APPLICATION OF AN ANT COLONY ALGORITHM
For Song Categorising using Metadata

Nadia Lachetar[1] and Halima Bahi[2]
[1]*Computer Science Department, Skikda University, Skikda, Algeria*
[2]*Labged Laboratory, Computer Science Department, Annaba University, Annaba, Algeria*

Keywords: Audio indexing, Naive Bayes algorithm, Ant colony algorithm, Song categorisation.

Abstract: Instead of the expansion of the information retrieval systems, the music information retrieval domain is still an open one. One of the promising areas in this context is the audio indexing databases. This paper addresses the problem of indexing database containing songs to enable their effective exploitation. Since, we are interested with songs databases, it is necessary to exploit the specific structure of the song in with each part plays a specific role. We propose to use the title and the artist particularities (in fact each artist tends to compose or sing a specific genre of music). In this article, we present our experiments in automated song categorisation, where we suggest the use of an ant colony algorithm. A naive Bayes algorithm is used as a baseline in our tests.

1 INTRODUCTION

Nowadays music is playing an important role in human's life, whereas digital catalogues rapidly become larger and more inconvenient and inefficient to access. If we do not have a good method to explore music, a large amount of music will be fallen into oblivion (Dang and Shirai, 2009). As the multimedia content is growing, and digital music libraries are expanding at an exponential rate, it is important to secure effective information access and retrieval.

In this paper, we propose to construct an automatic system of categorisation of songs by theme using metadata.

Currently many studies are devoted to the use of acoustic information to detect the theme of songs (Li and Ogihana, 2004; Chua and Lu, 2004; Liu and Zang, 2003). To index an audio document, the first step is to determine the type of information present. In the case of songs, many studies have been performed to detect the music, speech, or sound features (Scheirer and Slaney, 1997; Karjalainen and Tolonen, 1999). Very little has been done on the song (Arroabarrem et al., 2003).

Many aspects of the music itself (such as lyrics, genre, key or era) are shared on the Internet. Digital music can hold information such as artist, track name, year, and album in the source.

Our study is based on the use of metadata to detect a theme. We propose to use the song title that briefly describes its theme but also the characteristics of the artist, because every artist has a tendency to compose or sing a particular kind of music. We introduce methods for supervised learning classification of songs based on metadata and we propose the use of an ant colony algorithm for classification of songs using the title and the characteristics of the artist.

In section 2, we describe the state of the art. Then in section 3 we present our training data. Section 4 is devoted to categorize songs while section 5 presents the naive Bayes algorithm. In Section 6 we present our approach for indexing songs using the songs title and artist features. Section 7 presents the obtained results and a discussion.

2 STATE OF THE ART

Many works are devoted to the extraction of features of a song for the descriptions of its contents are generally guided by the acoustic analysis (Li and Ogihana, 2004; Chua and Lu, 2004; Liu and Zang, 2003). Knees and others have a pioneering work to build an automatic search that is able to find the music that satisfies arbitrary queries in natural

Lachetar N. and Bahi H..
APPLICATION OF AN ANT COLONY ALGORITHM - For Song Categorising using Metadata.
DOI: 10.5220/0003631303710376
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2011), pages 371-376
ISBN: 978-989-8425-79-9

language (Knees et al., 2000). Another work described in (Harb and Chen, 2003) based on an automatic segmentation of the soundtrack music or speech, using a technique of segmentation into sentences. The music segments are indexed in a way that allows a search by similarity. Other jobs using a classification according to the mood of the songs are described in (Dang, Shirai, 2009, Kanters, 2009, Laurier and al., 2008). The classification according to the mood does not seem interesting to apply it to a search engine for music because the mood is metadata subjective words are short and contain many metaphors that can be understood by humans.

Through this work, we introduce a new dimension of classification, considering contextual information about the artist. Thus, each artist sings songs with a specific emotion, such as Eric Clapton sings sad songs often but Bob Marley likes to sing happy songs.

3 CONSTRUCTION OF TRAINING DATA

In this section we describe how to prepare our training data, the collection of songs tagged with the theme described by the title and artist features.

A great blog site Live Journal (www.livejournal.com) is used, each blog entry is labeled with the theme of the song given by the title of this latter. The song title and artist features can be obtained by simple string matching with the database artist, obtained from open artist got from the music site (www.musicmoz.org). The lyrics may be obtained from the Site (www.lyrics.com).

4 SONG CATEGORIZATION

Research in the field of automatic categorization remains relevant today since the results are still subject to improvements. For some tasks, the automatic classifiers perform almost as well as humans, but for others the gap is even greater. At first glance, the main problem is easy to grasp. On one hand, we are dealing with a bank of songs and on the other with a set of categories. The goal is to make a computer application which can determine to which category belongs a song based on its contents. The set of categories is determined in advance. The problem is to group the songs by their similarity. There are two approaches to solving the problem of songs categorization: the information using either acoustic or verbal information. In this paper we will focus on the words comprising the title of the song to determine its theme and the characteristics of the artist to determine what kind of music.

The categorization process includes the construction of a prediction model that receives in input the title of the song, and as output it combines one or more labels.

Prior coding of song is necessary because there is currently no method of learning which can directly handle unstructured data in the model construction stage, or when used in classification.

For most learning methods, we must convert all texts in a PivotTable "individuals-variables".

In song categorization, we transform the title of the song into a vector $d_j = d_j\ (w_{1j}, w_{2j}, ..., w_{|T|j})$, where T is the set of terms (descriptors) that appear at least once in the corpus (the collection) learning. The weight w_{kj} correspond to the contribution of terms t_k to the semantics of title of song d_j .

Once we choose the components of the vector representing the song j, we must decide how to encode each coordinate of the vector d_j. There are different methods to calculate the weight w_{kj}. These methods are based on two observations:

More the term t_k is frequently in a title of song d_j, more it is relevant to the subject of this song.

More often the term t_k is in a collection, unless it is used as discriminating between songs.

The Coding terms frequency x inverse document frequency and Coding terms TFC are the most used.

5 NAIVE BAYES ALGORITHM

In machine learning, different types of classifiers have been developed to achieve maximum degree of precision and efficiency, each with its advantages and disadvantages. But, they share common characteristics (Sebastiani, 2002).

Naive Bayes Classifier is the most commonly used algorithm. When we apply the naïve Bayes for a song categorization task, we look for the classification that maximizes the probability of observing the words of titles of the songs.

During the training phase, the classifier calculates the probability that a new song belongs to this category based on the proportion of training songs belonging to this category. It calculates the probability that a given word is present in a title of the song, knowing that this song belongs to this category. Then as a new song should be classified, we calculate the probability that it belongs to each class using Bayes rule and the probabilities calculated in the previous step. The likelihood to be estimated is:

$$p(c_j|a_1,a_2, a_3, ..., a_n) \quad (1)$$

Where c_j is a category and a_i is an attribute

Using the Bayes theorem, we obtain:

$$\mathrm{p}(c_j|a_1, a_2, .., an) = \frac{P(a_1,a_2,...,a_n \backslash c_j)*P(c_j)}{P(a_1,a_2,...,a_n)} \quad (2)$$

$$p(a_1, a_2, a_3, ..., a_n \backslash c_j) = \prod_{i=1}^{n} p(a_i \backslash c_j) \quad (3)$$

6 APPLICATION OF AN ANT COLONY ALGORITHM FOR SONG CATEGORIZATION

6.1 Introduction

The originality of our approach is on adapting an algorithm of ant colony to song categorization.

The algorithm of ant colony optimization is inspired by the behaviour of ants searching for food. Its principle is based on the behaviour of individual ants; they are able to determine the shortest path between their nest and a food source using the pheromone which is a substance that ants lay on the floor when they move. When an ant has to choose between two directions, it chooses with higher probability (Solnon, 2005).

6.2 Principe of the Algorithm

It relies on the specific behaviour of ants, and determines the shortest path between the nest and a food source overall progress algorithm.

6.3 For Song Categorization

For the construction of the graph, the nodes represent titles of songs. The pheromone is a measure of similarity between titles of songs which may be the distance between these documents. The choice of distance is an important parameter.

We add to the vector of features once standardized the characteristics of the author (the type of music and the theme he sings in general).

6.3.1 Calculate the Distance between the Song Filed and the Songs constituting the Graph

For our approach we use the cosine similarity between two songs a and b defined by

$$\sum_{t \in T} \frac{p_t(a) * p_t(b)}{\sqrt{\sum_{t \in T} p_t(a)^2 * \sum_{t \in T} p_t(b)^2}} \quad (4)$$

Where:

- T is the set of attributes.
- $p_t(a)$ is the weight of term t in title of song a.
- $p_t(b)$ is the weight of term t in title of song b.

This measure allows comparing titles of songs of different lengths by normalizing their vector.

We use the cosine similarity between each title of song "a" of the graph of songs and the input title of song "b" to be classified.

The following algorithm computes the cosine similarity based on relevant attributes for the various couples forming the nodes of a graph and the input title of song. It takes as input the graph of songs and the document to classify and returns as output a similarity matrix based on relevant attributes.

Algorithm Cosine_Similarity

```
Begin
   Input:song_Graph,song_class   //graph
   of songs,Classified song;
   Output: Mat_Sim
   // similarity matrix based on the
   relevant attributes
   Mat_Sim ← 0;
   Begin
   For each node of song_Graph
   // Extract set of attribute nodes of
   the graph
   SIM=Calcul_Sim (node, song_class);
   Mat_Sim=Mat_Sim+Sim(node,song_class)
   ;
   Return Mat_Sim
   End.
End.
```

6.3.2 Ant Colony Optimization

To find the song category, we adopt the algorithm of ant colony optimization (ACO), proposed in (Solnon, 2005). Although the ant colony algorithm is originally designed for the travelling salesman problem, it finally offers great flexibility. Our choice is motivated by the flexibility of the metaheuristics which makes possible its application to different problems that are common to be NP-hard. Thus the use of a parallel model (colonies of ants) reduces the computing time and improves the quality of solutions for categorization.

Formalization of the problem: In our context, the problem of classifying a song reduces the problem of subset selection (Solnon, 2005), and we can formalize the pair (S, f) such that:

- S contains all the cosine similarities calculated between the graph of songs and the song to classify. It's "matrix similarity" mat_sim.
- F is defined by the function score, the score function is defined in (Solnon, 2005) by the formula.

$$score(s') = f(\text{song\_Graph} \cap \text{song\_class} - g(slits\ s')). \quad (5)$$

Splits (S') is the set of nodes in the graph which are more similar to the song to classify. So the result is a consistent subset S' of nodes, as the score function is maximized.

6.3.3 Construction of the Graph of Songs

To adapt our approach, a direct graph G(V,E) is drawn, where V is a set of vertices and E is a set of possible edges between these vertices as shown in figure 1 in this graph a number of ants is managed for tmax iteration.

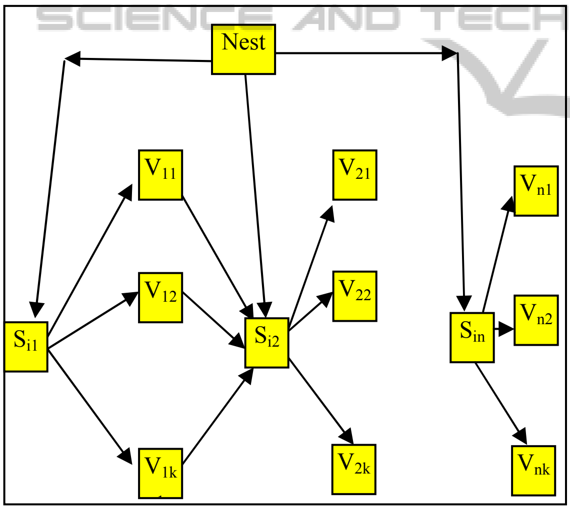

Figure 1: Graph representation of song categorization.

A graph G(V,E) is automatically generated from a text file that contains the problem's data (words of titles of songs).

- Graph's vertices are bundled with words of titles of songs .
- Graph's edges are bundled with the initial pheromone trails. The pheromone is a measure of similarity between titles of songs. We use the cosine similarity between two songs according to (4)

6.3.4 Description of the Algorithm

At each cycle of the algorithm, each ant constructs a subset. Starting from empty subset, ants at each iteration add a couple of nodes from the similarity matrix. S_k chosen among all couples not yet selected. The pair of nodes to add to S_k is chosen with a probability which depends on the trail of pheromones and heuristics. One aims to encourage couples who have the greatest similarity and the other is to encourage couples who are most increase the score function. Once each ant has built its subset, a local search procedure start to improve the quality of the best subset found during this cycle. Pheromone trails are subsequently updated based on the subset improved. Ants stop their construction when all pairs of candidate nodes are decreased the score subset.

Construction of a solution by an ant: The following code describes the procedure followed by ants to construct a subset. The first object is selected randomly. The following items are selected in all candidates.

Algorithm Construction_subset

Begin

Input:SS_problem(S, $s_{consistant}$, f)and an associated heuristic function: S*P(S)→ IR^+; heuristic pheromone; and an phenomenal factor δ and 2 heuristic factors φ_1 and φ_2;

Numeric parameter α, β_1 and β_2

Output: a consistent subset S'∈ S

Initialize pheromone trails to τmax

Repeat

For each ant k in 1 .. nbAnts, construct a solution S_k as follows:

1. Randomly select the first node O_i∈ S
2. $S_k \leftarrow \{o_i\}$
3. Candidat ← $\{o_i \in S\ /S_k \cup \{oj\} \in S_{consistent}\}$
4. While Candidates ≠ ∅ do
5. Choose a node o_i∈ candidat with probability
6. $p_{oi} = \sum_{t\in T} \frac{p_t(a)*p_t(b)}{\sqrt{\sum_{t\in T} p_t(a)^2 * \sum_{t\in T} p_t(b)^2}}$
7. $s_k \leftarrow S_k$ union $\{o_i\}$
8. Remove o_i from Candidates
9. Remove from candidates each node o_j as $S_k \cup \{o_j\} \in S_{consisting}$
10. End while
11. End for

Update pheromone trails according to $\{S_1, \ldots, S_{nbAnts}\}$

If a pheromone trail is less than τ_{min} then set it to τ_{min}

Else If a pheromone trail is greater than τ_{max} then set it to τ_{max}

Until maximum number of cycles reached or solution found.

End.

End.

7 RESULTS AND DISCUSSION

To evaluate performances of our suggestion, we make some experiments using two corpus one for the training and the other for the test. We also use the Naïve Bayes classifier as baseline one.

Table 1: Classes of corpus.

Class	Nb of songs in training stage	Nb of songs in test stage
National anthem	12	10
Loves songs	60	46
Religious songs	36	30
Sport songs	27	26
Learning songs	18	16

The results of classification stage are reported below for ant colony algorithm and naïve Bayes algorithm

Table 2: Results of tests with ant colony algorithm.

Clas	Nati Ant	Lov son	Reli son	Spor son	lear son	Total
Nati Ant	9	0	1	0	0	10
Lov Son	1	42	2	0	1	46
Reli Son	1	2	26	0	1	30
Spor Son	0	1	1	23	1	26
lea son	0	1	1	0	14	16

Table 3: Results of tests with naïf Bayes algorithm.

Clas	Nati ant	Lov son	Reli son	Spor son	lear son	Total
Nati Ant	7	2	1	0	0	10
Lov Son	1	40	2	1	2	46
Reli Son	2	4	22	1	1	30
Spor Son	0	2	1	22	1	26
lea son	1	1	1	0	13	16

Precision and recall are the most used measurements to evaluate information retrieval systems, they are defined as follow:

Table 4: Contingency table based evaluation of the classifiers.

	Song belonging to the category	Song not belonging to the category
Song assigned to the class by the classifier	a	b
Song rejected by the classifier	c	d

According to this table, we define:
Precision=a/(a+b), the number of correct assignments over the total number of assignments.

Recall=a/(a+c), the number of correct assignments over the number of assignments that should have been made.

When evaluating the performance of a classifier, precision or recall is not considered separately. So the F1 measure is defined which is used extensively by the formula:

F1 = 2*r*p/(p+ r) (r is the recall, and p is the precision).

It is a function which is maximized when the recall and precision are close.

Table 5 and table 6 present performances of ant colony and naïve Bayes in terms of recall, precision and F1

Table 5: Recall, precision, F1 for each class (ant colony algorithm).

Class	Recall	Precision	F1
National anthem	90.00	100.00	94.73
Loves songs	92.30	82.75	87.26
Religious songs	66.66	80.00	72.72
Sport songs	75.00	100.00	85.71
Learning songs	83.33	71.42	73.68

Table 6: Recall, precision, F1 for each class (naive Bayes algorithm).

Class	Recall	Precision	F1
National anthem	70.00	77.77	73.68
Loves songs	84.61	75.86	79.99
Religious songs	58.33	77.77	66.66
Sport songs	75.00	75.00	75.00
Learning songs	66.66	57.14	61.53

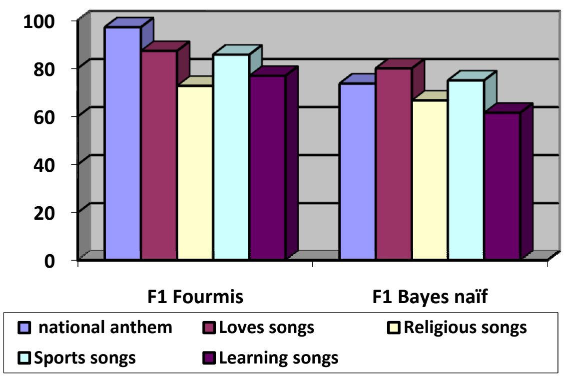

Figure 2: Classification rates for each category and both classifiers.

The histogram (figure 2) shows that the suggested ant colony algorithm outperforms Naïve Bayes algorithm in terms of recall and precision. This is not surprise since the graphical representation of the problem handled better relationship (characteristics of artist are taken into account during the construction of the graph).

REFERENCES

Chua, B. Y., Lu, G., 2004. Determination of perceptual tempo of music. In *CMMR*.

Dang, T. T., Shirai, K., 2009. *Machine learning approaches for mood classification of songs toward music search engine.*

Harb, H., Chen, T., 2003. *Cyndi: An indexing engine from the soundtrack by segmentation and semantic extarction keywords.* LIRIS, Lyon

Karjalainen, M., Tolonen, T., 1999. Multi-pitch and periodicity analysis model for sound separation and auditory scene analysis. In *ICASSP.IEEE,* vol. 2, pp. 929-932.

Knees, P., Pohle, T., Schedl, M., Widmer, G., 2000. A music search engine Built upon Audio-based and Web-based similarity Measures. In *SIGIR*.

Laurier, C., Grivolla, J., Herrera, P., 2008. Multimodal music mood classification using audio and lyrics. *ICMALA*, pp 668-693

Li, T., Ogihana, M., 2004. *Content-based music similarity search and emotion detection.*

Liu, D., Lu, L., Zang, H. J., 2003. *Automatic mood detection from acoustic music data.*

Scheirer, E., Slaney, M., 1997. Construction and evaluation of a robust multifeatures speech/music discriminator. In *ICASSP*. IEEE, vol.2, pp. 1331-1334.

Sebastiani, F., 2002. *Automated text categorization tools, techniques and application.* Renne France,.

Solnon, C., 2005. *Contribution to the practical problem-solving combinatorial-graph and the ants.* Thesis for habilitation research, Claude Bernard University Lion1.