OR-GATE: A NOISY LABEL FILTERING METHOD FOR SPEAKER VERIFICATION

Zhihua Fang\textsuperscript{1}  Hanhan Ma\textsuperscript{1}  Lin Li\textsuperscript{2}  Liang He\textsuperscript{1,3,†}

\textsuperscript{1} School of Information Science and Engineering, Xinjiang University, Urumqi, China
\textsuperscript{2} School of Electronic Science and Engineering, Xiamen University, Xiamen, China
\textsuperscript{3} Department of Electronic Engineering, Tsinghua University, Beijing, China

ABSTRACT

The deep learning models used for speaker verification are heavily dependent on large-scale data and correct labels. However, noisy (wrong) labels often occur, which deteriorates the system’s performance. Unfortunately, there are relatively few studies in this area. In this paper, we propose a method to gradually filter noisy labels out at the training stage. We compare the network predictions at different training epochs with ground-truth labels, and select reliable (considered correct) labels by using the OR gate mechanism like that in logic circuits. Therefore, our proposed method is named as OR-Gate. We experimentally demonstrated that the OR-Gate can effectively filter noisy labels out and has excellent performance.

Index Terms— speaker verification, speaker embedding, noisy label, OR-Gate

1. INTRODUCTION

In recent years, deep neural network based speaker models (e.g., TDNN \textsuperscript{1}, ResNet \textsuperscript{2}, and ECAPA-TDNN \textsuperscript{3}) have become the mainstream approaches for speaker verification, as well as deriving many variants (e.g., PSDTDNN \textsuperscript{4}, MFA-Conformer \textsuperscript{5}) with excellent performance. The success of these models depends on large-scale labeled datasets. Unfortunately, noisy (wrong) labels often occur during data collection, for example, Li et al. \textsuperscript{6} found about 1\% noisy labels in the NIST SRE04-10 dataset. Although these noisy labels can be corrected manually, this would be costly and impractical in large-scale databases.

The noisy labels are of three types \textsuperscript{6}: closed-set noise, open-set noise and mixed noise respectively. Closed-set noise means that the labels of the samples are incorrectly labeled as other types in the dataset, open-set noise means that the labels of the samples are incorrectly labeled as types that do not exist in the dataset, and mixed noise consists of both of the above error types.

In the presence of noisy labels in the dataset, the loss function contains wrong labels, which leads to the model to gradient descent towards a wrong direction. Noisy labels will hinder the model to learn stable speaker features. Therefore, it is essential to incorporate a model with stable training behavior to learn better speaker features from data with noisy labels.

Researchers have conducted numerous studies on image recognition with noisy labels, and these works can be roughly divided into four categories \textsuperscript{7}: robust architecture, robust regularization, robust loss function, and sample selection. The robust architecture \textsuperscript{8,9} models the noise transition matrix from a noisy dataset, the robust regularization \textsuperscript{10,11} aims to avoid overfitting of noisy labels by regularization techniques, the robust loss function \textsuperscript{12,13} reduces the contribution of noisy samples to network parameter updates by dynamically weighting different samples or obtains refurbished labels based on network predictions to calculate the loss function, the sample selection \textsuperscript{14,15} aims to select reliable samples from noisy data and then train with supervised or semi-supervised methods.

Image recognition with noisy labels has been well investigated, but there are relatively few studies \textsuperscript{6,16,17} on speaker verification with noisy labels. Since closed-set noise is more challenging \textsuperscript{18} and most studies have focused on closed-set noise, so we only focus on closed-set noise in this work.

In this paper, we propose a method to filter the noisy labels in the speaker dataset. The idea is based on trust in the model, and we judge whether labels are reliable by comparing model predictions with sample labels. The trust in the model has derived from the experimental observation that the deep learning model is robust to noisy labels during the initial stage of training. Therefore, we conduct the warm-up mechanism and use all data to train the model for a few epochs to get a speaker model with basic recognition ability. In each subsequent epoch, we use the OR-Gate with the top-k mechanism to split the data into a part with reliable labels and a part with unreliable labels, then only the data with reliable labels are used to update the network parameters. In summary, we have made the following contributions:

\begin{itemize}
  \item We propose the OR-Gate, which uses the comparison of model predictions and sample labels to determine the reliability of the labels and add the warm-up and top-k
\end{itemize}
mechanisms to the OR-Gate framework.

- We have demonstrated the superiority of the OR-Gate through extensive experiments and proved that the OR-Gate can farthest filter the correct labels from the noisy labels through tracing experiments.

2. RELATED WORK

Our work is inspired by SELF [15], which is a sample selection method of image recognition with noisy labels. The model attempts to identify correct labels progressively by self-forming ensembles of models and predictions. It uses a moving-average of ensemble models and predictions to improve filtering decisions, where the ensemble models are actually a vector $\bar{z}$. The moving-average is updated by the equation:

$$\bar{z}_j = \alpha \bar{z}_{j-1} + (1 - \alpha) \hat{z}_j$$

whereby $\bar{z}_j$ depicts the moving-average prediction for sample $k$ at epoch $j$, $\alpha$ is a momentum that represents the confidence in the model prediction as training progresses, $\hat{z}_j$ is the model prediction for sample $k$ in epoch $j$. In summary, $\bar{z}$ is the combined prediction of the model for sample $k$. By comparing the label $y$ of sample $k$ with the corresponding $\bar{z}$, we can evaluate whether the label $y$ is reliable or not.

SELF [15] maintains and updates $\bar{z}$. At each epoch, it compares the sample labels with the corresponding moving average prediction $\bar{z}$ whether they are consistent and thereby selects the data with reliable labels to train the network. Although SELF [15] achieved success in image recognition with noisy labels, it does not guarantee excellent performance for speaker verification with noisy labels.

3. METHOD

3.1. Network Architecture

During training, we maintain and update a variable $Z = \{\bar{z}_1, \bar{z}_2, \ldots, \bar{z}_n\}$, which records the number of successful matches of label and model prediction for each sample, and it is used to split the data by the OR-Gate.

Fig. 1 shows the framework of the OR-Gate. First, we train the model with all data $D$ for $w$ epochs, and then at each epoch, we divide the data into data $D_1$ with reliable labels and data $D_2$ with unreliable labels by the OR-Gate. $D_1$ is used to train the model and use the model prediction to update $Z$, while $D_2$ only makes the model prediction to update $Z$. At the next epoch, we continue to divide the data by the OR-Gate and repeat the above operation. For details, see Algorithm 1.

3.2. The OR-Gate mechanism

We evaluate the reliability of the labels by comparing the model predictions with the sample labels. Usually, the model fluctuates during training, and the predictions of the same sample may not be consistent across epochs. Therefore, we use the OR gate mechanism of analog circuits. During training, a label is considered reliable as long as the model prediction and the sample label are successfully matched. The above is repeated until the training is finished. Algorithm 1 illustrates the detailed training process.

3.3. Why adopt the warm-up and top-k mechanism?

Despite noisy labels negatively affecting the performance of speaker models, Nguyen et al. [15] discovered that deep learning first learns the samples that are easy to learn and then adapts the samples with noisy labels. Therefore, first training a few epochs with all the data can enable the model to learn a basic ability to recognize speakers.

Considering that the model is not completely accurate in recognizing the speaker after the warm-up, we use the top-$k$ mechanism when matching model predictions and sample labels in order to avoid overconfidence in the model. When us-
reliable label, in contrast, Y
_1 denotes the model prediction and Y_i denotes the one-hot label.

match(P_i, Y_i) 
match(P_i, Y_i) 
match(P_i, Y_i) 

...(a) The top-k mechanism. P_i denotes the model prediction and Y_i denotes the one-hot label.

(b) The OR-Gate. P_i^t denotes the model’s prediction of sample X_i in the t-th epoch and Y_i denotes the label of X_i.

**Fig. 2: The OR-Gate with top-k mechanism.** (a) Assume that the model output is 10-dimensional and k = 5. The different depths of the orange background in P_i are the top-k predictions. Y_i is in the top-k predictions of P_i, so Y_i is a reliable label, in contrast, Y_2 is unreliable. (b) Use the top-k mechanism to match P_i^t with Y_i. Y_i is considered reliable as long as the match succeeds once.

**Algorithm 1: OR-Gate.** Line 3-8: the warm-up; Line 10-11: splitting data by the OR-Gate; Line 6,15,20: matching model prediction and sample label by the top-k mechanism.

**4. EXPERIMENTS AND ANALYSIS**

4.1. Dataset and Implementation Details

We use the dev set and test set of VoxCeleb1 [19] respectively to train and evaluate, where the dev set contains 1211 speakers with 148,642 utterances. Since VoxCeleb1 has been manually checked, it can be considered as without noisy labels. In order to simulate different noisy label proportions, we add different proportions (0%, 5%, 10%, 20%, 30%, 50%) of closed-set noisy labels to the dev set for experiments.

We use ResNet34 [22] as the backbone, and attentive statistics pooling [20] and AM-Softmax [21] loss as the pooling layer and loss function respectively. The input of the network is 80-dimensional Fbank, the embedding dimension is set to 512 dimensions, and the mini-batch size is 128. The network parameters are optimized by the Adam optimizer, the initial learning rate is 0.0002 and every 5 epochs is reduced to 40% of the current one. The back-end takes cosine distance as a score. The warm-up hyperparameter has been set to w = 5 and the top-k mechanism used k = 90. For all experiments, the best EER of the last 5 epochs was used as the experimental result. All of our comparison experiments use the same network setup, except for the training strategy.

4.2. Results of comparison experiments

**Table 1**: EER(%) comparisons on VoxCeleb1 with different proportions of noisy labels added.

| System               | 0%   | 5%   | 10%  | 20%  | 30%  | 50%  |
|----------------------|------|------|------|------|------|------|
| Baseline             | 4.36 | 5.30 | 6.24 | 7.99 | 9.78 | 14.39|
| SELF [15] (Re-implented) | 5.52 | 5.94 | 5.90 | 6.34 | 7.50 | 11.90|
| LNCL+Sub-AM [16] + PLDA + Cosine | 4.64 | 4.83 | 5.11 | 5.45 | 5.56 | 6.32 |
| LNCL+Sub-AM [16] + NL-PLDA | 3.92 | 4.32 | 4.63 | 4.85 | 5.04 | 5.65 |
| OR-Gate              | 4.27 | 4.07 | 4.22 | 4.28 | 4.41 | 5.53 |

Table [1] shows the results of the comparison experiments. Firstly, it can be seen that the performance of baseline rapidly decreases with the increase of noisy labels, which proves that noisy labels do have a serious negative impact on the speaker model. Second, the OR-Gate achieved the best performance among all methods when the training data contained noisy labels. Finally, some seemingly counterintuitive data can be seen in the experiments, where the performance of the OR-Gate with a small number of noisy labels is better than the performance without noisy labels. It must be declared that all experimental data in this paper is authentic. We have published the performance curves for all experiments on the In-
and it can be seen that with fewer noisy labels, the amount of noisy data removed by the OR-Gate is relatively small, so the amount of data used for training does not differ much at noise rates of 0%, 5%, and 10%, respectively, and thus the performance of the model in these scenarios is neck-and-neck.

### 4.3. Results of filtering ability experiments

**Table 2:** F1 Scores for filtering noisy labels on VoxCeleb1 with different proportions of noisy labels added.

| Epoch | 0% | 5% | 10% | 20% | 30% | 50% |
|-------|----|----|-----|-----|-----|-----|
|       | 0.9756 | 0.9676 | 0.9562 | 0.9070 | 0.8248 | 0.4679 |
| 5     | 0.9973 | 0.9970 | 0.9966 | 0.9946 | 0.9887 | 0.8977 |
| 10    | 0.9988 | 0.9985 | 0.9981 | 0.9971 | 0.9939 | 0.9186 |
| 20    | 0.9990 | 0.9988 | 0.9983 | 0.9973 | 0.9941 | 0.9180 |
| 50    |      |      |      |      |      |      |
| 80    |      |      |      |      |      |      |

F1 scores are rarely used in the speaker verification field, but in order to evaluate the ability of the OR-Gate to select clean samples from a noisy labeled dataset, we add different tags for correct and wrong labels and track them during training. Specifically, we count the final $D_1$ contained true clean label and false clean label, and the $D_2$ contained false noisy label and true noisy label, these four statistics correspond to the TP, FP, FN, TN in the confusion matrix respectively, and then use the F1 Score, which synthesizes the precision and recall, to score the classification results. As shown in Table 2, the OR-Gate can largely extract the data with true label when the training data contains different proportions of noisy labels.

### 4.4. Results of robustness experiments

**Table 3:** EER(%) comparisons using different backbones.

| System            | Proportion of Noisy Labels |
|-------------------|-----------------------------|
|                  | 0%  | 5%  | 10% | 20% | 30% | 50% |
| Baseline          | 4.36| 5.30| 6.24| 7.99| 9.78| 14.39|
| OR-Gate           |     |     |     |     |     |     |
| $w = 1$           | 5.99| 6.27| 6.87| 8.37| 13.05| 25.83|
| $w = 2$           | 4.69| 4.41| 4.37| 4.68| 6.07 | 13.86|
| $w = 3$           | 4.16| 4.23| 4.11| 4.38| 4.83 | 8.10 |
| $w = 4$           | 4.17| 4.12| 4.19| 4.54| 4.60 | 6.41 |
| $w = 5$           | 4.27| 4.07| 4.22| 4.28| 4.41 | 5.53 |
| $w = 6$           | 4.07| 4.24| 4.47| 4.41| 4.52 | 5.37 |
| $w = 7$           | 4.19| 4.14| 4.31| 4.40| 4.55 | 5.20 |
| $w = 8$           | 4.44| 4.34| 4.23| 4.53| 4.66 | 5.16 |

Through further experiments, we explored the effect of different $w$ (or $k$) on the OR-Gate. As can be seen from Table 3 when the *warm-up* for 1 or 2 epochs is performed, the OR-Gate does not achieve good results because the training time is so short that the model does not learn sufficiently. However, after performing the *warm-up* for at least 3 epochs, the performance of the model can be greatly improved. Table 3 also illustrates that the $w$ value does not require meticulous selection and can achieve good performance over a wide range.

**Table 4:** EER(%) comparisons with different $k$ ($w = 5$).

| System         | Proportion of Noisy Labels |
|----------------|-----------------------------|
|                | 0%  | 5%  | 10% | 20% | 30% | 50% |
| Baseline       | 4.36| 5.30| 6.24| 7.99| 9.78| 14.39|
| OR-Gate        |     |     |     |     |     |     |
| $k = 1$        | 5.21| 4.81| 5.15| 5.58| 6.17| 8.55 |
| $k = 5$        | 4.67| 4.23| 4.31| 4.57| 5.21| 7.16 |
| $k = 10$       | 4.13| 4.48| 4.22| 4.62| 4.62| 6.56 |
| $k = 20$       | 4.26| 4.34| 4.37| 4.45| 4.64| 6.18 |
| $k = 30$       | 4.19| 4.15| 3.99| 4.45| 4.79| 6.28 |
| $k = 40$       | 4.33| 4.13| 4.31| 4.46| 4.77| 5.95 |
| $k = 50$       | 4.04| 4.21| 4.27| 4.17| 4.49| 5.74 |
| $k = 70$       | 4.32| 4.22| 4.16| 4.50| 4.63| 5.89 |
| $k = 90$       | 4.27| 4.07| 4.22| 4.28| 4.41| 5.53 |

We also explored the effect of the $k$ value for the OR-Gate performance at the same *warm-up* time. From Table 4 we can see that the OR-Gate’s performance is significantly improved by using the top-$k$ mechanism ($k > 1$) than not using the top-$k$ mechanism ($k = 1$). And over a wide range of $k$ value causes little deviation of performance, showing the robustness of $k$ value.

**Table 5:** EER comparisons using different backbones.

| System           | Proportion of Noisy Labels |
|------------------|-----------------------------|
|                  | 0%  | 5%  | 10% | 20% | 30% | 50% |
| TDNN             | 6.30| 7.27| 7.55| 9.13| 10.70| 13.61|
| ECAPA-TDNN       | 4.39| 5.57| 5.94| 7.83| 9.44 | 13.91|
| ResNet34         | 4.36| 5.30| 6.24| 7.99| 9.78 | 14.39|
| OR-Gate          |     |     |     |     |     |     |
| $w = 1$          | 6.33| 6.20| 6.43| 6.69| 7.03 | 8.34 |
| $w = 2$          | 4.33| 4.13| 4.31| 4.46| 4.77 | 5.95 |
| $w = 3$          | 4.26| 4.07| 4.22| 4.28| 4.41 | 5.53 |
| $w = 4$          | 4.07| 4.24| 4.47| 4.41| 4.52 | 5.37 |
| $w = 5$          | 4.27| 4.14| 4.31| 4.40| 4.55 | 5.20 |
| $w = 6$          | 4.44| 4.34| 4.23| 4.53| 4.66 | 5.16 |

Through further experiments, we explored the effect of different $w$ (or $k$) on the OR-Gate. As can be seen from Table 3 when the *warm-up* for 1 or 2 epochs is performed, the OR-Gate does not achieve good results because the training time is so short that the model does not learn sufficiently. However, after performing the *warm-up* for at least 3 epochs, the performance of the model can be greatly improved. Table 3 also illustrates that the $w$ value does not require meticulous selection and can achieve good performance over a wide range.

The OR-Gate is independent of particular network architecture, it can combine different backbones, pooling layers, and loss functions to achieve better performance, and is very portable. Table 5 demonstrates that different network architectures can be combined with the OR-Gate to achieve significant performance improvements.

### 5. CONCLUSIONS

In this paper, we propose a simple but effective method to filter the noisy labels from the speaker dataset. Experiments show that the OR-Gate achieves excellent performance with various proportions of noisy labels, as well as tracking experiments on noisy labels demonstrate the OR-Gate’s ability to filter noisy labels. Furthermore, the OR-Gate remains robust despite the increasing proportion of noisy labels and is suitable for various network architectures.

1 https://github.com/PunkMale/OR-Gate
6. REFERENCES

[1] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, “X-vectors: Robust dnn embeddings for speaker recognition,” in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 5329–5333.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[3] Brecht Desplanques, Jenthe Thienpondt, and Kris Dehmuyck, “ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification,” in Proc. Interspeech 2020, 2020, pp. 3830–3834.

[4] Zi-Kai Wan, Qing-Hua Ren, You-Cai Qin, and Qi-Rong Mao, “Statistical pyramid dense time delay neural network for speaker verification,” in ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022, pp. 7532–7536.

[5] Yang Zhang, Zhiqiang Lv, Haibin Wu, Shanshan Zhang, Pengfei Hu, Zhiyong Wu, Hung yi Lee, and Helen Meng, “MFA-Conformer: Multi-scale Feature Aggregation Conformer for Automatic Speaker Verification,” in Proc. Interspeech 2022, 2022, pp. 306–310.

[6] Lin Li, Fuchuan Tong, and Qingyang Hong, “When speaker recognition meets noisy labels: Optimizations for front-ends and back-ends,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 30, pp. 1586–1599, 2022.

[7] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee, “Learning from noisy labels with deep neural networks: A survey,” IEEE Transactions on Neural Networks and Learning Systems, pp. 1–19, 2022.

[8] Jiangchao Yao, Jiajie Wang, Ivor W. Tsang, Ya Zhang, Jun Sun, Chengqi Zhang, and Rui Zhang, “Deep learning from noisy image labels with quality embedding,” IEEE Transactions on Image Processing, vol. 28, no. 4, pp. 1909–1922, 2019.

[9] Kimin Lee, Sukmin Yun, Kibok Lee, Honglak Lee, Bo Li, and Jinwoo Shin, “Robust inference via generative classifiers for handling noisy labels,” in ICML, 09–15 Jun 2019, pp. 3763–3772.

[10] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz, “mixup: Beyond empirical risk minimization,” in International Conference on Learning Representations, 2018.

[11] Hongxin Wei, Lue Tao, RENCHUNZI XIE, and Bo An, “Open-set label noise can improve robustness against inherent label noise,” in Advances in Neural Information Processing Systems. 2021, vol. 34, pp. 7978–7992, Curran Associates, Inc.

[12] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey, “Symmetric cross entropy for robust learning with noisy labels,” in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 322–330.

[13] Lei Feng, Senlin Shu, Zhuoyi Lin, Fengmao Lv, Li Li, and Bo An, “Can cross entropy loss be robust to label noise?,” in Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, 7 2020, pp. 2206–2212.

[14] Junnan Li, Richard Socher, and Steven C.H. Hoi, “Divemix: Learning with noisy labels as semi-supervised learning,” in International Conference on Learning Representations, 2020.

[15] Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beguel, and Thomas Brox, “Self: Learning to filter noisy labels with self-ensembling,” in International Conference on Learning Representations, 2020.

[16] Fuchuan Tong, Yan Liu, Song Li, Jie Wang, Lin Li, and Qingyang Hong, “Automatic Error Correction for Speaker Embedding Learning with Noisy Labels,” in Proc. Interspeech 2021, 2021, pp. 4628–4632.

[17] Bengt J. Borgström and Pedro Torres-Carrasquillo, “Bayesian estimation of plda with noisy training labels, with applications to speaker verification,” in 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 7594–7598.

[18] Ragav Sachdeva, Filippe R. Cordeiro, Vasileios Belagiannis, Ian Reid, and Gustavo Carneiro, “Evidentialmix: Learning with combined open-set and closed-set noisy labels,” in 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), 2021, pp. 3606–3614.

[19] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, “Voxceleb: A large-scale speaker identification dataset,” in Proc. Interspeech 2017, 2017, pp. 2616–2620.

[20] Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda, “Attentive Statistics Pooling for Deep Speaker Embedding,” in Proc. Interspeech 2018, 2018, pp. 2252–2256.

[21] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu, “Additive margin softmax for face verification,” IEEE Signal Processing Letters, pp. 926–930, 2018.