A QUALITATIVE ASSESSMENT OF MEASURES FOR THE EVALUATION OF A COVER SONG IDENTIFICATION SYSTEM

Joan Serrà
Music Technology Group
Universitat Pompeu Fabra
jserra@iua.upf.edu

ABSTRACT

The evaluation of effectiveness in Information Retrieval systems has been developed in parallel to its evolution, generating a great amount of proposals to achieve this process. This paper focuses on a particular task of Music Information Retrieval: a system for Cover Song Identification. We present a concrete example and then try to elucidate which metrics work best to evaluate such a system. We end up with two evaluation measures suitable for this problem: *bpref* and *Normalized Lift Curves*.

1 INTRODUCTION

Before the final implementation of any Information Retrieval (IR) engine, we must carefully consider the quality of the end-product of our efforts. This step can be described as a performance evaluation of a proposed solution. IR techniques can be essentially seen as heuristics: we try to guess something as similar as possible to the right answer. So we have to measure how close to it we can come. Furthermore, evaluation methods are used in a comparative way to measure whether certain changes lead to any improvement in system performance. In particular, when tuning algorithm parameters, it is important to choose the evaluation measure that rewards what we think is a right answer (choosing a valid measure).

In the next sections we concentrate on evaluating an IR engine. More precisely, we focus on the evaluation of the effectiveness of a particular Music Information Retrieval (MIR) system. Our goal is to decide which measures are valid (construct validity) for a specific situation which is presented in subsequent sections.

This concrete case is related with Cover Song [1] Identification, a very active research topic within the last few years in the MIR community [3, 4, 6], as it provides a direct way of evaluating music similarity algorithms. Some efforts are then being devoted to compare and evaluate different alternatives for this purpose (as MIREX [2]).

[1] According to Wikipedia, in popular music, a cover song (a cover version, or simply cover) is a new rendition, performance or recording of a previously recorded song.

[2] http://www.music-ir.org/mirex2006/index.php/Audio_Cover_Song

2 EVALUATION MEASURES

We focus on the situation where a retrieval engine has an input query and it provides an output list of documents (preferably relevant to the query).

We find in the literature some measures from binary classification that might be useful for our purpose: *False / True Positives and Negatives* (*TP*, *FP*, *TN*, *FN*), *Sensitivity* and *Specificity*. We also consider here the *Fallout Rate*, the *Receiver Operating Characteristic* (*ROC*) curve, and the *Lift Curve* [8]. Finally, some popular IR measures we analyze are: *Precision*, *Recall*, the *Precision-Recall curve*, the *Break-even Point*, the *F-measure* and *Average Precision* (*AP*). We also consider *Reciprocal Rank* (*RR*), *Discounted Cumulative Gain* (*DCG*) and *Binary Preference-based measure* (*bpref* and *bpref-10*) [1, 2, 5, 7].

We do not study here other measures such as *Spearman's Rho* or *Kendall's Tau*, because our data does not fit to the models they were thought for. Basically, we do not have a true measure of similarity for the ground truth (our cover songs, originally, are not ranked from more similar to less similar, we just only know if they are a cover of a given query or not).

3 CASE STUDY: COVER SONG IDENTIFICATION SYSTEM

In this section we study the situation where there is a song database (D, the document collection), and we have to come up with an algorithm that, given a song title (query q), yields a list of potential cover songs (A, a list of their titles in descending order of similarity). Here, the query song is not retrieved (that is: $q \notin A$).

We should note that there is not a ground truth for song similarity. As a ground truth data, we label [3] all songs, indicating if they correspond to the same group (the same label is attached to the original song and covers of it) or not. Thus, our judgements are based on binary relevance.

For our concrete problem, we have a database of 2054 songs ($|D|$ = 2054), labelled into 451 different groups (or "canonical" song versions). The average number of covers per song is 4.24, ranging from 1 (the original song + 1 cover) to 14. Since the maximum number of covers

[3] Do not confuse the song titles (which are not relevant for us), with the label we attach to them after listening.

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	$\|R_q\|$
$q_1 \Rightarrow A_1$				$\star$											1
$q_2 \Rightarrow A_2$	$\star$	$\star$	$\star$		$\star$										7
$q_3 \Rightarrow A_3$						$\star$	$\star$	$\star$		$\star$					7
$q_4 \Rightarrow A_4$		$\star$		$\star$		$\star$		$\star$							14
$q_5 \Rightarrow A_5$	$\star$					$\star$	$\star$	$\star$							14
$q_6 \Rightarrow A_6$															4

Table 1. Test answer set example. It consists of 6 manually labelled answer sets (A_i) answering 6 hypothetical queries (q_i). These answer sets are composed of 14 ranked documents ($A_i = \{a_1, \ldots, a_{14}\}$), and they are ordered from most valuable (A_1) to less valuable (A_6). The "$\star$" symbol in (i, j) cell denotes that the a_j document is relevant for the i-th query. Last column ($|R_q|$) denotes the total number of covers for the query q_i that can be found in the database.

per canonical version is 14, the length of the answer set is set to this number in order to be able to present to a potential user all the relevant songs in a single output list ($|A| = 14$). A cutoff like this is typically introduced in an IR system because of the paginated presentation of search results.

3.1 Preliminary hypotheses

The order (ranking) in which the documents are presented in an answer set A will be relevant (as the algorithm attempts to partially define a similarity metric, and therefore, to provide the most similar songs at the beginning of the list). From this we hypothesize that rank-based measures (like *RR* or *DCG*) should perform well.

Another main objective of the desired algorithm is to maximize the amount of retrieved covers (like in any audio identification task). Then, we can also argue that *Recall*-based measures fit our requirements. Note that in our experiments we will know recall (as all the documents in the collection where we want to search for are labelled).

3.2 Test framework

We manually annotate and rank several synthetic sets of prototypical answers to different queries in order to try to elucidate which measure best fits our criteria.

For a set of queries $S_q = \{q_1, \ldots, q_{N_q}\}$, we define a set of answer sets $S_a = \{A_1, \ldots, A_{N_q}\}$, where each $A_k = \{a_{k,1}, a_{k,2}, \ldots, a_{k,14}\}$. In our experiments, the number of sets S_q is equal to 30 and N_q ranges from 4 to 8. We manually label the retrieved documents $a_{k,j}$ with a "$\star$" symbol, which denotes if we consider them to be relevant or not to the query.

An example of such an answer set is shown in table 1. This set has been chosen because it represents the typical situation we want to highlight (regarding ranking and recall of the answer sets), and it will be the main reference for our discussion in next subsection.

We intentionally rank the answers A_k from most to least important for us. This is the way we define the relevance of the answer sets. This also helps to observe which measures are more suitable. For instance, we prefer of our system to retrieve a single cover song if there is only one in the collection D, rather than retrieving four out of seven (see A_1 and A_2 in table1), even if they are ranked in the first positions. Also, on a situation with the same percentage of retrieved songs ($|R_a|/|R_q|$, where R_a corresponds to the set of relevant documents for the answer set A, and R_q corresponds to the set of all relevant documents in the entire collection D), it would be desirable that they were ranked at the first positions (see A_2 and A_3). Notice that A_6 is the worst answer because it does not retrieve any relevant document.

3.3 Evaluation measures for a Cover Song Identification system

We implemented all the cited measures and tested several synthetic sets of potential answers according to the mentioned framework. Table 2 presents the results corresponding to the example answer set of table 1. We now elaborate some comments on results such as these.

All the measures *TP*, *FP*, *FN* and *TN* provide us with important information but are not suitable for our task, as these features do not take into account the rank (position) of the correctly classified instances. Furthermore, they do not consider the total number of relevant documents per query ($|R_q|$), so that they do not care, for instance, about the difference in retrieving the only possible item of a set, or one of the largest labelled group (we want the former to have a higher reward than the latter).

Accuracy, *Specificity* and *Fallout Rate* suffer from the same kind of problems as the measures cited above, and, in addition, as our data is extremely skewed (in IR systems, over 99.9% of the documents are usually in the not-relevant category [1]), they allow few discernment and they are not discriminative enough between answer sets (for instance, A_4 and A_5).

If we plot the *ROC* and *Lift* curves, we find the same problems, as they are based in *TP*, *FP*, *FN* and *TN*. However, we have come with a useful variant (the *Normalized Lift Curve*, shown in figure 1), consisting of plotting the percentage of positive examples normalized by the total number of relevant queries ($\%Pos.Ex/|R_q|$) versus the ratio of examples normalized by the length of the answer set ($\%Ex./|A|$). Figure 1 provides an easy interpretable curve.

Measure	A_1	A_2	A_3	A_4	A_5	A_6
TP	1	4	4	4	4	0
FP	13	10	10	10	10	14
FN	0	3	3	10	10	14
TN	2040	2037	2037	2030	2030	2036
Accuracy	0.994	0.994	0.994	0.990	0.990	0.991
Sensitivity	1.000	0.571	0.571	0.285	0.285	0.000
Specificity	1.000	0.998	0.998	0.995	0.995	0.998
Fallout rate	0.006	0.005	0.005	0.005	0.005	0.007
Precision	0.071	0.286	0.286	0.286	0.286	0.000
Recall	1.000	0.571	0.571	0.286	0.286	0.000
Break-even point	0.3	0.7	0.3	0.5	0.4	0.1
AP	0.250	0.950	0.307	0.500	0.496	0.000
F-measure	0.133	0.381	0.381	0.286	0.286	0.000
RR	0.018	0.145	0.038	0.074	0.095	0.000
DCG	0.721	3.974	1.987	3.203	2.371	0.000
bpref	-2.000	0.550	0.143	0.235	0.194	0.000
bpref-10	0.727	0.563	0.395	0.256	0.232	0.000
bpref*	0.800	0.564	0.428	0.260	0.239	0.000

Table 2. Results for different measures for the test case example shown in table 1. The columns correspond to the value of the evaluation measure for the answer set A_i in the forementioned example set.

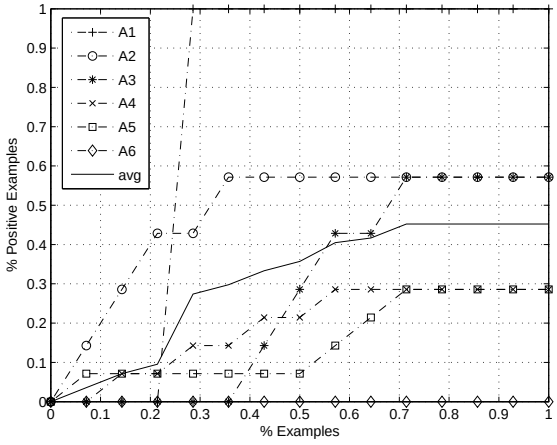

Figure 1. *Normalized Lift Curves* for the test answer set. Dash-dotted lines are the test answers (A_i) and the solid line corresponds their average. Performance is as good as the steepness of the curve that approximates to point (0,1).

Precision and *Recall* values seem to be good for our purposes (*Recall* better than *Precision*, as we had previously hypothesized), but they fail in taking into account the position (rank) of the correctly retrieved items (for instance, we cannot distinguish between A_2 and A_3 or between A_4 and A_5, which have the same $|R_a|$ and $|R_q|$).

The *Precision-Recall Curve* (figure 2) gives an idea about the ranking of the items, but it does not measure if we have retrieved all the possible elements. In addition, there are some problems when interpolating answers with just one relevant document (A_1). We also noticed some interpolation problems when there are equally spaced relevant documents in the answer set (A_4). Also, the *Precision-Recall Curve* has the problem that it is not just a value, and thus, it is a bit difficult to interpret it in some particular situations (looking at figure 2, which answer is better: A_3 or A_5?).

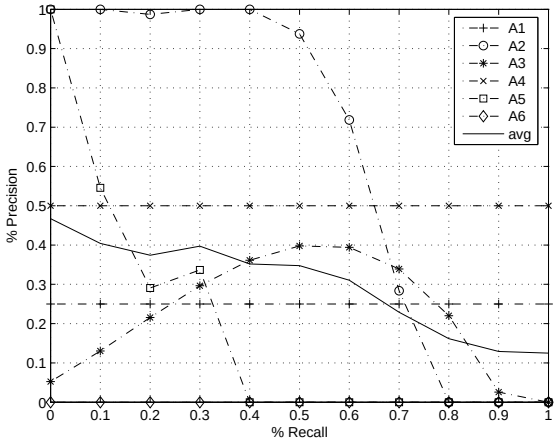

Figure 2. *Precision-Recall Curves* for the test answer set. Dash-dotted lines are the test answers (A_i) and the solid line corresponds to the average of them.

AP seems better than *Precision* or *Recall* alone (we are able to distinguish between differently ranked answers), but we feel that ranking matters a lot (see A_2). It also does not consider if we have retrieved all possible elements (R_q).

F-measure and other measures obtained by combining

Precision and *Recall* also suffer from their drawbacks.

Regarding *RR* and *DCG*, we find again that ranking matters a lot. We have problems in distinguishing between A_4 and A_5 for the former and between A_2 and A_3 for the latter. These measures also do not consider $|R_q|$.

In general, we can see that we need a measure which combines two different aspects: ranking and recall, as stated in our preliminary hypotheses. Looking at table 2, it would seem that we can come with such a single value just averaging recall and rank-based values. We decided therefore to implement some measures combining *AP* and *Recall* with a weighted mean and with the harmonic mean (in an *F-measure* fashion). We also tried with *RR* and *Recall* and with *DCG* and *Recall*. These new measures work well for several test sets, but, in the end, *bpref-10* or *bpref** fit better with our expectancies.

Bpref* stands for a variant of *bpref* [2] that seems to perform well for practically all the answer sets we have tested. When testing *bpref*, we found one of the problems mentioned in the above reference for A_1 (the number of relevant documents being very small), but we solved it with the variant mentioned in the same article (*bpref-10*) and also with a new one (*bpref**). This last leads to quite satisfactory results. The formulation of *bpref** is similar to *bpref-10*:

$$bpref^* = \frac{1}{|R_q|}\sum_{j=1}^{|R_a|}\left(1 - \frac{N_{nr}(j)}{|A| + |R_q|}\right) \qquad (1)$$

Where $|R_q|$ is the total number of relevant documents in the collection, $|R_a|$ is the number of relevant documents in the answer set, and $N_{nr}(j)$ is the number of judged non-relevant documents ranked before the j-th relevant retrieved document in the ordered set R_a.

If we consider the fact that *bpref* variants have the additional property of dealing with unjudged information (not labelled elements), which allows us, for instance, to introduce outliers to our database without affecting our evaluation measure [2], *bpref** becomes our choice of preference.

4 CONCLUSIONS

We have presented a particular MIR system focused on Cover Song Identification and discussed the suitability of different measures for evaluating it. Even though a complete formal analysis has not been presented, the discussion has guided us to assess the pros and cons of different measures, and to select those that seem to be more suitable to our problem (the ones to monitor when performing several implementations of a Cover Song Identification algorithm).

Furthermore, we have come with an adaptation of a particular evaluation measure, *bpref**, which we think reflects in many ways the retrieval performance we wanted to care about when considering the possible answer sets of our test case. Moreover, we have found that using *Normalized Lift Curves* can be very informative for our task.

In a broader sense, it is very difficult to tell someone which evaluation measure to use. Sometimes we take for granted that people know the differences between measures such as those we have dealt with. In some cases these differences are not so important. In other cases, one perhaps just take the measure that other people uses for a similar purpose or a similar database. This is good for comparison, but does not imply that the chosen measure is going to be the most appropriate one, and a possible problem arises when tuning algorithms to a value that you do not know if it measures the "information" you want.

Choosing an evaluation measure strongly depends on the problem you are focused, and every case has to be studied independently. So, we highly encourage researchers to make such analysis and reflections as we have made here when facing the evaluation of a concrete IR problem.

5 ACKNOWLEDGEMENTS

The author wishes to thank his colleagues at the MTG (UPF), specially Emilia Gómez and Perfecto Herrera for their constant support and helpful reviews, and Vanessa Murdock at Yahoo! Research (Barcelona).

This research has been partially funded by the EU-IP project PHAROS [4] .

6 REFERENCES

[1] R. Baeza-Yates and B. Ribeiro Neto. *Modern Information Retrieval*. ACM Press Books, 1999.

[2] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. *SIGIR'04*, (27), 2004.

[3] D. P. W. Ellis and G. E. Polliner. Identifying cover songs with chroma features and dynamic programming beat tracking. *Proc. ICASSP*, April 2007. (submitted, 4pp) - See also MIREX'06 poster.

[4] E. Gómez, B. S. Ong and P. Herrera. Automatic tonal analysis from music summaries for version identification. *Conv. of the Audio Engineering Society (AES)*, October 2006.

[5] C. D. Manning, R. Prabhakar and H. Schutze. *An introduction to Information Retrieval*. Cambridge University Press, Cambridge, England, preliminary draft ed., 2007. Online version at http://www.informationretrieval.org (last access: April 2007).

[6] M. Marolt. A mid-level melody-based representation for calculating audio similarity. *Proc. ISMIR*, 2006.

[7] E. M. Voorhees and L. P. Buckland. Common evaluation measures. *in Proc. of Text Retrieval Conference*, 2006. Appendix.

[8] N. Ye. *The handbook of Data Mining*. Lawrence Erlbaum Associates, 2003.

[4] http://www.pharos-audiovisual-search.eu/