VoiceID on the fly: A Speaker Recognition System that Learns from Scratch

Baihan Lin[1], *Xinxin Zhang*[2]

[1] Department of Applied Mathematics, University of Washington, Seattle, USA
[1] Department of Electrical & Computer Engineering, University of Washington, Seattle, USA
{sunnylin, zhangx43}@uw.edu

Abstract

We proposed a novel AI framework to conduct real-time multi-speaker recognition without any prior registration or pretraining by learning the speaker identification on the fly. We considered the practical problem of online learning with episodically revealed rewards and introduced a solution based on semi-supervised and self-supervised learning methods in a web-based application at https://www.baihan.nyc/viz/VoiceID/.

1. Introduction

Speaker recognition involves two essential steps: registration and identification. In laboratory setting, the state-of-the-art approaches usually emphasize the registration step with deep networks trained on large-scale speaker profile dataset [1]. However, in real life, requiring all users to complete voiceprint registration before a multi-speaker teleconference is hardly a preferable way of system deployment. Dealing with this challenge, speaker diarization is the task to partition an audio stream into homogeneous segments according to the speaker identity. Similarly, a preferable AI engine for a realistic speaker recognition system should (1) not require user registrations, (2) allow new user to be registered into the system real-time, (3) transfer voiceprint information from old users to new ones, (4) be up running without pretraining on large amount of data in advance. While attractive, assumption (4) introduced an additional caveat that the labeling of the user profiles happens purely on the fly, trading off models pre-trained on big data with the user directly interacting with the system by correcting the agent as labels. To tackle all these challenges, we formulated this problem into an interactive online learning problem with cold-start arms and episodically revealed rewards (the user can either reveal no feedback, approving the agent by not intervening, or correcting the agent). We built upon LinUCB [2] and proposed a semi-supervised learning variant to account for the fact that the rewards are entirely missing in many episodes. For each episodes without feedbacks, we applied a self-supervision process to assign a pseudo-action upon which the reward mapping is updated. Finally, we generated new arms by transferring the learned arm parameters for similar profiles given the user feedbacks. Unlike others, our VoiceID system is interactive, register-free, real-time and completely web-based (Figure 1).

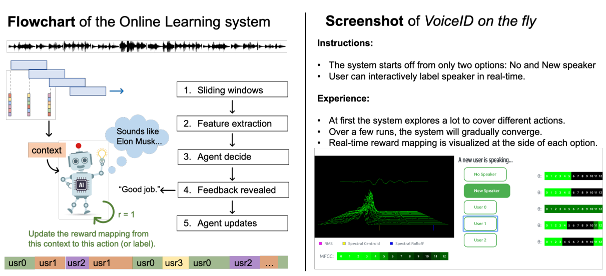

Figure 1: *Flowchart of the demo system: VoiceID on the fly.*

2. Problem Setting

Algorithm 1 presents at a high-level our problem setting, where $c(t) \in \mathbb{R}^d$ is a vector describing the context at time t, $r_a(t) \in [0, 1]$ is the reward of action a at time t, and $r(t) \in [0, 1]^K$ denotes a vector of rewards for all arms at time t. $\mathbb{P}_{c,r}$ denotes a joint probability distribution over (c, r), and $\pi : C \to A$ denotes a policy. Unlike traditional setting, in step 5 we have the rewards revealed in an episodic fashion (i.e. sometimes there are feedbacks of rewards being 0 or 1, sometimes there are no feedbacks of any kind). We consider our setting an online semi-supervised learning problem [3, 4], where the agent learns from both labeled and unlabeled data in online setting from scratch.

Algorithm 1 Online Learning with Episodic Rewards

```
1: for t = 1,2,3,···, T do
2:    (c(t), r(t)) is drawn according to P_{c,r}
3:    c(t) is revealed to the player
4:    Player chooses an action i = π_t(c(t))
5:    Feedbacks r_a(t) for all arms are episodically revealed
6:    Player updates its policy π_t
7: end for
```

3. Proposed Method

3.1. Episodically Rewarded LinUCB

We proposed Background Episodically Rewarded LinUCB (BerlinUCB) [5], a semi-supervised and self-supervised online contextual bandit which updates the context representations and reward mapping separately given the state of the feedbacks being present or missing (Algorithm 2). We assume that (1) when there are feedbacks available, the feedbacks are genuine, assigned by the oracle, and (2) when the feedbacks are missing (not revealed by the background), it is either due to the fact that the action is preferred (no intervention required by the oracle, i.e. with an implied default rewards), or that the oracle didn't have a chance to respond or intervene (i.e. with unknown rewards). Especially in the Step 15, when there is no feedbacks, we assign the context $\mathbf{x}_t$ to a class a' (an action arm) with the self-supervision given the previous labelled context history. Since we don't have the actual label for this context, we only update the reward mapping parameter $\mathbf{b}_{a'}$ and leave the covariance matrix $\mathbf{A}_{a'}$ untouched. This additional usage of unlabelled data (or unrevealed feedback) is especially important in our model.

3.2. The Self-Supervision and Semi-Supervision Modules

We construct our self-supervision modules given the cluster assumption of the semi-supervision problem: the points within the same cluster are more likely to share a label. We chose three popular clustering algorithms in modern speaker diarization task as the self-supervision modules: Gaussian Mixture Models (GMM), Kmeans and K-nearest neighbors (KNN).

Algorithm 2 BerlinUCB

1: **Initialize** $c_t \in \mathbb{R}_+, \mathbf{A}_a \leftarrow \mathbf{I}_d, \mathbf{b}_a \leftarrow \mathbf{0}_{d\times 1} \forall a \in \mathcal{A}_t$
2: **for** t = 1,2,3,$\cdots$, T **do**
3: Observe features $\mathbf{x}_t \in \mathbb{R}^d$
4: **for all** $a \in \mathcal{A}_t$ **do**
5: $\hat{\theta}_a \leftarrow \mathbf{A}_a^{-1}\mathbf{b}_a$
6: $p_{t,a} \leftarrow \hat{\theta}_a^\top \mathbf{x}_t + c_t\sqrt{\mathbf{x}_t^\top \mathbf{A}_a^{-1}\mathbf{x}_t}$
7: **end for**
8: Choose arm $a_t = \arg\max_{a\in\mathcal{A}_t} p_{t,a}$
9: **if** the background revealed the feedbacks **then**
10: Observe feedback $r_{a_t,t}$
11: $\mathbf{A}_{a_t} \leftarrow \mathbf{A}_{a_t} + \mathbf{x}_t\mathbf{x}_t^\top$
12: $\mathbf{b}_{a_t} \leftarrow \mathbf{b}_{a_t} + r_{a_t,t}\mathbf{x}_t$
13: **elif** the background revealed NO feedbacks **then**
14: **if** use self-supervision feedback
15: $r' = [a_t == \text{predict}(\mathbf{x}_t)]$ % clustering modules
16: $\mathbf{b}_{a_t} \leftarrow \mathbf{b}_{a_t} + r'\mathbf{x}_t$
17: **elif** % ignore self-supervision signals
18: $\mathbf{A}_{a_t} \leftarrow \mathbf{A}_{a_t} + \mathbf{x}_t\mathbf{x}_t^\top$
19: **end if**
20: **end if**
21: **end for**

3.3. The AI Engine for Online Speaker Diarization

To adapt our BerlinUCB algorithm to the specific application of speaker recognition, we first define our actions. There are three major classes of actions: an arm "New" to denote that a new speaker is detected, an arm "No Speaker" to denote that no one is speaking, and N different arms "User n" to denote that user n is speaking. Table 1 presents the reward assignment given four types of feedbacks. Note that we assume that when the agent correctly identifies the speaker (or no speaker), the user (as the feedback dispenser) should send no feedbacks to the system by doing nothing. In another word, in an ideal scenario when the agent does a perfect job by correctly identifying the speaker all the time, we are not necessary to be around to correct it anymore (i.e. truly feedback free). As we pointed out earlier, this could be a challenge earlier on, because other than implicitly approving the agent's choice, receiving no feedbacks could also mean the feedbacks are not revealed properly (e.g. the human oracle took a break). Furthermore, we note that when "No Speaker" and "User n" arms are correctly identified, there is no feedback from us the human oracle (meaning that these arms would never have learned from a single positive reward if we don't use the "None" feedback iterations at all!). The semi-supervision by self-supervision step is exactly tailored for a scenario like this, where the lack of revealed positive reward for "No Speaker" and "User n" arms is compensated by the additional training of the reward mapping $\mathbf{b}_{a_t}$ if context $\mathbf{x}_t$ is assigned to the right arm. To tackle the cold start problem, the agent grows it arms in the following fashion: the agent starts with two arms, "No Speaker" and "New"; if it is actually a new speaker speaking, we have the following three conditions: (1) if "New" is chosen, the user approves this arm by giving it a positive reward (i.e. clicking on it) and the agent initializes a new arm called "User N" and update $N = N + 1$ (where N is the number of registered speakers at the moment); (2) if "No Speaker" is chosen, the user disapproves this arm by giving it a zero reward and clicking on the "New" instead), while the agent initializes a new arm; (3) if one of the user arms is chosen (e.g. "User 5" is chosen while in fact a new person is speaking), the agent copies the wrong user arm's parameters to initialize the new arm, since the voiceprint of the mistaken one might be beneficial for the new profile.

Feedback types	(+,+)	(+,-)	(-,+)	*None*
New	$r = 1$	$r = 0$		
No Speaker	-	$r = 0$	$r = 0$	Alg. 2 Step 13
User n	-	$r = 0$	$r = 0$	

Table 1: *Routes given either no feedbacks, or a feedback telling the agent that the correct label is a∗. (+,+) means that the agent guessed it right by choosing the right arm; (+,-) means that the agent chose this arm incorrectly, since the correct one is another arm; (-,+) means that the agent didn't choose this arm, while it turned out to be the correct one. "-" means NA.*

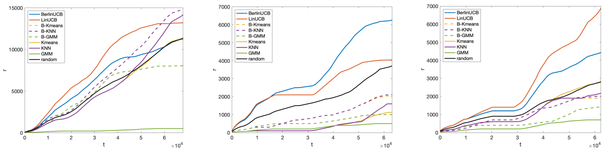

Figure 2: ***Rewards in MiniVox benchmark**. (a) C5-MFCC, p=0.01; (b) C10-MFCC, p=0.01; (c) C20-MFCC, p=0.01.*

4. Interactive System: VoiceID on the fly

As shown in Figure 1, our system "VoiceID on the fly" is an interactive continual learning web-based system. We computed the Mel Frequency Cepstral Coefficients (MFCC) in a sliding window fashion given the real-time audio input from microphone, with the MFCC bands color coded in the page. Figure 2 evaluated the system's performance in the MiniVox benchmark [6] where the labels are only revealed 1% of the time (p=0.01). We observe that although not pretrained on any dataset, our algorithms effectively learns "who speaks when", better than the baselines (GMM, Kmeans and KNN). This suggests a smooth deployment and a good user experience in the demonstration.

At the start, there are only two buttons available: "No Speaker" and "New Speaker". The agent chooses an arm by setting it be highlighted. If it is correct, we do not have to change it (unless it's "New Speaker", where we need to click on it to confirm creating a new arm). The feature band $\hat{\theta}_a$ of each arm is also color coded real-time to visualize how the agent learns across trials. If it is incorrect, we click on the right arm to give the system a feedback. This demonstration provides an intriguing example of how an AI agent can learn to recognize speaker identity (1) entirely escaping the necessity of registering user voiceprint beforehands, (2) effortlessly incorporating new users under an optimal exploration-exploitation trade-off, (3) effectively transferring representation of registered user features to new users, and (4) continually learning despite minimal involvement of human corrections (i.e. sparse and episodic feedbacks).

5. References

[1] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *INTERSPEECH*, 2017.

[2] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 661–670.

[3] B. Yver, "Online semi-supervised learning: Application to dynamic learning from radar data," in *2009 International Radar Conference (RADAR 2009)*, Oct 2009, pp. 1–6.

[4] I. Ororbia, G. Alexander, C. L. Giles, and D. Reitter, "Online semi-supervised learning with deep hybrid boltzmann machines and denoising autoencoders," *arXiv preprint arXiv:1511.06964*, 2015.

[5] B. Lin, "Online semi-supervised learning in contextual bandits with episodic reward," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2020.

[6] B. Lin and X. Zhang, "Speaker diarization as a fully online learning problem in minivox," *arXiv preprint arXiv:2006.04376*, 2020.