A cost constrained boosting algorithm for fast lesion detection and segmentation

Arne Militzer[a,b], Christian Tietjen[b], Joachim Hornegger[a]

[a]Pattern Recognition Lab, Department of Computer Science, Friedrich-Alexander-University Erlangen-Nuremberg, Martensstr. 3, 91058 Erlangen, Germany

[b]Siemens AG Healthcare, Computed Tomography, Siemensstr. 1, 91301 Forchheim, Germany

ABSTRACT

Machine learning techniques like pointwise classification are widely used for object detection and segmentation. However, for large search spaces like CT images, this approach becomes computationally very demanding. Designing strong yet compact classifiers is thus of great importance for systems that ought to be clinically used as time is a limiting factor in clinical routine. The runtime of a system plays an important role in the decision about its application. In this paper we propose a novel technique for reducing the computational complexity of voxel classification systems based on the well-known AdaBoost algorithm in general and Probabilistic Boosting Trees in particular. We describe a means of incorporating a measure of hypothesis complexity into the optimization process, resulting in classifiers with lower evaluation cost. More specifically, in our approach the hypothesis generation that is performed during the AdaBoost training is no longer based only on the error of a hypothesis but also on its complexity. This leads to a reduced overall classifier complexity and thus shorter evaluation times. The validity of the approach is shown in an experimental evaluation. In a cross validation experiment, a system for automatic segmentation of liver tumors in CT images, that is based on the Probabilistic Boosting Tree, was trained with and without the proposed extension. In this preliminary study, the evaluation cost for classifying previously unseen samples could be reduced by 83% using the methods described here without losing classification accuracy.

Keywords: CT, lesion detection, segmentation, machine learning, boosting, AdaBoost, Probabilistic Boosting Tree, complexity reduction

1. INTRODUCTION

Automatic detection and segmentation of liver tumors in CT images still pose a great challenge for researchers. The huge variability in appearance of both healthy liver tissue and tumors makes their distinction particularly difficult. In this field, machine learning techniques seem most promising as they can adapt to new cases more flexibly than other methods and thus generalize comparably well. Boosting techniques, such as the popular AdaBoost algorithm or its derivate, the Probabilistic Boosting Tree (PBT), have been used particularly successfully in this setting.[1–3]

There are two main approaches to lesion detection by classification: One can either train a classifier to identify image subwindows containing lesions or classify single voxels as belonging to a lesion or not. The former has the slight disadvantage of requiring assumptions about the expected lesion size, as a lesion usually has to lie completely inside a subwindow. The latter is more flexible in this respect, it can in principle handle lesions of arbitrary size and shape. It has, however, the slight drawback that it does not per se result in contiguous detections but rather class information about single points. These points then have to be combined into lesion masks. On the other hand, this provides a segmentation of the lesion with voxel accuracy at the same time. Both approaches have in common that their application to large images is computationally expensive.

In this paper, we present a technique that can reduce the computational complexity of a detection system based on AdaBoost classification considerably without losing quality. Carneiro et al.[4] presented a method for faster PBT training and application that is based on constraining the size of the classifier. Their method is, however, limited to the PBT, whereas we directly aim at the underlying AdaBoost procedure. The methods can thus be used complementarily or completely independently from each other.

Medical Imaging 2011: Computer-Aided Diagnosis, edited by Ronald M. Summers, Bram van Ginneken,
Proc. of SPIE Vol. 7963, 79631B · © 2011 SPIE · CCC code: 0277-786X/11/$18 · doi: 10.1117/12.877944

In the following sections, we describe the segmentation framework in which we developed and evaluated the proposed technique and shortly review the PBT and AdaBoost algorithms, before we present our extension to them. Finally, a comprehensive evaluation is provided that shows the validity of the approach.

2. METHODS

Our system[3] simultaneously detects and segments focal liver lesions in contrast enhanced CT images. To this end, first the liver is extracted using an automatic segmentation technique in order to reduce the search space. Next, intensities inside the liver are standardized to compensate for variations due to acquisition timing or patient specific perfusion differences. A PBT then assigns a lesion probability value to each individual point in the liver. After some post processing, a lesion mask is generated from the probability image by thresholding.

2.1 The Probabilistic Boosting Tree

The PBT is a two-class classifier proposed by Tu in 2005.[5] It has been successfully applied to various challenging classification problems such as polyp detection in virtual colonoscopy CT images,[6] detection of fetal anatomies in ultrasound images,[4] and segmentation of pediatric brain tumors in MR images.[7]

The PBT operates in a divide and conquer manner, resembling a soft decision tree. During learning stage it recursively builds a tree, training an AdaBoost[8] classifier for each node (Fig. 1).

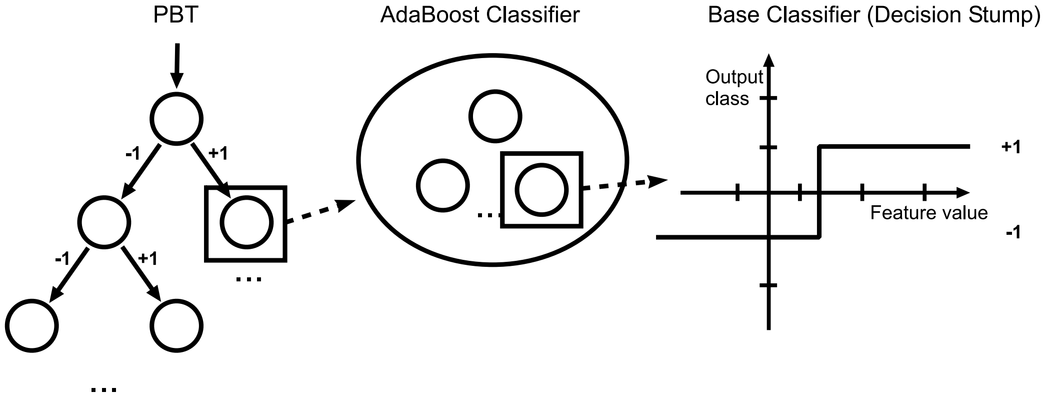


Fig. 1. Structure of the PBT: Each tree node contains an AdaBoost classifier, which consists of several base classifiers called weak learners (here: decision stumps).

AdaBoost constructs a strong classifier by repeatedly calling a weak learning algorithm and combining the hypotheses this algorithm generates. In each iteration, the new hypothesis is incorporated into the strong classifier, where its output is weighted based on its error on the training data. This procedure is repeated until a predefined maximum number of weak learners is reached or the hypothesis error exceeds a certain predefined level.
The resulting strong classifier

$$H(\mathbf{x}) = \sum_{i=0}^{N} \alpha_i h_i(\mathbf{x}), \tag{1}$$

after training $N \in \mathbb{N}$ weak learners h_i with weights $\alpha_i \in \mathbb{R}$, was shown to approach logistic regression[9] for posterior probabilities $p(y|\mathbf{x})$, $y \in \{-1, 1\}$ by

$$H(\mathbf{x}) \approx \frac{1}{2} ln \frac{p(y=+1|\mathbf{x})}{p(y=-1|\mathbf{x})}, \tag{2}$$

allowing the computation of approximate posterior probabilities for a sample $\mathbf{x}$ as

$$q(\pm 1|\mathbf{x}) = \frac{exp(\pm 2H(\mathbf{x}))}{1 + exp(\pm 2H(\mathbf{x}))}. \tag{3}$$

According to these probabilities the training samples are split at the newly trained node into a positive and a negative subset, putting ambiguous samples into both sets. A sample is considered ambiguous if its posterior probability q falls into the range $[\frac{1}{2} - \epsilon, \frac{1}{2} + \epsilon]$ for a user defined ϵ. The samples are then reweighted, putting less emphasis on the ambiguous ones, and the positive and negative subsets are used to train the right and left subtrees of the node. This procedure is repeated until a predefined maximum tree depth is reached.

Splitting the training set effectively divides the input space at each node, making this classifier particularly well suited for problems with high intra-class variance: Nodes high up in the tree have to make only rough decisions, while deeper nodes get to solve very specialized subproblems on smaller portions of the input space.

Applying the PBT to classify a new pattern $\mathbf{x}$ works in analogy to the training. Starting from the root node, the sample is handed down the tree. At each node the AdaBoost classifier calculates its posteriors $q(\mathbf{x})$ and depending on the result the pattern is recursively passed on to the subtrees. If the AdaBoost classifier is very certain about a sample's class, i.e. its probability $q(1|\mathbf{x})$ is outside the range $[\frac{1}{2} - \epsilon, \frac{1}{2} + \epsilon]$, the other class' subtree is not traversed. Finally, at each node the results from the subtrees are combined and returned up the tree, where for omitted subtrees the empirical class distribution from the training data is assumed as result. This simplification does not influence the overall result much, since at each node the subtrees' results are weighted based on the node classifier's posterior probability $q(1|x)$. In case this probability is high (or low) enough to omit a subtree, its result would receive very little weight anyway.

In this fashion the PBT combines the classification results of its internal nodes into the overall approximate posterior distribution $\tilde{p}(y|\mathbf{x})$ at the root node. This probability value can then be used to directly trade off the tree's sensitivity vs. its specificity via a single threshold.

2.2 Constrained Boosting

The PBT algorithm, due to its hierarchical nature, can generate much more compact classifiers than plain AdaBoost, without loss of discriminative power. Also, the mechanism of omitting certain subtrees when classifying a new sample, can yield a considerable speedup. In general, the classifier's complexity can be stated as a function of the number of weak learner evaluations when classifying new samples. There are two ways of further reducing this complexity.

2.2.1 Subtree Pruning

The implicit pruning performed during classifier application by not descending into certain subtrees can save a lot of computation time without losing much classification accuracy. Thus, better strong learners for the tree's nodes, which allow the pruning of more subtrees during application, could yield a reduction in complexity. However, this contradicts the idea of the PBT: The power of this classifier stems from the fact that it is hierarchical, working in a divide and conquer manner. Stronger classifiers in the nodes would mean giving up this concept and approximating a single flat classifier.

2.2.2 Cost Constrained Hypothesis Selection

Instead, we propose to reduce the PBT's overall complexity by modifying the AdaBoost training procedure. AdaBoost iteratively trains a set of weak learners or hypotheses h_t, at each iteration t incorporating the one with the best classification performance into the final classifier H. In our case decision stumps function as weak learners, reducing this process to selecting the single most discriminating feature at each iteration. Simply using less complex features could obviously reduce the overall complexity of the PBT considerably. However, as simpler features often don't have the same discriminative power, we do not want to abandon the more complex ones completely. Instead, we trade off complexity vs. discriminative power. This is achieved by defining a cost for each feature and combining it with the classification error to form a new selection criterion. So, for selecting the next hypothesis h_t for the ensemble, instead of optimizing a criterion based solely on the classification error with respect to the sample weight distribution D_t, such as

$$\psi_t(h) = \sum_{i:h(x_i)\neq y_i} D_t(i), \tag{4}$$

a cost term $c(h)$ is introduced, resulting in the optimization criterion

$$\psi_t(h) = \lambda \cdot c(h) + (1 - \lambda) \cdot \sum_{i:h(x_i) \neq y_i} D_t(i). \tag{5}$$

During PBT training, the weighting factor λ is decreased linearly with increasing tree depth until it reaches 0 on the last tree level. That way in higher nodes simple features are preferred, resulting in very fast classification. At the same time deeper nodes, that have to solve very detailed problems, can make use of the full feature set to generate more complex hypotheses where necessary. Being located deeper in the tree, these complex hypotheses will be evaluated less often, adding less to the overall complexity than hypotheses in higher nodes.

The hypothesis cost can be an arbitrary function defined on the hypothesis space $\mathcal{H}$ with $c : \mathcal{H} \mapsto [0, 1]$, may it be determined analytically, heuristically, or empirically. Also, this approach is not limited to classifiers based on decision stumps. For other types of weak learners, the cost term can be incorporated into the hypothesis generation process in a similar fashion.

Of course, using this combined optimization criterion can in general lead to the selection of suboptimal (with respect to their classification error) hypotheses. However, the only requirement AdaBoost makes for its hypotheses is that they have to be weak learners, i.e. their training errors have to be below 0.5. In the two-class case this can be formally ensured by using an odd number of samples. In the end, AdaBoost may have to generate more hypotheses than without the cost term. But the overall complexity can still be lower, if hypotheses with much lower individual costs are used.

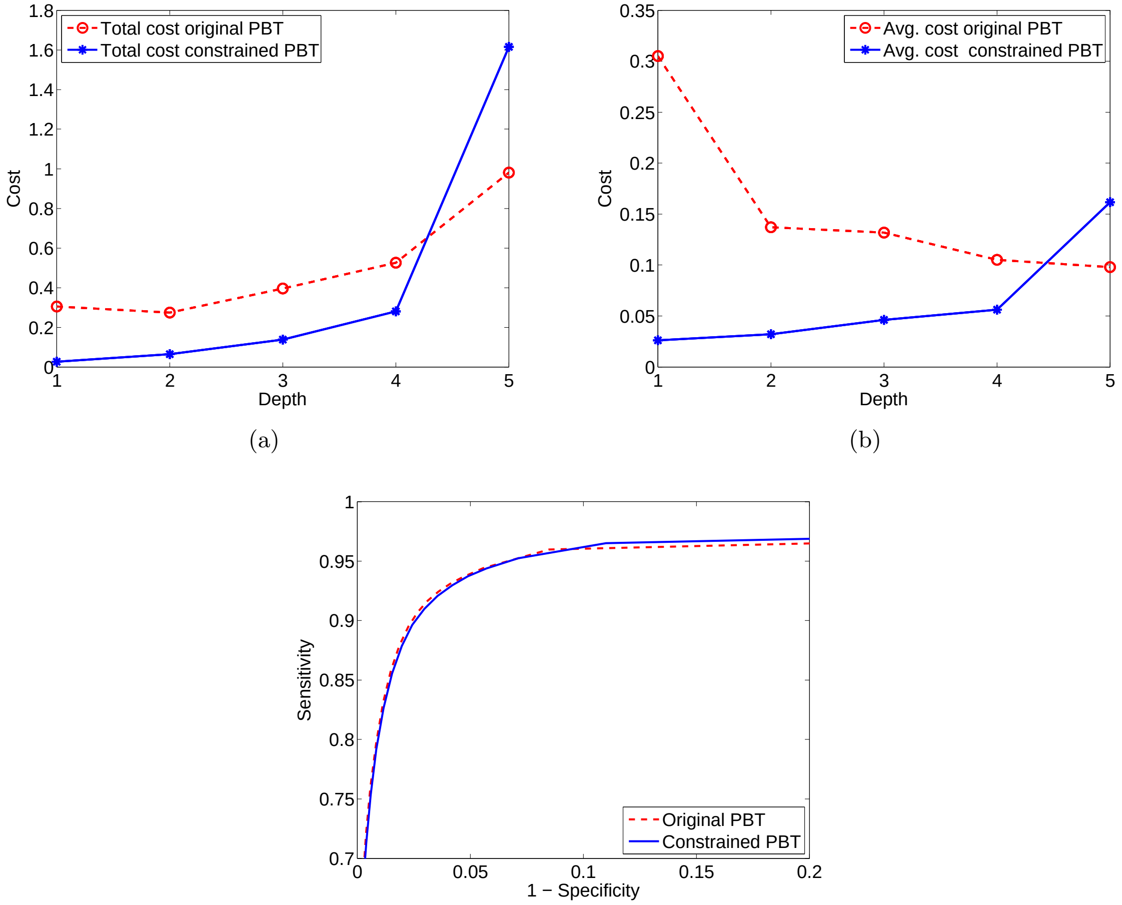
3. RESULTS AND DISCUSSION

The effect of the proposed constrained feature selection process was evaluated in a five fold cross validation experiment on a database of 15 CT datasets showing livers with venous contrast enhancement. All images contained hypodense liver lesions, such as cysts, various metastases or hepatocellular carcinoma. The system's detection and segmentation performance on this database has been evaluated in.[3] Here, we focused on comparing the evaluation costs of classifiers trained with the new feature selection process to those of classifiers trained with the original PBT algorithm.
To determine the hypothesis cost function to be used for optimization we chose a very hands-on approach. Each feature was calculated on the target machine and the elapsed time was measured, averaged over several thousand runs. The resulting values were scaled to the range $[0, 0.5]$. This range was used instead of $[0, 1]$ in order to have the cost values in the same range as the hypothesis errors, although the same effect on the training could be achieved by setting the weighting factor λ accordingly. Here, λ was initially set to 0.5 and then linearly decreased as described in section 2.2.

The overall cost of classifying a new sample with the PBT cannot be determined analytically because of the subtree pruning. Instead, one has to actually apply the PBT to the sample and accumulate the hypothesis costs along the way. This yields the PBT's cost $c_{pbt}(\mathbf{x})$ for a sample $\mathbf{x}$. The overall cost for an image, represented as the set of samples $X = \{\mathbf{x}_1, .., \mathbf{x}_n\}$ calculated from its voxels would then be $C_{pbt}(X) = \sum_{i=1}^{n} c_{pbt}(\mathbf{x}_i)$. In our cross validation experiment each image occurs in the test set once and only once, so we calculated the total evaluation cost of all 15 images in the test database. The experiment was run twice, once with and once without the proposed cost optimization, with all other parameters set identically.

Compared to the standard PBT training, the total evaluation cost could be reduced by 83% by the proposed feature selection method, without losing classification accuracy as can be seen from the receiver operating characteristic (ROC) curve in Fig. 2(c). This curve was generated by varying a threshold on the posterior probability output by the classifier and evaluating the resulting confusion matrix.

Figures 2(a) and 2(b) show, that the feature selection worked as expected: In high nodes, AdaBoost chose simpler (and thus cheaper) features than with the standard PBT version, thus the much higher total and average evaluation costs of the original PBT for depths $1-4$. In deeper nodes, which are rarely evaluated, the cost penalty in the constrained PBT algorithm was decreased. Hence, the AdaBoost procedure focused more on the weak learners' classification error and the average cost increased. The strong increase in cost at depth 5 is explained



(a)

(b)

(c)

Fig. 2. Comparison of classifier accuracies and costs. At each tree level, the cost of all hypotheses is (a) accumulated for one sample and (b) averaged over the number of AdaBoost classifiers. For the constrained PBT the weight of the cost factor is decreased linearly with increasing tree depth and thus the average cost per node increases. The standard PBT shows the opposite trend, as here only the hypothesis error is used for feature selection, ignoring costs. Hence, stronger but more expensive features are selected in higher tree nodes. (c) Comparison of classification performance of original PBT algorithm and our method by means of an ROC curve.

by the fact that in this experiment the variance in cost was very large between different features. The cost of the most expensive ones was more than 300 times bigger than that of the cheapest ones, which is why the difference in cost between these features was mostly larger than the difference in their hypothesis errors. Hence, the most expensive features could only be chosen at the deepest tree level, where the influence of the cost factor was set to 0. In other applications this behaviour might pose a problem, necessitating a smoother design of the cost function or a more sophisticated definition of λ.

The curve for the original PBT shows an opposite trend: As AdaBoost always chooses the best hypotheses, the more complex features, which have higher discriminative power, were chosen first and ended up in the highest tree nodes, resulting in high evaluation costs. At the same time these AdaBoost classifiers split up the input space very well, allowing the use of simpler features later in the PBT training process. That way the average evaluation cost decreased in deeper nodes.

The classification accuracy is the same for both versions (see Fig. 2(c)), but in the original PBT version the number of complex feature evaluations is considerably higher, resulting in a higher overall cost.

4. CONCLUSION

We introduced an extension to the AdaBoost algorithm that allows incorporating a user defined constraint into the hypothesis selection process during classifier training. In a preliminary study we showed that, using this method, the computational complexity of the PBT that forms the basis of our recently proposed liver tumor segmentation system could be reduced by 83% by incorporating a measure of feature cost. The same method can be applied to the AdaBoost cascade[10] or other, similar algorithms that contain a means of pruning parts of the classifier.

However, this extension to AdaBoost is by no means limited to complexity reduction. The algorithm is insensible to the semantics of the cost function. This method thus gives the user a powerful means of control over the AdaBoost procedure, allowing the simultaneous optimization on any additional criterion that can be defined at the hypothesis level.

DISCLAIMER

Approved for use in the SPIE conference. Not for commercial release or reprint outside SPIE conference.

REFERENCES

1. Shimizu, A., Narihira, T., Furukawa, D., Kobatake, H., Nawano, S., and Shinozaki, K., "Ensemble segmentation using AdaBoost with application to liver lesion extraction from a CT volume," *MIDAS Journal: Grand Challenge Liver Tumor Segmentation (MICCAI Workshop)* (Sept. 2008).
2. Li, Y., Hara, S., and Shimura, K., "A machine learning approach for locating boundaries of liver tumors in CT images," in [*Proc. ICPR*], 400–403 (Aug. 2006).
3. Militzer, A., Hager, T., Jäger, F., Tietjen, C., and Hornegger, J., "Automatic detection and segmentation of focal liver lesions in contrast enhanced CT images," in [*Proc. ICPR*], 2524–2527 (Aug. 2010).
4. Carneiro, G., Georgescu, B., Good, S., and Comaniciu, D., "Detection and measurement of fetal anatomies from ultrasound images using a constrained probabilistic boosting tree," *IEEE Transactions on Medical Imaging* **27**(9), 1342–1355 (Sept. 2008).
5. Tu, Z., "Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering," in [*Proc. ICCV*], **2**, 1589–1596 (Oct. 2005).
6. Tu, Z., Zhou, X. S., Barbu, A., Bogoni, L., and Comaniciu, D., "Probabilistic 3D polyp detection in CT images: The role of sample alignment," in [*Proc. CVPR*], **2**, 1544–1551 (June 2006).
7. Wels, M., Carneiro, G., Aplas, A., Huber, M., Hornegger, J., and Comaniciu, D., "A discriminative model-constrained graph cuts approach to fully automated pediatric brain tumor segmentation in 3-D MRI," in [*Proc. MICCAI*], 67–75, Springer-Verlag (Sept. 2008).
8. Freund, Y. and Schapire, R. E., "A decision-theoretic generalization of on-line learning and an application to boosting," in [*Proc. EuroCOLT*], 23–37, Springer-Verlag (Mar. 1995).
9. Friedman, J., Hastie, T., and Tibshirani, R., "Additive logistic regression: A statistical view of boosting," *Annals of Statistics* **28**(2), 337–407 (Apr. 2000).
10. Viola, P. and Jones, M., "Rapid object detection using a boosted cascade of simple features," in [*Proc. CVPR*], **1**, 511–518 (Dec. 2001).