Lossless Online Ensemble Learning (LOEL) and Its Application to Subcortical Segmentation*

Jonathan H. Morra, Zhuowen Tu, Arthur W. Toga, and Paul M. Thompson

Laboratory of Neuro Imaging, UCLA School of Medicine, Los Angeles, CA, USA

Abstract. In this paper, we study the classification problem in the situation where large volumes of training data become available sequentially (online learning). In medical imaging, this is typical, e.g., a 3D brain MRI dataset may be gradually collected from a patient population, and not all of the data is available when the analysis begins. First, we describe two common ensemble learning algorithms, AdaBoost and bagging, and their corresponding online learning versions. We then show why each is ineffective for segmenting a gradually increasing set of medical images. Instead, we introduce a new ensemble learning algorithm, termed Lossless Online Ensemble Learning (LOEL). This algorithm is lossless in the online case, compared to its batch mode. LOEL outperformed online-AdaBoost and online-bagging when validated on a standardized dataset; it also performed better when used to segment the hippocampus from brain MRI scans of patients with Alzheimer's Disease and matched healthy subjects. Among those tested, LOEL largely outperformed the alternative online learning algorithms and gave excellent error metrics that were consistent between the online and offline case; it also accurately distinguished AD subjects from healthy controls based on automated measures of hippocampal volume.

1 Introduction

The fields of data mining and biomedical engineering have recently seen a vast increase in the amount of available data. Ongoing medical imaging studies commonly analyze images from hundreds or even thousands of patients, sometimes scanned at multiple time-points. Many brain MRI studies focus on one particular brain region (hippocampus, caudate, etc.), and a first step in studying these structures is finding them on brain MRI, using a classification or segmentation algorithm. Boosting [1] and bagging [2] algorithms and their extensions have shown great promise for effective classification of voxels in images [3,4,5], but it is not always straightforward to create a training dataset for the algorithm

* This work was funded by the National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 RR021813 entitled Center for Computational Biology (CCB), the National Institute for Biomedical Imaging and Bioengineering, the National Center for Research Resources, National Institute on Aging, the National Library of Medicine, and the National Institute for Child Health and Development (EB01651, RR019771, HD050735, AG016570, LM05639).

to learn which features are relevant for classification. In studies where data acquisition is ongoing (such as the Alzheimer’s Disease Neuroimaging Initiative, ADNI [6], which scans 800 subjects every 6 months for 3 years), one may wish to begin to use a segmenter after each set of scans becomes available; in other applications, one may not have access to all previous scans used to train the algorithm in the past, or have time to retrain. In either case, an online algorithm is desirable. In imaging studies, the set of relevant features for classification may lie in a very high-dimensional space, so the algorithm must be able to use this information in a reasonable amount of time. Ensemble learning methods (such as bagging and boosting) are good candidate classifiers for combining information from thousands of potentially useful features: they combine weak classifiers - which individually may perform only slightly better than chance - to create a strong classifier that outperforms all of the component classifiers. These learning algorithms can be very effective for image segmentation as they can “select” important features, and overlook unimportant ones.

In such a situation, training data may arrive as a sequence of image sets, e.g., 100 volumes at a time. For example, in multi-site drug trials or longitudinal studies such as ADNI [6], it is vital to begin data analysis as soon as possible, while benefiting from the increasing pool of available scans. The original formulations of boosting and bagging required all training data to be available before training could begin; this is known as batch learning. The recently-developed online versions of bagging and boosting [7] have drawbacks because the algorithm focuses on updating the weights based on the sampled data each time, to simulate the batch training mode on a fixed set of weak learners. By just focusing on updating the weights (in boosting), and selecting the training samples (in bagging), these methods overlook the need to select appropriate features automatically, which is vital in image segmentation applications.

In this paper, we first use Oza’s versions of online boosting and online bagging and then introduce our new algorithm, LOEL, which extends the idea of online learning to medical image segmentation. LOEL is lossless and outperforms both boosting and bagging in the online case, and is comparable to both in the offline case.

2 Methods

2.1 Problem and Previous Work

When segmenting brain structures in 3D MRI scans, one seeks to assign each voxel to one or more regions of interest (ROI). Here we focus on the two-class case and, without loss of generality, we will study the segmentation of the hippocampus, a structure that degenerates in Alzheimer’s disease and is a target of interest in ongoing drug trials. Let $X = (x_1 \cdots x_N)$ be all the voxels in the manually labeled testing set and $Y = (y_1 \cdots y_N)$ be the ground truth labels for each x_n , such that $y_n \in [-1, 1]$. From discriminative learning point of view, we seek a classifier to minimize the error $e = \sum_i |y_i - F(X_{N_i})|$ where X_{N_i} is an image patch centered at voxel i . An ensemble learner essentially combines the outputs

of weak learners, which can be based upon a feature pool. Each weak learner h_n , which takes in X and outputs Y , pushes the overall solution towards the optimal solution. Combining these weak learners is the function of the specific ensemble algorithm. For instance in AdaBoost $F(X_{N_i}) = \sum \alpha_j h(X_{N_i})$, and in bagging $F(X_{N_i}) = \sum h(X_{N_i})$.

2.2 Background

To optimally combine a set of different weak learners from a feature pool, most ensemble methods either re-weight or iteratively modify the training data presented to each weak learner, bias the weak learners in some way, or both. Boosting (and its variants) can create a highly effective ensemble classifier as it keeps updating a weight w_i over the training examples in X to force weak learners to focus on difficult examples. While this is effective, it is not the only way to perturb the data. Bagging repeatedly resamples the training data, with replacement. After each resampling, a weak learner is created based on the resampled data, and the average prediction over all weak learners defines the strong learner. This resampling provides enough variation in the data to make each weak learner $h_n(X)$ different enough to contribute to the classification problem. Random forests [8] both resample the data and limit the search space from which to construct each $h_n(X)$. This provides randomization on both the dataset and the feature space. Even the extreme case of randomization has shown success, where extremely randomized trees [9] allow only one feature for each split of the tree and randomize the cut point of that feature.

2.3 Online Learning

First, we must use weak learners that can take advantage of sequentially presented training data. Unless the base learning algorithm can use data presented online, online ensemble learning becomes very difficult. For this paper, we will use both decision stumps and a modified 1-deep decision tree (explained later) that are both lossless in the online case. To coax an ensemble method into an online mode, the weak learner selection method must be changed.

To adapt bagging for online training, we use Oza's method [7] to sample the training data with replacement as more training data become available. In offline bagging, each sample is presented to each weak learner from 0 to N times, where N is fixed before learning. The number of times, K , that a sample is presented to each weak learner, may be modeled as a binomial distribution $P(K = k) = \binom{N}{k} \left(\frac{1}{N}\right)^k \left(1 - \frac{1}{N}\right)^{N-k}$. In the online case, we can view $N \rightarrow \infty$, and then the binomial distribution tends to a Poisson distribution with mean 1, $P(K = k) = \exp(-1)/k!$. Given a set of weak learners, running bagging in the online case is therefore equivalent to batch bagging. The online bagging algorithm is described in Fig. 1.

For online boosting, the algorithm is slightly more complicated. Again, we follow Oza's idea [7], which switches the roles of feature selecting and example gathering. Once a set of weak learners is obtained, the "difficulty" of an example

Given a set of weak learners H and a new example $(\mathbf{x}, y)$
 For each weak learner $h_n \in H, n = 1, \dots, N$
 • Set $p = \text{rand}(0, 1)$
 • while $p > e^{-1}$
 • Update classifier h_n with example $(\mathbf{x}, y)$
 • $p = p * \text{rand}(0, 1)$

Fig. 1. The online bagging training procedure from Oza [7]

Initialize $\lambda_n^{sc} = 0, \lambda_n^{sw} = 0$ before seeing any training examples.
 Given a set of weak learners H and a new example $(\mathbf{x}, y)$, set $\lambda = 1$
 For each weak learner $h_n \in H, n = 1, \dots, N$
 • Update h_n with example $(\mathbf{x}, y)$ and weight λ
 if $y = h_n(\mathbf{x})$ else
 • $\lambda_n^{sc} = \lambda_n^{sc} + \lambda$ • $\lambda_n^{sw} = \lambda_n^{sw} + \lambda$
 • $\epsilon_n = \frac{\lambda_n^{sc}}{\lambda_n^{sc} + \lambda_n^{sw}}$ • $\epsilon_n = \frac{\lambda_n^{sw}}{\lambda_n^{sc} + \lambda_n^{sw}}$
 • $\lambda = \lambda \frac{1}{2(1-\epsilon_n)}$ • $\lambda \frac{1}{2\epsilon_n}$

Fig. 2. The online boosting training procedure. λ_n^{sc} keeps track of the correctly classified examples, and λ_n^{sw} keeps track of the incorrectly classified examples per weak learner. λ attempts to model the weights w from batch AdaBoost.

is estimated by having each weak learner classify it, and then updating the weak learner and its weight based on the difficulty of that example. Online boosting is described in Fig. 2.

By Oza's own admission, the online boosting algorithm is not lossless compared to its batch mode; an online learning algorithm is lossless if it returns a model identical to that of the corresponding batch algorithm trained on the same examples. This can best be seen by example. Assume that weak learner n sees an example and correctly classifies it. That example's weight will decrease when weak learner $n + 1$ classifies it. Based on its decreased weight weak learner $n + 1$ may or may not classify it correctly. Then another example is presented to weak learner n , and it is updated such that it no longer correctly classifies the previously seen example. This means that the weight assigned for the previous example was incorrect, and it should have been assigned a higher weight when given to weak learner $n + 1$ (and all $m > (n + 1)$ weak learners). In addition to this drawback, online boosting does not lend itself to feature selection. Although Grabner [10] has shown a way to induce online boosting into feature selection, it still suffers a drawback in that it is lossy versus its batch mode. Additionally, Fern [11] has produced an online ensemble learning method based on Arc-x4, however it too suffers from the inability to generate new weak learners.

2.4 LOEL

The two main drawbacks of online bagging (no feature selection) and online boosting (it is lossy) make them less than ideal for online ensemble learning. We

Given a pool of weak learners $\mathbf{H}$ and a new example $(\mathbf{x}, y)$
 For each weak learner $h_n \in \mathbf{H}, n = 1, \dots, N$
 • Update h_n with example $(\mathbf{x}, y)$
 For $t = 1 \dots T$
 • Create a new feature pool, $\mathbf{H}_{\text{sm}}$, of size S by random sampling from $\mathbf{H}$
 For $s = 1 \dots S$
 • Randomize the parameters of h_s
 • $\epsilon_s = \text{error}(h_s)$
 • Add $h^* = \text{argmin}_{\mathbf{H}_{\text{sm}}} \epsilon_s$ to the model

Fig. 3. The LOEL training procedure. Test data are classified using an unweighted vote of each h^* . For all experiments in this paper, we set $\text{size}(\mathbf{H}_{\text{sm}}) = 0.3\text{size}(\mathbf{H})$.

overcame these limitations with the Lossless Online Ensemble Learning (LOEL) algorithm (Fig. 3).

The first for loop in LOEL is the weak learner updating loop, where the next example is added to all weak learners without weighting. Any weak learner can be used with LOEL, so long as it (1) is lossless, and (2) has a compact way of storing examples that is independent of the number of examples seen. An example of the second caveat is the decision stump. By storing two histograms of the data already seen (a histogram of the positive data, and a histogram of the negative data), each weak learner can keep track of the examples it has seen, independently of the total number of examples. By transforming these histograms into cumulative distribution functions, the error for a threshold can be estimated quickly, and in order to randomize the parameters, we just randomly choose a cut point.

Because each example is given exactly once to each weak learner, perturbations must be made on the weak learners themselves to differentiate between each run of the interior loop of Fig. 3. These perturbations were borrowed from both random forests (restricting the weak learner space) and from extremely randomized trees (randomizing the parameters of each weak learner) and as such should prove effective in LOEL.

Following the logic of Breimann [8], we can show that as more weak learners are added to the classifier we are fitting a more effective model. If $I(h_n(X) = Y)$ is an indicator function, then $\text{margin}_N(X, Y) = \sum_{n=1}^N I(h_n(X) = Y) - I(h_n(X) \neq Y)$. The margin is the confidence in each sample; increasing margin means a sample is more likely in a given class. We can then define the optimal classifier as $Y_N^* = P(\text{margin}_N(X, Y) < 0)$. In LOEL, $h_n(X) = h_n^{s,p}$ where s is the size of the resampled feature pool, and p are the randomized parameters. Then, following the law of large numbers $P_{X,Y}(P(h^{s,p}(X) = Y) - P(h^{s,p}(X) \neq Y) < 0)$.

LOEL is provably lossless as it is very similar to its batch version. The only difference between batch and online modes is the existence of the first loop in Fig. 3. So long as the weak learners conform to the specifications above, the construction of the weak learners will be no different in batch versus online modes. In fact, in online learning, the model that is output at each iteration is not even used when the next training example is presented, the compact representation of the whole weak learner pool is instead stored and updated.

This is in contrast to the online bagging and boosting methods where the model itself is updated. By having access to the entire feature pool and all examples already seen, LOEL can select weak learners that minimize the error over all weak learners and all examples.

3 Results

3.1 Tests on Standardized Data

We first compared AdaBoost, bagging, and LOEL in the offline case on a standard dataset from the UCI machine learning repository (archive.ics.uci.edu/ml/). We chose to use the breast cancer data because it presented a medical imaging two-class classification test. For this dataset we only aimed to show that LOEL is as effective as both AdaBoost and bagging in the offline case. We will reserve the online case to the full hippocampal segmentation task.

Fig. 4 shows how testing error varies as the number of weak learners grows. The error is defined as the number of incorrect examples divided by the total number of examples. For this test, we defined a weak learner as a decision stump, and randomly chose half the data set for training and half for testing. All three methods perform well on this data, with LOEL best at minimizing the testing error.

Fig. 4. Effects of varying the number of weak learners as a function of the testing error on a standard breast cancer dataset. LOEL outperforms both AdaBoost and bagging when minimizing error on the test set.

3.2 Hippocampal Segmentation

To apply LOEL to a real imaging problem, we segmented the hippocampus in a dataset from a study of Alzheimer’s disease (AD) which significantly affects the morphology of the hippocampus [12]. This dataset includes 3D T1-weighted brain MRI scans of individuals in three diagnostic groups: AD, mild cognitive impairment (MCI), and healthy elderly controls. All subjects were scanned on a 1.5 Tesla Siemens scanner, with a standard high-resolution spoiled gradient echo (SPGR) pulse sequence with a TR (repetition time) of 28 ms, TE (echo time) of 6 ms, 220mm field of view, 256x192 matrix, and slice thickness of 1.5mm. For training we used 20 subjects in a variety of disease states (AD, MCI, or Normals).

Given the ground truth segmentation (A) and an automated segmentation (B), both represented as binary sets, we define the following error metrics; $d(a, b)$ is the Euclidean distance between 2 points, a and b :

- Precision = $\frac{A \cap B}{B}$
- Recall = $\frac{A \cap B}{A}$
- Relative Overlap = $\frac{A \cap B}{A \cup B}$
- Mean = $\text{avg}_{a \in A}(\min_{b \in B}(d(a, b)))$

Fig. 5. Error metrics used to validate hippocampal segmentations

Fig. 6. Hippocampal segmentation errors as a function of the number of training brains. Offline training is the upper row and online, the lower row. The f-value is the average of precision and recall. In the offline case, all three methods show improvement as more training data is added. AdaBoost is not as effective in the offline case at minimizing the mean distance. In the online case, both bagging and LOEL show similar trends in each category. However, for bagging, this can be attributed to always choosing the prior feature. Since the prior improves as more brains are added the error metrics also tend to improve. The volatility of online AdaBoost arises from the fact that it is lossy.

For testing we used an independent set of 80 subjects (40 AD, 40 Normals). To assess segmentation accuracy, we present results that varied the number of training brains while testing on all 80 subjects. For the tests in this section, we slightly changed our weak learner formulation. Instead of a weak learner being a decision stump, it is instead a 1-deep modified decision tree. Some of our features are based on a “prior” image, which we define as the pointwise average of all the available training masks, which takes values in the range $[0,1]$. The features based on this image are so strong that they tend to overpower the other features. To provide more balance, we define the first level of the decision tree to be “is the prior less than 0.2.” This gives other features more prominence during the construction of weak learners. This formulation of the weak learners still follows the rules set out by the LOEL requirements because the root node is hardcoded. Instead of storing a positive and negative histogram per weak learner, we store

Table 1. This table shows p -values comparing the mean hippocampal volume of 40 AD vs. 40 normal subjects. Online bagging was not effective because every subject had the exact same hippocampal volume. Both AdaBoost and LOEL correctly distinguish AD from normal, but as shown by Fig. 6, AdaBoost is too volatile in the online case. Without the ability to make such a well known differentiation, a segmentation is not accurate enough for real use.

	AdaBoost	Bagging	LOEL
Offline	0.00023	0.034	0.028
Online	0.022	NA	0.0027

4 histograms, a positive and negative histogram for examples in which the prior is less than 0.2 and the same for examples in which the prior is greater than 0.2.

Fig. 6 shows our results in both the offline and online cases. In the offline case, all three methods are effective after 6 or 7 brains have been used for training. In the online case, AdaBoost is quite volatile, whereas both bagging and LOEL gradually improve with more training brains. Bagging and LOEL are close in the online case, but error metrics only tell half the story. Because the prior is such a good feature, by just choosing the prior for each weak learner, bagging is able to keep up with LOEL. Table 1 shows results of a 2-sample t -test comparing the mean hippocampal volume of AD to normal subjects. Each algorithm correctly differentiates AD from normal in the offline case, but LOEL and AdaBoost are the only algorithms that also do so in the online case. Bagging is just returning the prior as it cannot make any distinction between brains in the online case. AdaBoost can distinguish AD from normal, but the error metrics show that online AdaBoost is too volatile to be effective.

4 Conclusion

In this paper we developed a new ensemble learning method that is lossless in the online case. While this algorithm is not better than boosting or bagging in the offline case, it outperformed both of them in the online case. In the future, we hope to apply LOEL to more classification tasks to see if it generalizes well to other imaging problems and other domains.

References

1. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer Sys. Sci.* 55, 119–139 (1997)
2. Breiman, L.: Bagging predictors. *Machine Learning* 24(2), 123–140 (1996)
3. Morra, J., Tu, Z., Apostolova, L., et al.: Mapping hippocampal degeneration in 400 subjects with a novel automated segmentation approach. In: *ISBI*, May 2008, pp. 336–339 (2008)
4. Tu, Z., Narr, K., Dollar, P., et al.: Brain anatomical structure parsing by hybrid discriminative/generative models. *IEEE TMI* (2008)

5. Rohlfing, T., Maurer, C.R.: Multi-classifier framework for atlas-based image segmentation. *Pattern Recognition Letters* 26(13), 2070–2079 (2005)
6. Jack, C., Bernstein, M., Fox, N., et al.: The Alzheimer's Disease Neuroimaging Initiative (ADNI): The MR imaging protocol. *Journal of MRI* 27(4), 685–691 (2008)
7. Oza, N.: Online bagging and boosting. *IEEE Systems, Man, and Cybernetics* 3, 2340–2345 (2005)
8. Breiman, L.: Random forests. *Machine Learning* 45(1), 5–32 (2001)
9. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. *Machine Learning* 63(1), 3–42 (2005)
10. Grabner, H., Bischof, H.: On-line boosting and vision. In: *CVPR*, vol. 1, pp. 260–267 (2006)
11. Fern, A., Givan, R.: Online ensemble learning: An empirical study. In: *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 279–286. Morgan Kaufmann, San Francisco (2000)
12. Becker, J., Davis, S., Hayashi, K., et al.: 3D patterns of hippocampal atrophy in mild cognitive impairment. *Archives of Neurology* 63(1), 97–101 (2006)