Centroid-aware local discriminative metric learning in speaker verification.

Kekai Sheng, Weiming Dong, Wei Li, Joseph Razik, Feiyue Huang, Baogang Hu

Centroid-Aware Local Discriminative Metric Learning in Speaker Verification

Kekai Sheng[a,b], Weiming Dong[a,*], Wei Li[c], Joseph Razik[d], Feiyue Huang[c], Baogang Hu[a,b]

[a]*LIAMA-NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China*
[b]*University of Chinese Academy of Sciences*
[c]*Laboratory Youtu, Tencent Inc., Shanghai, China*
[d]*Laboratory LSIS, University of Toulon, Toulon, France*

Abstract

We propose a new mechanism to pave the way for efficient learning against class-imbalance and improve representation of identity vector (i-vector) in automatic speaker verification (ASV). The insight is to effectively exploit the inherent structure within ASV corpus — centroid priori. In particular: 1) to ensure learning efficiency against class-imbalance, the centroid-aware balanced boosting sampling is proposed to collect balanced mini-batch; 2) to strengthen local discriminative modeling on the mini-batches, neighborhood component analysis (NCA) and magnet loss (MNL) are adopted in ASV-specific modifications. The integration creates adaptive NCA (AdaNCA) and linear MNL (LMNL). Numerical results show that LMNL is a competitive candidate for low-dimensional projection on i-vector (EER=3.84% on SRE2008, EER=1.81% on SRE2010), enjoying competitive edge over linear discriminant analysis (LDA). AdaNCA (EER=4.03% on SRE2008, EER=2.05% on SRE2010) also performs well. Furthermore, to facilitate the future study on boosting sampling, connections between boosting sampling, hinge loss and data augmentation have been established, which help understand the behavior of boosting sampling further.

Keywords: Text-Independent ASV, Centroid-Aware Balanced Boosting Sampling, Adaptive Neighborhood Component Analysis, Linear MagNet

1. INTRODUCTION

Automatic speaker verification (ASV) is an important yet difficult pattern recognition task, and it can be divided into two categories: text-dependent ASV and text-independent one. We focus on the latter task in this work. An ASV system is usually composed of two modules: one is front-end for acoustic

*Corresponding author
Email address: weiming.dong@ia.ac.cn (Weiming Dong)

feature extraction and voice activity detection (VAD), and the other is back-end for representation extraction (e.g., i-vector [1]) and similarity measure.

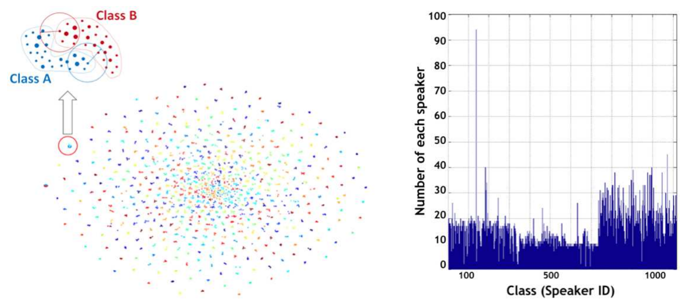

Figure 1: Feature visualization (t-SNE) [2] of i-vectors from the background corpus in National Institute of Standards and Technology speaker recognition evaluation (SRE) 2010 (*left*) and the corresponding histogram of numbers of instances of each speaker (*right*).

Technically, ASV systems are susceptible to intersession variability (intra-speaker variability [3, 4] and inter-speaker one [5]), causing local confusions (left of Fig. 1). Class-imbalance is another issue (right of Fig. 1), being counter-productive to learning as the training signal can be biased by class-imbalance data. To improve results against the issues, many methods [6, 7, 8, 9, 10] have been proposed. Amongst the previous literature of improved representation of i-vector space, it is observed that the adverse effect of class-imbalance seems to have received little attention. But we argue that class-imbalance does inhibit optimization progress and render the representation space of not discrimination enough, especially when the amount of data is limited (e.g., SRE 2008).

With the point in mind, we try to investigate how to improve the i-vector space and eschew class-imbalance simultaneously. The key idea here is to propose a mechanism efficiently exploiting the *centroid-priori* of ASV corpus [11]. In particular: 1) centroid-aware balanced boosting sampling collects class-balance mini-batches for efficient learning process, and 2) neighborhood component analysis (NCA) [12] or magnet loss (MNL) [13] strengthens local discriminative modeling on the mini-batches. Integrating the two modules creates adaptive NCA (AdaNCA) and linear MNL (LMNL). Comparisons with several typical metric learning methods demonstrate the effectiveness of our mechanism.

The main contributions of this work are summarized as follows:

1 Centroid-aware balanced boosting sampling is developed to mollify class-imbalance and pave the way for efficient learning, like hard example mining.

2 AdaNCA and LMNL based on NCA and MNL are developed with ASV-specific modifications to strengthen local discriminative modeling.

3 Connections are established between boosting sampling, hinge loss and data augmentation to further reveal the behavior of boosting sampling.

The rest of the paper is organized as follows. Section 2 introduces the related works. Section 3 describes the problem setting and the proposed mechanism.

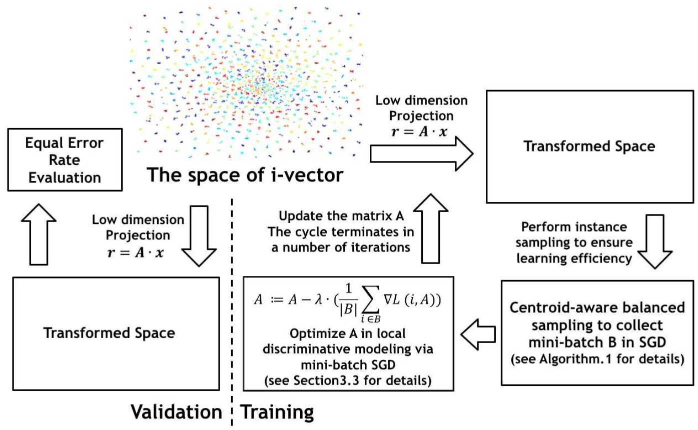

Figure 2: Illustration of the pipelines of the training stage of the mechanism (instance sampling + local discriminative modeling) and the validation stage in ASV task scenarios.

Section 4 presents details of the experiments and Section 5 summarizes useful knowledge in ASV. Section 6 carries out further investigation on the boosting sampling strategy. Section 7 elaborates the conclusions and future works.

2. RELATED WORKS

2.1. I-vector in ASV

Identity vector (i-vector) [1], also known as total variability modeling, aims to model the utterance variability in a low-dimensional space. Total variability originates from joint factor analysis, and the model is represented as

$$\mathbf{M}(s) = \mathbf{m} + \mathbf{T} \cdot \omega(s), \tag{1}$$

where s denotes a target speaker; $\mathbf{m}$ is a speaker-/channel-independent supervector, which is taken from a universal background model (UBM) supervector; $\mathbf{T}$ is the total factor matrix, which is an expanded subspace of speaker-/channel-dependent information; $\omega(s)$ is the i-vector extracted from the input utterance (details are available in [14]). I-vector has proved to be an effective representation for the inherent information of speakers (see e.g., [15, 16, 17]).

As previously noted, however, the performance degradation in ASV systems is attributed to the two issues. Many researchers attempt to mollify the issues with metric learning techniques, such as linear discriminant analysis (LDA), nearest neighbor discriminant analysis (NDA) [18] for low-dimensional projection on i-vector, or probabilistic LDA (PLDA) [7, 19] for intersession variability compensation. In this work, we resort to better representation of i-vector space and tackle class-imbalance issue at the same time to encourage the optimization to progress further, resulting in improved ASV result.

2.2. Distance metric learning

Distance metric learning aims at learning a transformation to a representation space where the distance corresponds with a task-specific notion of similarity. Classical examples of this research field are listed as follows.

Given the set of N (i-vector, label) pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, LDA is commonly used for low-dimensional projection on i-vector before PLDA in ASV for the sake of computational efficiency. NDA [20] models the variances based on K nearest neighbors (KNN) of $\mathbf{x}_i$, $NN_K(\mathbf{x}_i)$, to better preserve the local data structure. NCA [12] is another scheme used to train the linear transformation $\mathbf{A}$ on nearest neighbors modeling (i.e., p_{ij} in Eq. 2), which is akin to t-SNE [2] — i.e., a useful technique to unveil the structure in data.

$$f(\mathbf{A}) = \sum_{i=1}^{N} p_i \ = \sum_{i=1}^{N} \sum_{j \in \{l | y_l = y_i\}} p_{ij}, \ with \ p_{ij} = \frac{\exp(-\|\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_j\|_2^2)}{\sum_{k \neq i} \exp(-\|\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_k\|_2^2)} \ (p_{ii} = 0) \quad (2)$$

Weighted class-oriented linear regression [21] and fast NCA (FNCA) [22] leverage adaptive weighted learning to reinforce the local awareness of models. Large margin nearest neighbor (LMNN) [23] learns the transformation with two parts in Eq. 3 (μ denotes the tradeoff):

$$L(\mathbf{A}) = (1 - \mu) \cdot L_{pull}(\mathbf{A}) + \mu \cdot L_{push}(\mathbf{A}) \quad (3)$$

where $L_{pull}(\mathbf{A})$ pulls instances of identical class closer and $L_{push}(\mathbf{A})$ pushes data of different label farther. Comparatively, stochastic triple embedding [24] learns the representation in a triplet manner, and joint Bayesian model (JBM) starts from a different assumption to obtain success in face verification task [25]. Magnet loss (MNL) [13] and lifted structured feature embedding [26] are two latest well-designed methods for enhanced local discrimination modeling.

In ASV scenarios, NDA has proved to be a strong substitute for LDA [18]. It should be noted that, the exploitation of the local data structure is at the heart of the success of NDA. Inspired by NDA and the recently successful applications of local discriminative modeling (see e.g., [13, 27, 28]), we desire to improve the representation of i-vector space by virtue of the local discriminative modeling in exploiting the inherent structure in ASV data.

2.3. Sampling strategies in stochastic gradient descent

Mini-batch stochastic gradient descent (SGD) is an optimization algorithm and has proved to be beneficial to various tasks [29]. Eq. 4 shows the idea:

$$W_{k+1} = W_k - \lambda_k \cdot (\frac{1}{|B_k|} \sum_{x_i \in B_k} \nabla L(x_i, W_k)) \quad (4)$$

where B_k denotes the mini-batch, W_k is the weight being optimized at step k with learning rate λ_k; $L(x_i, W)$ refers to the loss given data x_i and weights W.

Since random shuffling is unable to ensure training efficiency against undesirable properties in data (e.g., noise or redundancy) in many tasks, many researchers strengthen the training signal via better mini-batches. For example, Shrivastava *et al.* [30] made it through an online hard example mining method.

Yang *et al.* [31] enhanced the optimization efficiency by proposing drop-sample algorithm. Miguel *et al.* [27] leveraged complete-linkage cluster to discover compact data cliques and enabled CNNs in exemplar learning. Canévet *et al.* [32] regarded the problem in collecting false positives as an exploitation (focus on false positives) versus exploration (use the entire data) dilemma and applied Monte Carlo tree search to solve it. They share one idea: well-selected unbiased mini-batch B_k possesses obvious advantage over randomly-selected ones in ensuring the quality and learning efficiency of models. With this view, the approach to collect B_k is generally task-specific and worthy of effort.

Inspired by the idea, to prevent the gradient estimates from being vitiated by the imbalance corpus and exploit the benefit in online mini-batch selection for learning efficiency, we need to develop an effective class-balance sampling strategy during the course of optimization in ASV setting.

3. PROBLEM SETTING AND MODIFICATIONS

3.1. Problem setting

Considering N (i-vector, label) pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i \in \mathcal{R}^{d_{ivector}}$ and $y_i \in \{1, \cdots, N_{class}\}$, we train a linear transformation $\mathbf{A}$ with an objective or loss (e.g., LMNL Eq. 8 and AdaNCA Eq. 7) for a transformed space $\{\mathbf{r}_i = \mathbf{A} \cdot \mathbf{x}_i\}_{i=1}^{N}$, where $\mathbf{r}_i \in \mathcal{R}^{d_{projection}}$, with a low equal error rate (EER, the error value when the false acceptance rate is equal to the false rejection rate after adjusting the threshold value). Fig. 3 shows the pipeline.

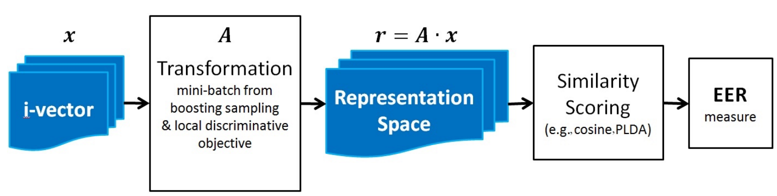

Figure 3: Workflow of experiments with distance metric learning in ASV.

In the test/validation stage, the similarity between two instances is:

$$d_{i,j}(\mathbf{A}) = \mathbf{r}_i^T \cdot \mathbf{r}_j = \frac{\mathbf{x}_i^T \cdot \mathbf{A}^T \mathbf{A} \cdot \mathbf{x}_j}{\|\mathbf{A}\mathbf{x}_i\|_2 \cdot \|\mathbf{A}\mathbf{x}_j\|_2}. \tag{5}$$

Cosine metric, Eq. 5, is adopted for similarity scoring to reveal the discrimination of the representation. To further ensure the necessity of the low-dimensional projection from $\mathbf{A}$ in the ASV community — i.e., whether lower EER can be achieved — PLDA should be performed on the transformed space.

As mentioned previously, the main purpose of this work is to better ASV results with improving representation of i-vector while boosting optimization efficiency against class-imbalance. We propose a mechanism to attain the goal by exploiting the inherent structure in ASV data. Specifically, centroid-aware sampling (see Sec. 3.2) generates genuinely hard unbiased mini-batch to ensure the learning efficiency, and local discriminative objective (see Sec. 3.3) is introduced to help strengthen local discriminative modeling on the mini-batches.

The motivations behind the combination are twofold. First, a well-designed sampling strategy is a promising solution to ensure the training efficiency of models, especially when there is class-imbalance issue or when there are much more easy examples than meaningfully hard ones, which would weaken the training signal. Second, a local discriminative objective can effectively absorb the information from mini-batches of the collected genuinely hard instances (e.g., high error) and result in a better representation space, leading the sampling process to mine for other confusable neighbor structures. Therefore, we believe that the integration of the solutions helps get the best of both worlds.

3.2. Centroid-aware balanced boosting sampling

In this section, we describe the centroid-aware balanced boosting sampling algorithm, which uncovers the internal structure within ASV corpus to combat class-imbalance for efficient learning. The pseudocode is shown in Alg. 1.

The motivation behind Alg. 1 is: clustering is a natural avenue to reveal the inherent data structure, as it can organize data into cliques and make it easy to detect the confusing regions, akin to hard example mining [30, 31]. However, it may also incur undesirable high computational workload. To avoid extra clustering computation cost, we propose three techniques as follows:

- Centroid similarity of two clusters of different speakers. Given clusters of two speakers (i.e., $C_i = \{l|y_l = i\}, C_j$), four metrics [33] in Eq. 6 are available : $S_{GA}(C_i, C_j)$ (group-average), $S_{SL}(C_i, C_j)$ (single-link), $S_{CL}(C_i, C_j)$ (complete-link) and $S_{CENT}(C_i, C_j)$ (centroid similarity).

$$S_{SL}(C_i, C_j) = \min_{a \in C_i, b \in C_j} < \mathbf{x}_a, \mathbf{x}_b >, \quad S_{CL}(C_i, C_j) = \max_{a \in C_i, b \in C_j} < \mathbf{x}_a, \mathbf{x}_b >$$
$$S_{CENT}(C_i, C_j) = \frac{1}{|C_i| \cdot |C_j|} \sum_{a \in C_i} \sum_{b \in C_j} < \mathbf{x}_a, \mathbf{x}_b > \tag{6}$$
$$S_{GA}(C_i, C_j) = \frac{1}{(|C_i| + |C_j|)(|C_i| + |C_j| + 1)} \left[\| \sum_{a \in C_i \cup C_j} \mathbf{x}_a \|_2^2 - (|C_i| + |C_j|) \right]$$

Fig. 4 illustrates these similarity measures for better understanding. In-

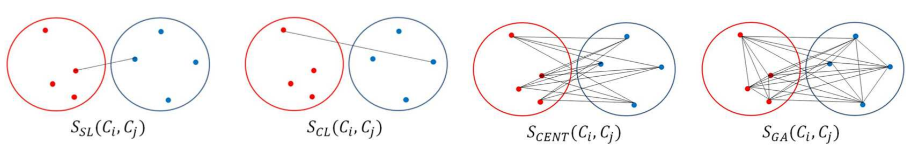

Figure 4: Schematic illustrations of the four similarity measures: single-link similarity, complete-link similarity, centroid similarity, and group-average similarity.

tuitively, different measures uncover different aspects of data, and it's the specific task that determines which is the best (e.g., $S_{SL}(C_i, C_j)$ for agglomerative clustering; $S_{CL}(C_i, C_j)$ for compact cliques detection [27]).

Algorithm 1 *Centroid-aware* balanced boosting sampling

Require: $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$; $\mathbf{A}$; $C = \{C_j = \{i \mid y_i = j\}\}_{j=1}^{N_{class}}$ (N_{class} refers to the total number of different classes in the corpus); $F = \{F_j \in \{0,1\}\}_{j=1}^{N_{class}}$ records whether the class j has been sampled during sampling for a single mini-batch, and $index_{(1:M \cdot D)}$ records data indexes in a single mini-batch; In each mini-batch: M is the number of different classes and D denotes the number of samples of identical label; N_{batch} refers to the number of mini-batches (for computational efficiency and stable training dynamics)

Ensure: vector of indexes of samples in mini-batches $index_{(1:M \cdot D)}^{mini-batch}$

1: Perform length-normalization $\{\hat{\mathbf{r}}_i \leftarrow \frac{\mathbf{A} \cdot \mathbf{x}_i}{\|\mathbf{A} \cdot \mathbf{x}_i\|_2}\}_{i=1}^{N}$ for feature comparability
2: Initialize the array of sampling flags $F_i \leftarrow 0$ ($i = 1, \cdots, N_{class}$)
3: Initialize $index_{(1:M \cdot D)}^{mini-batch}$ and $index_{(1:M \cdot D)}$ as empty vector
4: Generate $L \leftarrow \{(i, j, S_{CENT}(C_i, C_j)) \mid i, j \in \{1, 2, \cdots, N_{class}\}$ *and* $i < j\}$
5: Sort L in a descend manner by $S_{CENT}(C_i, C_j)$
6: **while** L *is not empty and* $|index_{(1:M \cdot D)}^{mini-batch}| < N_{batch}$ **do**
7:     pop one tuple $(i, j, S_{CENT}(C_i, C_j))$ from the top of L
8:     **if** $F_i == 0$ *and* $|C_i| \leq D$ *and* $|index_{(1:M \cdot D)}| < M \cdot D$ **then**
9:         randomly pick D indexes out of C_i and stack them into $index_{1:M \cdot D}$
10:         $F_i \leftarrow 1$ // 1 for being sampled yet
11:     **end if**
12:     do the same on class C_j as lines 8 to 11
13:     **if** $|index_{(1:M \cdot D)}| == M \cdot D$ **then**
14:         Stack $index_{(1:M \cdot D)}$ into $index_{(1:M \cdot D)}^{mini-batch}$
15:         re-initialize F_i (line 2) and reset $index_{(1:M \cdot D)}$ as an empty vector
16:     **end if**
17: **end while**
18: **return** $index_{(1:M \cdot D)}^{mini-batch}$
19: // The total computational complexity is : $O(N + N_{class}^2 \cdot log(N_{class}))$

In this case, we choose $S_{CENT}(C_i, C_j)$. As *centroid prior* is intrinsic in ASV data [11] and $S_{CENT}(C_i, C_j)$ can be aware of the centroid of speakers (evidence in Section. 5). That is why Alg. 1 is called *centroid-aware*.

- Class-balance boosting sampling with replacement. Essentially, class-imbalance decreases the ratio of hard examples as training progresses, and then models succumb to the biased training signal. So we'd better focus on the instances of large error to learn the fastest and eventually the best [34]. To this end, we generate mini-batches in a pairwise, hard-instance-first and class-balance manner. In particular, we evaluate $S_{CENT}(C_i, C_j)$ of pairs of different speakers (line 4 of Alg. 1), sort them in descending order (line 5 of Alg. 1), and maintain class-balance of each mini-batch (lines 6 to 17 of Alg. 1). Replacement strategy (according to $\{F_j\}_{j=1}^{N_{class}}$ in Alg. 1) is also adopted to equip models with the capability to reuse critical data points adaptively, and consequently, we can boost the training efficiency further.

- Random sampling as a regularizer. There is always some overfitting or oscillatory training dynamics coming with boosting sampling. To combat the detrimental effect, we perform random sampling (lines 9 to 10 of Alg. 1) in mini-batch generation. We believe that random sampling can help generate relatively unbiased mini-batches and make the gradient directions more explorative (e.g., less overfitting) to mitigate overfitting, working better than nearest neighbor sampling (e.g., NDA [18] or FNCA [22]).

Combine the aforementioned techniques and we propose Alg. 1, the sampling strategy in our optimization process. The algorithm is able to incorporate useful data structures, i.e., class-balance meaningfully hard mini-batches, into a local adaptive metric for deeper optimization while keeping the computational cost tolerable. We share the same idea with Allen-Zhu *et al.* [35] that the optimal sampling process corresponds to gathering more instances of larger gradient contribution. Furthermore, the encoded sampling encourages models to learn in an ensemble manner for high model stability and low generalization error.

Class-aware sampling [36] is most closely related to Alg. 1. However, our method is endowed with $S_{CENT}(C_i, C_j)$, boosting sampling and replacement strategy, which are useful techniques in encouraging the optimization to progress. No shuffle operation is performed on all the mini-batches, as our empirical results show that keeping the order of mini-batches boosts the training effectiveness.

3.3. Objective selection and modifications

As for the other module of our mechanism, we introduce two objectives — NCA [12] and MNL [13] — to strengthen local discriminative modeling. The reasons for the choices are twofold. First, a desirable representation modeling should originate from the inherent structure in data. Being independent of any task-free data assumption means that the model can get rid of the limitations from the predefined assumptions and mine the internal data structure directly. Consequently, the representation space from the model can be consistent with the real data distribution. Second, for the sake of local discrimination, a desired objective should be aware of the internal structure in data, neither simply pair-wise [23] nor triplet-wise [37]. Intuitively, such awareness can lead to a promising remedy for local discriminative modeling on i-vector space.

In this case, *centroid priori* is associated with ASV corpus [11] and readily available. Hence, NCA and MNL are two promising objectives to formalize the idea: NCA [12] performs the modeling on neighbor assignment akin to t-SNE, and MNL [13] offers a promising approach to take advantage of centroid prior. Given one mini-batch S_i in $index_{(1:M \cdot D)}^{mini-batch}$ from Alg. 1, their ASV-specific modifications, AdaNCA in Eq. 7 and LMNL in Eq. 8, are shown as follows.

$$
\begin{aligned}
&\hat{\mathbf{A}} = \arg\max_{\mathbf{A}} \frac{1}{Z} \sum_{j \in S_i} \sum_{k \in \{a | y_a = y_j, a \neq j\} \cap S_i} \frac{exp(-\frac{1}{2\sigma^2} \|\mathbf{A} \cdot (\mathbf{x}_j - \mathbf{x}_k)\|_2^2)}{\sum_{l \in S_i, l \neq j} exp(-\frac{1}{2\sigma^2} \|\mathbf{A} \cdot (\mathbf{x}_j - \mathbf{x}_l)\|_2^2)} \\
&with\ Z = M \cdot D,\ \mu(\mathbf{x}_i) = \frac{1}{D} \sum_{j \in \{l | y_l = y_i\}} \mathbf{x}_j,\ \sigma = \frac{1}{Z-1} \sum_{j \in S_i} \|\mathbf{A} \cdot (\mathbf{x}_j - \mu(\mathbf{x}_j))\|_2^2 \\
&s.t.:\ S_i = index_{(1:M \cdot D)}^{mini-batch}[i]\ from\ Alg.1
\end{aligned}
\tag{7}
$$

where $\mu(\mathbf{x}_i)$ refers to the centroid of class of $\mathbf{x}_i$, σ is a normalization factor to facilitate local discriminative modeling for a given mini-batch S_i.

$$
\begin{aligned}
&\hat{\mathbf{A}} = \arg\min_{\mathbf{A}} \frac{1}{Z} \sum_{m=1}^{M} \sum_{d=1}^{D} \{-\log \frac{exp(-\frac{1}{2\sigma^2} \|\mathbf{r}_d^m - \mu_m\|_2^2 - \alpha)}{\sum_{\mu: C(\mu) \neq C(\mathbf{r}_d^m)} exp(-\frac{1}{2\sigma^2} \|\mathbf{r}_d^m - \mu\|_2^2)}\}_+ \\
&with\ Z = M \cdot D,\ \mathbf{r}_d^m = \mathbf{A} \cdot \mathbf{x}_d^m,\ \mu_m = \frac{1}{D} \sum_{d=1}^{D} \mathbf{r}_d^m,\ \sigma = \frac{1}{Z-1} \sum_{m=1}^{M} \sum_{d=1}^{D} \|\mathbf{r}_d^m - \mu_m\|_2^2 \\
&s.t.:\ \mathbf{x}_d^m\ denotes\ the\ representation\ of\ d_{th}\ index\ of\ m_{th}\ class\ in\ S_i, \\
&\qquad S_i = index_{(1:M \cdot D)}^{mini-batch}[i]\ from\ Alg.1
\end{aligned}
\tag{8}
$$

where $C(\cdot)$ denotes the class ID of input (given a centroid μ, $C(\mu)$ refers to the class ID from which we calculate μ); α is the margin in hinge loss.

To our knowledge, this paper is the first to leverage MNL in ASV setting. We choose to adopt MNL without deep learning and provide mini-batches via Alg. 1 to work better in ASV scenarios. The results help better understand MNL and provide valuable knowledge for enhancing ASV systems.

4. EXPERIMENT CONFIGURATIONS

4.1. Corpora usage

Generally, the corpus for ASV task contains three disjoint sets: development set, enrollment set and trial set. The development set is utilized for training UBM, i-vector extractor and others techniques (e.g., LDA, PLDA) in the back-end; the remaining two sets are used to estimate the generalized performance of a given ASV system.

Information about the corpus usage is summarized in Table 1. All available corpora are collected to achieve good results, whereas a few useless (e.g., near silence or laughter) or poor (e.g., strong echo or noise) utterances are omitted to eliminate their negative effects on the training process.

Corpus	UBM	T-matrix	back-end for SRE2008	back-end for SRE2010
Switch Board	√	√		
SRE 2004	√	√	√	√
SRE 2005	√	√	√	√
SRE 2006	√	√	√	√
SRE 2008	√	√		√

Table 1: Corpora for different modules during experiments on SRE2008 and SRE2010.

4.2. Experimental setup: front-end and i-vector extraction

In the experiments, MFCCs of 20 dimensions (19 + energy) on the window of 20 ms with 10 ms shift are augmented with their delta and double delta coefficients, producing 60-dimensional feature vectors. Then, the feature vectors are subjected to feature warping. Prior to short-time cepstral mean and variance normalization, silence in recordings is trimmed with VAD. Subsequently, gender-independent UBM (full covariance 2048 component GMM) and gender-dependent i-vector ($d_{i-vector}$=600) extractors are trained. On the basis of i-vector (EER=6.41% on SRE2008 and EER=6.23% on SRE2010), several different metric learning methods are performed for improved representation space and better ASV results. The code for feature extraction, VAD, and i-vector processes (e.g., training and extraction) comes from kaldi [38], a popular toolkit for speech recognition and speaker verification. Hence our baseline ASV system is guaranteed to be realistic.

Male telephone data (det6) from core condition (short2-short3) on SRE2008 and male telephone data (det5) on SRE2010 are used in the evaluation, and EER is evaluated based on the scores from Eq. 5.

4.3. Configurations of methods in the experiments

Given the non-convexity in AdaNCA and LMNL, SGD with momentum is utilized for optimization purpose. For a good initialization of $\mathbf{A}$ to encourage convergence of training, we leverage the orthogonal matrix from LDA to initialize $\mathbf{A}$, similar to Dehak *et al.*[16], since LDA has demonstrated obvious improvements on i-vector and training stability. Besides, the normalization pre-process (zero-average normalization and length normalization) on i-vectors should be performed prior to the training to facilitate optimization progress.

In addition to the proposed mechanism, various metric learning methods [1] are taken into account (see Table 2 for the specific parameter setting) in comparison experiments to figure out the key in optimizing the transformation in ASV. Moreover, we believe that a wide range of methods, different assumptions (e.g., LDA, NDA [18], ITML [39] and JBM [25]) or representation modeling (e.g., LMNN [23], AdaNCA and LMNL), can collaborate precisely as exploratory techniques to understand the inherent structure in data and provide enlightening insights for further development in ASV systems.

[1] We adopt the code sources online: LDA in kaldi, Matlab toolbox for dimensionality reduction for LMNN, and Python implementations of metric learning algorithms for ITML.

Method	Parameters
LDA	output dimension=210
LMNN [23]	PCA pre-process, K=3, μ=0.5
ITML [39]	γ=1.0
NDA [18]	K=8
JBM [25]	PCA pre-process
AdaNCA	LDA-initial, M=60, D=4, N_{batch}=60, λ=0.14, momentum=0.9
LMNL	LDA-initial, M=200, D=4, N_{batch}=20, α=0.5, λ=0.1, momentum=0.6

Table 2: Parameter setting of various distance metric learning methods in experiments.

5. RESULT ANALYSIS

Important information of various metric learning methods (i.e., their properties and ASV results) is organized in Table 3 for comparison. To further verify the role of AdaNCA and LMNL in ASV community, we perform PLDA on the raw i-vectors (to check the necessity of low-dimensional projection) and the representations after low-dimensional projection (LDA, AdaNCA and LMNL).

	Locality Modeling	Parametric distribution	Sampling method	SRE2008 cosine (PLDA)	SRE2010 cosine (PLDA)
i-vector				6.41% (5.61%)	6.23% (1.98%)
LDA		$\surd$		4.69% (4.00%)	2.55% (1.99%)
LMNN [23]	$\surd$		random	8.88%	10.40%
ITML [39]		$\surd$		15.98%	12.75%
NDA [18]	$\surd$	$\surd$	nearest	5.49%	2.55%
JBM [25]		$\surd$		6.89%	3.08%
AdaNCA	$\surd$		Alg. 1	4.38% (4.03%)	2.34% (2.05%)
LMNL	$\surd$		Alg. 1	**4.24%** (**3.84%**)	**2.29%** (**1.81%**)

Table 3: Comparison of different metric learning methods on i-vector representation and their corresponding verification performance (EER) on SRE2008 and SRE2010. The dimension of the PLDA speaker subspace remains the same as the dimension of input representation.

The table shows considerable valuable knowledge on ASV as follows:

- Local structure modeling is critical in improving representation of i-vector. NDA (variances from *nearest neighbors*), AdaNCA (*neighbor assignments* with p_{ij}), and LMNL (*centroid priori* within $\hat{\mu}_m$) adopt different local discrimination modeling, and they generally outperform global discrimination methods in EER. Specifically, LMNL builds a better representation of i-vector to enable PLDA to achieve lower EER performance. Therefore, the proposed mechanism (boosting sampling + local discriminative modeling) is very competitive and efficient in ASV task scenarios.

  Besides the good results, it's also observed that when M or D increases too high or diminishes too small in Alg. 1, or when B_k are randomly collected, the learning efficiency is poor and the learned models always result in high EER. That is, the improper exploitation of the hard mini-batches (e.g., too greedy or too weak) results in poor explorative properties in SGD progress (e.g., underfitting), and subsequently, models tend to converge on local minima of high EER. Hence, it is clear that a proper approach for exploitation of the inherent structures in ASV corpus requires some efforts and we provide an applicable mechanism that is worthy of reference.

- Priority should be given to intra-class variation. In AdaNCA or LMNL, models prefer to pull instances of the same label closer than to push instances of impostor farther, as the gradient signal from neighbors of the target speaker generally surpasses that from non-target ones. Their successes imply that the pull force from target speakers makes more contributions to useful gradient estimates than the push force from non-target speakers does. Intuitively, the push force from a great number of non-target instances is prone to be contaminated by noise.

  To analyze the contribution of pull or push force quantitively, we resort to LMNN — $L_{pull}(\mathbf{A})$ and $L_{push}(\mathbf{A})$ easy the analysis. We train LMNN in various hyper-parameters ($K \in \{1, 3, 5\}$, $\mu \in \{0.1, \cdots, 0.9\}$ and PCA dimension $\in \{100, 110, \cdots, 400\}$), and the results are shown in Fig. 5. Two

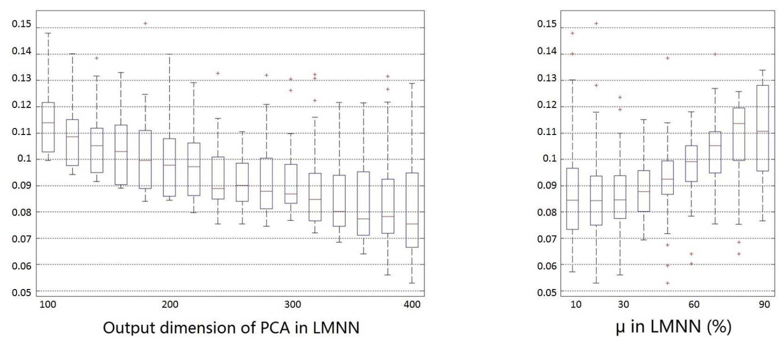

Figure 5: Box plots of EER of LMNN with different settings of parameters on SRE2008. EER goes with the output dimension of PCA (*Left*) and EER goes with the μ (*Right*).

  box plots show the influence of each hyper-parameter on the ASV performance. The tradeoff value, μ, in Eq. 3 of around 0.3 appears to be the best, suggesting that $L_{pull}(\mathbf{A})$ from the same speaker takes more credits than $L_{push}(\mathbf{A})$ from samples of different impostors for promising results.

- $S_{CENT}(C_i, C_j)$ reveals more useful internal structures for ASV than others in Eq. 6. As noted before, measures in Eq. 6 lead models to focusing on different aspects of data. Experimental results show that $S_{CENT}(C_i, C_j)$ exhibits training stability and low EER. Comparatively, $S_{GA}(C_i, C_j)$ presents higher EER (about 0.2% higher than S_{CENT}) and reaches the training convergence at a slower rate; $S_{SL}(C_i, C_j)$ and $S_{CL}(C_i, C_j)$ seem to be vulnerable to data noise and even render the optimization oscillatory, resulting in EER values just moderately lower than the baseline.

  Fig. 6 also buttress the role of centroid in ASV setting: mean-shift performs satisfactory in modeling the data distribution of i-vectors. In spite of some misfits, a large ratio of speakers properly center around each corresponding centroid. So it is arguable that exploiting *centroid prior* within the mini-batch generation or the objective function helps obtain accurate representation modeling and promote ASV performance.

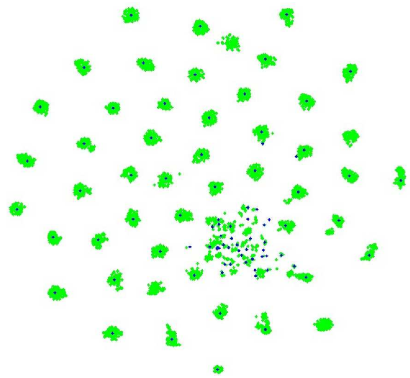

Figure 6: T-SNE visualization on i-vectors from the background set of SRE2010 (green points) and the results (blue points) of mean-shift. Although a few mismatches are found, large parts of the centroid successfully cover a single local region of i-vectors of the same speaker.

6. FURTHER INVESTIGATION ON BOOSTING SAMPLING

Since the integration of two solutions has proved to be effective for ASV task, a natural question follows: can we find some equivalence between sampling process (e.g., Alg. 1) and local discriminative objective (e.g., Eq. 3)? In what follows, we provide insights into the behavior of boosting sampling by establishing its connections with hinge loss and data augmentation. The insights can shed a new light on boosting sampling strategy.

6.1. Hinge loss and boosting sampling

As the visualization in Fig. 7 shows: with the help of sampling algorithm, models tend to pour more attention on regions of similar impostors while ignoring easy-to-classify areas. That is, both AdaNCA and LMNL are only triggered

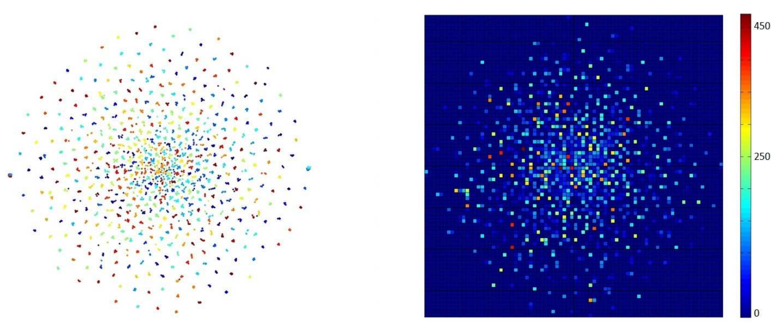

Figure 7: A schematic illustration of centroid-aware balanced boosting sampling working in a similar way to hinge loss. T-SNE visualization on i-vectors from SRE2008 (*left*) and the corresponding heatmap of frequency data being sampled as SGD iterations proceed (*right*).

by hard instances: AdaNCA conducts SGD iterations on mini-batches from

Alg. 1, and LMNL filters $index^{mini-batch}_{(1:M\cdot D)}$ further with hinge loss. Thus, some connections exist between boosting sampling strategy and hinge loss.

Another evidence comes from the derivative functions: hinge loss (Eq. 9; α is the margin) and loss function based on the collected set from sampling (Eq. 10) both emphasize on instances with large gradients to perform better near the decision boundary.

$$\begin{aligned} L(y) &= \sum_i max(0, \alpha - t_i \cdot y_i),\ (t_i = \pm 1,\ y_i\ for\ a\ classifier\ score) \\ \frac{\partial \ell}{\partial w} &= \sum_i -t_i \cdot \mathbb{1}\{\alpha > t_l \cdot y_l\} \cdot \frac{\partial y_i}{\partial w} = \sum_{i \in \{l | t_l \cdot y_l < \alpha\}} -t_i \cdot \frac{\partial y_i}{\partial w} \end{aligned} \tag{9}$$

$$\begin{aligned} L_{sampling}(y) &= \sum_{i \in collected\ set} L(y_i),\ (y_i\ for\ a\ classifier\ score) \\ \frac{\partial L_{sampling}}{\partial w} &= \sum_{i \in collected\ set} \frac{\partial L}{\partial y_i} \cdot \frac{\partial y_i}{\partial w} \end{aligned} \tag{10}$$

They are almost the same when the collected set — collected via a certain mini-batch generation algorithm — is $\{l | t_l \cdot y_l < \alpha\}$. Furthermore, they are also reciprocal: boosting sampling collects meaningful mini-batches from corpus for hinge loss, and hinge loss filters the mini-batches further (as LMNL does in Eq. 8). Many researchers integrate hard example mining into their models as an indispensable step for promising results. For instance, Chen *et al.* [40] proposed double-header hinge loss with hard quadruplets based on sampling for significantly large margin and computational efficiency.

On this basis, a connection is established between boosting sampling and hinge loss, and it can lead to better design of the optimization process.

6.2. Data augmentation and boosting sampling with replacement

There is also a link existing between boosting sampling and data augmentation. This deduction is best illustrated with the two case studies in Fig. 8. In the left branch, each utterance generates several artificial copies to appear in different mini-batches without changing the labels. In the right branch, boosting sampling with replacement enables the model to adaptively reuse critical instances. They both allow every instance to contribute to the training signal several times in each SGD iteration over the whole development set. Moreover, boosting sampling with replacement helps mollify the challenges in designing augmentation (e.g., how and how many). Thus, boosting sampling with replacement encodes data augmentation into optimization without the need for a specific augmentation strategy.

In fact, we tried to augment the extracted i-vectors from ASV corpus using a class invariant method [2], but models suffer from those artificial copies

[2]We randomly sample approximately 70% of the original frames from each utterance for 50 copies and extract the corresponding i-vector. Given that the artificial i-vectors are closely similar to the original one, the augmentation strategy will not change the correct class.

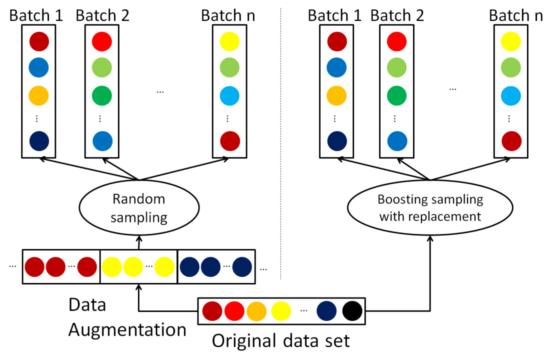

Figure 8: Data augmentation (*left*) and boosting sampling with replacement (*right*) allow a single training sample to contribute to the learning in different mini-batches.

and EER decreases. Therefore, boosting sampling with replacement strategy is a preferable choice in ASV and other SGD methods, especially no effective augmentation strategy is available.

7. CONCLUSION AND FUTURE WORKS

In this paper, an effective mechanism is proposed to ensure learning efficiency against class-imbalance and improve local discriminative modeling on the representation space of i-vector, resulting in competitive EER results. Specifically, we develop the centroid-aware balanced boosting sampling to gather class-balance hard mini-batches to pave the way for efficient optimization. Next, we choose NCA and MNL for representation modeling to absorb the meaningful information in the mini-batches. The combination results in AdaNCA and LMNL. AdaNCA (EER=4.03% on SRE2008, EER=2.05% on SRE2010) and LMNL (EER=3.84% on SRE2008, EER=1.81% on SRE2010) both enjoy strong competitive performances and keep the computational workload low at the same time. Several typical metric learning methods are also compared to ensure the conclusion, and comparison experiments offer useful knowledge for further development in ASV task. Besides, we investigate the behavior of boosting sampling strategy further by establishing its connections with hinge loss and data augmentation. The connections can help us design improved optimization process with boosting sampling.

The results in this work can be improved with deep learning techniques further, and call for additional efforts in designing better approaches to exploit the intrinsic data structure — e.g., more effective hard example mining or objectives using centroid priori — in ASV task.

ACKNOWLEDGEMENT

This research is supported in part by National Natural Science Foundation of China (NSFC) (Grant No.: 61573348, 61620106003, 61672520), and in part by the Institute of Automation Chinese Academy of Sciences (CASIA)-Tencent Youtu Joint Research Project. The authors are grateful to Fan Tang at CASIA, Fuzhang Wu, Xingming Jin, Peng Li, and others at Youtu Lab, Tencent Inc. for meaningful discussions and help. Besides, the authors thank all reviewers for their valuable comments on improving the quality of this paper.

References

[1] M. Senoussaoui, P. Kenny, N. Dehak, P. Dumouchel, An i-vector extractor suitable for speaker recognition with both microphone and telephone speech, in: Odyssey, 2010, p. 6.

[2] L. v. d. Maaten, G. Hinton, Visualizing data using t-SNE, Journal of Machine Learning Research 9 (Nov) (2008) 2579–2605.

[3] J. H. Hansen, Analysis and compensation of speech under stress and noise for environmental robustness in speech recognition, Speech communication 20 (1) (1996) 151–173.

[4] F. Kelly, A. Drygajlo, N. Harte, Speaker verification in score-ageing-quality classification space, Computer Speech & Language 27 (5) (2013) 1068–1084.

[5] N. Dehak, Z. N. Karam, D. A. Reynolds, R. Dehak, W. M. Campbell, J. R. Glass, A channel-blind system for speaker verification, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011, pp. 4536–4539.

[6] G. Liu, Y. Lei, J. H. L. Hansen, Robust feature front-end for speaker identification, in: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012, pp. 4233–4236.

[7] S. J. D. Prince, J. H. Elder, Probabilistic linear discriminant analysis for inferences about identity, in: IEEE 11th International Conference on Computer Vision, 2007, pp. 1–8.

[8] P. Kenny, Bayesian speaker verification with heavy-tailed priors, in: Odyssey, 2010, p. 14.

[9] A. O. Hatch, S. S. Kajarekar, A. Stolcke, Within-class covariance normalization for svm-based speaker recognition, in: INTERSPEECH 2006 - ICSLP, 2006.

[10] F. Richardson, D. Reynolds, N. Dehak, Deep neural network approaches to speaker and language recognition, IEEE Signal Processing Letters 22 (10) (2015) 1671–1675.

[11] M. Senoussaoui, P. Kenny, T. Stafylakis, P. Dumouchel, A study of the cosine distance-based mean shift for telephone speech diarization, IEEE/ACM Transactions on Audio, Speech, and Language Processing 22 (1) (2014) 217–227.

[12] J. Goldberger, G. E. Hinton, S. T. Roweis, R. Salakhutdinov, Neighbourhood components analysis, in: Advances in Neural Information Processing Systems, 2004, pp. 513–520.

[13] O. Rippel, M. Paluri, P. Dollar, L. Bourdev, Metric learning with adaptive density discrimination, in: International Conference on Learning Representations (ICLR), 2015.
URL `http://adsabs.harvard.edu/abs/2015arXiv151105939R`

[14] P. Kenny, G. Boulianne, P. Dumouchel, Eigenvoice modeling with sparse training data, IEEE Transactions on Speech and Audio Processing 13 (3) (2005) 345–354.

[15] G. Liu, J. H. Hansen, An investigation into back-end advancements for speaker recognition in multi-session and noisy enrollment scenarios, IEEE/ACM Transactions on Audio, Speech, and Language Processing 22 (12) (2014) 1978–1992.

[16] N. Dehak, P. A. Torres-Carrasquillo, D. A. Reynolds, R. Dehak, Language recognition via i-vectors and dimensionality reduction, in: Interspeech, 2011, pp. 857–860.

[17] M. H. Bahari, M. Mclaren, H. Van Hamme, D. A. Van Leeuwen, Speaker age estimation using i-vectors, Engineering Applications of Artificial Intelligence 34 (2014) 99–108.

[18] S. O. Sadjadi, S. Ganapathy, J. Pelecanos, The ibm 2016 speaker recognition system, in: Odyssey 2016: The Speaker and Language Recognition Workshop, pp. 174–180.

[19] S. Ioffe, Probabilistic linear discriminant analysis, in: European Conference on Computer Vision (ECCV), Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 531–542.

[20] N. Lack, Non-parametric discriminant analysis, in: Medizinische Informationsverarbeitung und Epidemiologie im Dienste der Gesundheit, Springer, 1988, pp. 320–322.

[21] Z. Liang, Y. Li, S. Xia, Adaptive weighted learning for linear regression problems via kullback-leibler divergence, Pattern Recognition 46 (4) (2013) 1209–1219.

[22] W. Yang, K. Wang, W. Zuo, Fast neighborhood component analysis, Neurocomputing 83 (2012) 31–37.

[23] K. Q. Weinberger, L. K. Saul, Distance metric learning for large margin nearest neighbor classification, Journal of Machine Learning Research 10 (Feb) (2009) 207–244.

[24] L. van der Maaten, K. Weinberger, Stochastic triplet embedding, in: IEEE International Workshop on Machine Learning for Signal Processing, 2012, pp. 1–6.

[25] D. Chen, X. Cao, L. Wang, F. Wen, J. Sun, Bayesian face revisited: A joint formulation, in: European Conference on Computer Vision, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 566–579.

[26] H. Oh Song, Y. Xiang, S. Jegelka, S. Savarese, Deep metric learning via lifted structured feature embedding, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4004–4012.

[27] M. A. Bautista, A. Sanakoyeu, E. Tikhoncheva, B. Ommer, CliqueCNN: Deep unsupervised exemplar learning, in: Advances In Neural Information Processing Systems, 2016, pp. 3846–3854.

[28] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, Z. Zhu, Deep speaker: an end-to-end neural speaker embedding system, arXiv preprint arXiv:1705.02304.
URL `https://arxiv.org/abs/1705.02304`

[29] L. Bottou, Large-scale machine learning with stochastic gradient descent, in: 19th International Conference on Computational Statistics, Physica-Verlag HD, Heidelberg, 2010, pp. 177–186.

[30] A. Shrivastava, A. Gupta, R. Girshick, Training region-based object detectors with online hard example mining, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 761–769.

[31] W. Yang, L. Jin, D. Tao, Z. Xie, Z. Feng, DropSample: A new training method to enhance deep convolutional neural networks for large-scale unconstrained handwritten chinese character recognition, Pattern Recognition 58 (2015) 190–203.

[32] O. Canévet, F. Fleuret, Large scale hard sample mining with monte carlo tree search, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 5128–5137.

[33] C. D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, New York, NY, USA, 2008.

[34] G. Montavon, G. Orr, K. Muller, Neural networks : tricks of the trade, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[35] Z. Allen-Zhu, Y. Yuan, K. Sridharan, Exploiting the structure: Stochastic gradient methods using raw clusters, arXiv preprint arXiv:1602.02151, preliminary version appeared in NIPS 2016.

[36] L. Shen, Z. Lin, Q. Huang, Relay backpropagation for effective learning of deep convolutional neural networks, in: European Conference on Computer Vision (ECCV), Springer Berlin Heidelberg, Berlin, Heidelberg, 2016, pp. 467–482.

[37] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006, pp. 1735–1742.

[38] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, et al., The kaldi speech recognition toolkit, in: IEEE 2011 workshop on automatic speech recognition and understanding, no. EPFL-CONF-192584, IEEE Signal Processing Society, 2011.

[39] J. V. Davis, B. Kulis, P. Jain, S. Sra, I. S. Dhillon, Information theoretic metric learning, in: Proceedings of the 24th International Conference on Machine Learning, ACM, 2007, pp. 209–216.

[40] C. Huang, C. C. Loy, X. Tang, Local similarity-aware deep feature embedding, in: Advances in Neural Information Processing Systems, 2016, pp. 1262–1270.