Using Closed-Set Speaker Identification Score Confidence to Enhance Audio-Based Collaborative Filtering for Multiple Users

Shepstone, Sven Ewan; Tan, Zheng-Hua; Kristoffersen, Miklas Strøm

Published in:
IEEE Transactions on Consumer Electronics

DOI (link to publication from Publisher):
10.1109/TCE.2018.2811250

Publication date:
2018

Document Version
Accepted author manuscript, peer reviewed version

Using Closed-set Speaker Identification Score Confidence to Enhance Audio-based Collaborative Filtering for Multiple Users

Sven E. Shepstone, *Member, IEEE*, Zheng-Hua Tan, *Senior Member, IEEE* and Miklas S. Kristoffersen, *Student Member, IEEE*

***Abstract*—In this paper we utilize a closed-set speaker-identification approach to convey the ratings needed for collaborative filtering-based (CF) recommendation. Instead of explicitly providing a rating for a given program, users use a speech interface to dictate the desired rating after watching a movie. Due to the inaccuracies that may be imposed by a state-of-the-art speaker identification system, it is possible to mistake a user for another user in the household, especially when the users exhibit similar or identical age and gender demographics. This leads to the undesirable effect of injecting unwanted ratings into the collaborative rating matrix, and when the users have different tastes, can result in the recommendation of undesirable items. We therefore propose a simple confidence-based heuristic that utilizes the log-likelihood scores from the speaker identification front-end. The algorithm limits the degree to which unwanted ratings negatively affect the integrity of the ratings information. Using real-speaker utterances over a range of age and gender demographics, we compare our approach against upper and lower-bound (non-speaker-identification-based) baseline systems. Results show that by taking the confidence into account of users, that we were able to improve upon the lower bound that unconditionally accepts ratings by a relative 6.9 %.**

***Index Terms*—collaborative filtering, confidence, i-vector**

I. Introduction

In the collaborative filtering paradigm, items, such as movies, are recommended based on similarities between users [1, 2, 3, 4, 5]. Users provide ratings for items, usually according to the well-known Likert scale [6], starred ratings [7], or more recently according to a simpler thumbs up / thumbs down strategy. From these ratings, for a given user, it is then possible to identify other users that have similar tastes. The most similar users form a so-called *neighborhood*, which ultimately allows previously unseen items to be recommended to people with similar tastes to the other users in the neighborhood. For example, if John happens to like spy movies, he might give high ratings to three movies A, B and C. Dave also happens to like spy movies, and has given fairly high ratings to the movies B, C and X. Since the recommender has no knowledge of what movies are spy movies, it cannot easily create content-based rules for example, to determine that there is an underlying common theme across both users[1]. However, by means of the high correlation between John and Dave's ratings, it quickly becomes apparent that they like the same movies. This makes it possible to recommend X to John and A to Dave. The fact that no knowledge of the actual items is required for collaborative systems, has made them hugely popular for commercial deployment in many sectors.

One area where collaborative filtering has been extensively utilized is in the streaming of movies to home users, where there are today a number of streaming services [8]. To improve future recommendations, people are typically assigned personal profiles (or sub-profiles), and given the opportunity to rate material. These ratings, as well as implicitly gathered information, such as viewing behavior, can over time help provide good recommendations to individual users [9]. However, in the process of watching a movie, few services distinguish between an audience that consists of only a single person, and one where multiple people (and hence multiple user profiles) could be present.

Very typically, the audience does indeed comprise more than one person. Consider for example, watching a movie together on a Friday night. The problem is that since only one login is utilized at a time, there is no easy way for everyone in the group to state their rating. This is primarily due to the ”session-based” approach that excludes the other (non-logged in) users from providing their ratings in a simple and effective manner. Even when each user in the household has their own sub-profile, the tediousness of switching in and out of sub-profiles can result in everyone using only a single person’s profile. Often, viewers are not even aware, or simply do not care, whose sub-profile is the one in use. In the group case, we cannot assume that everyone will express the same taste for a given movie, and especially in the case of larger groups, non-participation in ratings can lead to a large amount of unspecified ratings. When a user’s ratings are too scarce, the job of recommending something likable is less feasible since there is not a strong basis for correlating that user with other users’ ratings, thus only loosely connecting that user with others. Not using sub-profiles in the way intended, for

Work supported by Bang & Olufsen a/s.

S. E. Shepstone and M. S. Kristoffersen are with Bang & Olufsen a/s, Peter Bangs Vej 15, 7600 Struer, Denmark (e-mail: {ssh,mko}@bang-olufsen.dk).

Z-H. Tan is with the Department of Electronic Systems, Aalborg University, Fredrik Bajers Vej 7B, 9220, Aalborg (e-mail: zt@es.aau.dk).

[1] Content-based recommenders would have this kind of information.

example, person A watching person B's content, also has the downside that any content watched by others will affect any future recommendations for the person to whom the profile belongs. Looking at effective ways to help multiple viewers to easily contribute their ratings can help to provide the much-needed data for improving recommendation quality, and in the group context, a lot of research has already been devoted to this area [10], [11], [12], [13].

As machine intelligence moves forward, the line between human and machine is becoming blurred. For a long time now people have been making widespread use of speech-driven applications such as giving commands to smartphones for e.g. dictation, finding map directions and the like. Here, mostly it is the linguistic part, or content of speech that is relevant. The non-linguistic, or para-linguistic content of people's speech is however also very effective at conveying information about the speaker, where for example text- independent speaker recognition techniques can be used to detect their identity, or even age and gender. However, the paralinguistic component of speech is still somewhat overlooked in comparison to the actual content of speech when it comes to mainstream application deployment. Making use of people's speech in a collaborative filtering framework setting, where the non-linguistic speech components can be used to identify *who* spoke the rating, and not just the rating itself, can provide a powerful framework to address some of the shortcomings of the current methods. A major advantage could come about by replacing the session-based login paradigm with a sessionless one - as soon as someone is identified through their speech, their rating is immediately assigned to their profile, and can be subsequently used as input to the recommendation engine. This obviates the need for everyone needing to log on. In the group setting, after watching a movie, anyone and everyone in the group can now more easily state their rating.

In the speaker identification context, upon enrollment, each user in the household is assigned their own target model, and when someone speaks, the target model closest to the spoken speech identifies the user, making it fairly easy to identify each person in a group. Ideally, this identification happens at the local device itself, and without sending information identifying users to the cloud, to avoid privacy concerns. One problem with identifying people from their speech, however, is the probability of misidentifying a person in the household. The more speaker target models that need to be matched, the higher is the likelihood of misclassifying one person as another in the group. Furthermore, when the speaker target models are very close to one another in the speaker space, for example, with two children, the chance of a misidentification also increases. Finally, consider the use case in mind, i.e. personal recommendation, where each person rates a movie, and is then identified from their speech. The rating itself is most likely to just be a number from 1 to 5, embedded in a longer sentence, and resulting in a variable length speech utterance[2].

The above-mentioned issues can make using people's speech to provide recommendations more challenging since they increase the odds of inserting the mistaken person's ratings instead. Assume for example that John likes spy movies and Dave does not. After watching a good spy movie, John says that he would like to assign the movie he just watched a '5' rating. Now if John is mistaken for Dave, this will result in believing that Dave actually enjoys spy movies, when in fact, he does not. Even when users are of the same age and gender, there is no reason to believe that demographics and taste are mutually inclusive. Consider the case in point of two teenage girls, where one prefers thrillers and the other prefers drama. Whether or not we should use a person's ratings would seem to depend on how sure we can identify the person in mind.

Many works use and explore information derived from social, content and usage patterns to drive recommender systems, but few utilize users' speech in the recommender context. In [14], the authors compare a proposed text and speech-based natural language interfaces to request information from users in an open-ended manner. In [15], the authors utilize speech personality traits, extracted from acoustic and prosody features, to cold start a collaborative recommender system for providing recommendations to new users. There have been several works that have utilized text-independent attributes from speech to recommend items. In [16] the authors used age-and-gender profiles extracted from the speech of home users to recommend TV advertisements to them. In another study, emotions were extracted from speech and used in mood profiles to propose initial recommendations [17].

The contribution of this paper is a framework that uses people's speech to identify them in a group setting. By not having to individually select profiles or log-on, a sessionless recommender paradigm is enabled. The framework combines a speaker identification front-end with a recommender engine. To our knowledge, this is the first work that explicitly connects users' identity determined from their speech to their recommender-based profile.

We extend a classic and basic collaborative filtering framework with an additional frontend based on identifying people through a closed-set speaker identification approach. In closed-set speaker identification, all speaker models are assumed to be known upfront (there is no *unknown* category). In this work, we are primarily interested in the effect of known users being confused with one another, and therefore limit it to the closed-loop case. In a real-world setting, unknown users could be rejected if the likelihood of detection falls below a given predefined threshold. In the proposed system, people from a group simply consecutively state their ratings, during or after watching a movie. It is assumed that standard speech identification is used to extract the actual rating, a number from 1 to 5, and for the purpose of this study, we assume that the actual rating can be extracted from speech with a 100 % accuracy. The speech utterance is then reused to determine the speaker's identity. Depending on the group configuration, an additional confidence score is generated for each spoken

[2] Naturally, there would be additional context, such as a voice command prefix that can help to increase the robustness of identifying the user.

rating, and used to determine whether or not the rating should be admitted. For this work, the scope of the recommender algorithm will be limited to traditional collaborative filtering.

The rest of this paper is organized as follows: Sections II and III respectively introduce the collaborative filtering paradigm and speaker identification using i-vectors. Section IV introduces the proposed framework for applying the speaker confidence in a recommender context. The following section presents the experimental work that was carried out. Section VI discusses the results. The final two sections present conclusions and future work.

II. Collaborative Filtering

In the traditional collaborative filtering approach, each user i provides a set of ratings r_i. A simple, but effective method for determining the similarity between two users a and b is based on Pearson's correlation[3]. For a set of items J rated by both users a and b, it is given by:

$$\mathrm{Corr(a,b)} = \frac{\sum_{\mathrm{j}\in \mathrm{J}} (\mathrm{r_{a,j}} - m_a)(r_{b,j} - m_b)}{\sqrt{\sum_{\mathrm{j}\in \mathrm{J}} (\mathrm{r_{a,j}} - m_a)^2 (r_{b,j} - m_b)^2}} \tag{1}$$

where m_a and m_b are the mean ratings for user a and b respectively, which for user i is given as:

$$\mathrm{m_i} = \frac{1}{|\, r_i \,|} \sum_{j \in r_i} r_{i,j} \tag{2}$$

and where $r_{i,j}$ is the rating given by user i for item j.

In general, users with a high correlation to user a (that have accessed a large number of items in common) are said to be in the same neighborhood as a. For a large number of users, it is possible to limit the neighborhood to a preset size, which would then contain the top most similar users. For each item that user a has not seen yet, its rating is predicted as the weighted average of all other users in that neighborhood for the selected item. The items that received the highest ratings can then be set aside for recommendation.

III. Speaker Identification Using i-vectors

In text-independent speaker identification, people are recognized by their voice characteristics and manner of speaking [18]. Speaker verification is the detection task of determining whether a given test speech utterance belongs to a given speaker or not. Each speaker is characterized by a target model beforehand, and typically a likelihood score is computed that says to what extent the test model matches the target model. Speaker identification is simply the repeated process of applying speaker verification to multiple target speaker models, where the task is to determine who spoke the utterance.

Throughout the years, there have been many advances in text-independent speaker identification. Some of the well-known techniques include Gaussian Mixture Models (GMMs) [19], Support Vector Machines with GMM Supervectors [20], Joint Factor Analysis [21], i-vectors [22] and Deep Neural Networks [23]. In this work we shall assume an i-vector based system is used to carry out speaker identification of spoken ratings in the front-end, which has been shown to exhibit good all-round performance, especially in more realistic mismatched conditions [25, 26]. While good results have recently been reported for GMM-UBM, this is only the case for very short utterances of around 2s or under [27]. Note that for the use case in mind, the system should be able to accommodate longer utterances, and of variable length, and where users should not be constrained in how they might state their ratings. Therefore, saying "Give a rating of 2" while watching a movie and "Give a rating of 2 to the movie I watched last night" should be identical.

In the i-vector approach, a speech utterance, regardless of its length, is represented as a fixed-length low-dimensional i-vector [24]. The constant size and low dimension of i-vectors means that conventional classification and optimization techniques can be readily applied.

Assuming F-dimensional acoustical features and C mixture components, the FC-dimensional speaker dependent supervector can be modeled in the following way:

$$\mathbf{m}_\mathrm{i} = \mathbf{m}_0 + \mathbf{Tw} \tag{3}$$

where $\mathbf{m}_0$ is the speaker-independent supervector (this supervector is obtained by stacking the F-dimensional mean vectors of the Universal Background Model (UBM) [18]), $\mathbf{T}$ is a matrix of low rank and $\mathbf{w}$ is a hidden random variable assumed to have a standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The supervector $\mathbf{m}_\mathrm{i}$ is assumed to be normally distributed with mean $\mathbf{m}_0$ and covariance $\mathbf{TT^T}$. The i-vector is then just the MAP point estimate of this hidden variable. The matrix $\mathbf{T}$ is trained using an EM algorithm for eigenvoice matrices [24].

All speech processing happens in the local device and is not transmitted to the cloud. As more and more demands are placed on users' privacy, there is an increasing AI trend of processing all sensitive sensor data at the local device as much as possible.

IV. Proposed Framework

We shall now present the proposed framework. A system diagram is shown in Figure 1.

A. Detected Speaker Confidence

We introduce the notion of the detected speaker confidence. In a closed-set speaker identification system, when a test utterance is spoken, for each speaker model in the system, a score is generated. In a typical state-of-the-art speaker identification system, these scores can be log-likelihood scores, where the higher the score given to a specific speaker model, the more likely it is that the test utterance was spoken by the speaker the model represents.

[3] Results can change considerably when the notion of similarity changes.

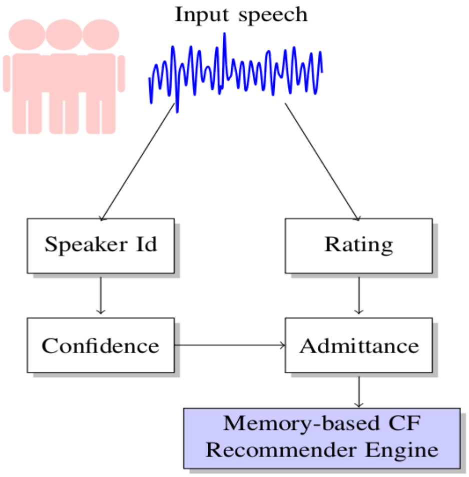


Fig. 1. System diagram showing the overall framework. The confidence determined when identifying a speaker is used to determine whether or not the speaker's rating should be admitted to the memory-based CF recommender.

For a given set of speaker models, we expect the target model, i.e. the speaker's true model, to generate a high score and the remaining non-target models to have lower scores. Without ground truth knowledge of which model is the target, the target model that matches the test utterance with the highest score is assumed to be that of the test speaker. However, when the speaker models are all very alike, for example, in the case of two children, the scores will be fairly similar (and typically lower), and the chance of misidentifying the correct user might increase.

For a test utterance, let at represent the assumed target (the score with the highest likelihood), and let ant_i, where $i <$ I and where I is the number of non-target scores, represent the score for each assumed non-target speaker[4]. Assuming all scores (both target and non-target) have been normalized to lie in the interval from 0 to 1, we can now compute the confidence as:

$$C_u = at - \sqrt{\frac{1}{I}\sum_{i=1}^{I}(ant_i)^2} \qquad (4)$$

The significance of this is that if there are a large number of non-target scores close to the target, the result will be a low confidence score. On the other hand, when there is a large gap between the target and the other non-target scores, then the confidence will be high. In a real-world setting, if the confidence value is too low, the system could react by simply asking the person who stated the last rating to restate their rating.

B. Applying confidence to recommendations

After we determine a corresponding confidence score that will accompany each trial, we can use that confidence value against a chosen threshold Θ to determine the usefulness of that score. The implication of this, in the recommendation context, is that every time a rating is spoken, the speaker is identified with a given confidence. Only ratings with a confidence score C_u that exceeds the threshold Θ will be admitted for that user. As will be discussed later, the value Θ can be found empirically. In the next section, we present the algorithm for enrolling and testing speakers for a given family unit, computing scores, and admitting or discarding user preferences.

V. Experimental Work

A. Datasets

For the experiments, we used the 1M MovieLens dataset [28], which has 1,000,029 ratings for 6040 users of 3900 movies. MovieLens is a popular dataset used for testing the effectiveness of collaborative filtering algorithms. A strong motivation for choosing the 1M dataset, as opposed to the newer 10M and 20M datasets was the fact that it includes both the age and gender for each of the 6040 users, making it possible to provide a more realistic matching with real speech utterances with the same age and gender.

The speech utterances themselves were taken from the aGender dataset [29]. The aGender corpus was supplied to participants in the Interspeech 2010 Paralinguistic Challenge to enhance the development of age and gender algorithms. The training part of the dataset contains 32527 utterances from 472 speakers, the development part contains 20549 utterances from 300 speakers and the testing part contains 17332 utterances. It comprises 4 age classes: children (7-14 years), young people (15-24 years), adults (25-54 years) and seniors (>55 years), and 3 gender classes: children, males and females, from a total of 954 speakers. Children are classed as their own gender since the voices of males are indistinguishable from females at that age. In more recent work, the age boundaries are slightly different, i.e. children (<13 years), young people (14-19 years), adults (20-54 years) and seniors (>55 years) [6]. The latter age boundaries,

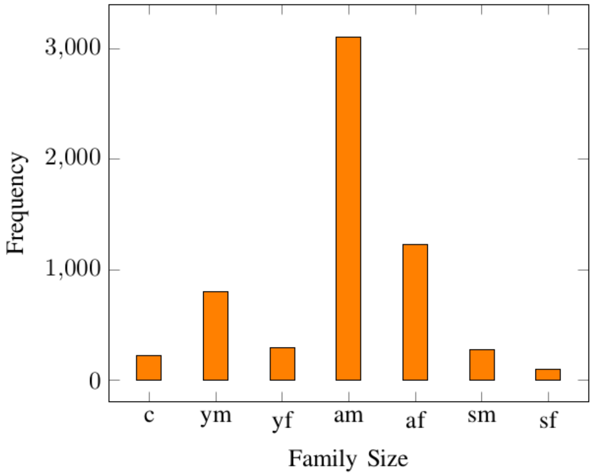


Fig. 2. Age and gender demographics for all 6040 users from MovieLens. Note the high proportion of users allocated to the *am* category. Legend: *c=child, ym=young male, yf=young female, am=adult male, af=adult female, sm=senior male, sf=senior female*.

[4] Assumed target and assumed non-target should not be confused with target and non-target, which are used to denote the ground truth of speakers.

corresponding to the recent work, were chosen[5]. Figure 2 shows a breakdown of the MovieLens users into these same seven age and gender classes.

From the MovieLens dataset, 30 % of the data was withheld as test data and was not modified in any way. The remaining 70 % was used as training data. To test the accuracy of a collaborative filtering engine, for all users and the recommended items, we measure the difference in Mean Absolute Error (MAE) between the real rating for an item (a Movie), as given by the test data, and the actual value as predicted by the recommendation engine, using the training data [2]. MAE is simply the sum of the differences between the actual rating (ground truth) and the predicted rating, divided by the number of content items taken into consideration. A lower MAE implies generally more accurate recommendations.

B. Experimental configurations

The approach taken in this work is to compare different configurations. For every configuration, the entire set of preferences from the above-mentioned training portion[6] of MovieLens, with associating user IDs, is used to build a CF recommender. The only parameter that varies from configuration to configuration is how the actual user IDs are determined.

For each configuration, the users from MovieLens were split into fixed-sized family units (without regard to their age and gender, meaning that it was possible for all users for a family unit to have the same age and gender category) with sizes ranging from 2 to 5 users. It is within these family units that the closed-loop speaker identification takes place, implying that there is a chance that one of the users from the family unit is confused with another. The range was chosen since it corresponds to typical family sizes in most populations. Family units of 1 person were not considered since the single-user case does not pose an interesting challenge - with only 1 person, no confusion can occur (and using speaker identification to identify users does not make sense). Family unit sizes larger than 5 users were also not considered, due to their low statistical representation in the population.

With this in mind, the following four experimental configurations were evaluated:

1) An upper bound baseline system, where the user IDs were not modified in any way. As these user IDs represent the ground-truth (no error in estimating who the real user is), this system is expected to achieve the highest accuracy in terms of MAE.

2) A lower bound system, where the user IDs within a family group are randomly exchanged with one another. This system is expected to perform the worst of all the proposed configurations.

3) The proposed closed-set speaker-identification system. For each preference of a given user u, the user ID itself is unconditionally replaced with a speaker-detected ID. More information on the proposed setup is given below.

4) The second proposed closed-set speaker-identification system. For each preference of a given user u, the user ID is only replaced with the speaker-detected ID when the confidence is above a predefined threshold. When the threshold is below Θ, the preference given is simply discarded.

For configurations 3 and 4, each unique user ID from a given family group was associated with a unique user from the aGender data set, taking into account the user's age and gender as well. The user's age and gender from MovieLens was used to identify one of seven predefined age and gender classes from aGender.

This allowed for matching of users' demographics from MovieLens to realistic speech utterances from aGender, reflecting a similar age and gender demographic profile, and a more realistic setting for the speech in a given family unit along with any associated confusion between speaker classes that might occur. Since there are a lot more users from MovieLens than compared to aGender (a ratio of 10:1), we allowed for random reassignment of users from aGender.

However, within a given family unit, this was done without replacement for each age and gender class to avoid using the same user's speech utterances from aGender for two different MovieLens users.

C. Audio system

In the aGender dataset, for each user, there are multiple speech utterances. Once a user was identified, the speech utterances were partitioned once again into a 70 % training and a 30 % test set. For each user in the family unit, the user's utterances from the training set were used to enroll that user in the i-vector system. The enrollment i-vector was computed using the mean of all the user's training i-vectors, which is a common enrollment strategy [30]. At this stage, for the current family unit being processed, with size N (which can vary in size from 2 to 5 users), a test speech utterance (a rating from 1 to 5) can now be evaluated and assigned to the closest matching user.

The i-vector system was constructed as follows: 13 Mel Frequency Cepstral Coefficients (MFCCs) (including log energy), first and second derivatives were extracted to give a fixed 39-feature frame for each 25 ms voice frame, with a 10 ms overlap for each frame. MFCCs are simply a compact representation of the spectral envelope of a speech signal. A 512-component GMM was trained using TIMIT. Using the data from the GMM, a total variability matrix was trained using the entire training portion of aGender. After this, for each utterance from the development part of aGender, a 300-dimensional i-vector was extracted from the total variability matrix. Once in the i-vector space, classification of the utterances was carried out using probabilistic linear discriminant analysis (PLDA) after performing normalization on the i-vectors. The performance in accuracy for different family sizes for the overall system is shown in Table I. Interestingly we note that a slightly higher performance was achieved for the family size of 3 as opposed to 2. We believe

[5] The original aGender age boundaries were chosen solely on the basis of marketing aspects, and not on any physiological aspects.

[6] The test data is kept intact across all configurations.

the reason for this to be the large proportion of the class adult male, making the task more challenging.

TABLE I
RESULTS FOR THE SPEAKER IDENTIFICATION SYSTEM FOR FAMILY SIZES OF 2, 3, 4 AND 5. DISTRIBUTION OF CLASSES FOR EACH FAMILY UNIT IS RANDOMLY GENERATED. RESULT OBTAINED USING DEVELOPMENT DATA FROM AGENDER AS TEST DATA, WITHOUT REPLACEMENT. THE LARGER THE FAMILY UNIT, THE LOWER THE NUMBER OF TESTED CONFIGURATIONS. EXECUTION TERMINATED WHEN THERE IS INSUFFICIENT DATA REMAINING TO TEST THE NEXT CONFIGURATION.

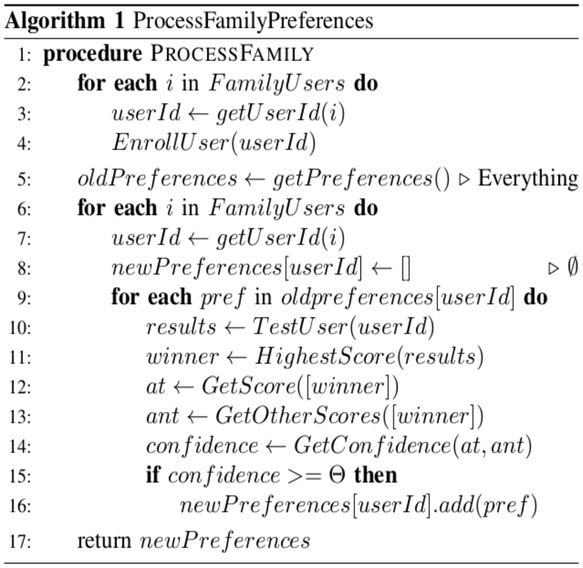
Family Unit Size	Tested Configurations	Accuracy %
2	144	84.38
3	89	85.39
4	64	73.82
5	35	70.29

D. Processing of Movie Ratings

The rating preferences from the training portion of MovieLens were processed to build a recommender in the following manner: For each movie preference for a given user, the actual user ID for that preference was connected to a random utterance from the 30 % test data from aGender for that same user. This test utterance was scored against all target models for the given family unit already enrolled, and the winning model was selected as the detected speaker, and mapped back to a MovieLens user ID. In a perfect system, the initial ID and final ID would therefore be the same. The resulting assumed target scores (i.e. the single winning score), and the remaining assumed non-target scores were then used according to (4) to compute a confidence score for the detected speaker (how likely we believe in the estimate). For the first speech configuration (configuration 3), for each preference, the original user ID was replaced with the user's newly estimated user ID within the family group (which most likely is the same as the original), without regard to the confidence of the prediction. For the second speech configuration (configuration 4), the original user ID was replaced with the user's newly estimated user ID with the family group, but only if the confidence was found to be above Θ. For all preferences where the confidence was below Θ, the preference was discarded, meaning that a lower number of training data points were used for the CF recommender. Once all preferences had been processed, it was then possible to evaluate the recommender's accuracy against the 30 % withheld test data from MovieLens.

E. Practical Implementation Details

The recommendation algorithm was implemented in Apache Mahout. The Pearson correlation method was used to compute the similarity between users. The neighborhood size for the most similar users was set to 25 users. The threshold Θ was set empirically to values of 0.2, 0.4, 0.6 and 0.8. The algorithm for enrolling and testing speakers for a given family unit, computing scores, and admitting or discarding user preferences is shown in Figure 3. Note that each user in the MovieLens dataset has a variable number of preferences (rated items), and for each preference given, the real user's ID is mapped to a speaker-ID counterpart.



```
Algorithm 1 ProcessFamilyPreferences
procedure PROCESSFAMILY
    for each $i$ in $FamilyUsers$ do
        $userId \leftarrow getUserId(i)$
        $EnrollUser(userId)$
    $oldPreferences \leftarrow getPreferences()$ ▷ Everything
    for each $i$ in $FamilyUsers$ do
        $userId \leftarrow getUserId(i)$
        $newPreferences[userId] \leftarrow []$ ▷ $\emptyset$
        for each $pref$ in $oldpreferences[userId]$ do
            $results \leftarrow TestUser(userId)$
            $winner \leftarrow HighestScore(results)$
            $at \leftarrow GetScore([winner])$
            $ant \leftarrow GetOtherScores([winner])$
            $confidence \leftarrow GetConfidence(at, ant)$
            if $confidence >= \Theta$ then
                $newPreferences[userId].add(pref)$
    return $newPreferences$
```

Fig. 3. Proposed algorithm for managing speakers and their preferences.

VI. RESULT AND DISCUSSION

To recap, we use the MAE between the suggested ratings given by the CF system for each of the four experimental configurations, and corresponding ratings for the test items from MovieLens. The results for the four configurations are shown in Table II.

TABLE II
RESULTS SHOWING THE MEAN ABSOLUTE ERROR (MAE) FOR EACH OF THE FOUR EXPERIMENTAL CONFIGURATIONS FOR FAMILY SIZES OF 2, 3, 4 AND 5. THE LOWER THE MAE, THE BETTER QUALITY THE RECOMMENDATION.

Config	MAE (2)	MAE (3)	MAE (4)	MAE (5)	MEAN / SD
1	0.8470	0.8470	0.8470	0.8470	0.8470 0.0000
2	0.8804	0.8878	0.8973	0.8954	0.8902 0.0077
3	0.8653	0.8706	0.8780	0.8750	0.8722 0.0055
4	**0.8614**	**0.8699**	**0.8708**	**0.8750**	0.8693 0.0057

Firstly, we notice that configuration 1 (upper bound) showed the best results for all family unit sizes, with the lowest average MAE. For configuration 1, the MAE is not affected by family size, since the ground truth user IDs are used. On the other end, configuration 2 (lower bound) shows the worst results, with highest average MAE.

For configuration 3, where we unconditionally replace the ground-truth speaker IDs in a family unit with their speaker recognition counterparts, we notice a trend whereby the larger the family size, the higher the MAE is, implying lower quality recommendation. This is an expected result since more users introduce a greater possibility for confusion, since there is a greater chance of confusing one user with another user.

In configuration 4, by discarding preferences for users detected with low confidence, for the given threshold shown, we were in three out of four cases able to improve the MAE compared to configuration 3. By considering the maximum possible performance as that given by configuration 1, and the worst by configuration 2, this corresponds to improving the performance by a further 6.9 % relative to configuration 3. There appears to be a benefit in utilizing confidence-based admittance techniques to improve recommendation accuracy.

Table III shows the experiment of configuration 4 repeated for family sizes of 3, 4 and 5, but with different threshold values[7]. Here, we see the larger the family size, the larger the percentage of ratings that are discarded by the system. In larger family sizes, the probability of assumed non-target scores lying in the vicinity of the target leads in general to a lower confidence overall, and hence more discarded ratings.

TABLE III
EXTENDED RESULTS FOR CONFIGURATION 4 (TAKING CONFIDENCE INTO ACCOUNT) SHOWING HOW THE PERCENTAGE OF RATINGS THAT ARE DISCARDED (D) DECREASES WITH INCREASING FAMILY SIZE AS WELL AS DECREASING THRESHOLD. THE TOTAL NUMBER OF RATINGS ACROSS ALL 6041 USERS WAS 699742. MAE IS MEAN ABSOLUTE ERROR AND % D IS PERCENTAGE DISCARDED

Family Size	MAE / % D T = 0.8	MAE / % D T = 0.6	MAE / % D T = 0.4	MAE / % D T = 0.2
3	0.8767 / 68.88	0.8645 / 33.43	**0.8639** / 12.00	0.8699 / 0.00
4	0.9272 / 86.99	**0.8600** / 49.62	0.8645 / 11.95	0.8708 / 0.00
5	1.0165 / 94.24	0.8686 / 59.83	**0.8654** / 14.80	0.8750 / 0.00

Another reason for this we believe is due to the unbalanced nature of the MovieLens dataset - a disproportionate number of users were assigned to the adult male category, increasing the likelihood of similar demographics (and hence decreasing the confidence somewhat). We believe therefore that it is in the larger family sizes, where the confidence-based admission technique is applied, where the most gain is to be expected.

What is interesting to note is that there is not a linear relationship between the threshold and the MAE error, where the best results achieved were 0.8639, 0.8600 and 0.8654 for family sizes of 3, 4 and 5, respectively. This suggests a trade-off between a very low threshold, with too many ratings being admitted corresponding to incorrectly detected users, and having a very discriminative threshold, where the much-reduced number of ratings starts to affect the recommender's ability. The results seem to indicate that performance in general is hurt significantly more when too many ratings are discarded.

Finally, we notice in general that the MAE results shown in both tables above are fairly similar across all configurations and thresholds. We believe this to be due to two factors:

1) The fairly high accuracy at which users can be correctly identified within each family unit implies that the majority of preferences are not assigned to incorrect users.

2) The family sizes chosen are fairly small (2-5). This limits the amount of confusion that occurs where preferences are assigned to the wrong user. It should also be noted that once all preferences have been submitted, the concept of the family unit disappears, meaning that each user would be part of a larger neighborhood beyond the limits of the family group.

VII. Conclusion

In this paper, we presented a closed-loop speaker identification system as a front-end to a collaborative filtering-based recommender, to address how to extract ratings from more than one user in a group. The gist of the proposal is that users assign ratings to items through conventional spoken dialogue, and are identified through their speech. Due to the inaccuracy of identifying people through their voice, a rating might be assigned to the wrong user in the group, which should be avoided. Using an additional confidence score can assist in determining whether a stated preference should be admitted or discarded. It seems the effectiveness of the algorithm depends on both family size and the chosen threshold, and that higher thresholds should be used for larger family sizes.

VIII. Future Work

Further work might look at taking the similarity of users from a given family unit into account - if they already have very similar profiles, a lower confidence threshold might be employed, since mixing up these two users (with similar tastes) is potentially less harmful. Another work might also consider using a weighted approach to admit ratings that fall below a predetermined threshold, instead of simply discarding them. Finally, from a privacy perspective, in this work, we do not address the issue of how the mapping of real users' speech utterances to logical IDs might be used to reveal their identify.

[7] The family size of 2 is not relevant here, since having only two scores means either they are equal, in which case the confidence is 0, or they are not equal, in which case the confidence is 1.

References

[1] G. Adomavicius and A. Tuzhilin, *Context-Aware Recommender Systems*. Boston, MA: Springer US, 2015, pp. 191–226.

[2] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," Knowledge-Based Systems, vol. 46, pp. 109–132, 2013.

[3] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender systems: An introduction*. Cambridge University Press, 2010, ch. 2.

[4] D. Véras, T. Prota, A. Bispo, R. Prudêncio, and C. Ferraz, "A literature review of recommender systems in the television domain," Expert Systems with Applications, vol. 42, no. 22, pp. 9046–9076, 2015.

[5] H.-J. Kwon and K.-S. Hong, "Personalized smart tv program recommender based on collaborative filtering and a novel similarity method," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, 2011.

[6] R. Likert, "A technique for the measurement of attitudes." Archives of psychology, 1932.

[7] C. A. Gomez-Uribe and N. Hunt, "The Netflix recommender system: Algorithms, business value, and innovation," *ACM Trans. Manage. Inf. Syst.*, vol. 6, no. 4, pp. 13:1–13:19, Dec. 2015.

[8] X. Amatriain and J. Basilico, “Past, present, and future of recommender systems: An industry perspective,” in *Proceedings of the 10th ACM Conference on Recommender Systems*, ser. RecSys ’16. New York, NY, USA: ACM, 2016, pp. 211–214.

[9] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *Computer*, no. 8, pp. 30–37, 2009.

[10] J. Masthoff, Group Recommender Systems: Aggregation, Satisfaction and Group Attributes. Boston, MA: Springer US, 2015, pp. 743–776.

[11] R. Sotelo, Y. Blanco-Fernandez, M. Lopez-Nores, A. Gil-Solla, and J. J. Pazos-Arias, “Tv program recommendation for groups based on muldimensional tv-anytime classifications,” *IEEE Transactions on Consumer Electronics*, vol. 55, no. 1, 2009.

[12] K. Verstrepen and B. Goethals, “Top-n recommendation for shared accounts,” in *Proceedings of the 9th ACM Conference on Recommender Systems*, ser. RecSys ’15. New York, NY, USA, 2015, pp. 59–66.

[13] Z. Yu, X. Zhou, Y. Hao, and J. Gu, “Tv program recommendation for multiple viewers based on user profile merging,” *User Modeling and User-Adapted Interaction*, vol. 16, no. 1, pp. 63–82, 2006.

[14] J. Kang, K. Condiff, S. Chang, J. A. Konstan, L. Terveen and F. M. Harper, “Understanding how people use natural language to ask for Recommendations,” *ACM Conference on Recommender Systems(RecSys),* 2017.

[15] X. Zhang, H. Zhao, “Cold-start Recommendation Based on Speech Personality Traits,” *Journal of Computational and Theoretical Nanoscience,* 2017, vol. 14, no. 3, pp. 1314-1323.

[16] S. Shepstone, Z-H. Tan, S. H. Jensen, “Audio-based age and gender identification to enhance the recommendation of TV content,” In *IEEE Transactions on Consumer Electronics,* 2013, vol. 59, no. 3, pp. 721-729.

[17] S. Shepstone, Z-H. Tan, S. H. Jensen, “Using audio-derived affective offset to enhance TV recommendation,” In *IEEE Transactions on Multimedia,* 2014, vol. 16, no. 7, pp. 1999-2010.

[18] T. Kinnunen and H. Li, “An overview of text-independent speaker recognition: from features to supervectors,” *Speech communication*, vol. 52, no. 1, pp. 12–40, 2010.

[19] D. Reynolds, T. Quatieri, and R. Dunn, “Speaker verification using adapted gaussian mixture models,” *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.

[20] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, “Support vector machines using gmm supervectors for speaker verification,” *Signal Processing Letters*, IEEE, vol. 13, no. 5, pp. 308–311, 2006.

[21] P. Kenny, “Joint factor analysis of speaker and session variability: Theory and algorithms,” CRIM, Montreal,(Report) CRIM-06/08-13, 2005.

[22] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 19, no. 4, pp. 788–798, 2011.

[23] Y. Lei, L. Ferrer, M. McLaren, and N. Scheffer, “A deep neural network speaker verification system targeting microphone speech,” in *INTERSPEECH*, 2014.

[24] P. Kenny, G. Boulianne, and P. Dumouchel, “Eigenvoice modeling with sparse training data,” *IEEE Transactions on Speech and Audio Processing,* vol. 13, no. 3, pp. 345–354, 2005.

[25] M. T. AI-Kaltakchi, W.L. Woo, S.S. Dlay, J.A. Chambers, “Comparison of i-vector and GMM-UBM approaches to speaker identification with TIMIT and NIST 2008 databases in challenging environments”, In *European Signal Processing Conference (EUSIPCO),* 2017

[26] R. Haraksim and A. Andrzej, S.S. Dlay, “Measuring performance in forensic automatic speaker recognition: VQ, GMM-UBM, i-vectors”, In *BioSig,* 2016.

[27] A. Poddar, M.. Sahidullah, G. Saha, “Improved i-vector extraction techniques for speaker verification with short utterances,” In *International Journal of Speech Technology,* 2017, vol. 1, no. 16.

[28] F. M. Harper and J. A. Konstan, “The movielens datasets: History and context,” *ACM Transactions on Interactive Intelligent Systems (TiiS),* vol. 5, no. 4, pp. 19, 2015.

[29] F. Burkhardt, M. Ekert, W. Johannsen, and J. Stegmann, “A database of age and gender annotated telephone speech,” *Proc. 7th International on Language Resources and Evaluation (LREC)*, pp. 1562– 1565, 2010.

[30] K.-A. Lee, A. Larcher, C. H. You, B. Ma, and H. Li, “Multi-session plda scoring of i-vector for partially open-set speaker detection.” In *INTERSPEECH*, 2013, pp. 3651–3655.

Sven E. Shepstone (M’11) received the B.S. and M.S. degrees in Electrical Engineering from the University of Cape Town in 1999 and 2002 respectively. From 2005-2010 he worked in the field of broadband communications for Ericsson a/s in Denmark, and has been at Bang and Olufsen a/s in Denmark since 2010. In 2015 he received the Ph.D degree from Aalborg University. Today he is a Research Specialist at Bang and Olufsen. His research interests include AI applied to consumer electronics. He was the recipient of the IEEE Ganesh N. Ramaswamy Memorial Student Grant at ICASSP 2015.

Zheng-Hua Tan (M'00--SM'06) received the B.Sc. and M.Sc. degrees in electrical engineering from Hunan University, Changsha, China, in 1990 and 1996, respectively, and the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 1999. He is a Professor in the Department of Electronic Systems at Aalborg University, Aalborg, Denmark, which he joined in 2001. He is also a Co-Head of the Centre for Acoustic Signal Processing Research (CASPR) at Aalborg University. He was a Visiting Scientist at the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, USA, an Associate Professor in the Department of Electronic Engineering at Shanghai Jiao Tong University, and a postdoctoral fellow in the Department of Computer Science at Korea Advanced Institute of Science and Technology, Daejeon, Korea. His research interests include machine learning, deep learning, pattern recognition, speech and speaker recognition, noise-robust speech processing, multimodal signal processing, and social robotics. He has authored or co-authored more than 170 publications in refereed journals and conference proceedings. He is a member of the IEEE Signal Processing Society Machine Learning for Signal Processing Technical Committee (MLSP TC). He has served as an Editorial Board Member/Associate Editor for Computer Speech and Language, Digital Signal Processing, and Computers and Electrical Engineering. He was a Lead Guest Editor of the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING and a Guest Editor of Neurocomputing. He has served as a General Chair, Program Co-chair, Area and Session Chair, and Tutorial Speaker of many international conferences.

Miklas S. Kristoffersen (StM’11) received the B.Sc. and M.Sc. degrees in Electronic Engineering with specialization in Vision, Graphics, and Interactive Systems in 2014 and 2016 from Aalborg University, Denmark. He is currently employed at Bang & Olufsen A/S in Denmark, where he is an industrial PhD candidate at Aalborg University. His main research interests include computer vision, machine learning, and context-aware recommender systems.