Abstract. The increasing popularity of music recommendation systems and the recent growth of online music communities further emphasizes the need for effective playlist management tools able to create, share, and personalize playlists. This paper proposes the development of generic playlists and presents a concrete scenario to illustrate their possibilities. Additionally, to enable the development of playlist management tools, a formal foundation is provided. Therefore, the concept of fuzzy lists is defined and a corresponding algebra is developed. Fuzzy lists offer a solution perfectly suited to meet the demands of playlist management.

1 Introduction

The proliferation of broadband Internet connections and the development of new digital music formats have led to the explosion of online music communities and music recommendation systems. The increasing popularity of these systems has created a strong demand for the development of playlist manipulation engines. Hence, playlist models are highly needed. However, playlists are by nature imprecise. One song could probably be replaced by another while preserving the essence of the playlist. Similarly, two songs can sometimes be permuted. The way playlists are built explains this phenomenon. For individual music lovers, the manual construction of a playlist results in some kind of consensus between the various aspects defining the songs. In large automated music recommendation systems, user collaborative filtering and co-occurrence analysis approaches are commonly used to construct playlists. While imprecise, each playlist can exactly characterize a trend, a dynamic, a mood. Additionally, playlists also have a subjective nature, i.e., they are highly dependent on their audience. Listeners might have strong musical preferences or may not have access to all the music. Therefore, playlist management tools have to include personalization mechanisms.

The contributions of this paper are twofold. (i) A new scheme for constructing and sharing playlists is described via a concrete scenario. The scenario illustrates how the imprecise and subjective characteristics of playlists can be handled in order to improve playlist management engines. (ii) Fuzzy lists, a generalization of lists and fuzzy sets, are defined and their corresponding algebra is developed. The proposed algebra is inspired by relational algebra in order to facilitate future implementation in an RDBMS. Fuzzy lists offer the formal foundation for the development of playlist management tools as the provided examples illustrate.

The remainder of this paper is organized as follows. The creation of fuzzy lists for playlist management is motivated by a scenario presented in Section 2. Section 3 discusses the related work. Section 4 provides a formal definition of fuzzy lists, their operators and functions. Concrete playlist manipulation examples are shown to underline
their utility. Finally, after evoking some implementation considerations in Section 5, conclusion and future work are presented in Section 6.

2 Motivation

In this section, a playlist management engine that respects both the imprecise and subjective nature of playlists is envisioned. The users of the system want to create and share playlists. The playlists are built by user communities in a collaborative fashion rather than by individual users. Automated classification systems could possibly be incorporated as well as classical users participating in the playlist building process. Additionally, the generated playlists are adaptable to the users’ profiles to respect their musical taste. The setup is as follows.

The objective is to create a playlist composed of a given number of songs and a theme. For example, 100 users are asked to build a playlist composed of ten songs, the first 3 songs should be rock, the next 3 should sound jazzy, the following 3 romantic, and the last one should be a blues song. Furthermore, the users are asked to provide smooth transitions between the songs, e.g., the third rock song should sound a bit jazzy. Agreement between the different users is achieved thanks to a voting mechanism. Finally, independently from the playlist building process, registered users have lists of songs they like and dislike.

The playlists created by the voters are merged into a single playlist, referred to as a generic playlist. The generic playlist stores the “election score,” referred to as the membership degree, obtained for each song and each position. The generic playlist can then be shared among all the registered users of the system. However, the generic playlist cannot be used directly by the users as for each position in the playlist many songs will probably coexist. Furthermore, if some songs have been previously tagged as the user’s favorites then they should preferably be played, or if the user has no access to a song, an accessible song should be alternatively chosen. The personalization mechanism selects songs from the generic playlist with respect to the user preferences and constraints. For each possible song, a preference grade is given; songs with a high grade should preferably be played while songs with a low grade should be played less often. A selection score is calculated using the user’s preference list and the generic playlist.

![Figure 1](image_url)

**Fig. 1.** Generic Playlists Usage Scenario