

# Making a Shallow Network Deep: Growing a Tree from Decision Regions of a Boosting Classifier

Tae-Kyun Kim\*  
 http://mi.eng.cam.ac.uk/~tkk22  
 Ignas Budvytis\*  
 ib255@cam.ac.uk  
 Roberto Cipolla  
 cipolla@cam.ac.uk

Department of Engineering  
 University of Cambridge  
 Trumpington Street, Cambridge  
 CB2 1PZ, UK  
 \*indicates equal contribution.

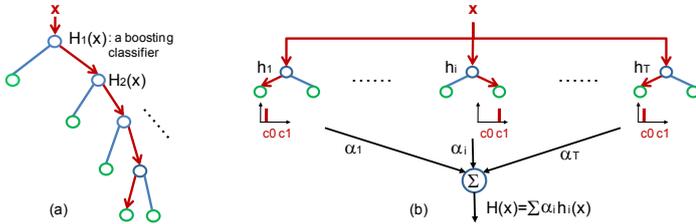


Figure 1: **Boosting as a tree.** (a) A boosting cascade is seen as an imbalanced tree, where each node is a boosting classifier. (b) A boosting classifier has a very shallow and flat network where each node is a decision-stump i.e. weak-learner.



Figure 3: **Boolean expression minimisation for an optimally short tree.** (a) A boosting classifier splits a space by binary weak learners (left). The regions are represented by the boolean table and the boolean expression is minimised (middle). An optimal short tree is built on the minimum expression (right).

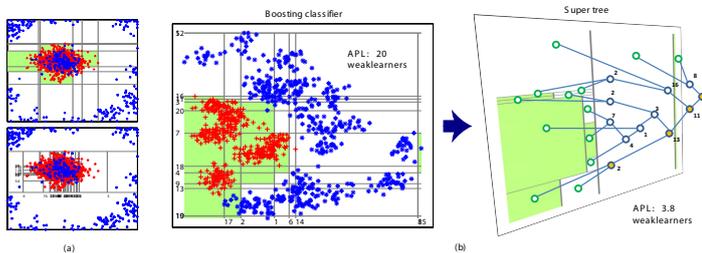


Figure 2: **Converting a boosting classifier into a tree for speeding up.** (a) The decision regions of a boosting classifier (top) are smooth compared to a conventional decision tree (bottom). (b) The proposed conversion preserves the Boosting decision regions and has many short paths speeding up 5 times.

This paper presents a novel way to speed up the classification time of a boosting classifier. We make the shallow (flat) network deep (hierarchical) by growing a tree from the decision regions of a given boosting classifier. This provides many short paths for speeding up and preserves the Boosting decision regions, which are reasonably smooth for good generalisation. We express the conversion as a Boolean optimisation problem.

**Boosting as a tree:** A cascade of boosting classifiers, which could be seen as a degenerate tree (see Figure 1(a)), effectively improves the classification speed. Designing a cascade, however, involves manual efforts for setting a number of parameters: the number of classifier stages, the number of weak-learners and the threshold per stage. In this work, we propose a novel way to reduce down the classification time of a boosting classifier not relying on a design of cascade. The chance for improvement comes from the fact that a standard boosting classifier can be seen as a very shallow network, see Figure 1(b), where each weak-learner is a decision-stump and all weak-learners are used to make a decision.

**Conversion of a boosting classifier into a tree:** Whereas a boosting classifier places decision stumps in a flat structure, a decision tree has a deep and hierarchical structure (see Figure 1(b) and 2(b)). The different structures lead to different behaviours: Boosting has a better generalisation via reasonably smooth decision regions but is not optimal in classification time. Whereas a conventional decision tree forms complex decision regions trying classification of all training points, a boosting classifier exhibits a reasonable smoothness in decision regions (see Figure 2(a)). We propose a method to grow a tree from the decision regions of a boosting classifier. As shown in Figure 2(b), the tree obtained, called *super tree*, preserves the Boosting decision regions: it places a leaf node on every region that is important to form the identical decision boundary (i.e. accuracy). In the mean time, Super tree has many short paths that reduce the average number of weak-learners to use when classifying a data point. In the example, super tree on average needs 3.8 weak-learners to perform classification whereas the boosting classifier needs 20.

**Boolean optimisation:** A standard boosting classifier is represented by the weighted sum of binary weak-learners as  $H(\mathbf{x}) = \sum_{i=1}^m \alpha_i h_i(\mathbf{x})$ ,

No. of weak learners	Boosting			Fast exit (cascade)			Super tree (cascade)		
	False positives	False negatives	Average path length	False positives	False negatives	Average path length	False positives	False negatives	Average path length
20	501	120	20	501	120	11.70	476	122	7.91
40	264	126	40	264	126	33.26	231	127	12.23
60	222	143	60	222	143	37.24	212	142	14.38
100	148	146	100	148(144)	146(149)	69.28(37.4)	(145)	(152)	(15.1)
200	120	143	200	120(146)	143(146)	146.19(38.1)	(128)	(146)	(15.6)

Figure 4: **Experimental results on the face images.** Example face images are shown in right.

where  $\alpha_i$  is the weight and  $h_i$  the  $i$ -th binary weak-learner in  $\{-1, 1\}$ . The boosting classifier splits a data space into  $2^m$  primitive regions by  $m$  binary weak-learners. Regions  $R_i, i = 1, \dots, 2^m$  are expressed as boolean codes (i.e. each weak-learner  $h_i$  corresponds to a binary variable  $w_i$ ). See Figure 3 for an example, where the boolean table is comprised of  $2^3$  regions. The region class label  $c$  is determined by the boosting sum. Region  $R_8$  in the example does not occupy the 2D input space and thus receives the *don't care* label marked "x" being ignored when representing decision regions. The boolean expression for the table in Figure 3 can be minimised by optimally joining the regions that share the same class label or *don't care* label as

$$\bar{w}_1 w_2 w_3 \vee w_1 \bar{w}_2 \bar{w}_3 \vee w_1 \bar{w}_2 w_3 \vee w_1 w_2 \bar{w}_3 \rightarrow w_1 \vee \bar{w}_1 w_2 w_3$$

where  $\vee$  denotes OR operator. The minimised expression has a smaller number of terms. Only the two terms,  $w_1$  and  $\bar{w}_1 w_2 w_3$  are remained representing the joint regions  $R_5 - R_8$  and  $R_4$  respectively. A short tree is then built from the minimised boolean expression by placing more frequent variables at the top of the tree (see Figure 3(right)).

Standard methods for Boolean expression minimisation, which has been previously studied for circuit design, are limited to a small number of binary variables i.e. weak-learners. Furthermore, all regions are treated with equal importance. We propose a novel boolean optimisation method for obtaining a reasonably short tree for a large number of weak-learners of a boosting classifier. The classifier information is efficiently packed by using the region coding and a tree is grown by maximising the region information gain. Further details are about a better way of packing the region information and the two stage cascade allowing the conversion with any number of weak-learners. See the paper for details.

**Experiments:** Experiments on the synthetic and face image data sets show that the obtained tree significantly speeds up both a standard boosting classifier and Fast-exit, a prior-art for fast boosting classification, at the same accuracy. The proposed method as a general meta-algorithm is also shown useful for a boosting cascade, since it speeds up individual stage classifiers by different gains. Figure 4 compares the average path lengths of the methods at the fixed accuracy at 0 threshold in the experiment using the face images. The proposed method is further demonstrated for rapid object tracking and segmentation problems. See the technical report at the authors' website.