ITC
irst

CENTRO PER LA RICERCA
SCIENTIFICA E TECNOLOGICA

38050 Povo (Trento), Italy
Tel.: +39 0461 314312
Fax: +39 0461 302040
e−mail: prdoc@itc.it − url: http://www.itc.it

COLLABORATIVE RADIO COMMUNITY

Avesani P., Massa P.,
Nori M., Susi A.

April 2002

Technical Report # 0204−07

LIMITED DISTRIBUTION NOTICE

This report has been submitted for publication outside of ITC and will probably be copyrighted if accepted for publication. It has been issued as a Technical Report for early dissemination of its contents. In view of the transfert of copy right to the outside publisher, its distribution outside of ITC prior to publication should be limited to peer communications and specific requests. After outside publication, material will be available only in the form authorized by the copyright owner.

Collaborative Radio Community

Paolo Avesani, Paolo Massa, Michele Nori, Angelo Susi

ITC-IRST,
Via Sommarive 18 - Loc. Pantè, I-38050 Povo, Trento, Italy
{*avesani,massa,nori,susi*}*@irst.itc.it*

Abstract. Recommender systems have been usually designed to support a single user in a one-to-one relation between a human and a service provider. This paper presents a collaborative radio community where the system delivers a personalization service on the fly, on the basis of the group recommending, promoting a shift from the one-to-one approach to a one-to-group scenario where the goal is assisting people in forming communities.

Keywords: recommender systems, web radio, learning preferences, multicast streaming, virtual community.

1 Introduction

Usually, recommender systems have been designed to support a single user in a one-to-one relation between a human and a service provider. Although advices are generated on the basis of the opinions of other users, the system doesn't support a direct relation between two users that play the different roles of recommender and recommended. Moreover, current personalization systems distinguish between the recommendation step and the use step (i.e. listening in the case of music).

The new technological landscape concerning connectivity has been exploited to conceive user centered services as Smart Radio [6], where an entire radio channel is devoted to a single user to deliver a fully personalized program. Very often in the past this emphasis on personalization has penalized the advantages that could arise from the interaction of a community of users. More recently a new awareness is developing that considers helping people to help each other a new challenge for recommender systems [7]. In this perspective we promote a shift from a one-to-one approach to a one-to-group-of-many scenario where the goal is assisting people in forming communities.

At the technological level the “group-of-many” can be managed with a peer-group approach that is receiving growing attention as new protocols become available e.g. JXTA [5].

In this paper we present a collaborative radio community where, taking advantage of a low band multicast streaming, the system delivers a personalization service on the fly devoted to group recommending. The users are involved both as listeners and as recommenders. The live interaction allows the users to elicitate their disagreement on the radio program. A different preference can be formulated providing alternative order relation among the sountracks. This kind of preferences are closely related to the current theme of radio program: in this context, a theme plays the role of a potential new category of music that should inform the selection of the contents.

Learning how to summarize partial order relations associated to a given theme into a global preference model allows the detection of new kind of non standard categories of music, i.e. the theme. This feature extends our previous work on *CoCoA* [1] a Compilation Compiler Advisor based on case-based reasoning, that supports the detection of the genres of use (from which the name *CoCoA-Radio* comes).

In the following we describe how *CoCoA-Radio* works and show a brief overview of the learning issues that arise when a community of users has to be recommended.

2 The Application

CoCoA-Radio is a *thematic interactive community* radio. It streams continuously radio programs of a fixed number of MP3s (usually 20); every radio program is relative to a given theme. The radio program is the same for every user of the community and can be thought as an hits list for the current theme where the songs at the top level can be considered more representative of this theme. Users can interact with *CoCoA-Radio* using a web browser expressing their preferences related to what they think it is the hits list for the scheduled theme. In this way they play the main roles of authoring, recommending and listening.

Let us introduce the main concepts of the radio. A *theme* is just a mnemonic label that refers to a common target feeling about music or a perspective to look at the music; themes can range from traditional “pop” or “women rock” to more fuzzy and undefined “hands moving”. A *playlist* is a user defined list of songs to be submitted to the radio. A *program* is the current list of songs that *CoCoA-Radio* is playing; it is relative to a theme and is a synthesis of the submitted playlists.

Let us now describe in details about how *CoCoA-Radio* works. It streams music programs for given themes continuously. The user can contribute to the radio program related to the next theme by submitting a playlist as a proposal. The system is in charge of summarizing a radio program taking into account the contributions of the users. The challenge is to schedule a radio program that best fits the scheduled theme and consequently best satisfies the listeners feeling.

It is important to stress that the goal of the radio is not to satisfy the user requests but the user expectation concerning the music for the current theme. We could argue that there are two mutual expectations. First, given a theme the users are constrained to submit playlists related to this theme. Second, the users can give their feedback, specifying the relative order of couples of songs in the syntesized program, when the program doesn’t satisfy their expectation on the music related to the given theme.

It is a goal of the system to adaptively modify its schedule in order to satisfy the users’ expectation and consequently to minimize the users’ feedback.

With *CoCoA-Radio*, what can be seen as a minus (the fact that every user gets the same music) becomes a plus; infact users can benefit of the presence of other users because they are all together concurring to form the hit lists for a theme. In essence *CoCoA-Radio* is a *social* application because users have a common goal and they are supposed to work in a cooperative manner to achieve it. Moreover, in a certain way users exploit their reciprocal musical knowledge. Of course this means that there should be a minimum agreement among users; but this could be overcome by replicating channels and clustering users on them depending on their musical preferences; this poses another interesting challenge.

3 Multicast streaming

Another advantage of streaming a single line of music for a community is the possible bandwidth saving. *CoCoA-Radio* is intended for (but not limited to!) Intranet LAN use and, for this reason, we used multicast IP to stream music. In this way, our streamer does not establish a connection with every client but sends a single stream of packets on the net in multicast; thus every machine that needs to play the music gets this packet. One can imagine the great bandwidth saving with, for example, 100 users! The protocol used for streaming is RTP[1] (short for Realtime Transport Protocol). It is an Internet protocol specifically designed for transmitting realtime data such as audio and video. Typically, RTP runs on top of the UDP protocol and so it allows multicast transmission.

4 Learning issues

In the following we describe some interesting open learning issues in the radio domain. In general users refer to music using categories that don't belong to the standard taxonomy of music genres and genres of use are not stable during the time, so a possible goal is to recognize which tracks can be classified under a given theme and how much a track is representative for it. Moreover the synthesis of the program starting from a collection of submitted playlists is a satisfiability problem that usually does not have solution due to possible unsolvable conflicts between users' playlists; in the context of the radio community it could be reasonable to avoid this problem detecting the reliability of the users and assigning less priority to playlists submitted by the less reliable of them. Both these aspects are analyzed in the learning algorithm included in *CoCoA-Radio* described in the next section.

5 Radio program synthesis

The *CoCoA-Radio* domain can be described as a set of songs to be ordered called $S = \{s_1, \ldots, s_N\}$, a set of radio listeners called $U = \{u_1, \ldots, u_M\}$, a set of themes, i.e. music categories $C = \{c_1, \ldots, c_L\}$, and a *ranking function R* (possibly induced from a playlist *P*) that is proposed by a user u_i on a given theme c_j, $R_{u_i}^{cj} : S \times S \rightarrow [0, 1]$, where $R_{u_i}^{cj}(s_1, s_2) = 1$ is a strong recommendation that s_1 should be ranked above s_2, 0 the opposite; a value closed to $1/2$ is interpreted as an abstention from making recommendation.

Starting from the user preference functions R_{u_i} can be derived a *preference function* of the form: $PREF(s_1, s_2) = \sum_{i=1}^{M} w_{u_i} R_{u_i}(s_1, s_2)$ where w_{u_i} are weights assigned to each listener u_i, determining the reliability of the user u_i respect to the given music theme c. A learning algorithm [2–4] can be designed to update incrementally the weight values.

Learning is assumed to take place in a sequence of rounds. On each round, we assume the learning algorithm is provided with a set of S^t of songs to be ranked. A combined preference function $PREF^t$ is computed and than used to produce a total ordering function $\rho_{c_j}^t$ said *Program* of songs S^t. After producing $\rho_{c_j}^t$ the radio receives *feedback* from the listeners. The feedback at time t, F^t, is a set of assertions "song s_1 should be preferred to song s_2" so F^t is a set of pairs (s_1, s_2). The feedback allows

[1] http://www.ietf.org/rfc/rfc1899.txt

to compute the function $Loss(R, F)$ that describes the loss of a preference function R given by a user respect to the users' feedback. The loss is the major component in the incremental adequacy of the weights w_{u_i} given by the learning algorithm to every radio user during the learning procedure.

At the end of the procedure for each theme c_j we have an order for the instances in S called ρ_{c_j}. This order is a sequence of instances. It is possible to map it in the Radio domain calling it *Program*, a sequence of S elements.

The *CoCoA-Radio* interaction can be summarized in three phases: *An initial phase of playlists submission*, given the set of possible songs in the database and a theme, users can submit their own playlist (an ordered list of songs) related to the given theme; *A phase of synthesis of a program on the basis of the submitted playlists*; *A final phase of users' feedback acquisition*, users can give information about the synthesized order in terms of songs pairs $< s_1, s_2 >$; the program will be then recomputed on the basis of the received feedback.

6 Future work

At the time of writing we have developed a first version of *CoCoA-Radio* using a J2EE software platform. The system has been deployed on a LAN with a few hundreds hosts and an archive of 6000 mp3 tracks.

Our primary goal in the short term is to test our architecture on the field with a restricted and controlled community of users, i.e. the researchers of our institute. We are interested in assessing on a real interactive environment both the technological and the collaborative architectures. The main focus of our experiment is the analysis of the learning curve at run time; our goal is to assess the relation between the feedback from users and the learning process. In this context, not only accuracy is important but also how fast is the convergence of the process.

A secondary goal is related to the acquisition of a meaningful dataset taking advantage of the contribution of the real users. This dataset could be exploited to evaluate alternative working hypothesis off line.

Currently, we have two the planned enhancements to extend the application: the first is the opportunity for the user to submit a proposal for the next theme, the second is the replication of the *CoCoA-Radio* with an added service of recommendation to help the user to choose what kind of channel to subscribe.

References

1. S. Aguzzoli, P. Avesani, and P. Massa. Compositional CBR via Collaborative Filtering. In *ICCBR'01 Workshop on CBR in Electronic Commerce*, Vancouver - Canada, 2001.
2. William W. Cohen, Robert E. Schapire, and Yoram Singer. Learning to Order Things. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.
3. Y. Freud, R. Iyer, R. Schapire, and Y. Singer. An Efficient Boosting Algorithm for Combining Preferences, 1998.
4. Y. Freund and R. Schapire. A Short Introduction to Boosting, 1999.
5. L. Gong. Project JXTA: A Technology Overview, 2001.
6. C. Hayes and P. Cunningham. Smart radio: Building music radio on the fly, 2000.
7. Loren Terveen and Will Hill. *HCI In The New Millenium*, chapter Beyond Recommender Systems: Helping People Help Each Other. Addison-Wesley, 2001.