Noname manuscript No.
(will be inserted by the editor)

Relational Learning helps in Three-way Classification of Alzheimer Patients from Structural Magnetic Resonance Images of the Brain

Sriraam Natarajan · Baidya Saha · Saket Joshi* · Adam Edwards · Tushar Khot [+] · Elizabeth M. Davenport · Kristian Kersting [**] · Christopher T. Whitlow · Joseph A. Maldjian

the date of receipt and acceptance should be inserted later

Abstract Magnetic resonance imaging (MRI) has emerged as an important tool to identify intermediate biomarkers of Alzheimer's disease (AD) due to its ability to measure regional changes in the brain that are thought to reflect disease severity and progression. In this paper, we set out a novel pipeline that uses volumetric MRI data collected from different subjects as input and classifies them into one of three classes: AD, mild cognitive impairment (MCI) and cognitively normal (CN). Our pipeline consists of three stages – (1) a segmentation layer where brain MRI data is divided into clinically relevant regions; (2) a classification layer that uses relational learning algorithms to make pairwise predictions between the three classes; and (3) a combination layer that combines the results of the different classes to obtain the final classification. One of the key features of our proposed approach is that it allows for domain expert's knowledge to guide the learning in all the layers. We evaluate our pipeline on 397 patients acquired from the Alzheimer's Disease Neuroimaging Initiative and demonstrate that it obtains state-of-the-art performance with minimal feature engineering.

1 Introduction

Alzheimer's disease (AD) is a progressive neurodegenerative condition that results in the loss of cognitive abilities and memory, with associated high morbidity and cost to society [21]. Accurate diagnosis of AD, as well as identification of the prodromal stage, mild cognitive impairment (MCI) is an important first step towards a cure and has been a focus of many neuroimaging studies.

Wake Forest School of Medicine, USA, * Oregon State University, USA,
[+] University of Wisconsin-Madison, USA, ** Fraunhofer IAIS, Germany

Magnetic resonance imaging (MRI) is a neuroimaging method that can be used for visualization of brain anatomy with a high degree of spatial resolution and contrast between brain tissue types. Structural MRI methods have been used to identify regional volumetric changes in brain areas known to be associated with AD and MCI, demonstrating the utility of such methods for studying this disease [21,25]. In particular, structural MRI has identified AD- and MCI-associated cross-sectional differences and longitudinal changes in volume and size of specific brain regions, such as the hippocampus and entorhinal cortex, as well as regional alterations in gray matter, white matter and cerebrospinal fluid (CSF) on a voxel-by-voxel basis [25]. More recently, MRI data have become the focus of machine learning experiments aimed at classifying subjects as AD vs cognitively normal (CN) or MCI vs CN. Recent approaches employ network analysis [21,22] or use machine learning directly on the voxels [25,4]. These approaches, however, only consider a two-way classification paradigm, AD vs CN, in which a clear decision boundary between these categories can be easily obtained. In reality, this progressive neurodegenerative disease is a continuum, with subjects spanning different stages from MCI to AD, making classification much more difficult.

We develop a novel data mining approach for the significantly more challenging problem of automatically classifying the subjects into one of three categories $\langle AD, MCI, CN \rangle$ given volumetric structural MRI data. Specifically, we propose a novel knowledge-based approach that allows the combination of state-of-the-art MRI data processing and modern machine learning techniques. Our pipeline consists of three stages – first is the *segmentation stage* that takes volumetric brain MRI data as an input and is then divided into anatomically relevant regions, second is a *relational mining stage* that uses the different segmenting information obtained over the image to build a series of binary classifiers and the final stage is the *combination stage* that combines the different classifiers to provide a single prediction.

The idea underlying this pipeline is simple and akin to the classical mixture of experts idea: rather than choose a single segmentation technique, we combine multiple segmentation techniques and different imaging data. For example, the knowledge-based segmentation method uses an atlas-based parcellation of the data into 116 anatomically relevant regions from which region-specific volumetric data can be extracted. Alternatively, one could employ a knowledge-free segmentation such as Expectation Maximization [5] (EM) that could result in different number of segments for different subjects depending on their brain characteristics. Hence, there is a *necessity* for employing learning algorithms that can be generalized across different number of segments or different modalities of the images. For this purpose, we employ a recently developed Statistical Relational Learning (SRL) [10] algorithm that can learn the structure and parameters of the combined model simultaneously [20]. SRL deals with machine learning in domains of inter-related objects where observations can be missing, partially observed, and/or noisy. It thus addresses the challenge of applying statistical inference and learning approaches to problems which involve rich collections of objects linked together in complex relational

networks. Given the importance of the brain network connectivity in identifying AD, SRL becomes a natural choice due to its ability to model relations such as neighborhood information. Note that if we employ a propositional classifier, we have to assume that all the subjects have equal number of segments, which is not the case in knowledge-free segmentations. As we show in our experiments, our methods outperform propositional classifiers. Also, the ability to use domain knowledge is one of the attractive features of SRL algorithms and is an essential attribute from a medical imaging perspective since the knowledge gained from decades of medical research can be very useful in guiding learning/mining algorithms.

Most SRL approaches are based on predicate logic that essentially employ binary classification, whereas here we are addressing the more challenging three-way classification. In order to still employ existing SRL approaches, we propose to solve this problem as a series of binary classification tasks (i.e., AD vs CN, AD vs MCI and MCI vs CN). This is inspired from the classical One-vs-One (OvO) classification approach that has long demonstrated to be very successful in machine learning [9,17]. The results are compared against a One-vs-all strategy (OvA) where a classifier is learned for each class separately and each class is discriminated from the others.

The essential idea in OvO is to divide the multi-class classification problem into a series of binary classification problems between pairs of classes, then combine the outputs of these classifiers in order to predict the target class. We use SRL-based classifiers for each binary classification and later combine them using a few different techniques (weighted combination, a meta-classifier, etc). The results are compared against a One-vs-all strategy (OvA) where a classifier is learned for each class separately and each class is discriminated from the other classes. We also employ two different types of segmentation algorithms (knowledge-based and knowledge-free) to demonstrate the general applicability of the pipeline.

We evaluate the pipeline on a real-world dataset, namely the Alzheimer's Disease Neuroimaging Initiative (ADNI) database of 397 subjects. It should be mentioned that in the experiments we report no subject selection took place (to identify good cases vs controls) and instead we used the complete set of subjects. This particular group of 397 subjects was selected based upon having both structural MRI and functional metabolic positron emission tomography data as part of a separate study. Similarly, we do not employ a careful feature selection but rather simply use resulting average tissue-type volume measurements obtained from the segmentation algorithms as features for our classification. Our results demonstrate that we have comparable or better performance than the current methods based upon individual binary and collective classification tasks with minimal feature engineering.

To summarize, the present paper makes the following major contributions: (1) We introduce a novel pipeline approach that combines several successful approaches in the learning and data mining communities – image segmentation, relational learning and OvO classification – to achieve very high classification performance on the very difficult task of 3-way classification for AD. (2) Our

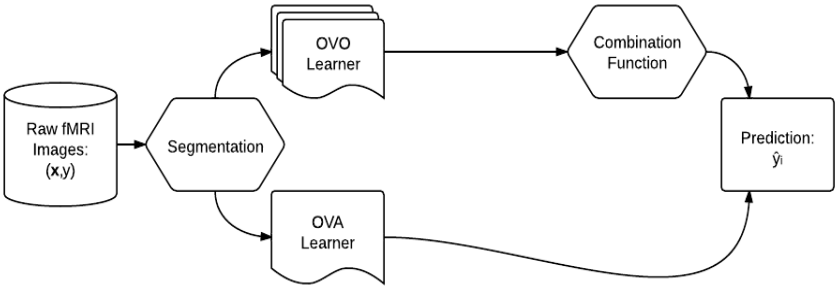

Fig. 1 Graphical Representation of the Pipeline.

approach makes it possible to include domain knowledge in the different stages of the pipeline. For instance, we can employ a knowledge-based segmentation algorithm or use knowledge of the segments themselves to guide the search in the SRL algorithm or provide relevance information when considering the neighbors of a particular region. (3) Our method allows for generalization across different number of regions for different subjects and across different imaging data types. (4) As far as we are aware, we are the first to use the classical OvO classification in the relational setting and demonstrate the usefulness of such an approach in a difficult imaging task. (5) Given the results of the different relational binary classifiers, we explore the use of different combination functions for combining them. This provides an opportunity to analyze and understand the nature of the task and that of the classifiers themselves. (6) We use a subset of the ADNI database without exclusion of cases or careful selection of controls for the harder 3-way classification task. (7) Finally, the introduction of this problem to the SRL community is itself a major contribution. As far as we know, this is the first SRL work that focuses on a medical imaging classification task, combined with state-of-the art image processing and segmentation algorithms for this purpose.

We proceed as follows. First, we outline the pipeline and its different stages. We then present the ADNI database and our experimental results comparing several different configurations of the pipeline. Finally, we conclude the paper by outlining some future research directions.

2 The Pipeline

We face the following problem:

> **Given** a set D of tuples $\{\langle \mathbf{x_1}, y_1\rangle, \cdots \langle \mathbf{x_n}, y_n\rangle\}$, where each $\mathbf{x_i}$ is a 3D voxel image corresponding to a subject and y_i is a class label (AD, MCI or CN). **Find** a function h that predicts y_i given $\mathbf{x_i}$ such that the predictions match the data drawn from the distribution that D has been drawn from.

Indeed, one is tempted to apply a standard classification approach. That is, we assume examples in D are drawn independently from identical distributions

(i.i.d.) and assume that there is a function g such that $g(\mathbf{x}) = y$, for every $\langle \mathbf{x}, y \rangle$ drawn from the distribution, and we aim to derive a function h that approximates g as closely as possible. Unfortunately, for a problem as complex as the 3-way classification of AD the standard approaches do not capture the visual aspects of the image data. They are, however, considered crucial when an MRI scan is observed by a physician.

Instead, we model the function g as a three stage pipeline i.e. $g(\mathbf{x})$ is approximated by $h_3(h_2(h_1(\mathbf{x})))$. Each stage h_i of the pipeline is designed to expose interesting and informative aspects of the data. We model the search for building the pipeline as a sequential search over individual stages. In particular we solve three problems **(1)**-**(3)**:

(1) Given the dataset D **generate** the dataset $D' = \{\langle h_1(\mathbf{x_1}), y_1 \rangle, \cdots, \langle h_1(\mathbf{x_n}), y_n \rangle\}$ where each $h_1(\mathbf{x_i})$ is a representation of the image $\mathbf{x_i}$ segmented into regions.

Each $h_1(\mathbf{x_i})$ is a set of vectors $\langle \langle s_{i,1}, f(s_{i,1}) \rangle \ldots \langle s_{i,m}, f(s_{i,m}) \rangle \rangle$ where each $s_{i,j}$ is a segmented region and $f(s_{i,j})$ is a vector of features and neighborhood information for $s_{i,j}$. Intuitively, $h_1(\mathbf{x_i})$ can be viewed as a graph where each $s_{i,j}$ is a node, and there is an edge between two nodes if the corresponding regions are neighbors in the original image. An important thing to note here is that two examples in D' need not have the same number of regions. Also, two regions need not have the same number of features (because each region can have a different set of neighbors). This makes it difficult — if not impossible — to represent D' by a flat feature vector without extensive feature engineering. A relational representation, however, is ideally suited.

(2) Given the dataset D' **train** a relational probabilistic classifier on D' that given example $\langle h_1(\mathbf{x}), y \rangle$ generates example $\langle h_2(h_1(\mathbf{x})), y \rangle$ where $h_2(h_1(\mathbf{x}))$ is a distribution over the classes AD, MCI and N, thus creating datasets D''_{train} and D''_{test}.

A single classifier in this stage can be replaced by a set of classifiers trained to produce a distribution between every pair of classes (OvO). As mentioned earlier, we use a relational classifier for this purpose. Since D' is a relational database, we cannot use propositional classifiers and have to resort to relational methods. Additionally relational methods are extremely well suited to leverage neighborhood information.

(3) Given the classifiers learned from the previous stage, i.e., h_2 for the three different combinations **design** a combination function h_3 that combines their results of the multiple classifiers in the previous step.

The resulting pipeline **(1)**-**(3)** is illustrated in Fig. 1. Next we explain each of the stages **(1)**-**(3)** in detail.

2.1 Stage 1 — Image Segmentation

To segment volumetric brain MRI data into a number of regions, we used two different segmentation techniques, namely (1) a knowledge based segmentation

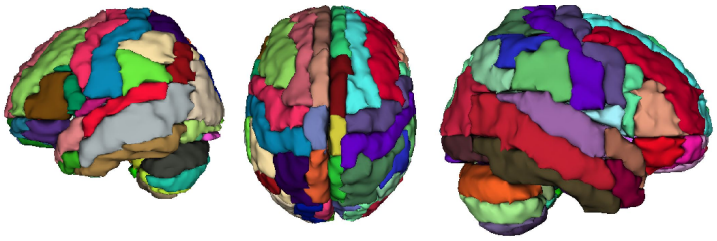

Fig. 2 AAL atlas segmentation showing the different regions of interest in the brain (Best viewed in color)

method using an anatomic atlas and (2) a knowledge-free segmentation technique based on Expectation Maximization (EM). (1) The atlas-based method parcellates the MRI data into different anatomically relevant regions whereas (2) The EM method divides the brain MRI data into different homogeneous regions based on T1-weighted voxel signal intensity (which represents the combination of three different cerebral tissues: gray matter (GM), white matter (WM) and cerebro-spinal fluid (CSF)). While the former method incorporates the knowledge of anatomical parcellations of volumetric brain data, the output segments generated by the latter method are free from a priori knowledge and clinical anatomical significance.

Atlas-based Segmentation:

The individual subject MR images were segmented into GM, WM and CSF regions, then spatially normalized to Montreal Neurologic Imaging (MNI) space and modulated with the Jacobian determinants of the warping procedure to generate volumetric tissue maps using the Dartel high-dimensional warping and the SPM8 new segment procedure as implemented in the VBM8 toolbox[1]. The resulting modulated tissue volumetric maps were further parcellated into 116 regions using the Automated Anatomic Label (AAL) atlas [2,1] as implemented by the *wfu pickatlas*[18]. Fig. 2 shows some of the AAL regions. The volumetric data from each AAL region was used as features for input of SRL based classifiers. We present these features in the next section.

Expectation Maximization: For EM, we use voxel intensity of spatially normalized volumetric T1-weighted MRI to find natural clusters within images. EM depends on soft assignment of voxels to a given set of partitions. Every voxel is associated with every partition through a system of weights based on how strongly the voxels should be associated with a particular partition. The Expectation step is defined by:

$$E[z_{ij}] = \frac{p(x = x_i|\mu = \mu_j)}{\sum_{n=1}^{k} p(x = x_i|\mu = \mu_n))} = \frac{e^{-(x_i-\mu_j)^2/(2\sigma^2)}}{\sum_{n=1}^{k} e^{-(x_i-\mu_n)^2/(2\sigma^2)}}$$

This equation states that the expectation or weight z_{ij} for voxel i with respect to partition j equals the probability that x is voxel x_i given that μ is parti-

[1] http://dbm.neuro.uni-jena.de/vbm.html

tion μ_j divided by the sum over all partitions k of these probabilities. The σ^2 in the second expression represents the covariance of the voxel intensity. Once the E-step has been performed and every voxel has a weight or expectation for each partition, the Maximization step begins. This step is defined by $\mu_j \leftarrow \sum_{i=1}^{m} E[z_{ij}]x_i$. This equation states that the partition value j is changed to the weighted average of the voxel values where the weights are the weights from the E step for this particular partition. This EM cycle is repeated for each new set of partitions until the partition values no longer change by a significant amount. Note that the EM algorithm assumes that the initial partition values are close to the natural clusters of the given voxels. We select the initial partitions randomly. Then, we run the EM algorithm for a given number of partitions and choose the number of partitions having minimum Akaike Information Criterion (AIC) [13]. We assign each voxel to a particular partition having largest posterior probability for the voxel, weighted by component probability. Finally, we find the segments or connected components from each volumetric T1-weighted MRI by assigning the neighboring voxels belonging to the same partition into the same segment.

Now, we have everything necessary to begin stage **(2)**.

2.2 Stage 2 — Boosted Relational Models

Recall that the output of the stage **(1)** is the complex network of the brain with information about each region. Such networked information can elegantly be represented using predicate logic. To so, we have to decide on the vocabulary, i.e., the predicate and constant symbols. We convert this data to predicate logic. Some of the predicates we used are presented in Table 1. The predicate names denote the attributes while the parameters are variables that can take values from a certain domain. Note that the attributes of the regions are defined in a logical form that allows for different number of regions for different persons. Similarly, the predicate *adj* allows for neighborhood definitions and this will allow us to encode an arbitrary network structure of the brain and does not constraint the number of neighbors for a region. We denote all the query predicates (*ad, cn, mci*) as y and all other ones as non-query predicates as $\mathbf{x}$.

Now, to solve problem **(2)**, we employ functional gradient boosting.Assume that the training examples are of the form $(\mathbf{x}_i, y_i)$ for $i = 1, ..., N$ and $y_i \in \{1, ..., K\}$. The goal is to fit a model $P(y|\mathbf{x}) \propto e^{\psi(y,\mathbf{x})}$. The standard method of supervised learning is based on gradient-descent where the learning algorithm starts with initial parameters θ_0 and computes the gradient of the likelihood function. Dietterich et al. [6] used a more general approach to train the potential functions based on Friedman's [8] gradient-tree boosting algorithm where the potential functions are represented by sums of regression trees that are grown stage-wise. Since the stage-wise growth of these regression trees are similar to the Adaboost algorithm [7], it is called as *gradient-tree boosting.*

Predicate	Explanation
hasregion(p,r)	Person p has region r
centroidx(P, R, X)	centroid of region R is X
avgSpread(P, R, S)	average spread of R is S
size(P,R, S)	Size of R is S
avgWMI(P, R, W)	Avg intensity of white matter in R is W
avgGMI(P, R, G)	Avg intensity of gray matter in R is G
avgCSFI(P, R, C)	Avg intensity of CSF in R is C
variance(P, R, V)	variance of intensity in R is V
entropy(P, R, E)	entropy of R is E
adj(R1,R2)	R1 is adjacent to R2
ad(P)	P has AD
mci(P)	P has MCI
cn(P)	P is cognitively normal

Table 1 Examples of predicates used in the pipeline. Here, P stands for a patient and R for a region. The last three predicates are the query predicates that are discriminated by our classifiers.

Functional gradient ascent starts with an initial potential ψ_0 and iteratively adds gradients Δ_i. This is to say that after m iterations, the potential is given by

$$\psi_m = \psi_0 + \Delta_1 + ... + \Delta_m \tag{1}$$

Here, Δ_m is the functional gradient at episode m and is

$$\Delta_m = \eta_m \times E_{x,y}[\partial / \partial \psi_{m-1} log\, P(y|x; \psi_{m-1})] \tag{2}$$

where η_m is the learning rate. Dietterich et al. suggested evaluating the gradient at every position in every training example and fitting a regression tree to these derived examples i.e., fit a regression tree h_m on the training examples $[(x_i, y_i), \Delta_m(y_i; x_i)]$. They point out that although the fitted function h_m is not exactly the same as the desired Δ_m, it will point in the same direction (assuming that there are enough training examples). So ascent in the direction of h_m will approximate the true functional gradient. The same idea has later been used to learn relational models [20], relational policies [16,19], relational CRFs [11] and relational sequences [15].

We denote all the non-query predicates as $\mathbf{x}$ and the query predicates (*ad, nor, mci*) as y. Hence, we are interested in learning $P(y|\mathbf{x})$ where $P(y|\mathbf{x}) = e^{\psi(y;\mathbf{x})} / \sum_y e^{\psi(y;\mathbf{x})}$. The main idea in the gradient-tree boosting is to fit a regression-tree on the training examples at each gradient step. In this work, we replace the propositional regression trees with relational regression trees.

Theorem 1 *The functional gradient with respect to $\psi(y_i = 1; \mathbf{x_i})$ of the likelihood for each example $\langle y_i, \mathbf{x}_i \rangle$ is*

$$\frac{\partial \log P(y_i; \mathbf{x_i})}{\partial \psi(y_i = 1; \mathbf{x_i})} = I(y_i = 1; \mathbf{x_i}) - P(y_i = 1; \mathbf{x_i}) \tag{3}$$

where I is the indicator function that is 1 if $y_i = 1$ and 0 otherwise.

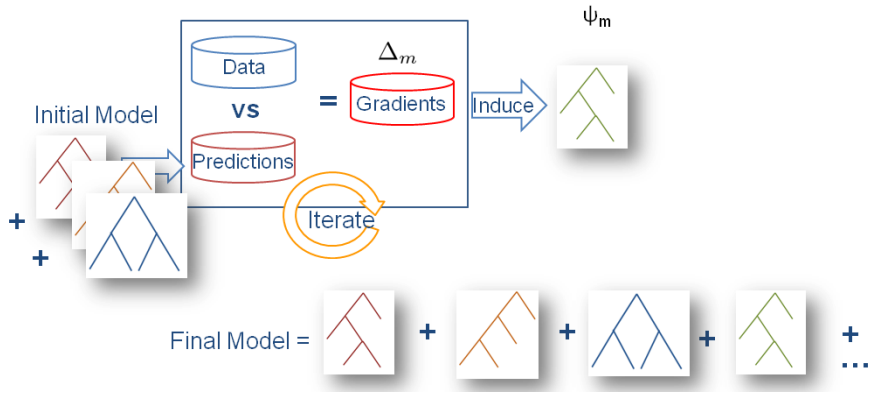

Fig. 3 Relational Functional Gradient Boosting. This is similar to the standard FGB where trees are induced in stage-wise manner the key difference being that the trees are relational regression trees.

The proof involves derivation of the gradient w.r.t ψ and is quite easy to prove. Following prior work [20], we use *Relational Regression Trees* (RRTs)[3] to fit the gradient function for every training example. In our case, a training example is a patient. These trees upgrade the attribute-value representation used within classical regression trees. Each RRT can be viewed as defining several new feature combinations, one corresponding to each path from the root to a leaf. The resulting potential functions still have the form of a linear combination of features but the features can be quite complex.

This idea is illustrated in Figure 3. First a tree is learned from the training examples and this tree is used to determine the weights of the examples for the next iteration (which in this case is the difference between the true probability of being true and the predicted probability). Once the examples are weighted, a new tree is induced from the examples. The trees are then considered together and the regression values are added when weighing the examples and the process is repeated.

At a fairly high level, the learning of RRT proceeds as follows: The learning algorithm starts with an empty tree and repeatedly searches for the best test for a node according to some splitting criterion such as weighted variance. Next, the examples in the node are split into *success* and *failure* according to the test. For each split, the procedure is recursively applied further obtaining subtrees for the splits. We use weighted variance on the examples as the test criterion. In our method, we use a small depth limit (of at most 3) to terminate the search. In the leaves, the average regression values are computed. We augment RRT learner with aggregation functions such as *count, max, average* that are used in the standard SRL literature [10] in the inner nodes thus making it possible to learn complex features for a given target. These aggregators are pre-specified and the thresholds of the aggregators are automatically learned from the data. We restrict our aggregators to just the three mentioned earlier.

In the case of continuous features such as *intensity* level, *size*, *spread*, etc., we discretize them into bins.

Finally, to prepare for stage **(3)**, we represent the distribution over the classes as a set of RRTs on the features. For example, when we classify AD vs. CN patients, we learn 20 RRTs for predicting if the person has AD. Since it is a binary classification, it is sufficient to learn one set of 20 trees for the class AD. Similarly, we learn two other sets of 20 trees each for predicting AD vs MCI and CN vs MCI leading to a final model with three sets of 20 trees each. In the case of OvA, there will be three sets of 20 trees - one each for predicting AD, MCI, and CN given the rest of the classes.

Now, we have everything together for the final stage **(3)** of our pipeline.

2.3 Stage 3 — Combining Classifiers

We investigated two alternatives. We first present our One-vs-One (OvO) method before explaining the One-vs-All (OvA) strategy. The result of previous step is a set of probabilistic classifiers for each pair of classes from AD, MCI and CN (in essence, 3 classifiers). So, now there is a need to combine these multiple classifiers. For a detailed review, see [9]. Let us denote each classifier as $c^k, k = 1, 2, 3$. We have used the following combination functions:

– **Voting:** Each c^k outputs a prediction and the class has the maximum vote i.e., $argmax_c \sum_k [I(y^k = c)]$, where y^k is the predicted label of the k^{th} classifier and c is the class.
– **Weighted Voting:** In this case, $class = argmax_c \sum_k [w^k \cdot P(y^k = c)]$. We derived a gradient for the log likelihood of the training data and also used a grid search over the weight space and report the results of the different methods.
– **Pairwise Coupling:** We considered the PC method [12] where the goal is to determine the posterior over each of the classes from the estimated joint distributions.
– **Classifier method:** We used the output of each OvO classifier to train a proposional classifier such as SVM, Bagging etc. that combines the output of these different classifiers to make its final prediction. The input of the new classifier is essentially the predictions of the classifiers of the previous stage. More precisely, the input is a set $P = \langle p_1^1, p_2^1, ..., p_3^3 \rangle$ for each patient i, where p_j^k is the posterior probability of the class j as predicted by the classifier k. Hence, we aim to learn a function h_3 such that $h_3(P) = y_i$ where y_i is one of AD, CN or MCI. The advantage is that while the use of weights assumes that the OvO results are combined using a linear function, the use of classifiers makes it possible to use non-linear combinations of the OvO results leading to more expressive models.

The OvA strategy employs three classifiers. Each of them discriminate class j from $j' \in class \setminus j$. We use a simple aggregation method called as *Maximum confidence strategy* which is similar to the voting strategy presented earlier. The output class is taken from the classifier that has the largest posterior

probability $argmax_c\, p_c$. For more details on the OvA aggregation, please refer to [9].

Given the above combination functions, the net result is the prediction of the disease state for the patient given the T_1 weighted scan. Hence the resulting classifier h is essentially a nested classifier $h_3(h_2(h_1(\mathbf{x})))$; the final output of our pipeline **(1)**-**(3)**.

3 Experimental Setup

In order to investigate the performance of the proposed pipeline for three-way classification of Alzheimer patients from structural magnetic resonance images of the brain, we followed the following experimental protocol.

ADNI Subjects. Data used in this study were obtained from the Alzheimer's disease Neuroimaging Initiative (ADNI) database (www.loni.ucla.edu/ADNI) sponsored by the NIH and industrial partners. The primary goal of ADNI is to test whether serial MRI, PET, other biological markers, and clinical and neuropsychological assessment can measure the progression of mild cognitive impairment and early Alzheimer's disease. Further information can be found at www.adni-info.org. We used data available from 102 CN (average age 75.8, 62 male, 40 female), 92 AD (average age 75.5, 55 male, 37 female), and 203 MCI (average age 74.8, 137 male, 66 female) participants.

Set up. Each subject's T_1-weighted MRI data was used (2122945 voxels). We used the spatially normalized voxel data to run EM and the pre-processed modulated segmentation maps for the AAL segmentation algorithm. For each segment, several features were extracted - *avgWMI*, *avgGMI*, *avgCSFI* (which are the average value across voxels from the WM, GM and CSF modulated tissue maps generated in the SPM new segment procedure), size, centroid and spread of each segment. The centroid is a three-dimensional attribute. Also included were the neighborhood information about the segments, where the number of neighbors for each segment can be very different and necessitates the use of SRL-based algorithms.

We used 10-fold cross validation in all our experiments. For the OvO based learners, we created training sets for each classifier (AD vs CN, AD vs MCI, MCI vs CN). Hence the cases and controls were chosen separately for each classifier. To ensure correctness of comparison, we went through the entire data base and created 10 different folds such that each subject was in the test set for one fold and in the training for the rest. Given this, for each fold, we used the training set data to create three different training sets for the OvO classifier. For instance, when creating the training set for the first fold of AD vs CN classifier, we remove all the subjects who had MCI in that training fold. This ensured that we trained on the same set of subjects for all the three classifiers in each fold and that the test examples were never seen by any of the three classifiers.

Once the individual classifiers were learned, we used the common training set to learn the combination function for combining their predictions. The

common training set is the union of the three training sets and do not contain a single example from the test fold. Once the combination function is learned the predictions were made on the test fold and the results averaged over 10 runs. For the OvA classifier, things are simpler in the sense that we can use just the 10 training and test sets and evaluate the performance as there is no need for creating smaller training sets from the given training set. For the propositional classifiers we used the default functions of Weka and LibSVM to create the 10 folds.

It should be reiterated that these training and test folds were chosen at random – no careful selection of cases vs controls was performed. Also, we did not perform any major feature selection. The features of each segment were used as they are. We preprocessed the data only to convert it into predicate logic format. Also, since most SRL methods are based on predicate logic and almost all the features are real numbers in our problem, we had to discretize these features. Each feature was discretized into several bins based on the histograms of values and natural points for discretization were picked automatically using filters in Weka. Using domain knowledge in this step (clinically relevant discretizations) remains one very interesting future direction.

4 Results

We compare several versions of the algorithms in this section, including the list of propositional classifiers on the AAL segmented data and the relational classifiers using both segmentation methods (EM and AAL) as well as different combination functions. To understand the need for segmentation, we used modulated gray matter voxel data with LibSVM. We report these results as well. We did not use any segmentation algorithm for this setting of SVM, which we will denote as SVMMG.

– **Propositional Classifiers** - Naive Bayes (NB), Decision Trees (J48), SVMs, AdaBoost and Bagging on the AAL data and SVMs with gray matter data (that we denote as SVMMG).
– **Relational OvO with AAL segmentation** - Using various combination functions: Weighted voting with grid search (AALGS), gradient descent (AALGD), bagging (AALB), AdaBoost (AALA) and Pairwise coupling (AALPC).
– **Relational OvO with EM segmentation** - Also using various combination functions: Weighted voting with grid search (EMGS), gradient descent (EMGD), bagging (EMB), AdaBoost (EMA) and Pairwise coupling (EMPC).
– **Relational OvA** - With AAL (OvAAAL) and EM (OvAEM).

The best parameter settings for propositional classifiers are presented in Table 2 and these were obtained using cross-validation. First, we compared the propositional classifiers with AAL segmentation. Results are presented in Figure 4.*a*. We used Weka and used the multi-class classification setting. As can be seen from the figure, the propositional algorithms do not show a good performance using the AAL data. We also present the results of running LibSVM on the voxel data (i.e., without any segmentation - SVMMG). As can

Classifier	Parameters
J48	C 0.25 M 2
SVM	C 1.0, L 0.01, P 1E12, N 0, V 1, W 1 RBF
AdaBoost	P 100, S 1, I 10
Bagging	P 100, S 1, I 10, M 2
Logistic	R 1.0E-8, M -1

Table 2 Parameter settings for propositional classifiers.

be seen, the performance is slightly better but still is not comparable to the performance of the best relational + segmentation algorithm (AAALB) which is presented for comparison purposes. Measuring accuracy over the entire data set can be misleading [14], hence, we also compute the area under the curve for the Receiver Operating Characteristics curv (AUC-ROC). The AUC-ROC has long been viewed as an alternative single-number measure for evaluating the predictive ability of learning algorithms. This is because the AUC-ROC is independent to the decision threshold and invariant to the priors on the class distribution.

We also evaluate different versions of the relational learning algorithms. Results are presented in Figure 4.*b*. We also included the OvA classifiers with AAL and EM in the results. It can be seen that the best performing algorithms use AAL and some combination function based on a classifier. AALB has the best results among the different algorithms presented in the figure. The other classification functions did not have nearly as good a performance as bagging but are significantly better than the propositional algorithms. This clearly shows that treating the problem as a multi-class classification problem may not be the best solution (OvA methods also do not perform well). Instead, posing the problem as a slightly more complex OvO problem significantly improves performance.

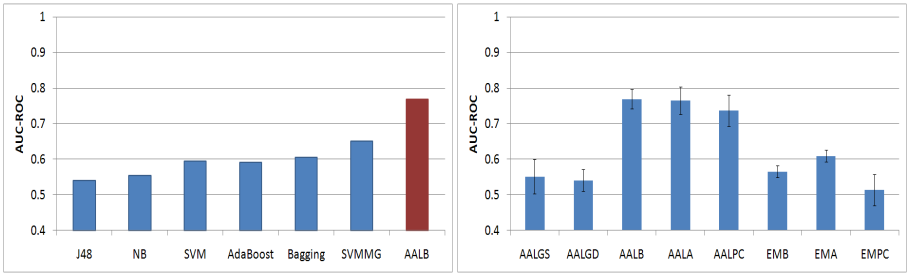

Fig. 4 Classification performances in terms of "Area under the ROC curve" of the different algorithms: (a) propositional classifiers (blue) compared against the relational AALB (red), and (b) relational classifiers. (c) Parameter settings for propositional classifiers

A consolidated version of the results for all the classifiers is presented in Table 3. In general, the relational methods have a superior performance com-

Classifier	AUC-ROC	Classifier	AUC-ROC
J48	0.540 ± 0.01	SVM	0.595 ± 0.01
NB	0.554 ± 0.01	Bagging	0.606 ± 0.01
Adaboost	0.591 ± 0.01	SVMMG	0.619 ± 0.01
EMPC	0.514 ± 0.04	OvAAAL	0.605 ± 0.01
EMB	0.565 ± 0.02	EMA	0.609 ± 0.03
OvAEM	0.544 ± 0.02	AALGS	0.551 ± 0.09
AALGD	0.541 ± 0.06	AALPC	0.737 ± 0.09
AALA	0.765 ± 0.07	AALB	0.769 ± 0.05

Table 3 Consolidated results of the classifiers.

pared to the propositional algorithms with AAL segmentation. For instance, AALA and AALB have scored the best on this data compared to the standard classifier methods (last row of the table). Comparatively, the first three rows present the standard machine learning classifiers on the same data. As can be seen there is significant difference between the first three rows and the last row of the table.

Note that AALPC (2nd last row, third column) which is the method that uses pairwise coupling as against a classifier has a competitive performance compared to AALB. This justifies observations made earlier [9] that pairwise coupling can be a very promising method to combine multiple OvO classifiers. The knowledge-based segmentation algorithm of AAL also has a higher performance than the knowledge-free EM algorithm. It remains an interesting future direction to explore the use of domain knowledge to guide the EM algorithm to better segment the images in order to increase performance. While this may not be very useful in our current task, in other problems such as identifying MCI patients who are likely to develop AD, it may be potentially useful to combine the clinical knowledge for guiding the segmentation algorithm and the classifier.

To understand how the methods performed on individual classification tasks (AD vs CN, AD vs MCI, MCI vs CN), we also present the confusion matrices in Table 4. We include a single confusion matrix for each of the three OvO classifiers using the AAL segmentation method. Consideration of the matrices will show that, while we can achieve a relatively high true positive rate (TPR) and true negative rate (TNR) when classifying AD v CN and AD v MCI, classification of MCI v CN is a more difficult task. Hence, we see a proportionally larger number of false negatives in the third confusion matrix. It can be clearly seen that while we are tackling the hard problem of 3-class classification, it also helps in the two-class classification case. More precisely, learning in the harder task helps the classifiers to improve on the easier task.

We present the segments that are used in our learned models in Figure 5. These are the regions that discriminate between the classes and are identified by learned algorithms and correspond to the medically relevant regions as verified by our Neuroradiologists. In order to construct this Figure, we extracted the regions used by the trees in the internal nodes and plotted these regions on the different views. The goal of this exercise was to evaluate if the learned

Confusion Matrices						
	AD v CN		AD v MCI		MCI v CN	
	Pos	Neg	Pos	Neg	Pos	Neg
Pos	64	18	27	60	149	44
Neg	16	86	30	168	76	26

Table 4 Confusion matrices for the three classifiers

models confirm to known regions or if they are considering non-important regions for classification.

The first, second and third columns represent coronal, sagittal and axial view of the same slice of a patient respectively while the rows correspond to predicting AD (vs CN), AD (vs MCI) and MCI (vs CN) respectively. Our proposed algorithm shows consistency in detecting the regions that are known clinically to be affected by AD [21] (regions of interest – for example, number 37 - 40 hippocampus, 49-55 occipital, 59-62 parietal and 81-85 temporal). This shows that the learning algorithms perfectly compliment the segmentation algorithms in this task. While in previous methods the neurologists had to use the specific regions for correlations, our method identifies them automatically and uses them in the prediction models.

Our results show that SRL algorithms better interact with the segments created by AAL. It is also clear that while learning to predict three classes, individual classifiers are themselves quite predictive. Finally, it is very encouraging that the algorithms are able to identify the segments that are known to be clinically interesting in predicting AD. Most of the methods to-date have computed the correlation between the regions for predicting AD, but our methods automatically identify the interesting segments for this 3-class task.

5 Conclusion and Future work

We have addressed a challenging three class classification problem from MRI images. Specifically, we proposed to solve the problem of classifying patients into one of AD, MCI or CN using a pipeline that consists of three different stages. First is a segmentation stage where the regions are grouped into (medically relevant) regions. The second stage is a relational learning stage which learns on the network created by the previous stage. This stage essentially uses a series of binary classifiers for multi-class classification. The final stage is the combination stage that combines the results of these multiple classifiers. The use of a graph network (with varying number of nodes) in the first stage necessitates the use of a relational learning algorithm. Our extensive experimental results demonstrate that the pipeline obtains state-of-the-art performance with minimal feature engineering. The pipeline is the first application of SRL to MRI classification, and the results clearly illustrate the benefits of using a relational representation in the first and second stage of the pipeline. It naturally accounts for varying numbers of segments, suits a knowledge-based segmentation, and scales well from the two-class to the three-class problem;

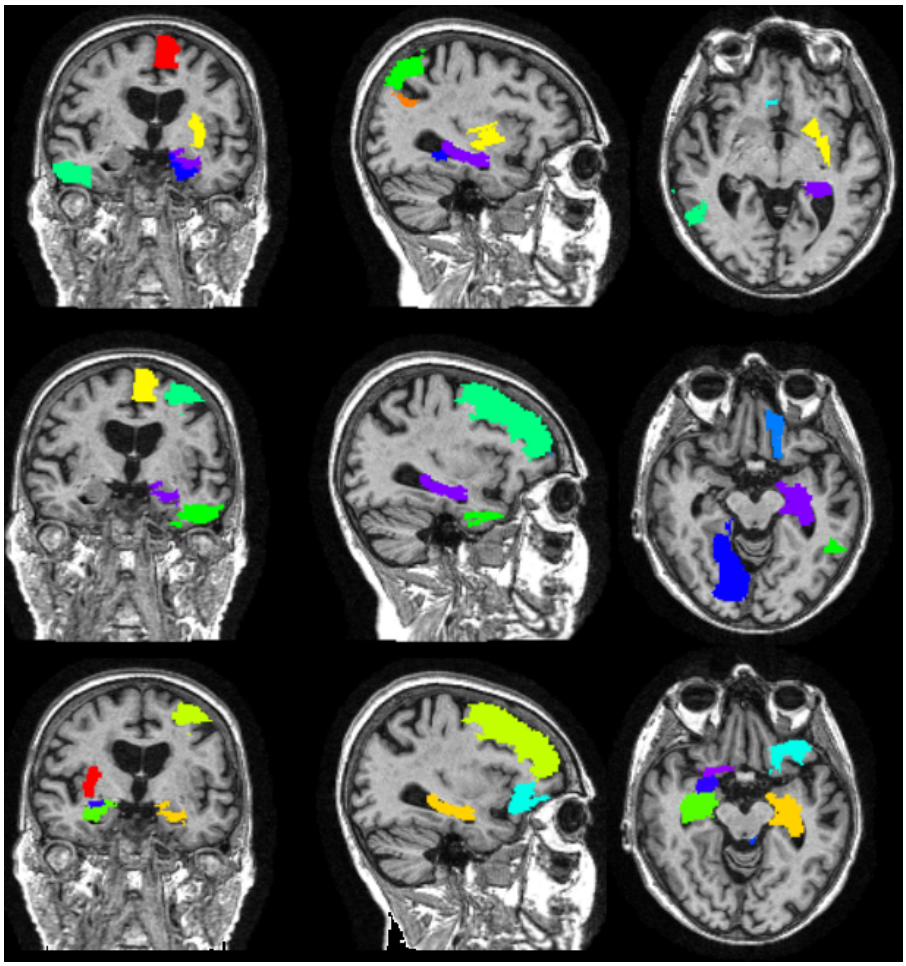

Fig. 5 Predictive segments as identified by our pipeline (different colors indicate different regions).

while having a reasonable performance in the two-class setting, propositional approaches yield a significantly lower performance in the three-class case.

Our work provides several interesting avenues for future work. One of our future directions is to use the knowledge of the domain experts in guiding the segmentation algorithms to identifying more clinically relevant regions. Due to the use of logical variables and unification, statistical relational models generalize well. Hence, we plan to apply the algorithms to more challenging tasks such as identifying those MCI patients who are affected by AD later in life. Our initial results indicate that clinical data can be very useful for this task when combined with MRI scans. We plan to combine the two different data types to determine whether their combination is an improvement over either of them separately. Also, our current work considers only T_1 weighted images.

There are other types such as T_2 and FLAIR and we intend to generalize the algorithms to determine if including these other weighted image types can improve the prediction task. It would also be interesting to employ local learning methods such as the one presented in Tang et al. [23] for learning to classify from a single image. Finally, it might also be possible to perform a two stage learning task by running algorithms such as PCA to reduce the dimensionality and then learn a similarity metric between the classes [24].

Acknowledgment

We would like to thank Ben Wagner for help with programming and creating the data set. SN acknowledges the support of Translational Science Institute of Wake Forest School of Medicine. KK was supported by the Fraunhofer ATTRACT fellowship STREAM.

References

1. http://prefrontal.org/blog/2008/05/brain-art-aal-patchwork.
2. http://www.slicer.org.
3. H. Blockeel and L. De Raedt. Top-down induction of first-order logical decision trees. *Artificial Intelligence*, 101:285–297, 1998.
4. K. Chen, E. M. Reiman, G. E. Alexander, D. Bandy, R. Renaut, W.R. Crum, N. C. Fox, and M. N. Rossor. An automated algorithm for the computation of brain volume change from sequential mris using an iterative principal component analysis and its evaluation for the assessment of whole-brain atrophy rates in patients with probable alzheimer's disease. *Neuroimage*, 22(1):134–143, 2004.
5. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B.39:pp. 1–38, 1977.
6. T.G. Dietterich, A. Ashenfelter, and Y. Bulatov. Training conditional random fields via gradient tree boosting. In *ICML*, 2004.
7. Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *ICML*, 1996.
8. J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.
9. Mikel Galar, Alberto Fernández, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recogn.*, 44:1761–1776, August 2011.
10. L. Getoor and B. Taskar. *Introduction to Statistical Relational Learning*. MIT Press, 2007.
11. B. Gutmann and K. Kersting. TildeCRF: Conditional Random Fields for Logical sequences. In *ECML*, 2006.
12. T. Hastie and R. Tibshirani. Classification by pairwise coupling. In *NIPS*, pages 507–513, 1998.
13. Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2001.
14. J. Huang and L. C.X. Using auc and accuracy in evaluating learning algorithms. *Knowledge and Data Engineering, IEEE Transactions on*, 17(3), 2005.
15. A. Karwath, K. Kersting, and N. Landwehr. Boosting Relational Sequence alignments. In *ICDM*, 2008.

16. K. Kersting and K. Driessens. Non–parametric policy gradients: A unified treatment of propositional and relational domains. In *ICML*, 2008.
17. Stefan Knerr, Léon Personnaz, and Gérard Dreyfus. Single-layer learning revisited: A stepwise procedure for building and training a neural network. In F. Fogelman Soulié and J. Hérault, editors, *Neurocomputing: Algorithms, Architectures and Applications*, volume F68, pages 41–50. Springer-Verlag, 1990.
18. J. A. Maldjian, P. J. Laurienti, R. A. Kraft, and J. B. Burdette. An automated method for neuroanatomic and cytoarchitectonic atlas-based interrogation of fMRI data sets. *NeuroImage*, 19(3):1233–1239, 2003.
19. S. Natarajan, S. Joshi, P. Tadepalli, K. Kristian, and J. Shavlik. Imitation learning in relational domains: A functional-gradient boosting approach. In *IJCAI*, 2011.
20. S. Natarajan, T. Khot, K. Kersting, B. Guttmann, and J. Shavlik. Gradient-based boosting for statistical relational learning: The relational dependency network case. *Machine Learning*, 2011.
21. Liang Sun, Rinkal Patel, Jun Liu, Kewei Chen, Teresa Wu, Jing Li, Eric Reiman, and Jieping Ye. Mining brain region connectivity for alzheimer's disease study via sparse inverse covariance estimation. In *In KDD*, 2009.
22. Kaustubh Supekar, Vinod Menon, Daniel Rubin, Mark Musen, and Michael D. Greicius. Network analysis of intrinsic functional brain connectivity in Alzheimer's disease. *PLoS Computational Biology*, 4(6):e1000100, 2008.
23. Y. Tang, P. Yan, Y. Yuan, and X. Li. Single-image super-resolution via local learning. *International Journal of Machine Learning and Cybernetics*, 2(1):15–23, 2011.
24. X. Xu, W. Liu, and S. Venkatesh. An innovative face image enhancement based on principle component analysis. *International Journal of Machine Learning and Cybernetics*, 3(4):259–267, 2012.
25. Jieping Ye, Gene Alexander, Eric Reiman, Kewei Chen, Teresa Wu, Jing Li, Zheng Zhao, Rinkal Patel, Min Bae, Ravi Janardan, and et al. Heterogeneous data fusion for alzheimer?s disease study. *In KDD*, page 1025, 2008.