CAVIAR: CLASSIFICATION VIA AGGREGATED REGRESSION AND ITS APPLICATION IN CLASSIFYING OASIS BRAIN DATABASE

Ting Chen, Anand Rangarajan, Baba C. Vemuri

Department of CISE, University of Florida, Gainesville, FL 32611

ABSTRACT

This paper presents a novel *cla*ssification *vi*a *a*ggregated *r*egression algorithm – dubbed CAVIAR – and its application to the OASIS MRI brain image database. The CAVIAR algorithm simultaneously combines a set of weak learners based on the assumption that the weight combination for the final strong hypothesis in CAVIAR depends on *both* the weak learners and the training data. A regularization scheme using the nearest neighbor method is imposed in the *testing* stage to avoid overfitting. A closed form solution to the cost function is derived for this algorithm. We use a novel feature – the histogram of the deformation field between the MRI brain scan and the atlas which captures the structural changes in the scan with respect to the atlas brain – and this allows us to automatically discriminate between various classes within OASIS [1] using CAVIAR. We empirically show that CAVIAR significantly increases the performance of the weak classifiers by showcasing the performance of our technique on OASIS.

Index Terms— aggregated regression, classifier ensemble, OASIS, dementia

1. INTRODUCTION

Brain MRI analysis and its associated application in the diagnosis and treatment of brain-based diseases has attracted immense attention in the past two decades. Dementia is an example of such a neurological disorder, that may occur at any stage of adulthood and lead to long-term decline in cognitive function. Therefore, a technique that is able to detect the changes in brain structures due to the onset of dementia and use this information to classify the subjects is of great value. One of the main challenges is that it is not easy to find a single classifier that achieves a low error rate. Due to the difficulty in obtaining good features from MRI, the classifiers we obtain are actually weak classifiers. However, can a set of weak classifiers create a single strong learner? Numerous variants of algorithms for classifier ensembles have been proposed in literature, for instance, boosting and bagging. Generally speaking, the boosting method iteratively refines each weak learner and re-adjusts the weighted combination of the training data after each iteration. Another type of algorithm is called bootstrap aggregation (bagging) proposed by Breiman [2]. Here bootstrap samples are trained using weak learners and the output of the weak predictors are combined by averaging or voting. In Adaboost [3], it is required that the performance of weak learners be slightly better than mere random assignment. Therefore, a "very weak" weak classifier is not acceptable. Besides, this algorithm also needs a large number of weak learners in order to converge. Meanwhile, due to the equally weighted voting scheme, bagging also has its limitation in terms of improving on linear models.

In this paper, we propose a novel ensemble classifier - called CAVIAR - which is different from the bagging and boosting mentioned above and offers a significantly better performance. CAVIAR is a regression based classification algorithm, which assumes that the weights for combining the weak learners depend not only on the weak learners but also on the training data. *It is analogous to a medical consultation carried by a group of doctors on a number of patients. We think that it's well-justified to assume that each patient's personal condition has a different affect on the experts' consensus final decisions.* Since in CAVIAR the weights vary over both the weak learners and the training set, it is prone to overfitting. We impose a regularization scheme wherein the weights corresponding to closed training patterns are forced to be similar. Furthermore, in the testing phase, a strong classifier is constructed by choosing the weights corresponding to the nearest neighbors of the test pattern among the training data set. *Intuitively speaking, as a new patient comes in for medical consultation, we expect that a good strategy for the experts involves searching through similar case studies in order to arrive at a consensus diagnosis*. To the best of our knowledge, this kind of data adaptive testing technique has never been reported in literature. We demonstrate the stability and effectiveness of the classification model, along with its novel testing technique via numerous experiments on the OASIS database.

2. CLASSIFICATION VIA AGGREGATED REGRESSION (CAVIAR) ALGORITHM

Formally speaking, let X be the set of training feature vectors and Y be the set of labels, where (x_n, y_n) are drawn

This research was in part funded by the NIH grant RO1 NS046812.

randomly from $X \times Y$ based on some unknown distribution. In the case of two-class classification, Y is a binary set containing two labels $\{+1, -1\}$. Assume h is the weak hypothesis applied to the instance taken from the sample set X, with its magnitude $|h|$ being the confidence of the prediction and its sign distinguishing the class to which it belongs. Let $W = \{w_{nt}\}$, with $n = 1, 2, \ldots, N$ and $t = 1, 2, \ldots, T$, be the weight matrix that corresponds to the training samples and weak learners. The goal in the classification problem is to find a proper combination $C(x_n) = \sum_t w_{nt} h_t(x_n) = \mathbf{w}_n \cdot \mathbf{h}(x_n)$ for *each data* that minimizes a given classification error discriminant function $\sum_n Dist(C(x_n), y_n)$ for the whole training data set. To simplify the notation, we use $\mathbf{w}_n$ to denote the vector $(w_{n1}, w_{n2}, \ldots, w_{nT})^t$ and let $\mathbf{h}$ be the vector $(h_1(x_n), h_2(x_n), \ldots, h_T(x_n))^t$.

So far, the curious reader may have two concerns. First, since the dimension of the weights to be optimized in CAVIAR is $N \times T$ which is huge for a large data set, the optimization might be difficult and time consuming. Second, this training technique is very likely to induce over-fitting. We now relegate the discussion of the optimization issue to the next section and address in detail our method for solving the over-fitting problem here.

It is justified to assume that the weights $\mathbf{w}_n$ and $\mathbf{w}_m$ are expected to be similar if the training samples $\mathbf{x}_n$ and $\mathbf{x}_m$ are close to each other in the feature space. We therefore adopt an aggregated regularization term in order to prevent over-fitting. A nearest neighbor graph G is pre-computed (once and for all for the training data) and stored, with $G(n, m) = 1$ if x_n and x_m are neighbors (appropriately symmetrized) and $G(n, m) = 0$, otherwise. We regularize our objective function using $\lambda \sum_{n,m=1}^{N} G(n, m) Dist(\mathbf{w}_n, \mathbf{w}_m)$.

Assume we obtain a set of weights for combining the weak learners in the training stage. In the testing phase, a filtering procedure is imposed in order to construct a strong learner based on the test data and to overcome over-fitting. Instead of using all the training results, we only use the learned weights that are associated with the training patterns which are in the nearest neighborhood of the test pattern. Formally, let x be an incoming test instance. We then take into account a set of nearest neighbors of x from the training data set X. Moreover, each neighbor is assigned a different weight according to its distance to the instance x. We choose the weight to be inversely proportional to the distance which accords with the assumption that the training data being more similar to the test sample x should have more contribution to the final strong classifier for that particular x.

The final hypothesis H of CAVIAR for a test sample x is the weighted combination of the T weak hypotheses weighted by the K nearest neighbors' contribution, that is, $H(x) = sign(\sum_{k=1}^{K} \alpha_{s_k} \sum_{t=1}^{T} w^*_{s_k t} h_t(x))$. Here $w^*_{s_k t}$ is the optimal weight obtained from the training. The pseudo code of 2-class CAVIAR algorithm is listed in Algorithm 1.

Note that any distance measure can be used to define the objective function in step 3 of the training stage. The most popular one is the L2 distance which leads to a closed form solution in the optimization.

Algorithm 1 CAVIAR : for 2 classes

Training stage:

1: **Input** N labeled training samples $\langle (x_1, y_1), \ldots, (x_N, y_N) \rangle$, where $y_i \in \{-1, 1\}$ and T weak learners $h_1, \ldots, h_T$, $h_t : X \to [-1, 1]$

2: **Initialize** the weight matrix: $W = \{w_{nt}\}$, for $n = 1, 2, \ldots, N$ and $t = 1, 2, \ldots, T$

3: **Minimize** the following objective function:

$$W^* = \arg\min_{\mathbf{w}} \sum_{n=1}^{N} Dist(\mathbf{w}_n \cdot \mathbf{h}(x_n), y_n) + \lambda \sum_{n,m=1}^{N} G(n, m) Dist(\mathbf{w}_n, \mathbf{w}_m) \quad (1)$$

Testing stage:

1: **Input** the test sample x

2: **Compute** the nearest neighbors of x: $x_{s_1}, x_{s_2}, \ldots x_{s_K} \in X$, attained from X within the distance threshold $\bar{d}$

3: **Assign** weights to the chosen training samples using:

$$\alpha_{s_k} = \frac{\exp(\beta d_{s_k})}{\sum_{k=1}^{K} \exp(\beta d_{s_k})} \quad (2)$$

where $d_{s_k} = Dist(x, x_{s_k})$

4: **Output** the strong hypothesis

$$H(x) = sign(\sum_{k=1}^{K} \alpha_{s_k} \sum_{t=1}^{T} w^*_{s_k t} h_t(x)) \quad (3)$$

Next, we show how to *generalize this algorithm to the multi-class classification case*, where each training sample corresponds to a particular class label from the set of integers $Y = \{1, 2, \ldots, C\}$, and C is the number of classes. In our approach, we define a C-dimensional label vector $\mathbf{L}_n$ for each training sample x_n, with the c^{th} entry being 1 only if the label of x_n is c and the remaining entries being -1. Meanwhile, the weak learner h_t in this case is a vector function, which maps an instance into a C-dimensional vector space with a positive entry indicating that the data belong to that class, negative otherwise and the magnitude being the confidence of the prediction. The final hypothesis will assign class c to the test data if the c^{th} entry has the maximum positive value in the following vector $\sum_{k=1}^{K} \alpha_{s_k} \sum_{t=1}^{T} w^*_{s_k t} h_t(x)$. Therefore, the objective function is as follows,

$$W^* = \arg\min_{\mathbf{w}} (\sum_{n=1}^{N} Dist(\mathbf{w}_n \cdot \mathbf{h}(x_n), \mathbf{L}_n)) + \lambda \sum_{n,m=1}^{N} G(n, m) Dist(\mathbf{w}_n, \mathbf{w}_m). \quad (4)$$

We do not present the detailed pseudo code here since the whole structure of the algorithm is similar to the 2-class case.

3. OPTIMIZATION

In this section, we briefly discuss the optimization technique and derive the closed form solutions for both 2-class and multi-class algorithms when using the L2 distance.

By using the L2 distance as the distance measure in the 2-class objective function, we obtain

$$E = \sum_{n=1}^{N} ||\mathbf{w}_n \cdot \mathbf{h}(x_n) - y_n||^2 + \lambda \sum_{n,m=1}^{N} G(n,m)||\mathbf{w}_n - \mathbf{w}_m||^2.$$

Expanding the right hand side of the equation and adopting the following notations, we get: Let $H_n = \mathbf{h}(\mathbf{x}_n)\mathbf{h}^t(\mathbf{x}_n)$ and denote the column vector of W as $W^t = [\mathbf{w}_1^t, \mathbf{w}_2^t, \ldots, \mathbf{w}_N^t]$, the matrix B_k as $B_k = H_k + 2\lambda(\sum_{n=k,m\neq 1} G(n,m) + \sum_{n\neq 1,m=k} G(n,m))I_{T\times T}$ and the column vector b as $b^t = [y_1\mathbf{h}^t(\mathbf{x}_1), y_2\mathbf{h}^t(\mathbf{x}_2), \ldots, y_N\mathbf{h}^t(\mathbf{x}_N)]$.

The cost function can be re-arranged into a matrix form as follows:

$$E = W^t \begin{pmatrix} B_1 & \ldots & -2\lambda G(1,N)I_{T\times T} \\ -2\lambda G(2,1)I_{T\times T} & \ldots & -2\lambda G(2,N)I_{T\times T} \\ \ldots & \ldots & \ldots \\ -2\lambda G(N,1)I_{T\times T} & \ldots & B_N \end{pmatrix} W - 2b^t W + \sum_{n=1}^{N} y_n^2.$$

Taking the derivative of E w.r.t. W and setting the equation to 0, we have, $\frac{\partial E}{\partial W} = (D^t + D)W - 2b^t = 0$, with D being the matrix in the equation above that contains B_k as diagonal. The problem is finally reduced to solving the following linear system $(D^t + D)W = 2b^t$.

For the multi-class case, assume there are C classes, for each weak learner h_t, the output is a C dimensional vector denoted as $[h_{t1}(x_n), \ldots, h_{tC}(x_n)]$. We re-arrange those T C-dimensional vectors by taking the corresponding c^{th} entry of each and stacking them in to one vector, and get $\mathbf{h}_c(x_n) = [h_{1c}(x_n), \ldots, h_{Tc}(x_n)]$. Recall that the true label for each data is a C dimensional vector. Similar to the 2-class case, we change some notations and define $H_n = \mathbf{h}_1(\mathbf{x}_n)\mathbf{h}_1^t(\mathbf{x}_n) + \mathbf{h}_2(\mathbf{x}_n)\mathbf{h}_2^t(\mathbf{x}_n) + \cdots + \mathbf{h}_C(\mathbf{x}_n)\mathbf{h}_C^t(\mathbf{x}_n)$, which is a $T\times T$ matrix, $B_k = H_k + 2\lambda(\sum_{n=k,m\neq 1} G(n,m) + \sum_{n\neq 1,m=k} G(n,m))I_{T\times T}$ and $b_c^t = [y_1\mathbf{h}_c^t(\mathbf{x}_1), y_2\mathbf{h}_c^t(\mathbf{x}_2), \ldots, y_N\mathbf{h}_c^t(\mathbf{x}_N)]$.

A similar closed form solution as in the 2-dimensional case is obtained, one that requires us to solve the linear system $(D^t + D)W = 2(b_1^t + b_2^t + \cdots + b_C^t)$.

Note that in both cases, $D^t + D$ are sparse matrices. Finally, the overall optimization reduces to one basic problem: solving a sparse linear system $Ax = b$ [4] with A being $D^t + D$ in the previous equations. Sparse Cholesky factorization is used when A is a positive definite matrix. Since the positive definite property is not guaranteed for small λ, we resort to the LDL^t factorization when A is indefinite.

4. EXPERIMENTS

In this section, we empirically validate our proposed algorithm by classifying the OASIS MRI database into the constituent classes namely, young (Y), old (O), middle aged (M), control and very mild to moderate Alzheimer disease (AD).

4.1. Feature Selection

In addition to the behavioural assessments and cognitive tests, a morphological marker for the Alzheimer disease is the enlargement of ventricles and the shrinkage of cortex and hippocampi. In [5], Mert *et al.* revealed the structural changes in the brain across different age groups and between the healthy and patients with dementia. This led us to the hypothesis that a good feature might be one that captures the structural differences among the brains. Therefore, we constructed the 3D histogram of the deformation field (vectors in the 3D space) required to co-register an emerging atlas to a sample MR brain scan as our feature. This was achieved by performing a group wise registration [6] of the MR images within the OASIS data set. The number of bins in each direction was set to $(6 \times 6 \times 6)$ for constructing the histograms of the vectors.

4.2. Weak Learners

In order to demonstrate the power of CAVIAR, we choose the most simple weak learners by randomly selecting a component from the feature vector and picking a random threshold. The samples were assigned to particular classes based on their values in that chosen component of the feature vector by comparing to the threshold. For instance, assume the data x has a d-dimensional feature. The weak learner $h(x)$ assigns the data to class 1 if the k^{th} dimension (randomly chosen) of the feature vector is larger than a random value t, and assigns to class 2 otherwise.

4.3. Model Selection

The free parameters involved in our CAVIAR algorithm include the regularization parameter λ, the number of nearest neighbors K in setting up the nearest neighbor graph G, the weighting parameter β and the distance threshold $\bar{d}$. The experiments indicate that only $\bar{d}$ is crucial for CAVIAR's performance, and requires tuning with the remaining parameters fixed *a priori*, without sacrificing performance. The parameter λ is used to tune the amount of similarity of the weights $\mathbf{w}_n$ and $\mathbf{w}_m$ corresponding to the nearest neighbors $\mathbf{x}_n$ and $\mathbf{x}_m$. CAVIAR works well in the region $\lambda \in (0, 1)$ and we set it to be 0.01 in the experiments. Empirically, K is chosen

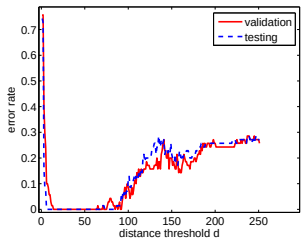

Fig. 1. The classification errors for both validation and test data sets w.r.t. different $\bar{d}$ (indicated by the bins on the x-axis) for Middle aged vs. Old.

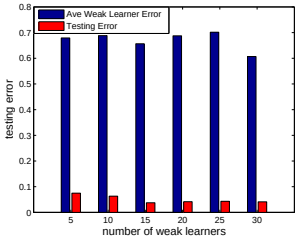

Fig. 2. The classification errors of the weak learners and the final strong hypothesis w.r.t. different number of weak learners for AD vs. Control.

to be approximately 5% of the number of the training data. β is used to adjust the weights of the chosen training data as in Eqn.(2). A setting of $\beta = 0$ implies that the data are equally weighted and a $\beta = \infty$ corresponds to a single nearest neighbor choice. Stable performance is achieved when β is inversely proportional to the average distance among the samples.

The *most crucial* parameter for this algorithm is the distance threshold $\bar{d}$. Note that Euclidian distance is used as our distance measure here. A larger than necessary $\bar{d}$ means more training data are involved in the testing which results in overfitting. Meanwhile a smaller than necessary $\bar{d}$ leads to a small portion of training results being used which pushes the algorithm closer to being a nearest neighbor method. Therefore, a proper choice of $\bar{d}$ is of great importance. To this end, we discretize the search space for $\bar{d}$ into l bins and this bin number only affects the resolution of the search. By searching over the l different $\bar{d}$ values and compute the error, we find the best $\bar{d}$ for each data set. The optimization curves of the classification errors w.r.t. $\bar{d}$ (indicated by l bins) for both the validation and test data are shown in Fig.1. We obtain the best $\bar{d}$ according to the validation error curve and use this $\bar{d}$ in the testing stage.

4.4. Experimental Results

The OASIS data set contains cross-sectional collection of 416 subjects aged 18 to 96. We divided the subjects into three groups, with ages below 40 designated as young, above 60 as old and the ones in between as middle aged. Among the old people, we took 70 subjects, and 35 of them were diagnosed with very mild to moderate AD while the rest were controls. For each experiment, we randomly divided the data set into 5 portions and took 4 of them for training and 1 for testing, with 1 of the training data selected as a validation set. For different numbers of weak learners, the experiments were repeated 20 times and the averages were taken as the result to be reported.

We first classified the data set between any two age groups using the $2-$class version of CAVIAR and then demonstrated the performance of our multi-class classification algorithm by classifying the three age groups simultaneously. Finally, we presented a more serious challenge to our algorithm by classifying the healthy and the very mild to mild AD patients. For comparison, the results of the Adaboost algorithm with the same weak learner settings are also reported.

In Table 1, we show the average error for the weak learners in the second column and the test error of the final strong hypothesis in the third column, followed by the improvement of the performance of our algorithm w.r.t. the weak learners, and reported the test error for Adaboost in the last column. The error is measured by the mis-classification rate.

To illustrate the change of performance w.r.t. the increasing number of weak learners, we show the experimental results of AD vs. Control in Figure.2.

Table 1. Testing results with 20 weak learners(WL)

	Ave. WL Err	Test Err	Improve	Ada. Err
Y vs. M	0.3220	**0.0233**	92.76%	0.0400
M vs. O	0.6746	**0.0164**	97.57%	0.0320
O vs. Y	0.4770	**0.0086**	98.19%	0.0125
Y vs. M vs.O	0.7925	**0.0375**	95.27%	0.0912
AD vs. Control	0.6876	**0.0417**	93.94%	0.0975

The experimental results indicate that the deformation field captures the structural changes across the brains very well and the CAVIAR algorithm significantly improves the performances of the weak classifiers, when presented with a small number of simple weak learners in both 2-class and multi-class cases.

5. REFERENCES

[1] D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris and R. L. Buckner, "Open Access Series of Imaging Studies (OASIS): Cross-Sectional MRI Data in Young, Middle Aged, Nondemented, and Demented Older Adults," *Journal of Cognitive Neuroscience*, vol. 19(9), pp. 1498-1507, 2007.

[2] L. Breiman, "Bagging Predictors," *Machine Learning*, vol. 24(2), pp. 123-140, 1996.

[3] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting," *Journal of Computer and System Sciences*, vol. 55, pp. 119-139, 1997.

[4] I. S. Duff, "MA57-A Code for the Solution of Sparse Symmetric Definite and Indefinite Systems," *ACM Trans. Math. Software*, vol. 30(2), pp. 118-144, 2004.

[5] M. R. Sabuncu, S. K. Balci, M. E. Shenton and P. Golland, "Image-Driven Population Analysis Through Mixture Modeling," *IEEE Transactions on Medical Imaging*, vol. 28(9), pp. 1473-1487, 2009.

[6] S. Joshi, B. Davis, M. Jomier and G. Gerig, "Unbiased Diffeomorphic Atlas Construction for Computational Anatomy," *NeuroImage*, vol. 23, pp. S151-S160, 2004.