ADAPTIVE CUTS FOR EXTRACTING SPECIFIC WHITE MATTER TRACTS

Nagesh Adluru[⋆] *Vikas Singh*[†] *Andrew L. Alexander*[⋆]

[⋆] Waisman Center, University of Wisconsin-Madison
[†]Dept. of Biostatistics, University of Wisconsin-Madison

ABSTRACT

Extracting specific white matter tracts (e.g., uncinate fasciculus) from whole brain tractography has numerous applications in studying individual differences in white matter. Typically specific tracts are extracted manually, following replicable protocols which can be prohibitively expensive for large scale studies. A tract clustering framework is a suitable computational framework but from a neuroanatomical point of view, one of the key challenges is that it is very hard to design a universal similarity function for different types of white matter tracts (e.g., projection, association, commissural tracts). In this paper, we propose an adaptive cuts framework in which, using normalized cuts motivated objective function, we adaptively learn tract-tract similarity for each specific tract class using atlas based training data. Using the learnt similarity function we train an ensemble of binary support vector machines to extract specific tracts from unlabeled whole-brain tractography sets.

Index Terms— Tract specific analyses, tract clustering, specific white matter pathways, normalized cuts, feature weighting, ensemble SVMs.

1. INTRODUCTION

Tract specific analysis (TSA) is an emerging branch of region-of-interest analyses that can be used to investigate specific *a priori* hypotheses in white matter studies [1]. To perform TSA one needs to establish tract correspondence across individuals in a population. Typically, tractography is performed on the population specific atlas and specific tracts are identified manually in the atlas following protocols such as [2]. These template tracts are then warped back into the native space of the individuals relying primarily on the quality of spatial normalization for correspondence. An alternative approach is to obtain correspondences using tract clustering [3].

Hence there has been a strong continued interested in the tract clustering problem. Most existing tract clustering algorithms depend on some slight variation of the Hausdorff distance [4] for defining tract-tract similarity. For example, [5, 6] use symmetrized Hausdorff distance to define similarity and use normalized cuts for clustering. More recently, incorporating prior using white matter tract atlases has been gaining attention because of the the arbitrariness of similarity functions [7]. Although [7] incorporates prior into their clustering algorithm their similarity function is based on B-splines using a 20 knot representation. For each subject, specific white matter tracts are obtained by finding nearest neighbor from the atlas tracts using B-spline similarity. Although their approach seems to be an atlas based tract clustering, they are essentially relying on B-spline similarity which is not sufficient for all types of white matter bundles as they mention in their future work section.

[8] designed a meta similarity function and employed locally linear embedding (LLE) method for clustering. They first construct a minimum spanning tree (MST) on the tracts using pairwise Chamfer and Hausdorff distances between the tracts. Then tract-tract similarity is based on the paths in the MST. Their heuristic is that the Chamfer and Hausdorff capture the local similarity, while the paths in the MST can capture global structure. However, their similarity does not adapt to the specific anatomy of the tracts to be extracted. [9] defines intrinsic distance between two tracts using shortest geodesic paths on the so called "induced graphs". These graphs are again built using pair-wise distances (e.g., Chamfer) and Kullbleck-Leibler divergence to a prior. [10] introduces an extension of the distance transforms idea ([11, 12, 13]) for using distance between curves to model the probability distribution rather than just the distance between point coordinates in 3D space. Using this they incorporate prior as voxelwise label probabilities.

To summarize, various distinct similarities have been used for tract clustering, and each paper makes an argument for a specific similarity measure but the final clustering solution will depend on how good that measure is. Also based on the neuroanatomical complexity of the fiber tracts it seems apparent that its hard to design a single universal similarity function that works well for different types of tracts. The key contribution of the proposed approach is to use available training data and adaptively learn similarities for each specific tract as a weighted linear combination of a set of basic similarities, in a computationally efficient way. This approach can result not only in potentially more robust and flexible clustering model, but also gives some insight into the geometric properties of a tract based on the adaptive weights. There are similarities between the ideas in [14] and our proposed approach but there are key technical and application differences. [14] uses ker-

nel target alignment based objective and a grid search based optimization, while our approach is based on normalized cuts type ratio objective and a semi-definite program based optimization. Furthermore, their similarities are defined using diffusion tensor invariants rather than tractography data.

2. METHODS

2.1. Adaptive Cuts

In this section we present the main formulations that allow us to learn feature weights using an adaptive normalized cuts objective. Let $\overline{K} = \sum_{f=1}^{M} \alpha_f K_f$, where $\sum_{f=1}^{M} \alpha_f = 1$ and K_f is an $n \times n$ matrix that represents pairwise similarities of n tracts according to a feature representation, f. These feature representations form the bases for learning the tract-tract similarity (see §2.4 for specific set of feature representations used in our experiments). $\alpha = \{\alpha_f\}_{f=1}^{M}$ is referred to as the set of feature-weights. $\overline{K}$ is referred to as a *weighted* similarity matrix or kernel. If $\overline{K}$ were given, the binary class normalized cut tries to find paritions, P_1, P_2 of the n tracts by maximizing the following ratio of the intra-class to the inter-class similarities:

$$\underset{P_k}{\operatorname{argmax}} \sum_{k=1}^{2} \frac{\sum_{i,j \in P_k} \overline{K}_{ij}}{\sum_{i \in P_k, j \notin P_k} \overline{K}_{ij}} \tag{1}$$

But in our case P_ks are known from the training data and the goal becomes to estimate $\overline{K}$. Hence the following ratio needs to be optimized:

$$\underset{\alpha}{\operatorname{argmax}} \sum_{k=1}^{2} \left[\frac{\sum_{i,j \in P_k} \sum_{f=1}^{M} \alpha_f K_f(i,j)}{\sum_{i \in P_k, j \notin P_k} \sum_{f=1}^{M} \alpha_f K_f(i,j)} \right] \tag{2}$$

By exchanging the sums and simplifying the notation we obtain:

$$\underset{\alpha}{\operatorname{argmax}} \sum_{k=1}^{2} \left[\frac{\sum_{f=1}^{M} \alpha_f V(k,f)}{\sum_{f=1}^{M} \alpha_f U(k,f)} \right], \text{ where} \tag{3}$$

$$V(k,f) = \sum_{i,j \in P_k} K_f(i,j) \text{ and } U(k,f) = \sum_{i \in P_k, j \notin P_k} K_f(i,j)$$

$$= \underset{\alpha}{\operatorname{argmax}} \frac{\sum_{f=1}^{M} \alpha_f \sum_{k=1}^{2} V(k,f)}{\sum_{f=1}^{M} \alpha_f \sum_{k=1}^{2} U(k,f)}$$

$$= \underset{\alpha}{\operatorname{argmax}} \frac{\overline{V}^T \alpha}{\overline{U}^T \alpha}, \text{ where } \overline{V}_f = \sum_{k=1}^{2} V(k,f), \overline{U}_f = \sum_{k=1}^{2} U(k,f)$$

The above *single* ratio optimization can be solved using a linear program formulation. However in the adaptive cut setting since we have N bootstraps of training examples (see §2.3), the objective becomes the following *multi*-ratio function which is NP-hard to optimize:

$$\underset{\alpha}{\operatorname{argmax}} \sum_{b_i=1}^{N} \frac{\overline{V}_{b_i}^T \alpha}{\overline{U}_{b_i}^T \alpha} \tag{4}$$

We transform this problem into a quadratic program (QP) and then relax it to a semi-definite program (SDP). We first obtain an *even* number of ratios in the above multi-ratio function by (equivalently) optimizing:

$$\underset{\alpha}{\operatorname{argmax}} \sum_{b_i=1}^{N} \frac{\overline{V}_{b_i}^T \alpha}{\overline{U}_{b_i}^T \alpha} (N-1) \tag{5}$$

Now we can reformulate the above pairs-of-ratios into QP as follows:

$$\underset{\alpha}{\operatorname{argmax}} \sum_{b_i, b_j, i \neq j} \frac{\overline{V}_{b_i}^T \alpha}{\overline{U}_{b_i}^T \alpha} + \frac{\overline{V}_{b_j}^T \alpha}{\overline{U}_{b_j}^T \alpha} \tag{6}$$

$$= \underset{\alpha}{\operatorname{argmin}} \sum_{b_i, b_j, i \neq j} \frac{\alpha^T \left(\overline{U}_{b_i} \overline{V}_{b_j}^T + \overline{V}_{b_i} \overline{U}_{b_j}^T \right) \alpha}{\alpha^T \left(\overline{V}_{b_i} \overline{V}_{b_j}^T \right) \alpha}$$

$$= \underset{\alpha}{\operatorname{argmin}} \sum_{b_i, b_j, i \neq j} \frac{\alpha^T A_{b_i b_j} \alpha}{\alpha^T B_{b_i b_j} \alpha}, \text{ where} \tag{7}$$

$$A_{b_i b_j} = (\overline{U}_{b_i} \overline{V}_{b_j}^T + \overline{V}_{b_i} \overline{U}_{b_j}^T), B_{b_i b_j} = \overline{V}_{b_i} \overline{V}_{b_j}^T$$

We simplify the above hard problem into minimizing the pairs of *differences* between numerators and denominators as:

$$\underset{\alpha}{\operatorname{argmin}} \sum_{b_i, b_j, i \neq j} \alpha^T D_{b_i b_j} \alpha, \tag{8}$$

where $D_{b_i b_j} = A_{b_i b_j} - B_{b_i b_j}$. Letting $D = \sum_{b_i, b_j, i \neq j} D_{b_i b_j}$, we obtain the "standard QP" for the simplified problem as:

$$\underset{\alpha}{\operatorname{argmin}} \alpha^T D \alpha = \underset{\alpha}{\operatorname{argmin}} \alpha^T \left(\frac{D + D^T}{2} \right) \alpha = \underset{\alpha}{\operatorname{argmin}} \alpha^T Q \alpha.$$

Since even the standard QP for this sum of differences is NP-hard in general case, we minimize the following *trace* (`tr`) leading to a standard semi-definite program (SDP):

$$\underset{\alpha\alpha^T}{\operatorname{argmin}} \mathtt{tr}(Q \alpha \alpha^T), \; s.t. \sum_{f=1}^{M} \sum_{f'=1}^{M} (\alpha\alpha^T)_{ff'} = 1, \alpha\alpha^T \succeq 0, \alpha \geq 0$$

Finally, Alg. 1 shows the steps to obtain the set of feature weights, α^*. For analyses on the conditions for optimiality of the SDP relaxation, please refer to [15]. Because of the complexity of the technical details, we summarize the flow of key ideas in modeling the adaptive cuts in Fig. 1.

Algorithm 1 Finding α^* in Adaptive-Cuts

1: **Input:** $\{K_f\}_{f=1}^{M}, \{P_1^{b_i}, P_2^{b_i}\}_{b_i=1}^{N}$.
2: Compute Q.
3: Find $\left(\alpha\alpha^T\right)^*$ by solving the above SDP.
4: Compute α^* as $\{\alpha_f^* \leftarrow \sqrt{(\alpha\alpha^T)_{ff}^*}\}_{f=1}^{M}$.
5: $\forall f \in \{1, \ldots, M\}, \alpha_f^* \leftarrow \frac{\alpha_f^*}{\sum_{f=1}^{M} \alpha_f^*}$.
6: **Output:** $\{\alpha_f^*\}_{f=1}^{M}$.

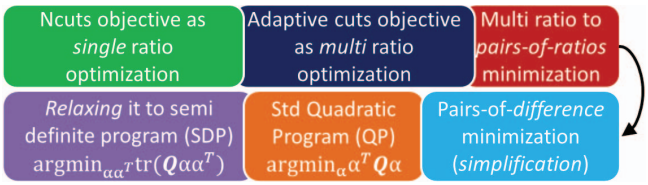


Fig. 1. The flow of key ideas in modeling the adaptive cuts framework (top: left→right, bottom: right→left).

2.2. Ensemble of Pairwise SVMs

Once we obtain the feature weights (α^*), there are several options of using these weights in extracting specific tracts: (1) One can apply these weights and obtain a weighted kernel and perform either a k-way multicut or iteratively perform normalized cuts. However there are challenges in terms of loading of approximately 250000×250000 matrix into memory. Even the Nystrom based methods face computational challenges when dealing with such large number of tracts. (2) Another option is to use the weighted kernel as a custom kernel for support vector machines (SVMs) and train a multi-class classifier. But not only is learning using 250000 examples computationally expensive, but also the classes are heavily unbalanced as can be seen from the tract counts in parantheses in Fig. 2. (3) One can train an ensemble of one vs. rest binary SVMs. This option is attractive except that we need adaptive reweighting of the loss on the training examples because of the class imbalances. (4) We use an ensemble of *pairwise* classifiers (such as Arcuate vs. IFO, Arcuate vs. UNC). Although this option may not be completely optimal in terms of a multi-class loss function, it is computationally more effective and hence we choose this option for our experiments. To learn the SVMs, we use the same bootstrapped sets (16 total, see §2.3) of pairwise classes of training tracts that are used in learning the feature weights (α^*). For each class (9 total) and each bootstrap we learn 5-fold crossvalidated SVMs which gives us a total of $9 \times 16 \times 5 = 720$ SVMs. Thus for each new tract to be classified, the output from each of 720 SVMs is compared and is assigned the class that has the maximum of all these outputs.

2.3. Bootstrapping the Atlas Tracts

In the absense of training data from individual whole-brain tractographies, we use the eight different tracts on an atlas (as shown in Fig. 2) labeled by neuroanatomists as our training data. We bootstrap the training data to generate pairwise binary classes for learning feature weights as well as training the ensemble of binary SVMs. For each specific class of tract, we sample 1000 fibers from the tract of interest and 2000 fibers from the each of the rest of 8 classes including the background tracts and repeat the whole procedure twice. Thus for each specific tract, we have $b_i = \{1, \ldots, N(=16)\}$ training sets.

2.4. Bases for Tract Similarity

It is important to have a good set of *basic* similarity functions $\left(\{K_f\}_{f=1}^{M}\right)$ between pairs of individual fibers to be able to learn robust similarity functions adaptively for each specific tract. We use six different features to define fiber similarities. Cosine bases [16] are used to represent each of the 3D coordinates, curvature and torsion functions for each fiber. By using cosine bases we can compute similarities directly using the coefficients without having to worry about the point-correspondences between the fibers (for e.g., as in Hausdorff distance). It is important to note that cosine representation is not rotation invariant. But in neuroimaging studies it is not uncommon to spatially normalize the data thus reducing the dependence on rotational invariance. For each of these three representations, we also add spatial prior using mid-points thus giving a total of $M = 6$ different feature similarities that would to adaptively combined. In principle the larger M is and the larger the training data, the better is the learnt similarity in the sense of approaching some "true" similarity function.

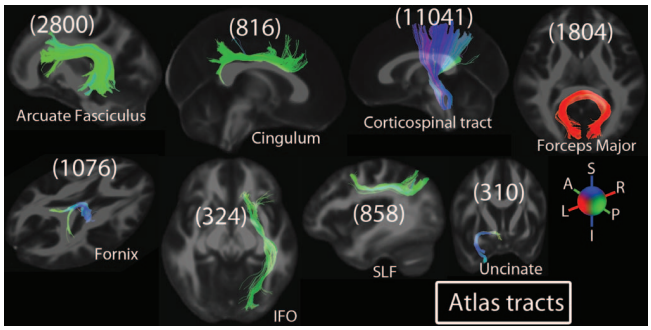


Fig. 2. The eight different white matter pathways extracted by neuroanatomists as part of the Pittsburgh Brain Connectivity (PBC) competition. The atlas pathways are used in adaptively learning the similarity and training the ensemble SVMs for automatic extraction of the pathways in individual subjects. The number of tracts in each of the specific class of tracts is shown in the parantheses. There are a total of 230971 tracts in the background and the total number of tracts in the whole brain is 250000.

3. RESULTS

The adaptive cuts framework reveals more information than just extracting specific tracts: it also provides us with a ranking of the features used to represent the tracts and thus gives us a way of interpreting geometric/shape complexity of the specific tracts. Fig. 3 shows the α^*s for each of the 9 classes of tracts and one can see that most tracts are well represented using the 3D coordinates and their mid-points and not really depend on torsion suggesting that the amount of *twisting* is not discriminative between different classes of tracts. Also as can be seen from the same figure, curvature plays a more important role for association fibers such as asoarcuate fasciculus and cingulum, which have a bend going from posterior to anterior part of the brain, compared to projection fibers such as corticospinal tract, which projects nearly straight going from inferior to superior part of the brain. The PBC competition data included two test brains in addition to the one training brain (Fig. 2). Fig. 4 shows the results of extraction

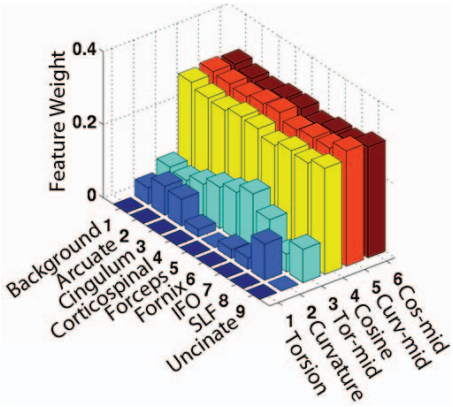


Fig. 3. Feature weights for different tracts. One can notice that torsion does not play an important role in distinguishing tracts.

of the 8 specific tracts (excluding the background class) on the two test brains using our pairwise adaptively-weighted kernel SVMs. These tracts are extracted from the whole-brain tractography data on those two brains. One can observe the high level of visual consistency between these and the atlas tracts in Fig. 2. For tracts like uncinate which have very small number of training tracts (only 310 out of 250000), the results are understandably weak. Having more training data from several individual tractography sets would help in capturing the variance in the structure for more accurate extraction.

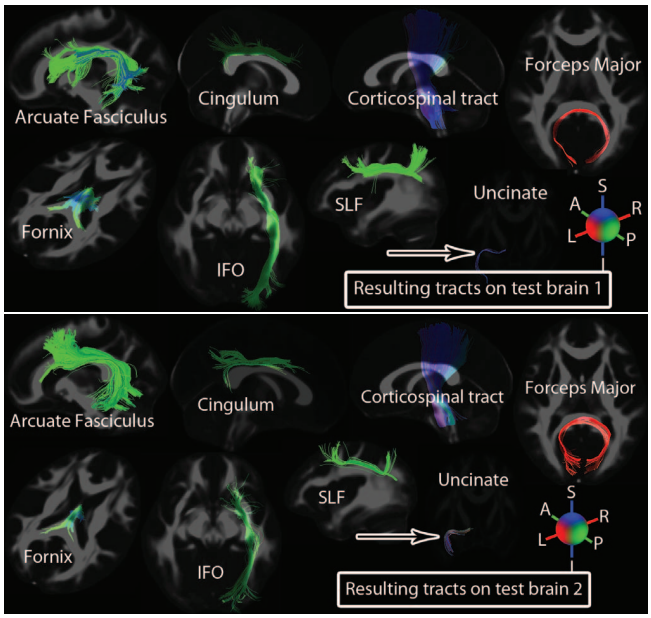


Fig. 4. The eight different tracts extracted by the proposed adaptive cuts framework from a whole-brain tractography on two different test brains. The uncinate, being a small tract, is pointed using white arrows.

4. CONCLUSION AND DISCUSSIONS

In this paper we proposed a novel way of extracting specific white matter tracts. We adapt the normalized cuts objective for each tract specifically using a semi-definite programming framework to adaptively learn feature weights. These weights provide a two-fold advantage: (1) to learn an ensemble of binary kernel-SVMs and (2) to gauge the geometric complexities of different types of tracts. We demonstrated the applicability of the framework on the PBC competition data. Due to limited number of whole-brain tractography sets and lack of accurate brain volume header information, we report only qualitative performance of our framework. More comprehensive quantitative comparisons using additional feature representations, training data from individual subjects as well as comparisons with other feature weighting based formulations such as multiple-kernel learning are part of our on-going work.

Acknowledgements

Research partially supported by NSF IIS 1116584, NIH R21AG034315, NIH P50-MH84051, NIH RO1 MH08026, NIH P30 HD003352, and the Henry M. Jackson Foundation.

5. REFERENCES

[1] P. Yushkevich and et al., "Struc-specific stat map of WM tracts," *NIMG*, vol. in press, pp. 954–966, 2008.

[2] M. Catani and M. Thiebaut, "A DTI tract. atlas for virtual in vivo dissections," *Cortex*, pp. 1105–1132, 2008.

[3] L. O'Donnell and C. Westin, "WM tract clustering and correspondence in populations," in *MICCAI*, 2005.

[4] D. Huttenlocher and et al., "Comparing images using the hausdorff dist.," *PAMI*, pp. 850–863, 1993.

[5] L. O'Donnell and et al., "A method for clustering WM fiber tracts," *AJNR*, pp. 1032–1036, 2006.

[6] L O'Donnell and C Westin, "Automatic tract. seg. using a high-dim. WM atlas," *TMI*, pp. 1562–1575, 2007.

[7] Maddah M. and et al., "Automated clustering of WM fiber tracts from DT-MRI," in *MICCAI*, 2005.

[8] Tsai A. and et al., "Fiber tract clustering on manifolds with dual rooted-graphs," in *CVPR*, 2009, pp. 1–6.

[9] Wassermann D. and Deriche R., "Sim. manifold learning & clustering of WM tracts," *MICCAIW.*, 2008.

[10] M. Maddah and et al., "Modeling of anat. info. in clustering of WM trajecs. using Dirichlet distribution," in *CVPRW*, 2008.

[11] A. Rosenfeld and J. L. Pfalz, "Distance functions on digital pictures," *PR*, pp. 33–61, 1968.

[12] M. Maddah and et al., "Probabilistic clustering and quant. analysis of WM fiber tracts," in *NIMG*, 2007.

[13] M. Maddah and et al., "A unified framework for clustering & quant. anal. of WM fiber tracts," *MIA*, 2008.

[14] P. Rodrigues and et al., "Adaptive dist. learning scheme for DTI using kernel target align.," in *MICCAIW*, 2008.

[15] L. Mukherjee and et al., "Learning kernels for variants of NCuts: Convex relaxations & appls," in *CVPR*, 2010.

[16] M. Chung and et al., "Cosine series rep. of 3D curves & its app. to WM fiber bundles in DTI," *SII*, 2010.