Validation Study: Segmentation of Gray and White Matter from T1-Weighted MRI Images and MS Lesion Segmentation by Mulitchannel Tissue Classification

Sayan Pathak
spathak@insightful.com
Insightful Corporation, 1700 Westlake Ave N, Suite 500, Seattle, WA 98109

Sept 30, 2002

Abstract:
MRI brain tissue classification is a difficult task because of the effects of noise and shading artifacts and relatively low contrast-to-noise ratio between white matter, gray matter and cerebrospinal fluid. Typical classification methods are based on pixel classification using statistical classifiers combined with Markov Random Field assumptions. Another application is the identification of Multiple Sclerosis lesions in the brain and quantifying the total lesion load.

In this study we will validate our ITK implementation of an unsupervised clustering algorithm using an unsupervised (k-Means) and a supervised algorithm (a simple Gaussian classifier) as in the Classifier Framework. We will also combine the Markov Random Field (MRF) image filter with the classifier framework and characterize the performance of the various algorithms. The results of the classification are compared with published results and other sources of available ground truth.

1. Introduction

Volumetric analysis of the brain from MR images has emerged as an important biomedical research tool to study diseases such as Alzheimer's disease, Huntington's disease, and attention deficit disorder. Segmentation of the brain parenchyma and its constituent tissue types, the gray and white matter, is necessary for volumetric information in longitudinal and cross-sectional studies. On a similar note, management of diseases like the multiple sclerosis (MS) requires quantification of lesion load.

Manual brain/lesion segmentation and registration is a tedious task that requires substantial time and effort by trained personnel and also suffers from large inter-observer variability and poor reproducibility.

In the literature, different classification techniques have been developed for brain tissue segmentation purposes. In this study, we have evaluated the classification of brain parenchyma tissue using an unsupervised and a supervised classification algorithm. We have tried to improve the results of the classification using Markov-random-fields based image filter. We have also applied the algorithms towards segmentation of MS lesions and evaluated its performance on a MS lesion-mimicking phantom with known lesion load.

2. Algorithms

2.1. Classification Framework

Typically, a classification process can be characterized by

(a) a function that defines the membership of an unclassified pixel to different tissue types,

(b) a decision rule that categorizes each unclassified pixels by assigning it a label, and

(c) a third component that precedes the membership function are the model estimators. These estimators populate the parameters of the membership functions based on the choice of model by the user.

In ITK, this factorization is used to divide the code into the modular components. Components are implemented so that components in the same category have a consistent interface (or calling convention). This consistency allows components to be swapped thereby allowing rapid prototyping of new variants of the classification algorithms.

In this validation study, we will test the combination of

- Membership function (class `MahalanobisDistanceMembershipFunction and class DistanceToCentroidMembershipFucntion`)
- Decision rule (class `MinimumDecisionRule`)
- Estimators (class `ImageKmeansModelEstimator`)

We have also use class `itkMRFImageFilter` for removing noise from the segmented image and integrate spatial adjacency information in the segmentation process.

Class `ImageClassiferBase` drives the classification process by connecting together the components, starting the calculation for the membership of each class to a corresponding tissue type and returning the final classification by using the decision rule chosen by the user.

2.1.1. Membership Function

The membership function classes operate on vector input (**x**), where **x** represents a pixel value. For a single channel image, **x** is a scalar value. However, in multichannel data **x** is a vector where the number of elements in the vector is equal to the number of channels in the input data. For instance, if we use a FLAIR (Fluid Attenuated Inversion Recovery) and a T2 series, as is the case for the MS lesion classification experiments, then each vector will have two elements.

We have used two membership functions classes in our experiments

(a) class `DistanceToCentroidMembershipFunction:` This class calculates the euclidean distance between two vectors. Each tissue type is represented by their class means ($\boldsymbol{\mu}$) or the centroids of each cluster in a given data set. The result is a vector (**d**), where $\mathbf{d} = \mathbf{x} - \boldsymbol{\mu}$. The dimension of the vector (**d**) is equal to the number of tissue types that are classified. This function is used with the K-means classifier.

(b) class `MahalanobisDistanceMembershipFunction:` This class calculates the Mahalanobis distance. When each tissue type is represented by their class means and covariance ($\boldsymbol{\mu}, \boldsymbol{\Sigma}$), the result is a vector (**d**), where

$$\mathbf{d} = (\mathbf{x} - \boldsymbol{\mu})\, \boldsymbol{\Sigma}^{-1}\, (\mathbf{x} - \boldsymbol{\mu})^{T}.$$

The dimension of the vector (**d**) is equal to the number of tissue types that are classified. This function is used with the Gaussian classifier. This is particularly relevant in multi-channel MR data where the two or more channels are correlated and incorporation of the covariance matrix improves the classification results.

The choice of the membership function is dependent on the speed vs. prformance issues. While `DistanceToCentroidMembershipFunction` is fast, the `MahalanobisDistanceMembershipFunction` performs better in the face of overlapping clusters.

2.1.2. Decision Rule

Once the memberships of a given pixel to the different tissue types have been evaluated the next step is to apply a decision rule to label each pixel with a tissue type. There can be many different decision rules such as one that assigns the tissue type to the pixel, which has the maximum probability of belonging to a particular class or as in our case the one that assigns the tissue tyoe to a pixel, which is closest to a tissue type cluster.

This is achieved by a simple function integrated with the class `MininumDecisionRule`. For each vector (**d**), this function identifies the minimum distance to the membership function. The position of the minimum distance is assigned as the class label for the corresponding tissue type. For example, if there are 5 different tissue types, the vector (**d**) will have 5 entires. Say the 3rd entry is the one with the smallest distance. The resulting pixel is classified as belonging to the 3rd tissue type. The advantage of this method is its simplicity and speed. However, more complex techniques using fuzzy logic have been developed and the existing framework enables easy integration of these advanced techniques.

2.1.3. Model or Membership Function Estimator

This is an optional step and if used it usually precedes step 2.1.1 and 2.1.2. The base class functionality is defined in `ImageModelEstimatorBase` class. We have derived two classes from this base class: (1) ImageGaussianModelEstimator and (2) `ImageKmeansModelEstimator`. These classes are relevant when the user expects automatic generation of the membership function.

The Gaussian model estimator is straightforward; it requires the user to provide some training data where the classifications into different tissue types are done *a priori*. The algorithm then calculates the means and the covariances for the different classes and creates the membership functions, which can then be plugged into the classifier framework. We have not used this estimator in our supervised classification validation study involving Gaussian model, instead we provide the model generated-apriori directly to the classification framework.

The K-means model estimator is a simple algorithm in our case; there are more advanced algorithms in the code/Numerics/Statistics. Here we use the `KmeansImageModelEstimator` class for generation of the K-means [GER92]. This object performs partitioning of data sets into different clusters either using a user provided seed points as initial guess or generates the clusters using a recursive approach when the user provides the number of desired clusters. Each cluster is represented by its cluster center. The two algorithms used are the generalized Lloyd algorithm (GLA) and the Linde-Buzo-Gray algorithms. The cluster centers are also referred to as codewords and a table of cluster centers is referred as a codebook.

As required by the GLA algorithm, the initial seed cluster should contain approximate centers of clusters. The GLA algorithm generates an updated cluster centers that result in a lower distortion than the input seed cluster when the input vectors are classified / labeled using the given codebooks.

If no codebook is provided, the Linde-Buzo-Gray algorithm is used. This algorithm uses the GLA algorithm at its core to generate the centroids of the input vectors (data). However, since there is no initial codebook, LBG first creates a one-word codebook (or centroid of one cluster comprising of all the input training vectors). The LBG uses codeword/or centroid splitting to create increasing number of clusters. Each new set of clusters is optimized using the GLA algorithm. The number of clusters increases as 2^n where n= 0, 1, ... The codebook is a vnl matrix, where there are N rows with each row representing the cluster mean of a given cluster. The number of columns in a codebook is equal to the input image vector dimension.

The threshold parameter controls the "optimality" of the returned codebook where optimality is related to the least possible mean-squared error distortion that can be found by the algorithm. For larger thresholds, the result will be less optimal. For smaller thresholds, the result will be more optimal. If a more optimal result is desired, then the algorithm will take longer to complete. A reasonable threshold value is 0.01. If, during the operation of the algorithm, there are any unused clusters or cells, the OffsetAdd and OffsetMultiply parameters are used to split the cells with the highest distortion. This function attempts to fill empty cells up to 10 times (unless the overall distortion is zero). If the GLA is unable to resolve the data into the desired number of clusters or cells, only the codewords are returned. In terms of clustering, codewords are cluster centers, and a codebook is a table containing all cluster centers. The GLA produces results that are equivalent to the K-means clustering algorithm.

2.1.4. Markov-Random Fields Post-filtering

We have also used `MRFImageFilter` class to further refine the classification of both the single and multichannel validation studies. This algorithm uses the maximum a posteriori (MAP) estimates for modeling the MRF. The object traverses the data set and uses the model generated by a classifier to gets the distance between each pixel in the data set to a set of known classes, updates the distances by evaluating the influence of its neighboring pixels (based on a MRF model) and finally, classifies each pixel to the class, which has the minimum distance to that pixel (taking the neighborhood influence under consideration). The algorithm details has been published by Besag [BES86].

3. Validation

3.1. Data

For this validation study, we used images from two sources:

(1) For the single channel classification, we used data from the "Internet Brain Segmentation Repository" (IBSR) website. This data set consists of 20 normal T1-weighted volumes (1mm x 1mm pixels, 3mm slice thickness) with manually segmented brain volume. The images were obtained using different scanners and scanning parameters.

(2) For the multichannel MS lesion segmentation study, Insightful has produced a phantom simulating the brain including structures of known volume that mimicked MS lesions. The phantom was scanned using the standard MR imaging protocol: FLAIR, T2-weighted, and PD-weighted images of the phantom were acquired. These images are used for multi-channel tissue classification validation. We have used only FLAIR and T2-weighted images, since in a previous in-house study we found this to be the optimal combination for detecting MS lesions using the Gaussian classifier.

3.2. MS phantom

An MR phantom was designed to mimic the MR properties of ventricles filled with CSF, white matter tissue and MS lesions. The phantom was constructed at the University of Washington using an in-house recipe of Jello, formaldehyde and bleach (as preservative), and Gadodiamide. By controlling the concentrations of these ingredients, the T1 and T2 MR properties of the mixture could match those of various tissues. Test tubes filled with distilled water were inserted to mimic the the CSF found in the brain ventricles. Another mixture was made with MR properties similar to MS lesions. After the "white matter" substrate had set, 29 cc of the "lesion" mixture was injected at multiple locations. The phantom was scanned on a 1.5 T MR unit (Signa Echo Speed, General Electric, Milwaukee, WI, USA). Images were scanned in an axial plane using 3 mm interleaved slices, a 20 cm field of view (FOV) with pixel size being 0.78mm. The pulse sequences used were as follows: FLAIR (TR 10000 ms; TE 120 ms; TI 1250 ms), and Dual-echo, PD and T2 fast spin echo (FSE) (TR 4000 ms; TE 15/105 ms; Echo Train Length of 8).

3.3. Procedure

The complex nature of the ground truth segmentation of the brain tissue types made the validation study with the IBSR data extremely challenging. We developed an interface to read the images and the ground truth data. For the IBSR data, the segmented tissue image was compared with the ground truth (manual segmentations), the similarity index (SI, described later) and the total volume of each tissue type were written to a file. For the phantom data, the true lesion volume known a priori is used as the ground truth.

Performance of our brain volume segmentation was assessed using the following similarity index between two sets A and B:

$$\frac{2|A \cap B|}{(|A| + |B|)},$$

where $|\cdot|$ represents the size of the set, and $\cap$ represents the intersection of the two sets. This index ranges from 0 (no overlap) to 1 (perfect alignment). The numerator of this metric is a measure of the overlap between the two sets and the denominator is the mean volume of the two sets. This index takes into account both the size and location of the overlap. The sets A and B in our case are the sets of segmented voxels extracted by our atlas-based segmentation and manual outlining, respectively.

We have not used the Hausdorff distance as a validation metric in this case since it measures the maximum deviation between the boundaries (or surfaces) between to segmented regions. This metric does not take into account the possibility of outliers existing in a data set and hence, in our experience during the validation were found to be inappropriate metric to evaluate the algorithms performance for the brain tissue classification with out applying any post-processing filters such as connected component filters that remove the noise.

For the MS lesion detection using multichannel classification study, the total lesion load was compared with the known lesion load in the phantom data. Our earlier plans for making available two patient datasets were obstructed due to unavailability of permission of the Instituitional Review Board. However, we have tested our algorithm on real patient images the University of Washington in-house studies and have been found to reduce the delineation time significantly.

3.4. Parameters

The validation studies were performed using the IBSRClassificationApp files and the MSClassificationApp files, currently located in directories `Examples\IBSRValidation\IBSRClassification` and `Examples\MultichannelTissueClassificationValidation`, respectively.

The two sub-validation studies have slightly different parameter files. The details of each of the syntax in which the parameters are passed are summarized in the `Inputs\ReadMe.txt`. Only the Gaussian classifier uses the parameter files where the mean and the covariance matrix are needed for the supervised classification. The Kmeans classification uses the `ImageKmeansModelEstimator` to generate the Kmeans from the data itself. No additional parameters were needed for the IBSR data set but for the MS lesion detection found that having a initial codebook affected the performance of the algorithm. To override the default (no initial codebook), an initial codebook was in-build within the application. This initial codebook is supposed to be insensitive towards the final Kmeans estimate and hence, has been made transparent to the user by embedding it inside the application. However, if at a later time the need for controlling the initial codebook is necessary, the classifier interface allows for appropriate changes.

The general format for the Test*.txt files are slightly different for the single channel and multichannel experiments. For the IBSR classification (single channel classification) a validation study parameter file is assumed to contain 6 or more lines with the following format:

- Line 1 specifies the path to the “20Normal_T1” directory containing the images of the 20 normal subject dataset as provided by IBSR.

- Line 2 specifies the path to the “20Normal_T1_brain” directory containing the manual brain segmentation mask for the 20 normal subject dataset as provided by IBSR.
- Line 3 specifies path to the “20Normal_T1_seg” directory containing the manual brain segmentation for gray mater, white mater and cerebrospinal fluid (CSF) for the 20 normal subject dataset as provided by IBSR.
- Line 4 specifies the output file where results of the validation are written.
- Line 5 specifies the number of channels in the dataset. In this case it is set to 1 (for single channel dataset).
- Line 6 and onwards specifies parameters for the specific data set. Each line should contain six strings:
  - The patient ID number,
  - The starting slice number of the brain images dataset,
  - The number of slices,
  - The starting slice for the segmented brain structure,
  - The number of segmented slices,
  - The path specifying the file location for the model parameters for the classifier.

For the multichannel MS study a validation study parameter is assumed to contain 6 or more lines with the following format:

- Line 1 specifies the path to the multichannel data.
- Line 2 & 3 specifies the filename extensions one for each channel. The file name format of the data is similar to the IBSR, i.e., #patientID+ “_##.raw” Since we chose to use only 2 channels there are twp extensions “_fl.raw” and “_t2.raw.”
- Line 4 specifies the output file where results of the validation are written.
- Line 5 specifies the number of channels in the dataset. In this case it is set to 2 (for multichannel dataset). Note: the application is hardwired to handle 2 channel data since current itkVector containers that hold the multichannel data needs to be fixed at compile time.
- Line 6 and onwards specifies parameters for the specific data set. Each line should contain three strings:
  - The patient ID number,
  - The number of slices in the dataset,
  - The path specifying the file location for the model parameters for the classifier.

3.5. Results and Discussions

In this validation study, our goal was to characterize the performance of the classifier without any manual editing, which is often an integral part of a medical imaging procedure. The goal of the study was to determine if the classification framework in ITK could be applied to medical images for tissue classification. Furthermore, we did not apply any pre-processing such as in-homogeneity correction or post-processing such as connected component noise filtering. Hence, while the results we report here reasonable but a clinical application would require both pre- and post-processing with manual editing for arriving at a clinically acceptable delineation.

Four algorithms were applied (1) Gaussian supervised classifier, (2) Kmeans unsupervised classifier, (3) Gaussian supervised classifier with MRF image filter, and (4) Kmeans unsupervised classifier with MRF image filter. These 4 algorithms were applied to the IBSR data sets and the Multichannel MS lesion mimicking phantom.

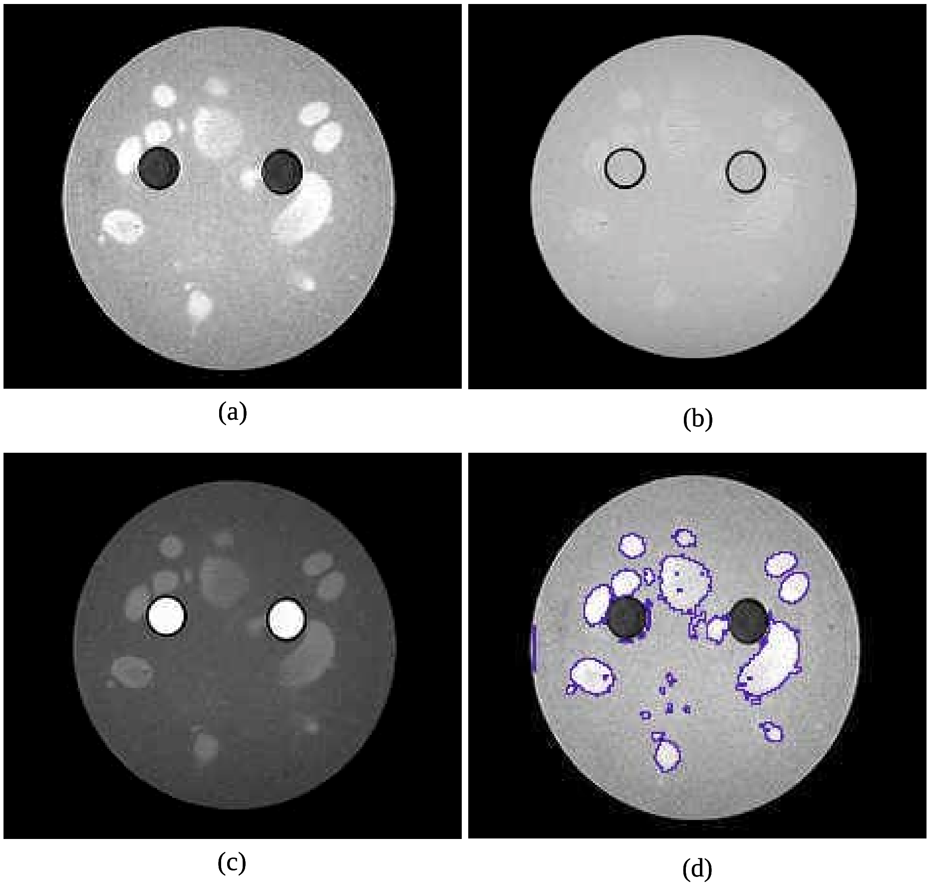
Table 1: Summary of the gray and white mater segmentation on the IBSR data.

Tissue Type	**Algorithm**	**Mean SI ± St.Dev**	**SI Range**
Gray Mater			
	Gaussian classifier	0.78 ± 0.08	0.60-0.90
	K-means classifier	0.75 ± 0.05	0.62-0.80
	Gaussian classifier with MRF filtering	0.81 ± 0.07	0.62-0.89
	K-means classifier with MRF filtering	0.76 ± 0.04	0.67-0.81
White Mater			
	Gaussian classifier	0.82 ± 0.05	0.74-0.88
	K-means classifier	0.83 ± 0.04	0.73-0.88
	Gaussian classifier with MRF filtering	0.78 ± 0.04	0.69-0.83
	K-means classifier with MRF filtering	0.83 ± 0.04	0.74-0.87

In the IBSR data sets, there were a few sets with irregular intensity jumps between slices and were treated as outliers. The result of the classification is summarized in Table 1. The performances of the different segmentation algorithms were comparable and the differences between them were statistically insignificant. The similarity index measure for the gray and white matter classification was approximately 0.8.

The algorithm had a large number of false positives for the segmentation of the cerebrospinal fluid (CSF). The mean similarity index with the Gaussian Classifier was only 0.2 which increased to 0.32 after applying a MRF image filter. The MRF filter was able to reduce the false positives significantly. We believe using T1 channel for segmentation of the CSF is sub-optimal. Incorporating more channel of information in our experience would significantly improve the performance. Also, there are advanced model building methods in the toolkit such as ones based on expectation maximization could potentially improve clustering capabilities.

For the multichannel MS phantom classification, Fig. 1 shows a cross-sectional image of the MS phantom for the FLAIR, t2, pd-weighted images along with the segmenation of the lesions based on the classification algorithm. The ground truth is known a priori to be 29.5cc for this phantom. Table 2 summarizes the results of the classification. The best result was obtained using the Gaussian classifier with an error of only 0.3%. However adding a MRF filter resulted in an increase in the error. The large adjacent non-lesion pixels influence the classification and eliminated the true lesions. However, with the unsupervised K-means algorithm adding the MRF filter reduced the error by a factor of 2. We have also experimented with adding a third channel of proton density (PD) weighted channel and also tried other possible combination s of FLAIR, T2 and PD images. In our experience, the FLAIR and T2 were found optimal for MS lesion quantification. However, the algorithm needs to be validated on real patient MR images for identifying conclusive trends in future.



(a) (b)

(c) (d)

Figure 1. Sample slice from the MS Phantom MRI showing (a) FLAIR, (b) PD-weighted, and (c) T2-weighted images and (d) the results of segmentation (using only the FLAIR and T2-weighted images) overlaid on the FLAIR image.

Table 2: Summary of the lesion segmentation in the MS phantom where the know lesion volume was 29.5 cc.

Algorithm	Estimated Lesion volume (in cc)	%error
Gaussian classifier	29.6	0.3
K-means classifier	34.7	17.6
Gaussian classifier with MRF filtering	24.5	16.9
K-means classifier with MRF filtering	26.9	8.8

4. Conclusions

In this validation study, we evaluated the effectiveness of the ITK implementation of a supervised and an unsupervised classification of both single channel and multichannel MR data sets. In the single channel validation experiment, we found both the classification approaches to be suitable for gray and white matter classification. However, the discrimination of CSF was limited. We believe that adding additional channels of information could improve the results. In multichannel validation experiment, the Gaussian classifier performed the best. However, the true potential of the algorithms remain to be tested with real patient images.

Acknowledgement

We thank Dr. Ken Maravilla to help us design and scan the MS lesion mimicking MR phantom.

Reference

[BES86] J. Besag, “On the Statistical Analysis of Dirty Pictures,” *J. Royal Stat. Soc. B*, Vol. 48, 1986.

[GER92] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston, MA, 1992.