Unsupervised 3-D Feature Learning for Mild Traumatic Brain Injury

Po-Yu Kao[1(✉)], Eduardo Rojas[1], Jefferson W. Chen[2], Angela Zhang[1], and B.S. Manjunath[1]

[1] University of California, Santa Barbara, Santa Barbara, CA 93106, USA
{poyu_kao,manj}@ece.ucsb.edu
[2] University of California, Irvine, Irvine, CA 92697, USA
jeffwc1@uci.edu

Abstract. We present an unsupervised three-dimensional feature clustering algorithm to gather the mTOP2016 challenge data into 3 groups. We use the brain MR-T1, diffusion tensor fractional anisotropy, and diffusion tensor mean diffusivity images provided by the mTOP2016 competition. A distance-based size constraint method for data clustering is used. The proposed approach achieves 0.267 adjusted rand index and 0.3556 homogeneity score within the 15 labeled subjects, corresponding to 10 correctly classified data items. Based on visual exploration of the data, we believe that a localized analysis of the lesion regions, using the computed tractography data, is a promising direction to pursue.

1 Introduction

This paper addresses the challenge of feature detection and classification of subject data based on brain imaging, as described in the mTOP challenge. The imaging data include the MR-T1 and diffusion weighted images (DWI). While there is extensive work on applying unsupervised learning to clustering 2-D image features [1–3,6], the problems posed by the mTBI data set are particularly challenging since the features of interest are likely very localized. Furthermore, the subject categorization is derived not necessarily from the image data but from other observations, making this problem very distinct from the traditional works in natural image processing.

We propose a fully unsupervised methodology to learn the 3-D features from the data, a 3-D convolutional network to extract the feature representation for each subject, and a distance-based size constraint methodology for data clustering.

2 Unsupervised 3-D Feature Learning

Our proposed workflow includes four stages. The first stage performs data preparation and pre-processing on mTOP 2016 data set. The second stage performs learning 3-D features from brain MR-T1, diffusion tensor fractional anisotropy

A. Crimi et al. (Eds.): BrainLes 2016, LNCS 10154, pp. 282–290, 2016.
DOI: 10.1007/978-3-319-55524-9_26

(DT-FA) and diffusion tensor mean diffusivity (DT-MD) images from 27 subjects of mTOP2016 data set. The third stage performs feature representation for each subject, and the last stage performs group clustering based on these feature representations.

2.1 Data Preparation and Pre-processing

The mTOP data consists of MR-T1, DT-FA and DT-MD images, see Fig. 1. This data set contains 27 subjects belonging to 3 different categories (healthy, patient category 1 or patient category 2) each consisting of 9 subjects. mTBI Patients are categorised into one of two groups based on their long term recovery status following the injury. The imaging data includes for MR-T1 image at $182 \times 218 \times 182$ voxels, with $1\,\text{mm} \times 1\,\text{mm} \times 1\,\text{mm}$ voxel resolution, and the dimension for DT-FA and DT-MD image is $91 \times 109 \times 91$ with $2\,\text{mm} \times 2\,\text{mm} \times 2\,\text{mm}$ voxel resolution.

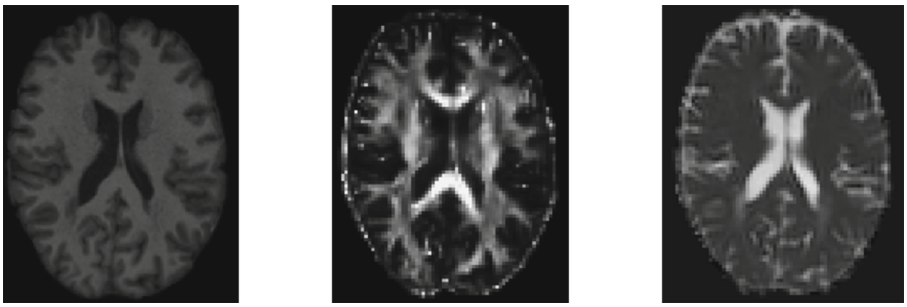

Fig. 1. Left: MR-T1 image, Middle: DT-FA image, Right: DT-MD image

Data preparation for MR-T1 images is shown in Fig. 2. For MR-T1 images, we consider $8 \times 8 \times 8$ voxel volume represented as a 512 dimensional vector of voxel values, $\tilde{x}_{T1}^{(i)} \in \mathbb{R}^{512}$, where i indexes the 3-D patch. The overlap between the volumes in a sliding window is 50%, and those volumes that have more than 75% zero values are discarded. Thus, a large number of data vectors are generated that are the organized as column vectors in a matrix.

Moreover, these vectors are normalized to zero mean and unit standard deviation:

$$x^{(i)} = \frac{\tilde{x}^{(i)} - mean(\tilde{x}^{(i)})}{std(\tilde{x}^{(i)})}$$

where $\tilde{x}^{(i)}$ is a unnormalized column vector and "mean" and "std" are the mean and standard deviation of the element of $\tilde{x}^{(i)}$. Let X_{T1} represent this matrix that includes data from all of the 27 subjects. Similarly, two other matrices X_{FA} and X_{MD} are constructed. However, since the spatial resolution of the data for

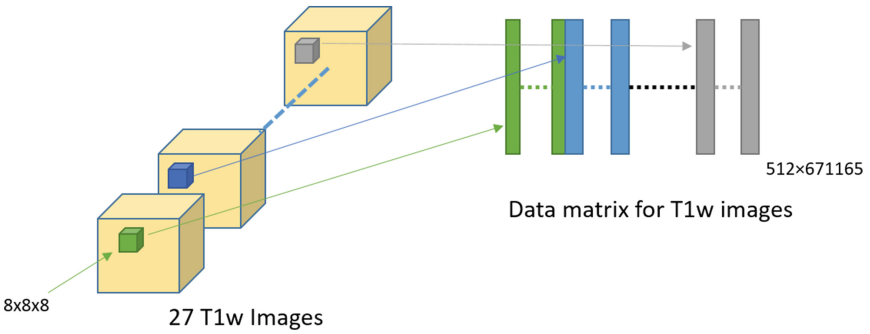

Fig. 2. Data preparation for MR-T1 images

these two cases are different from the MR-T1, we use a $4 \times 4 \times 4$ voxel volume. Therefore, the data vectors all represent a $512\,\mathrm{mm}^3$ spatial volume.

After normalization, we apply the standard *Zero Component Analysis* (ZCA) whitening transform [5] on each of the datasets X_{T1}, X_{FA}, and X_{MD}. This helps minimize the correlation among the components of the column vectors. For contrast-normalized data, we set the whitening parameter ϵ_{zca} to 0.01 for $8 \times 8 \times 8$ voxel patches and 0.1 for $4 \times 4 \times 4$ voxel patches.

2.2 Dictionary Learning via K-means Clustering

The next step is to learn a dictionary for each of the data matrices using the standard K-means clustering. A separate dictionary is learned for each of the three matrices. Let the data matrix be $X \in \mathbb{R}^{N \times M}$ and the corresponding dictionary be $D \in \mathbb{R}^{N \times K}$. Then,

Loop until convergence:

$$c_j^{(i)} = \begin{cases} D^{(j)\top} x^{(i)}, & \text{if } j = \arg\min_l |D^{(l)\top} x^{(i)}| \quad \forall i, j. \\ 0, & \text{otherwise.} \end{cases}$$

$$D := XC^\top + D$$

$$D^{(j)} / ||D^{(j)}||_2 \quad \forall j$$

where $c_j^{(i)}$ is the code vector associated with the input $x^{(i)}$ (i^{th} column of X), and $D^{(j)}$ is the j^{th} column of the dictionary D that is a 3-D feature we learned. In the end, we will learn K 3-D features from a dataset ($D \in \mathbb{R}^{N \times K}$). Note that $C \in \mathbb{R}^{K \times M}$. Let the three corresponding dictionaries be D_{T1}, D_{FA}, and D_{MD}.

2.3 Feature Representation

Feature computation workflow schematic is shown in Fig. 3. Input data includes the three types: brain MR-T1, DT-FA and DT-MD, for each of the subjects.

Each of these datasets is first normalized by subtracting the mean voxel value and dividing by the standard deviation within the brain region. The dictionary code words learned from the K-means clustering above are used as the weights for the first convolutional layer. The stride for MR-T1 is 2 voxels, and for DT-FA and DT-MD is 1 voxel. This is followed by a 3-D max-pooling layer of size $3 \times 3 \times 3$. The final merge layer concatenates the features from the three different pooling layers, thus constructing a single feature vector for each of the subjects. The dimensions of the resulting 3-D feature vector is $1536 \times 25 \times 32 \times 23$.

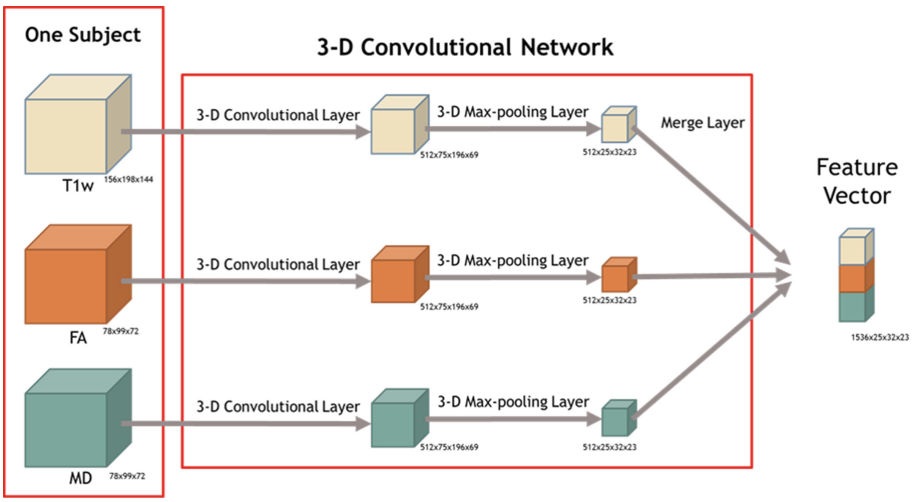

Fig. 3. 3-D Convolutional network for feature extraction

2.4 Group Clustering with Size Constraints

Ideally one would like to train the convolutional network to adjust the weights for discriminating the three different classes. However, given the number of data points, this is currently not feasible. We explored training an SVM with cross-validation but the initial results were not promising. Instead, we now consider this problem as one of unsupervised clustering in the feature space computed by the above hand-tuned convolutional network.

For clustering, we use the standard K-means clustering with distance-based size-constraint, building upon the method described in [8]. However, [8] does not provide a unique solution as it only uses the cluster labels. Instead, we modify the method to account for both labels and distances to the centroid as follows.

Given a dataset of N objects with P centroids (number of clusters), let $Dist$ be the $N \times P$ distance matrix,

$$Dist = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1P} \\ d_{21} & d_{22} & \dots & d_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1} & x_{N2} & \dots & d_{NP} \end{bmatrix} \tag{1}$$

where d_{ip} is the distance between i object and p-th centroid. The objective is to compute a constrained $P \times N$ binary label matrix L,

$$L = \begin{bmatrix} l_{11} & l_{12} & \dots & l_{1N} \\ l_{21} & l_{22} & \dots & l_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ l_{P1} & l_{P2} & \dots & l_{PN} \end{bmatrix} \tag{2}$$

such that

$$\sum_{i=1}^{P} l_{ij} = 1, \quad j = 1, \dots, n, \quad \text{and} \quad \sum_{j=1}^{N} l_{ij} = N_i, \quad i = 1, \dots, p \tag{3}$$

where $l_{ij} = 1$ if the j-th object is assigned to cluster i, and cluster i is constrained to have exactly N_i points. This results in the following problem statement:

$$\text{minimize} \sum_{k=1}^{n} Dist_{(k)} L^{(k)} \tag{4}$$

where $Dist_{(k)}$ is the k^{th} row of $Dist$, and $L^{(k)}$ is the k^{th} column of L. This binary integer linear programming problem can be easily solved by any existing solver. The mTOP2016 data set has 27 subjects which are belonged to three different classes, and each class has nine subjects. Therefore, for this data set, we set $N = 27, P = 3$, and $N_i = 9$ for each class.

3 Experiments and Discussions

Experiments are carried out with the following parameter settings: (i) whether to use whitening (ii) the size of 3-D patches (iii) the size of 3-D max-pooling kernel (iv) the number of 3-D features. We use adjusted rand index (ARI) [4] and homogeneity score (HS) [7] to measure the performance. The adjusted rand index measures the similarity of two assignments (clustered labels vs. ground truth labels), which is invariant to permutations and normalised to chance. Similarity score is between 1.0 and -1.0. Random labelings have a ARI close to 0.0, and 1.0 stands for perfect match. Homogeneity score measures the purity of ground truth labels within cluster. HS is between 1.0 and 0.0. 1.0 stands for perfectly homogeneous labeling.

3.1 Effect of Whitening

In general, the whitening transformation helps improve the accuracy. Figure 4 shows some example dictionary elements learnt from K-means clustering and contrasts that to the original data. We observe that the ZCA transformation results in a sharper dictionary kernel. Figure 5 shows the clustering performance with and without whitening. The x-axis here shows the size of the dictionary. With the ZCA transform the results improve considerably as evidenced by the corresponding ARI and HS scores. This experiment used a stride size of 4 voxel and $8 \times 8 \times 8$ patch size for MR-T1 images, a stride size of 2 voxel and $4 \times 4 \times 4$ patch size for DT-FA and DT-MA image, and a $25 \times 32 \times 23$ kernel in the max-pooling layers.

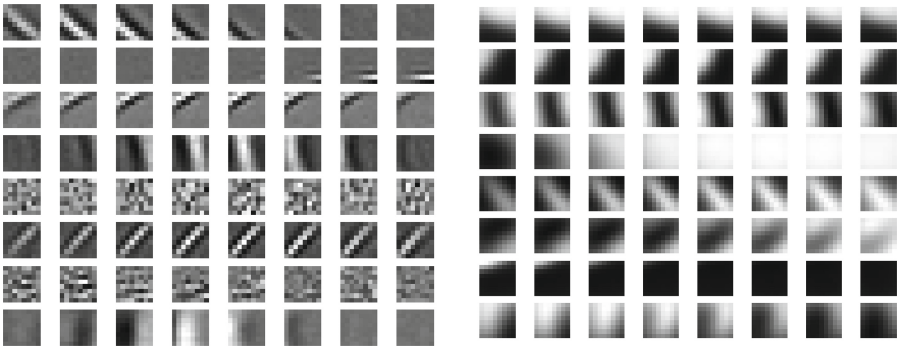

Fig. 4. 3-D features learned by K-means algorithm from MR-T1 images. Each row stands for a 3-D feature and different columns stand for different axial planes. Left: Learned from whitened image patches. Right: Learned from un-whitened image patches

3.2 Effect of 3-D Patch Size

We also computed features at different 3-D patch (volume) size settings and the results are plotted in Fig. 6. Similar to the previous figure, the x-axis shows the size of the dictionary. The 3-D feature size in the inset corresponds to the MR-T1 images. This experiment used ZCA transformed (whitened) data and $3 \times 3 \times 3$ kernels in max-pooling layers, 2 voxel stride size for MR-T1 image and 1 voxel stride size for DT-FA and DT-MD images. Overall, the $8 \times 8 \times 8$ features for MR-T1 image and the $4 \times 4 \times 4$ features for DT-FA and DT-MD image worked best. Therefore, increasing the max-pooling kernel decreased the classification accuracy.

3.3 Effect of the Size of 3-D Max-pooling Kernel

In Fig. 7, we compared the results between $3 \times 3 \times 3$, and $25 \times 32 \times 23$ maximum pooling kernel size. The x-axis also shows the size of the dictionary. In our

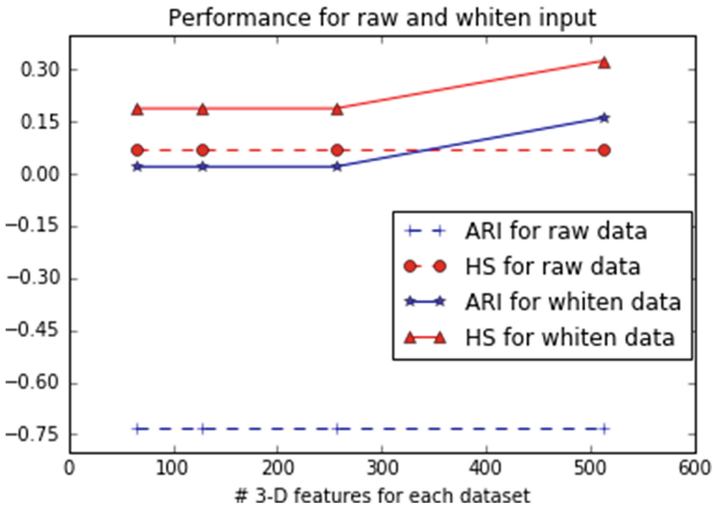

Fig. 5. The effect of whitening

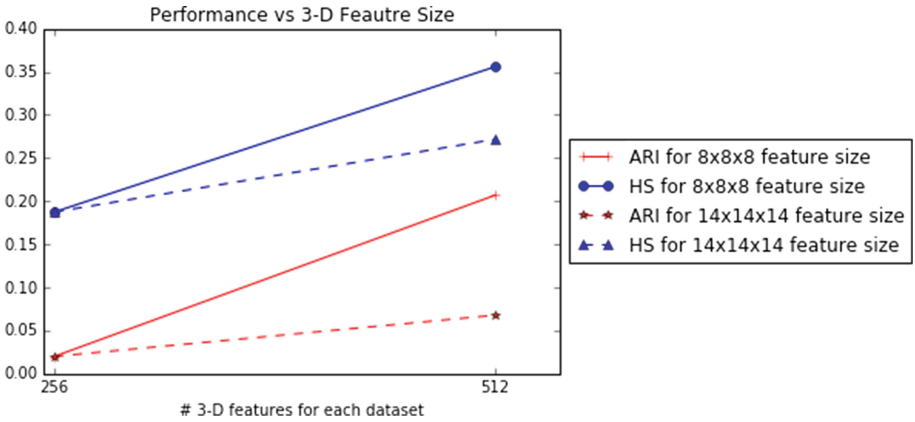

Fig. 6. The effect of 3-D features size

experiments we observe that $3 \times 3 \times 3$ maximum pooling kernels have the best performance. This experiment used whitened data sets, $8 \times 8 \times 8$ feature kernels, and a stride size of 2 voxel for MR-T1 image, and $4 \times 4 \times 4$ feature kernels and 1 voxel for DT-FA and DT-MD images.

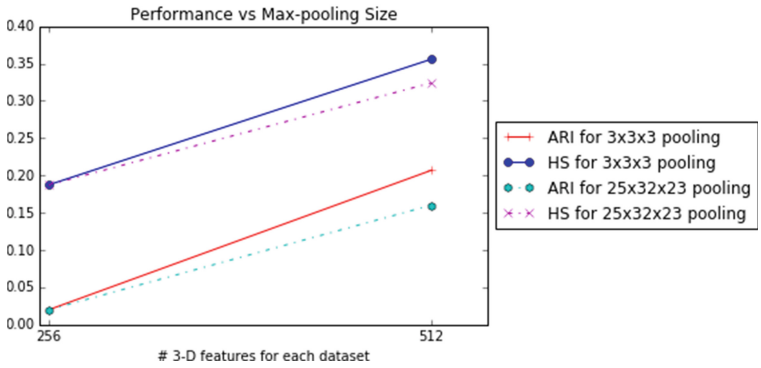

Fig. 7. The effect of max-pooling size

3.4 Effect of Dictionary Size

We considered feature representations with 64, 128, 256, and, 512 3-D dictionary items. Figures 5, 6 and 7 clearly show that a dictionary size of 512 gives the best results. Going beyond 512 did not result in much improvement.

4 Conclusion

We explored unsupervised classification of the mTBI challenge data set. Given the small number of samples, it is not feasible to train a deep learning network for feature extraction and classification. Instead we focused on computing volume features and using it for classification. In the end, the best classification results correctly classified 10 out of 15 samples for which the labels are known, and the corresponding unsupervised clustering scores are ARI = 0.267 and HS = 0.3556. We are currently working on extending this to use the tractography data computed from the DWI. Here we notice that there are significant discontinuities in the computed tracks at several potential lesion locations. Future work includes developing automated methods to detect such discontinuities and score them.

Acknowledgments. This research was partially supported by HD059217 from the National Institutes of Health.

References

1. Coates, A., Lee, H., Ng, A.Y.: An analysis of single-layer networks in unsupervised feature learning. Ann. Arbor **1001**(48109), 2 (2010)
2. Coates, A., Ng, A.Y.: Learning feature representations with K-means. In: Montavon, G., Orr, G.B., Müller, K.-R. (eds.) Neural Networks: Tricks of the Trade. LNCS, vol. 7700, pp. 561–580. Springer, Heidelberg (2012). doi:10.1007/978-3-642-35289-8_30
3. Hinton, G.E., Osindero, S., Teh, Y.-W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (2006)

4. Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**(1), 193–218 (1985)
5. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images (2009)
6. Olshausen, B.A.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature **381**(6583), 607–609 (1996)
7. Rosenberg, A., Hirschberg, J.: V-Measure: a conditional entropy-based external cluster evaluation measure. In: EMNLP-CoNLL, vol. 7 (2007)
8. Zhu, S., Wang, D., Li, T.: Data clustering with size constraints. Knowl. Based Syst. **23**(8), 883–889 (2010)