etc. The application of fuzzy cluster analysis to pure text partition is mature and successful. However, fuzzy clustering is rarely applied in semi-structured data classification, such as XML documents. In TV-Anytime, the program description metadata is represented with XML, which includes both textual and structured

* This work was supported by the Doctorate Foundation of Northwestern Polytechnical University of China.

information.

In this paper, we apply fuzzy clustering to TV program classification in a TV-Anytime environment. We firstly describe the XML based program description metadata representation. Secondly we present the program feature extraction and presentation method. Then we propose the method of using fuzzy clustering for program classification.

2. Program Metadata Representation

Generally, metadata is data about data, such as the title, genre, and language of a television program. In the context of TV-Anytime, each program has a metadata, which acts as program description. TV-Anytime uses the MPEG-7 Description Definition Language (DDL) to describe the metadata structure as well as the XML encoding of metadata. XML (eXtensible Markup Language) [6], which is developed by W3C, has emerged as a standard information exchange mechanism on the Internet. XML allows the encoding of structural information within documents.

The program description metadata is corresponding to ProgramInformationTable, the first part of content description metadata in TV-Anytime. It describes items of content, such as the title of the program, the genre it falls under, and a list of keywords that can be used to match a search [7]. For instance, a simple example of the program description metadata is shown in Figure 1.

```
<?xml version="1.0"?>
<ProgramInformationTable>
 <ProgramInformation>
  <BasicDescription>
<Title>Gone with the Wind</Title>
<Synopsis>
Scarlett's first marriage was for spite, and her second marriage was for money, but one man kept weaving in and out of Scarlett's life, the dashing captain Rhett Butler.
</Synopsis>
<Keyword>Love</Keyword>
<Keyword>Marriage</Keyword>
< Keyword>War</Keyword>
<Genre>Romance</Genre>
<ChannelNo>cctv6</ChannelNo>
<StartTime>2002-09-22T20:30:00+08:00</StartTime>
<Language>en</Language>
  </BasicDescription>
 </ProgramInformation>
</ProgramInformationTable>
```

Figure 1. Program description metadata example

In this example, the program is a movie whose name is "Gone with the Wind". The synopsis concludes a textual description of the program. The keywords are used to describe what the program is about. The TV-Anytime gives a genre dictionary [8], which defines the normative TV-Anytime set of genres. For example, in the 3.4 section of the genre dictionary, FICTION, there are all together 19 kinds of genres such as: General light drama, Soap, and Romance etc. In DTV, there are multiple channels simultaneously broadcast to consumer, the channel number is used to identify different channels. The start time of the content means the time when the content is broadcast. This factor is important because maybe you won't get up in the small hours of the day; say 2:30AM to watch a program even though you may show interest in it in the other time of the day. The start time string format is compliant with ISO 8601 [9]. The language of the content is also important to users. For example, news broadcast in English may have little or even no attraction to a little boy who does not know English at all, but it would be the desired content to a young college student who want to improve his/her English listening ability and get informed of the daily news at the same time.

3. Program Feature Extraction

We adopt the Vector Space Model (VSM) [10] as the feature extraction and object information presentation method. In the VSM paradigm, object information is presented as vectors.

In program metadata, there are some terms in Title, Synopsis and Keyword fields. We can gather these terms in all the program metadata by alphabet as a dictionary, and represent it as a vector.

$$D = (term_1, term_2, \ldots term_n) \quad (1)$$

To compute the dictionary vector, usually these steps are followed [11]. First the individual words occurring in the metadata are identified. Words that belong to the *stop list*, which is a list of high-frequency words with low content discriminating power, like "a", are deleted. Then a stemming routine is used to reduce each remaining word to word-stem form, that is, the remaining words are reduced to their stem by removing prefixes and suffixes. For instance the words "computer", "computers", "computing" and "computability" could all be reduced to "comput". This is used for decreasing redundancy.

For example, through above steps, we get a dictionary vector as follows:

D=(accomplic, anim, captain, climb, cup, danger, dash, enemi, football, fox, gone, jump, lake, life, love, man, marriag, match, monei, murder, music, photo, spite, spy, visit, war, wash, weav, wind, world)

With the dictionary vector, we can define a program as

two vectors P_a and P_b:

$$P_a = (t_1, t_2, \ldots t_n) \quad (2)$$

$$P_b = (G, C, S, L) \quad (3)$$

P_a is a vector with n (total number of terms in above dictionary) items, where t_i is the weight assigned to $term_i$ ($1 \le i \le n$) in the dictionary vector. The weight t_i is assigned complying with this rule: if $term_i$ is included in Title, Synopsis or Keyword field of the program's metadata, then t_i=1, otherwise t_i=0. For the film "Gone with the Wind", P_a=(0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0). P_a implies what the program is about.

P_b contains four items: G, C, S, and L, where G stands for genre of the program; C stands for channel number; S stands for start time of the program; L stands for Language. P_b reflects broadcasting characteristics of the program. The values in expression (3) can be gained as follows.

TV-Anytime genre dictionary defines totally 101 second-level genres. We number them from the first genre to the last genre with 1 to 101. So the G value of *i*-th genre can be assigned as *i*. For instance, since *Romance* is the fifty-fifth genre in the genre dictionary, so G(Romance) is 55.

We number the channels with positive integers. Relevant channels are numbered with closer integers. For instance, in CCTV, there are 12 channels, and then we can number cctv1 to cctv12 with 1 to 12; so C(cctv6) is 6.

We can divide the start time of the programs into several domains, and number them with positive integers. The time that people have more possibility to watch TV will be assigned larger integers. The partition and evaluation is as follows: 0:00-4:00: 1; 4:00-6:00: 2; 8:00-12:00: 3; 14:00-16:00: 4; 6:00-8:00: 5; 16:00-18:00: 6; 12:00-14:00:7; 22:00-24:00: 8; 18:00-22:00:9; So S(20:30) = 9.

There are 6703 categories of language in the world wide according to [12] in 1996. The languages can be ranking by population speaking. The top 5 is as follows: Chinese North, English, Spanish, Bengal, and Indian. We can assign L value of the language with its order in the ranking. So L(en) =2, en stands for English.

With above definition, we can get P_b of the film "Gone with the Wind", P_b=(55, 6, 9, 2).

4. Program Classification Using Fuzzy Clustering

To illustrate the fuzzy clustering process, we take five synthesized program examples, whose feature vectors are supposed as follows.

Program 1("Gone with the Wind"): P_{1a}=(0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0), P_{1b}=(55, 6, 9, 2).

Program 2: P_{2a}= (1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0), P_{2b}=(59, 6, 5, 2).

Program 3: P_{3a}= (0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0), P_{3b}=(20, 2, 7, 1).

Program 4: P_{4a}= (0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0), P_{4b}=(85, 8, 3, 9).

Program 5: P_{5a}= (0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1), P_{5b}=(11, 5, 6, 1).

4.1 Original Data Normalization

Since data in P_b is not obtained with universal measurement, we should do normalization for it.

First, calculate the average value and standard deviation of each factor in P_b.

$$\bar{x}_j = \frac{1}{5} \sum_{i=1}^{5} x_{ij} \quad (4)$$

$$S_j = \sqrt{\frac{1}{5} \sum_{i=1}^{5} (x_{ij} - \bar{x}_j)^2} \quad (5)$$

Then, the original feature value can be *normalized* to its normalized value.

$$x'_{ij} = \frac{x_{ij} - \bar{x}_j}{S_j} \quad (6)$$

Such normalized value is not always on a standard universe, say [0, 1]. In order to compress the normalized value into [0, 1], we need to adopt extremum normalization equation to accomplish it.

$$x''_{ij} = \frac{x'_{ij} - x'_{j\min}}{x'_{j\max} - x'_{j\min}} \quad (7)$$

where

$$x'_{j\max} = \max(x'_{1j}, x'_{2j}, ..., x'_{5j}) \quad (8)$$

$$x'_{j\min} = \min(x'_{1j}, x'_{2j}, ..., x'_{5j}) \quad (9)$$

Through above processing, we get normalized value of P_b in table 1.

Table 1: Normalized value of P_b

Program ID	G	C	S	L
1	0.5946	0.6667	1	0.1250
2	0.6486	0.6667	0.3333	0.1250
3	0.1216	0	0.6667	0
4	1	1	0	1
5	0	0.5	0.5	0

4.2 Building Fuzzy Similarity Matrix

In fuzzy similarity matrix, r_{ij} is the similarity between object i and j. A commonly used similarity metric is the cosine of the angle between two objects.

$$r_{ij} = \frac{\sum_{k=1}^{m} x_{ki} x_{kj}}{\sqrt{(\sum_{k=1}^{m} x_{ki}^2)(\sum_{k=1}^{m} x_{kj}^2)}} \quad (10)$$

We can get two fuzzy similarity matrices, one is R_a, which is calculated from P_a, and the other is R_b, which is calculated from P_b. Since the two matrices are symmetric, we just give the lower triangle part of each matrix. When calculating R_b, we use the normalized data in Table 1.

$$R_a = \begin{pmatrix} 1 & & & & \\ 0.3478 & 1 & & & \\ 0.6731 & 0.3015 & 1 & & \\ 0.6731 & 0.4020 & 0.4444 & 1 & \\ 0.4348 & 0.1816 & 0.3015 & 0.4020 & 1 \end{pmatrix}$$

$$R_b = \begin{pmatrix} 1 & & & & \\ 0.8791 & 1 & & & \\ 0.8097 & 0.4461 & 1 & & \\ 0.5943 & 0.8349 & 0.1036 & 1 & \\ 0.8751 & 0.7100 & 0.6955 & 0.4082 & 1 \end{pmatrix}$$

With these two matrices, we can construct our final fuzzy similarity matrix R according the following equations.

$$R = W_a \times R_a + W_b \times R_b \quad (11)$$

$$W_a + W_b = 1 \quad (0 \le W_a \le 1, 0 \le W_b \le 1) \quad (12)$$

where W_a means the weight of R_a, and W_b means the weight of R_b.

The final fuzzy similarity matrix can be calculated as (Supposing W_a=0.6, W_b=0.4):

$$R = \begin{pmatrix} 1 & & & & \\ 0.5603 & 1 & & & \\ 0.7277 & 0.3593 & 1 & & \\ 0.6416 & 0.5742 & 0.3081 & 1 & \\ 0.6109 & 0.3931 & 0.4591 & 0.4045 & 1 \end{pmatrix}$$

4.3 Clustering

Although the fuzzy similarity matrix has reflexivity and symmetry, it has not transitivity. If a relation has reflexivity, symmetry, and transitivity, we define it as an equivalence relation. Only when the relation is an equivalence relation, set *X* can be classified according to it. Using equivalence relation for classification, an element belongs to one and only one cluster.

The fuzzy equivalent matrix $R^{(n)}$ can be calculated by using Warshall algorithm [13], which is outlined in the following steps:

(0) set $R^* = [r_{ij}^*]$, $R^* = R$;

(1) $k = 1$;

(2) $r_{ij}^* = r_{ij}^* \vee (r_{ik}^* \wedge r_{kj}^*)$;

(3) $k = k + 1$;

(4) If $k \le n$, go to (2), otherwise exit.

Now, we get R^*, $R^{(n)} = R^*$.

$$R^{(n)} = \begin{pmatrix} 1 & & & & \\ 0.5742 & 1 & & & \\ 0.7277 & 0.5742 & 1 & & \\ 0.6416 & 0.5742 & 0.6416 & 1 & \\ 0.6109 & 0.5742 & 0.6109 & 0.6109 & 1 \end{pmatrix}$$

Assigning the threshold λ with different value, we can get different clustering result to the program set. The clustering result is shown in Table 2.

Table 2: Clustering result

Threshold (λ)	Classification (Programs in a pair of braces belong to one cluster)
0.55	{1, 2, 3, 4, 5}
0.60	{1, 3, 4, 5}, {2}
0.64	{1, 3, 4}, {2}, {5}
0.75	{1}, {2}, {3}, {4}, {5}

From the clustering result, we can see that if λ is too small, all programs will belong to one cluster; if λ is too large, each program will belong to individual clusters. Here, we choose λ=0.64, and the programs are partitioned into 3 clusters. We set group CRID of cluster {1, 3, 4} as groupFiction. Since "Gone with the Wind" belongs to this cluster, we can add MemberOf element to its metadata. The revised metadata is shown in Figure 2.

```
<?xml version="1.0"?>
<ProgramInformationTable>
 <ProgramInformation>
  <BasicDescription>
 <Title>Gone with the Wind</Title>
 …
 <Language>en</Language>
 <MemberOf xsi:type="MemberOfType"
crid="groupFiction"/>
  </BasicDescription>
 </ProgramInformation>
</ProgramInformationTable>
```

Figure 2. Program metadata with MemberOf element

5. Conclusions

This paper introduces our research and design on application fuzzy clustering to TV program classification in TV-Anytime environment. With program classified on broadcaster side, the program group recommendation becomes possible. The data sets used for experimentation and performance analysis is the MovieLens dataset [14], which consists of a total of 1682 movie descriptive information. The clustering result proves that adopting fuzzy cluster analysis in program classification is feasible and effective.

References

[1] TV-Anytime Requirements on Environment, TV035r6, TV-Anytime Forum, Aug. 2000

[2] TV-Anytime Specification on System Description, SP002v13, TV-Anytime Forum, Feb. 2003

[3] TV-Anytime Specification on Content Referencing, SP004v12, TV-Anytime Forum, Jun. 2002

[4] Zheng Zhikai, Zhang Guangfan and Shao Huihe. Data Mining and Knowledge Discovery: An Overview and Prospect. Information and Control, 1999, 28(5): 357-365.

[5] Frank Höppner. Fuzzy Cluster Analysis. http://www.fuzzy-clustering.de/, Aug. 2001

[6] T Bray, J Paoli, and C M Sperberg-McQueen, "Extensible Markup Language (XML) 1.0", *http://www.w3.org/TR/ REC-xml*, Oct. 2000

[7] TV-Anytime System Description Document (Informative with mandatory appendix B), WD608, TV-Anytime Forum, Aug. 2002

[8] TV-Anytime Metadata Specifications Document, SP003v12 Part A AppendixB, TV-Anytime Forum, Jun. 2002

[9] ISO 8601, "Data elements and interchange formats - Information interchange - Representation of dates and times".

[10] G Salton. Automatic Text Processing: The transformation, analysis, and retrieval of information by computer. Addison-Wesley, Massachusetts, USA, 1989

[11] Tak W Yan and Hector Garcia-Molina. "Index Structures for Information Filtering Under the Vector Space Model", Proceedings of the Tenth International Conference on Data Engineering, Houston, USA, 1994

[12] Languages in the World. http://www.pacint.hf.ah.cn/gb/sjdyy.htm

[13] Stephen Warshall. A theorem on Boolean matrices. Journal of the ACM. 1962, 9(1): 11-12.

[14] GroupLens Home Page, http://www.cs.umn.edu/Research/GroupLens/data/ ml-data.tar.gz