On the Benefits of Representing Music Objects as Fuzzy Sets

Klaas Bosteels[1,2] Elias Pampalk[2] Etienne E. Kerre[1]

1. Fuzziness and Uncertainty Modelling Research Group
Dept. of Applied Mathematics and Computer Science, Ghent University, Ghent, Belgium
2. Last.fm Ltd., London, UK
Email: klaas.bosteels@ugent.be, elias@last.fm, etienne.kerre@ugent.be

Abstract— *Over the past few years, fuzzy set theory has become a well-established mathematical theory that has been applied in many practical applications. In this paper, we elaborate on the use of fuzzy sets in the context of music information retrieval. First, we demonstrate that music objects like artists and songs can straightforwardly be represented as fuzzy sets, and then we explain why it is beneficial to represent them in this way. In particular, we illustrate the potential of this fuzzy approach by describing how we relied on it to build the popular "Multi Tag Search" demonstration on* `http://playground.last.fm`.

Keywords— Music information retrieval, fuzzy sets, fuzzy comparison measures.

1 Introduction

The way in which the objects are represented is an important aspect of most music information retrieval applications. Simple vectors are used in many cases, but certain more advanced representations like statistical distributions are popular too. In this paper, we advocate for the usage of fuzzy sets for this purpose. As we will explain in detail, representing a music object as a fuzzy set in some universe U, i.e., as a $U \to [0,1]$ mapping that associates a degree of membership with each u from U, is a natural and mathematically sound approach that offers many possibilities and advantages. To enforce our claims, we will describe how we relied on fuzzy set theory for building the "Multi Tag Search" demonstration on `http://playground.last.fm`, the section of the CBS-owned music community website Last.fm[1] where new ideas and experimental technologies are showcased.

2 Representing music objects as fuzzy sets

There are three main ways to obtain descriptions for music objects:

1. Hiring music experts to manually annotate the objects.

2. Assembling a large community of music enthusiasts and then tapping into the wisdom of crowds.

3. Applying signal processing techniques to generate descriptions automatically.

Hence, song descriptions can come from experts, communities, or computers. In each case, it is often possible to convert them to fuzzy sets.

[1]http://www.last.fm

2.1 From experts

When music experts are hired to generate descriptions, they are usually asked to provide values for a predefined list of attributes. For instance, an expert could be used to assign values to the attributes "genre" and "complexity" for a particular song. Since the latter attribute is numerical, it can straightforwardly be converted to a gradual element of a fuzzy set. For example, if a song is 80% complex, then we would give "complexity" a membership degree of 0.8 in the fuzzy set that corresponds with this song. In case of a categorical attribute like "genre", however, we cannot interpret its values as membership degrees. Nevertheless, it is still possible to convert such attributes, namely, by regarding the values as binary attributes. For example, we can express that a song is an alternative rock song by assigning the membership degree 1 to "alternative rock" and setting the membership degrees of all other possible genres to 0. In fact, it might even be more appropriate to directly present the different genre values as separate numerical attributes to the expert, since a song often belongs to several genres to some extent. The expert can then express that, e.g., a particular song is mainly alternative rock, but also incorporates some notable blues and jazz influences. The well-known online radio station Pandora[2] uses 150-500 so called "genes" to describe songs [1, 2], where each gene is a numerical attribute that takes a value from the interval $[0, 5]$. Hence, the descriptions generated by Pandora's music experts could easily be interpreted as fuzzy sets.

2.2 From communities

The data that powers Last.fm's recommendation engine is mainly generated by a technique called "scrobbling". When a user listens to a song, the name of this song is sent to Last.fm and added to his or her profile. By counting how many times each user listens to each song, implicit user ratings can be obtained from these "scrobbles". Since such ratings can easily be interpreted as membership degrees, they can be used to associate a fuzzy set of users with each song. For instance, the users that listened the most to the song in question can be given membership degree 1, and the membership degrees of the remaining users can be determined by dividing their play counts by the greatest play count, e.g., if there are only three users and their play counts for a particular song are 10, 5, and 1, then the membership degrees of these users in the fuzzy set for this song would be 1, 0.5, and 0.1, respectively. Comparing such fuzzy sets by means of the cosine similarity measure

[2]http://www.pandora.com

00s **alternative** **alternative rock** amazing awesome ballad **beautiful** best songs ever bittersweet brit pop brit rock **british** **britpop** calm chill **chillout** classic rock **coldplay** easy listening emo emotional favorite favorite songs **favorites** favourite songs favourites **indie** indie rock **love** love songs lovely male vocalists melancholic **melancholy** **mellow** nostalgic **piano** piano rock **pop** pop rock relax relaxing **rock** **sad** scientist sentimental slow soft rock uk wicker park

(a) Tag cloud for "The Scientist" by "Coldplay".

60s **70s** 80s acoustic alternative anthem anti-war awesome ballad beatles **beautiful** best songs ever **british** britpop chill classic **classic rock** easy listening favorite favorite songs favorites favourites folk genius imagine **john lennon** legend legendary lennon love male vocalist male vocalists melancholic melancholy mellow **oldies** peace **piano** political **pop** pop rock psychedelic relax **rock** rock ballad sad **singer-songwriter** slow soft rock **the beatles**

(b) Tag cloud for "Imagine" by "John Lennon".

Figure 1: Two examples of tag clouds.

is equivalent to the basic "item-to-item collaborative filtering" recommendation algorithm [3].

Last.fm also uses social tagging [4] as an additional community-driven way to generate descriptions. Social tags are individual keywords or phrases that people associated with objects, usually for organizational purposes. When the tags applied by a large community are combined, rich and complex descriptions emerge. A popular way to visualize such a description is by means of a tag cloud, i.e., an alphabetically ordered list of tags in which the font size is proportional to the number of people that associated the tag in question with the object. Fig. 1 shows two examples. By interpreting the font sizes in a tag cloud as membership degrees, we obtain a fuzzy set of tags that describes the corresponding object.

2.3 *From computers*

Using algorithms that analyze audio signals, song descriptions can be obtained in a fully automatic way [5]. However, these descriptions are usually very low-level and difficult to interpret. A possible solution for this problem is to teach computers how to add high-level annotations to songs by feeding the automatically generated low-level descriptions into machine learning algorithms that were trained on descriptions from experts or communities [6, 7], or by developing algorithms that propagate annotations by analysing the low-level descriptions [8]. Although it is certainly not impossible to convert the low-level descriptions to fuzzy sets [9], it makes more sense to apply such a conversion to the high-level ones obtained by incorporating machine learning techniques. Since these automatically generated high-level descriptions are intended to resemble the descriptions generated by humans, they can be converted to fuzzy sets as explained above.

3 The Benefits

In the 40 years since Zadeh introduced the concept of a fuzzy set [10], tens of thousands of papers and plenty of books have been written on this seminal topic, and more than 20 international journals were established in the corresponding field. The general advantage of representing songs as fuzzy sets is

alternative awesome ballad **beautiful** best songs ever **british** britpop chill classic rock easy listening favorite favorite songs favorites favourites love male vocalists melancholic melancholy mellow **piano** **pop** pop rock relax **rock** sad slow soft rock

Figure 2: Intersection of the tag clouds shown in Fig. 1.

that we can rely on this vast amount of research for dealing with such representations. In this section, we describe a few more concrete benefits that ensue from this general one.

3.1 *Fuzzy aggregation operators*

Fuzzy aggregation operators are mathematical objects that aggregate multiple membership degrees into one. Formally, a fuzzy aggregation operator $\mathcal{H}$ of arity n, with $n \in \mathbb{N} \setminus \{0\}$, is an increasing $[0,1]^n \to [0,1]$ mapping that satisfies $\mathcal{H}(0,0,\ldots,0) = 0$ and $\mathcal{H}(1,1,\ldots,1) = 1$. Triangular norms [11], Quasi-arithmetic means [12], Sugeno and Choquet integrals [13, 14], and ordered weighted averaging (OWA) operators [15] are just a few examples of fuzzy aggregation operators that could be valuable for combining fuzzy descriptions of music objects. We refer to [16] for an introductionary overview of the extensively studied field of fuzzy aggregation operators. Fig. 2 provides a very simple example. It shows the result of applying the minimum operator to compute the intersection of the tag clouds in Fig. 1, which could be used as an intuitive visualization of the similarities between the songs that correspond with these tag clouds.

3.2 *Fuzzy comparison measures*

Expressions like, e.g.,

$$\frac{|A \cap B|}{|A \cup B|}, \qquad \frac{2\,|A \cap B|}{|A| + |B|}, \qquad \text{and} \qquad \frac{|A \cap B|}{\sqrt{|A|\,|B|}}$$

are commonly used to determine the similarity between two (ordinary) sets A and B. By defining the cardinality of a fuzzy set A in U as $|A| = \sum_{u \in U} A(u)$ and generalizing the classical set-theoretic operations to fuzzy sets, such expressions can be extended to similarity measures for fuzzy sets [17]. More generally, a measure that compares crisp sets can often be extended to fuzzy sets. We call such a measure a fuzzy comparison measure. A wide plethora of fuzzy comparison measures can be found in the literature, and the properties of many of these measures have been studied extensively. For instance, the measures M for which $1 - M$ satisfies the triangle inequality have been characterized for a broad family of fuzzy similarity measures [18], which can be very useful in practice since the triangle inequality allows to reduce the required number of comparisons [19]. Another family of fuzzy comparison measures that has practical value is the one developed in [20, 21, 22], since it can be used to systematically construct measures that satisfy certain properties. We rely on this family in the practical example described in Section 4.

3.3 Fuzzy relations

A (binary) fuzzy relation on U is a $U \times U \rightarrow [0,1]$ mapping that associates a membership degree with each pair of elements from U, i.e., a fuzzy set in $U \times U$. Since fuzzy comparison measures are usually fuzzy relations on the (ordinary) set of all possible fuzzy sets, the approach described in the previous subsection has the additional advantage that the obtained similarities are also membership degrees, which implies that fuzzy aggregation operators can be used to combine them. For example, a fuzzy relation that models the similarities between the songs in the considered collection is a prerequisite of the general framework for defining dynamic playlist generation heuristics introduced in [23], because it relies on fuzzy set theory to make the definitions systematic, formal, and intuitively clear. Since music similarity measures based on a fuzzy similarity measure are fuzzy relations already, we do not have to apply any normalization procedures on them in order to be able to use this framework.

4 Practical example

In this section, we describe a practical example that illustrates the potential of representing songs as fuzzy sets. More precisely, we explain how we relied on fuzzy set theory to build the “Multi Tag Search”, a web application that allows users to find music by supplying one or more tags. Fig. 3 shows the first results returned by this application when popular artists that match with the tags “british” and “rock” are requested. Any set of keywords or phrases can be used to search for either songs or artists, and the options “up-and-coming” and “free downloads” are also available, in addition to “popular”.

4.1 Basic method

Putting it in one sentence, the “Multi Tag Search” ranks all relevant music objects from Last.fm’s database by representing the query as a fuzzy set of tags and comparing this fuzzy set with the fuzzy tag set associated with each object. More formally, the ranking score for object i is given by:

$$score(\widetilde{Q}, \widetilde{O}_i) = w(i)\, M(\widetilde{Q}, \widetilde{O}_i) \tag{1}$$

with $\widetilde{Q}$ and $\widetilde{O}_i$ the fuzzy sets corresponding to the query and object i, respectively, M a fuzzy comparison measure, and w a function that associates a real number with each object index. The weighting function w allows us to push certain objects up or down in the generated ranking. It is mainly used to implement the “popular”, “up-and-coming”, and “free downloads” search options, but we also utilize it for some small optimizations like penalizing objects to which only very few tags have been applied.

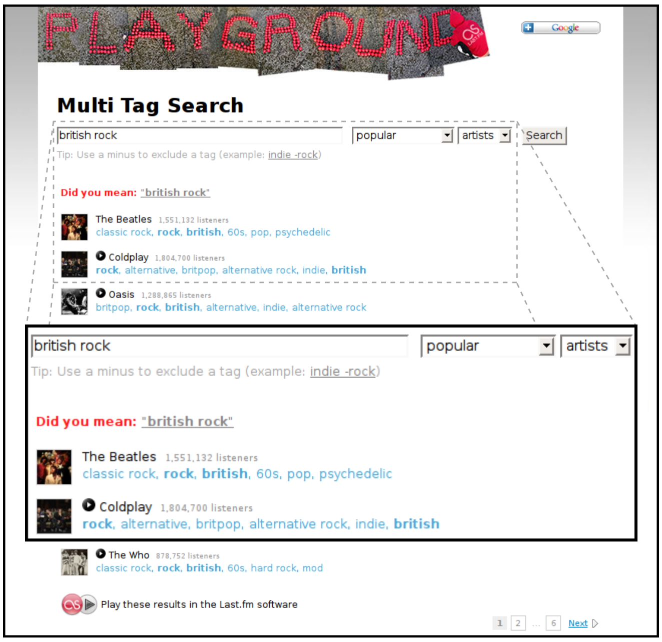

Figure 3: Screenshot of the “Multi Tag Search”.

We used the following general definitions of the fuzzy tag sets $\widetilde{Q}$ and $\widetilde{O}_i$:

$$\widetilde{Q}(t) = \begin{cases} \dfrac{q(t)}{\max_{u \in T} q(u)} & \text{when } t \in Q \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$\widetilde{O}_i(t) = \begin{cases} \dfrac{o_i(t)}{\max_{u \in T, j \in I} o_j(u)} & \text{when } t \in O_i \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

for all $t \in T$ and $i \in I$, with T the universe of all possible tags, I the set of all object indexes, Q the set of query tags, O_i the set consisting of all tags that have been applied to object i, and q and o_i, for each $i \in I$, a $T \rightarrow \mathbb{R}$ mapping. For the values of the parameters q and o_i in these definitions, we drew some inspiration from text retrieval, namely, we put $q = idf$ and $o_i = ntf_i$, with idf the so called “inverse document frequency”, and ntf_i the “normalized term frequency” [24]:

$$idf(t) = \log\left(\frac{\text{total number of objects}}{\text{number of objects tagged with } t}\right) \tag{4}$$

$$ntf_i(t) = \frac{\text{number of taggings with } t \text{ for object } i}{\text{total number of taggings for object } i} \tag{5}$$

for each $t \in T$ and $i \in I$. Hence, tags that are not frequently used get a high membership degree in $\widetilde{Q}$, and the membership degree of the tags in $\widetilde{O}_i$ is proportional to the number of times they have been applied to object i. Many other term-weighting schemes can be found in the text retrieval literature [25], but using the inverse document frequency for the query and the normalized term frequency for the objects appeared to work very well for Last.fm’s tag data.

4.2 Constructing a suitable comparison measure

High-quality tag data is by far the most important requirement for a good tag-based search engine. Improving the quality of the underlying data is generally more worthwhile than trying to find a better comparison measure, and which comparison measure is used exacly becomes less important when the tag data is of high quality. "Worry about the data first before you worry about the algorithm," as Google's director of research Peter Norvig puts it [26]. Nevertheless, it is of course important to use a suitable comparison measure that has the right properties, but it makes sense to choose the simplest possible measure that satisfies these properties, especially when you have to deal with millions of objects and users.

For the "Multi Tag Search", we wanted the fuzzy comparison measure M to satisfy the following properties:

$$M(A,B) = 0 \Leftarrow \operatorname{supp} A \cap \operatorname{supp} B = \emptyset \qquad \text{(exclusive)}$$
$$M(A,B) = 0 \Rightarrow \operatorname{supp} A \cap \operatorname{supp} B = \emptyset \qquad \text{(coexclusive)}$$
$$M(A,B) = M(A/_{\operatorname{supp} A}, B/_{\operatorname{supp} A}) \qquad \text{(left-restrictable)}$$

for all A and B from the class $\mathcal{F}(U)$ of all fuzzy sets in U, with $\operatorname{supp} A$ the support of $A \in \mathcal{F}(U)$, i.e., $\operatorname{supp} A = \{u \in U \mid A(u) > 0\}$, and $A/_V$, for $A \in \mathcal{F}(U)$ and $V \subseteq U$, the restriction of A to V, i.e., the $V \to [0,1]$ mapping that associates $A(v)$ with each $v \in V$. Putting it in words, we want the value returned by M to be 0 if and only if none of the query tags have been applied to the object in question, and we do not want the returned value to depend on the membership degrees of the tags that are not part of the query. The latter is particularly important, mainly because it is advantageous from a computational point of view, but also because it ensures that we cannot obtain different values for objects with fuzzy tags sets in which each query tag always has the same membership degree. For instance, if we would use the well-known cosine similarity measure, which is not left-restrictable, then the query $Q = \{\text{"british"}, \text{"rock"}\}$ would lead to a different value for the fuzzy sets obtained by directly interpreting the font sizes in the tag clouds shown in Fig.1 as membership degrees, while there really is no reason to prefer one over the other in this case since the font sizes for "british" and "rock" are exactly the same in both clouds. In the final ranking, one of the corresponding songs would have to be put before the other of course, but we want that decision to only depend on the value returned by w.

The family of fuzzy comparison measures introduced in [21], which includes the popular cosine similarity measure, makes it very easy to systematically construct fuzzy comparison measures that satisfy given properties. All members of this family are instances of a triparametric general form, and several properties can be ensured by imposing constraints on the parameters $\mathcal{H}$, φ_1, and φ_2 of this general form, where $\mathcal{H}$ is a binary fuzzy aggregation operator, i.e., an increasing $[0,1]^2 \to [0,1]$ mapping such that $\mathcal{H}(0,0) = 0$ and $\mathcal{H}(1,1) = 1$, and φ_1 and φ_2 are two $[0,1]^3 \to \mathbb{R}$ mappings.[3] As value of the first parameter $\mathcal{H}$, we chose the product t-norm T_{P} given by $T_{\mathrm{P}}(x,y) = x \cdot y$, for all $x,y \in [0,1]$, because it is a very simple operator that is interactive [27], i.e., a modification of x or y always implies an alteration of $T_{\mathrm{P}}(x,y)$ if $x \neq 0$ and $y \neq 0$. The general form from [21] then reduces to:

$$M(A,B) = \frac{\varphi_1(\|A \cap_{T_{\mathrm{P}}} A\|, \|B \cap_{T_{\mathrm{P}}} B\|, \|A \cap_{T_{\mathrm{P}}} B\|)}{\varphi_2(\|A \cap_{T_{\mathrm{P}}} A\|, \|B \cap_{T_{\mathrm{P}}} B\|, \|A \cap_{T_{\mathrm{P}}} B\|)} \qquad (6)$$

for all $A, B \in \mathcal{F}(U)$, with $\|.\|$ the $\mathcal{F}(U) \to [0,1]$ mapping given by $\|A\| = |A|/|U|$ for all $A \in \mathcal{F}(U)$, and $\cap_{T_{\mathrm{P}}}$ an infix notation for the pointwise extension of T_{P}, i.e., $(A \cap_{T_{\mathrm{P}}} B)(u) = T_{\mathrm{P}}(A(u), B(u))$ for all $u \in U$, with A and B fuzzy sets in U. Using the proofs provided in [21], it can easily be verified that this $\mathcal{F}(U) \times \mathcal{F}(U) \to \mathbb{R}$ mapping is an exclusive, coexclusive, and left-restrictable fuzzy comparison measure when the formulas

$$z \leq \sqrt{x \cdot y} \implies 0 \leq \varphi_1(x,y,z) \leq \varphi_2(x,y,z)$$
$$\varphi_1(x,y,0) = 0 \qquad \varphi_1(x,u,z) = \varphi_1(x,v,z)$$
$$\varphi_1(\dot{x},\dot{y},\dot{z}) > 0 \qquad \varphi_2(x,u,z) = \varphi_2(x,v,z)$$

are all satisfied for every $x,y,z,u,v \in [0,1]$ and $\dot{x},\dot{y},\dot{z} \in]0,1]$. The simplest solution that we could find for this is $\varphi_1(x,y,z) = z$ and $\varphi_2(x,y,z) = 1$, for all $x,y,z \in [0,1]$, which leads to:

$$M(A,B) = \frac{\|A \cap_{T_{\mathrm{P}}} B\|}{1} = \frac{|A \cap_{T_{\mathrm{P}}} B|}{|U|} \qquad (7)$$

for all $A, B \in \mathcal{F}(U)$, i.e., a fuzzification of the Russell and Rao coefficient [28]. We then have

$$\begin{aligned} M(\widetilde{Q}, \widetilde{O}_i) &= M(\widetilde{Q}/_{\operatorname{supp}\widetilde{Q}}, \widetilde{O}_i/_{\operatorname{supp}\widetilde{Q}}) && (M \text{ left-restr.}) \\ &= M(\widetilde{Q}/_Q, \widetilde{O}_i/_Q) && (\operatorname{supp}\widetilde{Q} = Q) \\ &= \frac{1}{|Q|} \sum_{t \in Q} \widetilde{Q}(t)\, \widetilde{O}_i(t) && (\text{definition } M) \\ &\propto \sum_{t \in Q} \mathit{idf}(t)\, \mathit{ntf}_i(t) && (\text{def. } \widetilde{Q} \text{ and } \widetilde{O}_i) \end{aligned}$$

and thus:

$$\mathit{score}(\widetilde{Q}, \widetilde{O}_i) \propto w(i) \sum_{t \in Q} \mathit{idf}(t)\, \mathit{ntf}_i(t) \qquad (8)$$

Hence, we merely need to compute $w(i) \sum_{t \in Q} \mathit{idf}(t)\, \mathit{ntf}_i(t)$ to determine the ranking of object i for query Q.

4.3 User feedback

Last.fm's Playground was launched in May 2008, with the "Multi Tag Search" as one of the initial demonstrations. Since then, this demonstration has rather consistently been the most popular one over a period of several months. It also seemed to be more "sticky" than the other initial demonstrations, since it was the only one for which the number of page views increased after launch. Moreover, we received substantially more positive direct user feedback for it than any of the other initial demonstrations. Hence, the users seemed to like the "Multi Tag Search" quite a lot, which suggests that it must at least work reasonably well. The great variety in the supplied queries confirms this observation. We received about 25 thousand unique queries so far, and, as illustrated by Table 1, several different types of tags occur in these queries.

[3] We required φ_1 and φ_2 to be increasing in their first and second argument in [21], but since the proofs for the constraints considered in this paper do not rely on this prerequisite, we can omit it here.

Table 1: Results of a manual classification of the 250 most frequently occurring query tags.

Type	Frequency	Examples
Genre	51%	rock, pop
Style	26%	experimental
Locale	7%	british, detroit
Instrument	5%	piano, guitar
Mood	4%	sad, romantic
Time	3%	90s, oldies
Opinion	2%	beautiful, love
Other	2%	free, cover

5 Conclusion

We have shown that fuzzy set theory has a lot to offer to the music information retrieval community. By representing music objects as fuzzy sets, the possibility arises to exploit the rich mathematical toolset provided by this theory, including fuzzy aggregation operators, fuzzy comparison measures, and fuzzy relations. As a practical example, we explained how we employed fuzzy set theory to build the popular "Multi Tag Search" demonstration on Last.fm's Playground. Using previous work on fuzzy comparison measures, we were able to systematically construct a suitable comparison measure for this tag-based search engine, instead of merely relying on intuition and trial-and-error. Rather than arbitrarily choosing one of the many comparison measures available in the literature and checking if it works as expected, we started from the desired properties and systematically constructed a simple and computationally efficient measure that satisfies these properties. Since different applications have different requirements, such a systematic way of constructing comparison measures can be very valuable for building applications in which objects need to be compared, as is commonly the case in music information retrieval as well as in certain other domains.

Acknowledgement

The first author thanks the Fund for Scientific Research–Flanders (FWO) for funding this research.

References

[1] W. Glaser, T. Westergren, J. Stearns, and J. Kraft. Consumer item matching method and system. US Patent Number 7003515, 2006.

[2] D. Turnbull, L. Barrington, and G. Lanckriet. Five approaches to collecting tags for music. In *Proceedings of the ISMIR International Conference on Music Information Retrieval*, 2008.

[3] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7:76–80, 2003.

[4] P. Lamere. Social tagging and music information retrieval. *Journal of New Music Research*. To appear.

[5] E. Pampalk. *Computational models of music similarity and their application in music information retrieval*. PhD thesis, Vienna University of Technology, 2006.

[6] D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green. Automatic generation of social tags for music recommendation. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in neural information processing systems 20*. MIT Press, 2008.

[7] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE Transactions on Audio, Speech, and Language Processing*, 16:467–476, 2008.

[8] M. Sordo, C. Lauier, and O. Celma. Annotating music collections: How content-based similarity helps to propagate labels. In *Proceedings of the ISMIR International Conference of Music Information Retrieval*, 2007.

[9] K. Bosteels and E. Kerre. Fuzzy Audio Similarity Measures Based on Spectrum Histograms and Fluctuation Patterns. In A.-E. Hassanien, A. Abraham, and J. Kacprzyk, editors, *Computational Intelligence in Multimedia Processing: Recent Advances*. Springer, 2008.

[10] L. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[11] E. Klement, R. Mesiar, and E. Pap. *Triangular norms*. Kluwer Academic Publishers, 2000.

[12] J. Aczel. On mean values. *Bulletin of the American Mathematical Society*, 54:392–400, 1948.

[13] M. Sugeno. *Theory of fuzzy integrals and its application*. PhD thesis, Tokyo Institute of Technology, 1974.

[14] G. Choquet. Theory of capacities. *Annales de L'Institut Fourier*, 5:131–295, 1953.

[15] R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Transactions on Systems, Man and Cybernetics*, 18:183–190, 1988.

[16] M. Detyniecki. Numerical aggregation operators: State of the art. In *International Summer School on Aggregation Operators and their Applications*, 2001.

[17] B. De Baets and H. De Meyer. Transitivity-preserving fuzzification schemes for cardinality-based similarity measures. *European Journal of Operational Research*, 160(3):726–740, 2005.

[18] S. Janssens. *Bell inequalities in cardinality-based similarity measurement*. PhD thesis, Ghent University, 2006.

[19] J. Barros, J. French, W. Martin, P. Kelly, and T. Cannon. Using the triangle inequality to reduce the number of comparisons required for similarity-based retrieval. In *Storage and Retrieval for Image and Video Databases*, 1996.

[20] K. Bosteels and E. Kerre. A biparametric family of cardinality-based fuzzy similarity measures. *New Mathematics and Natural Computation*, 3:307–319, 2007.

[21] K. Bosteels and E. Kerre. A triparametric family of cardinality-based fuzzy similarity measures. *Fuzzy Sets and Systems*, 158:2466–2479, 2007.

[22] K. Bosteels and E. Kerre. On a reflexivity-preserving family of cardinality-based fuzzy comparison measures. *Information Sciences*. To appear.

[23] K. Bosteels and E. Kerre. Fuzzy heuristics for dynamic playlist generation. Submitted.

[24] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24:513–525, 1988.

[25] H. Blanken, A. de Vries, H. Blok, and L. Feng, editors. *Multimedia retrieval*. Springer, 2007.

[26] P. Norvig. Theorizing from data: Avoiding the capital mistake. CITRIS Distinguished Speaker Series, UC Berkeley, 2006.

[27] D. Dubois and H. Prade. *Fuzzy sets and systems: Theory and applications*. Academic Press, 1980.

[28] B. Sarker. The resemblance coefficients in group technology: A survey and comparative study of relational metrics. *Computers and Industrial Engineering*, 30:103–116, 1996.