Speaker Clustering Using Dominant Sets

Feliks Hibraj*
Ca' Foscari University
Venice, Italy
feliks.hibraj@gmail.com

Sebastiano Vascon*
Ca' Foscari University
Venice, Italy
sebastiano.vascon@unive.it

Thilo Stadelmann
ZHAW Datalab
Winterthur, Switzerland
stdm@zhaw.ch

Marcello Pelillo
Ca' Foscari University
Venice, Italy
pelillo@unive.it

***Abstract*—Speaker clustering is the task of forming speaker-specific groups based on a set of utterances. In this paper, we address this task by using Dominant Sets (DS). DS is a graph-based clustering algorithm with interesting properties that fits well to our problem and has never been applied before to speaker clustering. We report on a comprehensive set of experiments on the TIMIT dataset against standard clustering techniques and specific speaker clustering methods. Moreover, we compare performances under different features by using ones learned via deep neural network directly on TIMIT and other ones extracted from a pre-trained VGGVox net. To asses the stability, we perform a sensitivity analysis on the free parameters of our method, showing that performance is stable under parameter changes. The extensive experimentation carried out confirms the validity of the proposed method, reporting state-of-the-art results under three different standard metrics. We also report reference baseline results for speaker clustering on the entire TIMIT dataset for the first time.**

I. Introduction

Speaker clustering (SC) is the task of identifying the unique speakers in a set of audio recordings (each belonging to exactly one speaker) without knowing who and how many speakers are present altogether [1]. Other tasks related to speaker recognition and SC are the following:

- *Speaker verification* (SV): A binary decision task in which the goal is to decide if a recording belongs to a certain person or not.
- *Speaker identification* (SI): A multiclass classification task in which to decide to whom out of n speakers a certain recording belongs.

SC is also referred to as *speaker diarization* when a single (usually long) recording involves multiple speakers and thus needs to be automatically segmented prior to clustering. Since SC is a completely unsupervised problem (the number of speakers and segments per speaker is unknown), it is straightforward to note that it is considered of higher complexity with respect to both SV and SI. The complexity of SC is comparable to the problem of image segmentation in computer vision, in which the number of regions to be found is typically unknown.

The SC problem is of importance in the domain of audio analysis due to many possible applications, for example in lecture/meeting recording summarization [2], as a pre-processing step in automatic speech recognition, or as part of an information retrieval system for audio archives [3]. Furthermore, SC represents a building block for speaker diarization [4].

The SC problem has been widely studied [5], [6]. A typical pipeline is based on three main steps: *i.a)* acoustic feature extraction from audio samples, *i.b)* voice feature aggregation from the lower-level acoustic features by means of a speaker modeling stage, and *ii)* a clustering technique on top of this feature-based representation.

The voice features after phase *i)* have been traditionally created based on Mel Frequency Cepstral Coefficient (MFCC) acoustic features modeled by a Gaussian Mixture Model (GMM) [7], or i-vectors [8], [9]. More recently, with the rise of deep learning, the community is moving towards learned features instead of hand-crafted ones, as surveyed by Richardson et al. [10]. Recent examples of deep-feature representations for SI, SV, and SC problems come for example from Lukic et al. [11], after Convolutional neural networks (CNN) have been introduced in the speech processing field by LeCun et al. already in the nineties [12]. McLaren et al. used a CNN for speaker recognition in order to improve robustness to noisy speech [13]. Chen et al. used a novel deep neural architecture to learn speaker specific characteristics directly from MFCC features [14]. Yella et al. exploited the capabilities of an artificial neural network of 3 layers to extract features directly from a hidden layer, which are used for speaker clustering [15].

However advanced phase *i)* has become during the last years, the clustering phase *ii)* still relies on traditional methodologies. For example, Khoury et al. demonstrated good results for speaker clustering using a hierarchical clustering algorithm [16], while Kenny et al. report hierarchical clustering to be unsuitable for the speaker clustering stage in a speaker diarization system [17]. In [18] they performed clustering with K-means on dimensionality-reduced i-vectors which showed to work better than spectral clustering as noted in [4].

In this paper, we therefore improve the results of the speaker clustering task by first using state-of-art learned features and then, a different and more robust clustering algorithm, dominant sets (DS) [19]. The motivation driving the choice of dominant sets is the following: *a)* no need for an a-priori number of clusters; *b)* having a notion of compactness to be able to automatically detect clusters composed of noise; *c)* for each cluster the centrality of each element is quantified (centroids emerge naturally in this context); and *d)* extensive

* = Equal contribution

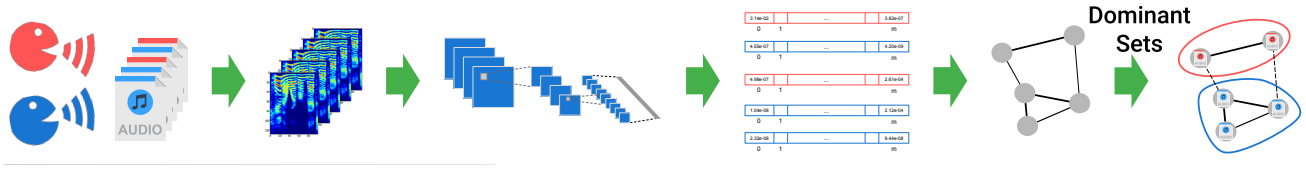

Fig. 1. Pipeline of the overall sequence of elaborations: voices → spectrograms → CNN → feature vectors → graph → Dominant Set → clusters.

experimentations and the underlying theory prove a high robustness to noise [19]. All the aforementioned properties perfectly fit the SC problem.

The contribution of this paper is three-fold: first, we apply the dominant set method for the first time in the SC domain, outperforming the previous state of the art; second, it is the first time that the full TIMIT dataset [20] is used for SC problems, making this paper a reference baseline in this context and on this dataset; third, we use for the first time a pre-trained VGGVox[1] network to extract features for the TIMIT dataset, obtaining good results and demonstrating the capability of this embedding.

The remainder of this paper is organized as follow: in Sec II the proposed method is explained in detail (with Sec II-A having explanations for the different feature extraction methods, and Sec II-B having an introduction to the theoretical foundations of DS). In Sec III the experiments that have been carried out are explained and in Sec IV we discuss the results before drawing conclusions in Sec V together with future perspectives.

II. Speaker Clustering with Dominant Sets

Our proposed approach, called SCDS is based on the two-phase schema (see Fig.1): the first part in which features are extracted from each utterance and the second one in which from this feature-based representation the dominant sets are extracted. In this section, the specific parts are explained.

A. Features extraction

We use two different feature extraction methods in this work that we call CNN-T (derived from embeddings based on the TIMIT dataset), and CNN-V (based on a model trained on VoxCeleb [21]):

1) CNN-T features: Features are extracted from the CNN[2] described in detail by Lukic et al. [22], specifically from the dense layer L7 therein. The network has been trained on 590 speakers of the TIMIT database that have been fed to the net as spectrograms derived from the corresponding utterances, and yields 1,000-dimensional feature vectors.

2) CNN-V features: Features are extracted from the published VGGVox model trained on the 100,000 utterances of the VoxCeleb dataset [21]. Since the domain of VoxCeleb and TIMIT are similar, we expect to have good performances on the latter. VGGVox is based on the VGG-M convolutional architecture [23] which was previously used for image data, adapted for spectrogram input. We get 1,024-dimensional features from the FC7 layer as in the original publication.

[1]https://github.com/a-nagrani/VGGVox
[2]https://github.com/stdm/ZHAW_deep_voice

B. Dominant Set clustering

Dominant set clustering is a graph-based method that generalizes the problem of finding a maximal clique to edge-weighted graphs. A natural application of this method is for partitioning (clustering) a graph into disjoint sets. In this framework, a dataset is modeled as an undirected edge-weighted graph $G = (V, E, w)$ with no self loops, in which the nodes V are the items of the dataset (represented by feature vectors). The edges $E \subseteq V \times V$ are the pairwise relations between nodes and their weight function $\omega : E \rightarrow \mathbb{R}_{\geq 0}$ calculates pairwise similarities. The $n \times n$ symmetric adjacency matrix $A = (a_{ij})$ is employed to summarize G:

$$a_{ij} = \begin{cases} w(i,j) & \text{if } (i,j) \in E \\ 0 & \text{otherwise.} \end{cases}$$

Typically with every clustering method two properties shall hold: the *intra-cluster* homogeneity is high while *inter-cluster homogeneity* is low. These two properties are important in order to separate and group objects in the best possible way. They are directly reflected in the combinatorial formulation of DS (see [19] for the details). Pavan and Pelillo propose an intriguing connection between clusters, dominant sets and local solutions of the following quadratic problem [19]:

$$\begin{aligned} &\text{maximize} && \mathbf{x}^T A \mathbf{x} \\ &\text{subject to} && \mathbf{x} \in \triangle^n \end{aligned} \tag{1}$$

where A is the similarity matrix of the graph and $\mathbf{x}$ is the so-called *characteristic vector* which lies in the n-dimensional simplex $\triangle^n$, that is, $(\sum_i \mathbf{x}_i = 1, \forall i \; x_i \geq 0)$. The components of vector $\mathbf{x}$ represent the likelihood of each element to belong to the cluster, the higher the score the higher the chance of being part of it. If $\mathbf{x}$ is a strict local solution of (1) then its support $\sigma(\mathbf{x}) = \{i \in V | x_i > 0\}$ is a dominant set.

In order to extract a DS, a local solution of (1) must be found. A method to solve this problem is to use a result from evolutionary game theory [24] known as *replicator dynamic* (RD) (see Eq. 2).

$$x_i(t+1) = x_i(t) \frac{(A\mathbf{x}(t))_i}{\mathbf{x}(t)^T A \mathbf{x}(t)} \tag{2}$$

RD is a dynamical system that operates a selection process over the components of the vector $\mathbf{x}$. At convergence of Eq. 1 ($|| \, (\mathbf{x}(t) - \mathbf{x}(t+1)) \, ||_2 \leq \epsilon$), certain components will emerge ($x_i > 0$) while others will get extinct ($x_i = 0$). In practical cases, if these last components of x are not exactly equal to zero then a thresholding ($x_i > \theta$) is performed. The convergence of the process is guaranteed if the matrix A is non-negative and symmetric. The dynamical system starts at

the barycenter of the simplex and its components are updated using Eq. 2.

Deciding upon a cutoff threshold θ is not obvious. Instead of using a predefined value, we prefer to employ the approach proposed by Vascon et al. [25], [26]. The parameter is computed based on the following idea: it decides the minimum degree of participation of an element to a cluster and is relative to the participation of the centroid. The support for each dominant set becomes $\sigma(\mathbf{x}) = \{i \in V | x_i > \theta * max(x)\}$ with $\theta \in [0, 1)$ (see Sec. IV-E for sensitivity analysis on the parameters).

At each iteration a dominant set is extracted and its subsets of nodes are removed from the graph (this is called *peeling-off* strategy). The process iterates on the remaining nodes until all are assigned to a cluster.

C. Similarity measure

To compute weights on edges of graph G we use the *cosine* distance to construct a similarity function. The cosine distance has been chosen because it showed good performance on SC tasks [16], [18], [21]. Given two utterances and their m-dimensional feature vectors f_i and f_j, we apply the following function:

$$\omega(f_i, f_j) = exp\left\{-\frac{d(f_i, f_j)}{\sigma}\right\} \tag{3}$$

where d is the cosine distance between given features, and σ is the similarity scaling parameter.

Setting the parameter σ is often problematic and typically requires a grid search over a range of plausible values or a cross-validation. In this work, we decided to use a principle heuristic from spectral clustering [27] which proved to work well also in other works [28], [29]. Based on [27] and [29] we tested a local scaling parameter σ_i for each utterance to be clustered. This means that in (3) our parameter $\sigma = \sigma_i \sigma_j$ depends on local neighborhoods of given features f_i and f_j and it is determined as follows:

$$\sigma_i = \frac{1}{|N_i|} \sum_{k \in N_i} d(f_i, f_k) \tag{4}$$

where N_i represents the nearest neighborhood of element i. In our experiments we used $|N_i| = 7$ as in [29].

D. Cluster labeling

Once all dominant sets are extracted, the final step is to label each partition such that each speaker is in one-to-one correspondence with a cluster. The labels of the data are then used to perform the assignment. We tested two approaches for cluster labeling:

1) Max: a prototype selection method which assigns cluster labels using the class of the element with *maximum* participation in the characteristic vector [25]. Labels are unique, and in case 2 different clusters share their labels, the latter one is considered completely mistaken, increasing error in the evaluation.

TABLE I
DATASETS USED IN THIS PAPER.

	Acronym	#POIs	#Utt/POI	Utterances
TIMIT Small [31]	TimitS	40	2	80
TIMIT Full [20]	TimitF	630	2	1260

2) Hungarian: with this approach, each cluster is labeled using the Munkres (aka Hungarian) method [30]. The cost $c_{i,j}$ of assigning a cluster i to a particular label j is computed as the number of elements of class j in the cluster i. Since the method minimizes the total cost of assignments, the value of $c_{i,j}$ is changed to $\hat{c}_{i,j} = \max(c) - c_{i,j}$. This turns the minimization problem to a maximization one, where $\max(c)$ is the maximum cost over all the assignments.

III. EXPERIMENTS

A. Datasets & data preparation

We evaluate our method on the TIMIT dataset, presented as *TIMIT Small* and *TIMIT Full* (see Table III-A). The dataset is composed of 6,300 phrases (10 phrases per speaker), spoken by 438 males (70%) and 192 females (30%). Speakers coming from 8 different regions and having different dialects. The phrases of each speaker have been divided into 2 parts in accordance with previous research [11], [22], [31]. In our experimentation we used the same 40 speakers dataset as reported by these earlier attempts (here called TIMIT Small), and the full TIMIT set composed by 630 speakers. Note that *TIMIT Small* is disjoint with the training set of CNN-T. This dataset is suited to our work because we are not dealing with noise, segmentation or similar diarization problems.

B. Comparison to other methods

The proposed method has been compared with the state of the art [11], [22], [31] and with other clustering techniques like spectral clustering (SP), k-means (KM) and hierarchical clustering (HC). Given the fact that our proposed method is completely unsupervised (in particular, there is no knowledge a-priori of the number of clusters), we tested some heuristics to estimate k also for the aforementioned algorithms. Specifically, the Eigengap heuristic [32] and the number of clusters found by our method are used. Moreover, we chose affinity propagation (AP) [33] and HDBSCAN [34] because they do not require an a-priori k. In order to fairly compare our method, we tested them with the best settings. Specifically for HC and KM, cosine distance was the best choice, while for SP we used RBF kernel with γ parameter found through an extensive grid search. The cut on HC has been set such that the error is minimized as in [22]. For AP we used the same similarity measure of SCDS while for HDBSCAN the Euclidean distance and minimum number of points per cluster equal to 2 were used.

C. Evaluation criteria

To evaluate the clustering quality we used three distinct metrics: the *misclassification rate* (MR) [35], the *adjusted RAND index* (ARI) [36] and the *average cluster purity* (ACP)

[37]. The usage of different metrics is important because each of them gives a different perspective on results: MR quantifies how many labels of speakers are inferred correctly from clusters while ARI and ACP are measures of grouping/separation performance on utterances.

Formally, given a one-to-one mapping between clusters and labels (see Sec II-D), MR is defined as $MR = \frac{1}{N}\sum_{j=1}^{N_s} e_j$ where N is the total number of audio segments to cluster, N_s the number of speakers, and e_j the number of segments of speaker j classified incorrectly. *Cluster purity* is a measure to determine how pure clusters are. If a cluster is composed of utterances belonging to the same speaker, then it is completely pure, otherwise (i.e., other speakers are in that cluster, too) purity decreases. Formally, average cluster purity is defined as:

$$acp = \frac{1}{N}\sum_{i=1}^{N_c} p_{i.} \cdot n_{i.} \text{ , where } p_{i.} = \sum_{j=1}^{N_s} n_{ij}^2 / n_{i.}^2$$

N_c is the number of clusters, n_{ij} utterances in cluster i spoken by speaker j and $n_{i.}$ is the size of cluster i. The ARI finally is the normalized version of RAND index [38], with maximum value 1 for perfectly assigned clusters with respect to the expected ones.

D. Experimental setup

Our proposed method is evaluated in experiments composed as follows: given a set of audio utterances, features are extracted following one of the methods in Sec II-A and the affinity matrix is computed as in Sec II-C. Subsequently, the DS are found on top of this graph-based representation. Labeling is performed on each cluster following the methodology proposed in Sec II-D. The goodness of clusters are then evaluated using the metrics in Sec III-C. The summarized results are reported in Tables II and III and discussed in the next section. The hyper parameters for all experiments are set to $\theta = 0.1$ and $\varepsilon = 1e-6$.

IV. Results

In this section, the results of a series of analyses are reported, followed by an overall discussion.

A. Initialization of k in the competitors

DS does not need an a-priori number of clusters, while the supervised competitors do. In order to make a fair comparison with standard approaches (HC, KM and SP), we used as k: the correct number of clusters to be found (symbol $\diamondsuit$ in tables II and III), the number of clusters found by DS (symbol k in the tables) and the number of clusters estimated with *Eigengap* (symbol # in tables).

Experimental results show that even when the correct number of clusters is provided, SCDS still achieves more desirable results (see tables). This means that not only our method is able to recover a number of clusters close to the right one, but also that it is able to extract much more correct partitions. And when the number of clusters found by DS is given to the other methods, results obtained are plausible, showing that our method is able to grasp a good number of clusters while with standard heuristics the performance drops strongly.

B. Analysis of different feature extraction methods

In the next experiments, we tested the two CNN-based features, CNN-T and CNN-V. Both provide good features in term of capacity to discriminate speakers. With the CNN-T features, the performance of our method saturates on TIMIT Small (see last rows of Table II) and reaches almost perfect performances on TIMIT Full (see last rows of Table III). This is mainly explained by the fact that the network used to extract the CNN-T embeddings has been trained in using the remaining 590 of the 630 TMIT speakers [22], thus biasedly performing well on the entire dataset.

Surprisingly, features obtained from VGGVox are so generic that they allow almost the same performances for SCDS. This approach is also beneficial for competitors, and in fact all of them have better performances in term of MR/ARI/ACP with CNN-V features rather than CNN-T ones (except for KM).

C. Cluster labeling

We tested two methods for labeling clusters for our approach (see *Max* and *Hungarian* in Sec II-D), while for all the other competitors we used only the *Hungarian* algorithm since *Max* is a peculiarity of DS. Under all conditions and datasets both labeling methods perform the same (see last rows of results where *Max* = SCDS$\star$, *Hungarian* = SCDS). Labeling with *Max* method comes for free directly from DS theory, while applying the Hungarian method has its computational cost.

D. Metrics comparison

The three metrics (MR, ARI and ACP) are important to be analyzed in conjunction because they capture different aspects of the result. Having the lowest MR in the final results in both datasets emphasize the fact that we are correctly labeling clusters and that the number of miss-classified samples is extremely low. On the other side, reaching the highest value in ARI shows that our method obtains a good partitioning of the data with respect to the expected clusters. Furthermore, having the higher ACP confirms that clusters extracted with SCDS are mainly composed by utterances from the same speaker.

The proposed method reaches best scores on all metrics simultaneously. Indeed, other methods reach similar performances, in particular on TIMIT Small (like HC, AP), but none of them work as well as our in the most complex experimental setting used, TIMIT Full with VGGVox features (where no knowledge of the actual voices to be clustered could possibly enter the features and thus the clustering system).

E. Sensitivity analysis

Finally, we report the results of a sensitivity analysis on the two free-parameters of our method under two metrics (see Fig 2 and 3), the precision ε of Replicator Dynamics (see Eq. 2) and the relative cut-off θ (see Sec. II-B). The analysis has been carried out on TIMIT Full with VGGVox features because under this setting a certain amount of variability on results is observed, which made this analysis interesting. The search space for the parameters is as follows: $\theta \in [0.0,\ 0.9995]$ and $\varepsilon \in [1e-11,\ 1e-2]$. The choice has been made on these

TABLE II
CLUSTERING RESULTS ON THE TIMIT SMALL DATASET.

SMALL TIMIT	CNN-T Features MR ↓	ARI ↑	ACP ↑	CNN-V Features MR ↓	ARI ↑	ACP ↑
HC ♢	0.0250	0.9259	0.9667	0.0000	1.0000	1.0000
HC [22]	0.0500	-	-	-	-	-
Adadelta 20k [11]	0.0500	-	-	-	-	-
Adadelta 30k [11]	0.0500	-	-	-	-	-
ν-SVM [31]	0.0600	-	-	-	-	-
GMM/MFCC [31]	0.1300	-	-	-	-	-
SP ♢	0.0750	0.8422	0.9500	0.0000	1.0000	1.0000
KM ♢	0.0250	0.9259	0.9667	0.0375	0.9390	0.9750
HC k	0.0250	0.9259	0.9667	0.0000	1.0000	1.0000
SP k	0.0750	0.8422	0.9500	0.0000	1.0000	1.0000
KM k	0.0250	0.9259	0.9667	0.0375	0.9390	0.9750
HC #	0.4500	0.4234	0.5500	0.6750	0.2466	0.3250
SP #	0.4500	0.0827	0.5500	0.6750	0.1751	0.3038
KM #	0.4500	0.3543	0.5267	0.6750	0.1746	0.3193
AP	0.0500	0.8951	0.9416	0.0000	1.0000	1.0000
HDBS	0.1000	0.8056	0.8833	0.0750	0.8422	0.9083
SCDS	**0.0000**	**1.0000**	**1.0000**	**0.0000**	**1.0000**	**1.0000**
SCDS⋆	**0.0000**	**1.0000**	**1.0000**	**0.0000**	**1.0000**	**1.0000**

TABLE III
CLUSTERING RESULTS ON THE TIMIT FULL DATASET.

FULL TIMIT	CNN-T Features MR ↓	ARI ↑	ACP ↑	CNN-V Features MR ↓	ARI ↑	ACP ↑
HC ♢	0.0770	0.8341	0.9283	0.0571	0.8809	0.9484
SP ♢	0.2294	0.0432	0.8355	0.0675	0.5721	0.9488
KM ♢	0.1071	0.7752	0.9071	0.1286	0.6982	0.8730
HC k	0.0762	0.8343	0.9280	0.0706	0.8502	0.9295
SP k	0.2341	0.0421	0.8332	0.0635	0.4386	0.9427
KM k	0.1079	0.7682	0.9007	0.1429	0.6646	0.8485
HC #	0.9921	0.0050	0.0079	0.9984	0.0000	0.0016
SP #	0.9921	0.0003	0.0075	0.9984	0.0000	0.0016
KM #	0.9921	0.0052	0.0076	0.9984	0.0000	0.0016
AP	0.0753	0.8330	0.9030	0.1396	0.7127	0.8222
HDBS	0.1825	0.6214	0.7825	0.3000	0.4112	0.6527
SCDS	**0.0048**	**0.9897**	**0.9947**	**0.0349**	**0.9167**	**0.9578**
SCDS⋆	**0.0048**	**0.9897**	**0.9947**	**0.0349**	**0.9167**	**0.9578**
SCDSbest	**0.0032**	**0.9929**	**0.9966**	**0.0024**	**0.9944**	**0.9974**

extremal points for the following reasons: a low value, e.g. $\theta = 0.0005$, means that a point belongs to a cluster if and only if its level of participation in the cluster with respect to the centroid is at least $\theta\times$centrality of the centroid. Instead, $\theta = 0.9995$ means that the centroid and the sample must be almost exactly the same. In the first case we are creating clusters which span widely in terms of similarities of its elements, while in the latter case we create clusters composed by very similar elements. Regarding the parameter ε, when it is set to $1e-11$, it requires that two successive steps in Eq. 2 are very close to each other while in the case $1e-2$ we allow for a coarse equilibrium point.

Changes in both variables showed that the area in which the performances are stable is large (see the blue area in Fig 2 and yellow area in Fig 3). Only when extremal values of parameters are used the performances drops. The best parameter choice (CNN-T: $\theta = 0.15$, $\varepsilon = 1e-6$; CNN-V: $\theta = 0.67$, $\varepsilon = 1e-7$) is shown in Table III as SCDSbest.

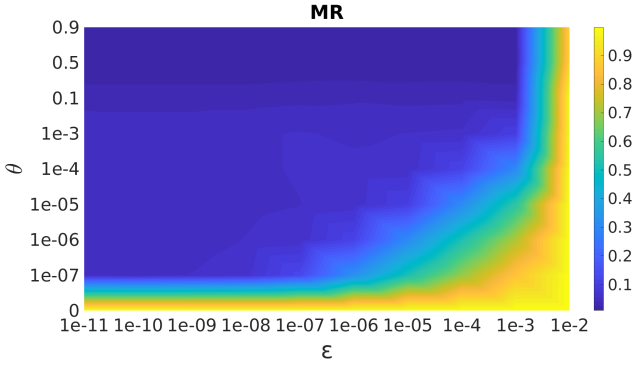

Fig. 2. Sensitivity of ε and θ with respect to the MR measure.

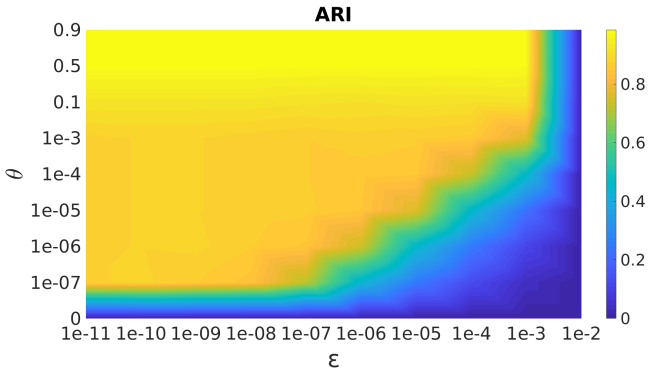

Fig. 3. Sensitivity of ε and θ with respect to the ARI measure.

F. Overall discussion

From a global perspective we can say that the proposed SCDS method performs better than the alternatives on the used datasets, outperforming the state-of-the-art and showing a more adaptive response also with a pre-trained model on a different dataset. In particular, this is evident in TIMIT Full, where better performances than competitors are achieved even when they are given the right number of clusters to be found. It is worth to note that our clustering method has only two parameters to set, which are both very insensitive to variation as shown in the sensitivity analysis.

Interesting to note, an analysis of misclassified speakers shows that if a speaker is wrongly clustered by DS, it is also wrongly clustered by all other methods. This gives rise to the assumption that in these cases the extracted features may be the reason for the error rather than the clustering approach used.

V. CONCLUSIONS

In this paper, we have proposed a novel pipeline for speaker clustering. The proposed method is based on the dominant set clustering algorithm which has been applied to this domain for the first time. It outperforms the previous state of the art and other clustering techniques.

We proposed a method which is almost parameter-less – the two free parameters do not affect too much the results, testifying to its stability. Moreover, we successfully used

a pre-trained CNN model on a different dataset and report reasonable speaker clustering performance on the TIMIT Full dataset for the first time (after the 99.84% MR reported by Stadelmann and Freisleben using a classical pipeline [31]). Now that we reached a good starting point with noise-free utterances we can start considering more complex datasets with their relatively more challenging tasks (noise, segmentation, cross-talk etc.). Future work also includes improving the features using the siamese network proposed by Nagrani et al. [21] to extract similarities directly.

Code available at https://github.com/feliksh/SCDS

ACKNOWLEDGMENT

The authors would like to thank Y. Lukic for providing the MLSP features and J. Salamone for contributing to first results.

REFERENCES

[1] H. Beigi, *Fundamentals of speaker recognition*. Springer Science & Business Media, 2011.

[2] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 356–370, 2012.

[3] J. Ajmera and C. Wooters, "A robust speaker clustering algorithm," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2003.

[4] S. Shum, N. Dehak, and J. Glass, "On the use of spectral and iterative methods for speaker diarization," in *Annual Conference of the International Speech Communication Association*. ISCA, 2012, pp. 482–485.

[5] H. Jin, F. Kubala, and R. Schwartz, "Automatic speaker clustering," in *Proceedings of the DARPA speech recognition workshop*, 1997, pp. 108–111.

[6] S. O. Sadjadi, T. Kheyrkhah, A. Tong, C. Greenberg, E. S. Reynolds, L. Mason, and J. Hernandez-Cordero, "The 2016 nist speaker recognition evaluation," *Interspeech*, pp. 1353–1357, 2017.

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

[8] N. Dehak, R. Dehak, P. Kenny, N. Brümmer, P. Ouellet, and P. Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification," in *Annual Conference of the International Speech Communication Association*, 2009.

[9] H.-S. Lee, Y. Tsao, H.-M. Wang, and S.-K. Jeng, "Clustering-based i-vector formulation for speaker recognition," in *Annual Conference of the International Speech Communication Association*, 2014.

[10] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, 2015.

[11] Y. Lukic, C. Vogt, O. Durr, and T. Stadelmann, "Learning embeddings for speaker clustering based on voice equality," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2017, pp. 1–6.

[12] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The Handbook of Brain Theory and Neural Networks*, vol. 3361, no. 10, p. 1995, 1995.

[13] M. McLaren, Y. Lei, N. Scheffer, and L. Ferrer, "Application of convolutional neural networks to speaker recognition in noisy conditions," in *Annual Conference of the International Speech Communication Association*, 2014.

[14] K. Chen and A. Salman, "Learning speaker-specific characteristics with a deep neural architecture," *IEEE Transactions on Neural Networks*, vol. 22, no. 11, pp. 1744–1756, 2011.

[15] S. H. Yella, A. Stolcke, and M. Slaney, "Artificial neural network features for speaker diarization," in *IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2014, pp. 402–406.

[16] E. Khoury, L. El Shafey, M. Ferras, and S. Marcel, "Hierarchical speaker clustering methods for the nist i-vector challenge," in *Odyssey: The Speaker and Language Recognition Workshop.*, 2014.

[17] P. Kenny, D. Reynolds, and F. Castaldo, "Diarization of telephone conversations using factor analysis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 6, pp. 1059–1070, 2010.

[18] S. Shum, N. Dehak, E. Chuangsuwanich, D. Reynolds, and J. Glass, "Exploiting intra-conversation variability for speaker diarization," in *Annual Conference of the International Speech Communication Association*, 2011, pp. 945–948.

[19] M. Pavan and M. Pelillo, "Dominant sets and pairwise clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 167–172, 2007.

[20] W. M. Fisher, G. R. Doddington, and K. M. Goudie-Marshall, "The DARPA Speech Recognition Research Database: Specifications and Status," in *Proceedings of DARPA Workshop on Speech Recognition*, 1986, pp. 93–99.

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

[22] Y. Lukic, C. Vogt, O. Drr, and T. Stadelmann, "Speaker Identification and Clustering Using Convolutional Neural Networks," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Sept 2016, pp. 1–6.

[23] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *British Machine Vision Conference (BMVC)*, 2014.

[24] J. W. Weibull, *Evolutionary game theory*. MIT press, 1997.

[25] S. Vascon, M. Cristani, M. Pelillo, and V. Murino, "Using dominant sets for k-nn prototype selection," in *International Conference on Image Analysis and Processing (ICIAP)*, A. Petrosino, Ed. Springer Berlin Heidelberg, 2013, pp. 131–140.

[26] L. Dodero, S. Vascon, L. Giancardo, A. Gozzi, D. Sona, and V. Murino, "Automatic white matter fiber clustering using dominant sets," in *International Workshop on Pattern Recognition in Neuroimaging (PRNI)*. IEEE, 2013, pp. 216–219.

[27] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Advances in neural information processing systems (NIPS)*, 2005, pp. 1601–1608.

[28] R. Tripodi, S. Vascon, and M. Pelillo, "Context aware nonnegative matrix factorization clustering," in *IEEE International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 1719–1724.

[29] E. Zemene, L. T. Alemu, and M. Pelillo, "Dominant sets for "constrained" image segmentation," *CoRR*, vol. abs/1707.05309, 2017. [Online]. Available: http://arxiv.org/abs/1707.05309

[30] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics (NRL)*, vol. 2, no. 1-2, pp. 83–97, 1955.

[31] T. Stadelmann and B. Freisleben, "Unfolding speaker clustering potential: a biomimetic approach," in *International Conference on Multimedia*, 2009, pp. 185–194.

[32] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[33] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *science*, vol. 315, no. 5814, pp. 972–976, 2007.

[34] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Advances in Knowledge Discovery and Data Mining*, J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 160–172.

[35] M. Kotti, V. Moschou, and C. Kotropoulos, "Review: Speaker segmentation and clustering," *Signal Process.*, vol. 88, no. 5, pp. 1091–1124, May 2008.

[36] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193–218, 1985.

[37] A. Solomonoff, A. Mielke, M. Schmidt, and H. Gish, "Clustering speakers by their voices," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1998.

[38] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical association*, vol. 66, no. 336, pp. 846–850, 1971.