Fast AdaBoost Training Using Weighted Novelty Selection

Mojtaba Seyedhosseini, Antonio R. C. Paiva and Tolga Tasdizen

www.sci.utah.edu
THE UNIVERSITY OF UTAH

INTRODUCTION

- Boosting is a general learning concept to train a strong learner by combining a set of weak learners.
- AdaBoost training can be time consuming for large datasets and convergence can be slow for problems with complex decision boundary.
- We propose a new learning framework which speeds up the training of AdaBoost, or any other boosting based algorithms.
- It consists of two parts: weighted novelty selection and AdaBoost.

WEIGTHED NOVELTY SELECTION

- WNS is the pre-processing sampling method in the WNS-AdaBoost which provides the boosting algorithm with a concise summary of the training dataset.
- WNS summarizes the dataset with a set of representative points and corresponding weights that show the importance of the representative points.
- WNS picks a data point as a representative point if the smallest distance to all previous representative points is larger than a threshold.
- Choosing an appropriate threshold, WNS ensures that enough points are picked to cover the whole space while keeping the number of them to a minimum.

$$X=\{x_1,\ldots,x_N\} \rightarrow \boxed{\text{WNS}} \rightarrow \left.\begin{matrix} X^R=\{x'_1,\ldots,x'_M\} \\ W=\{w_1,\ldots,w_M\}\end{matrix}\right\} \rightarrow \boxed{\text{AdaBoost}} \rightarrow \begin{matrix} H=\{h_1,\ldots,h_T\} \\ A=\{\alpha_1,\ldots,\alpha_T\}\end{matrix}$$

Fig 1: Illustration of the WNS-AdaBoost training model.

WNS-AdaBoost

- WNS-AdaBoost takes the outputs of the WNS to train the AdaBoost classifier.
- AdaBoost is given a smaller number of training points together with prior information about the importance of them.
- WNS speeds up AdaBoost in the training stage by reducing the number of training samples and maintains also the performance of AdaBoost at a good level by providing prior information about the importance of the selected representative points.

Given a training set $X=\{x_1, \ldots, x_N\}$ and corresponding labels $L=\{l_1, \ldots, l_N\}$, $l_i \in \{-1, 1\}$.

1. Separate the classes and make two sets: $X1=\{x_i \mid l_i=-1\}$, $X2=\{x_i \mid l_i=1\}$.
2. Choose a δ and run WNS for $X1$ and $X2$. The output of WNS, i.e. representative points and weights for each class are: $X1 \rightarrow (X_1^R, W_1)$, $X2 \rightarrow (X_2^R, W_2)$.
3. Construct a new training set $X^R=\{X_1^R, X_2^R\}$ and $W=\{W_1, W_2\}$.
4. Normalize W so it will be a probability distribution.
5. Use X^R, W to train AdaBoost classifier.

Fig 2: WNS-AdaBoost training algorithm

RESULTS

- Poker hand classification

TABLE I
TRAINING TIME AND PERFORMANCE FOR THE "POKER HAND" DATASET.

Method	δ	Number of training samples	Time for applying WNS (s)	Time for training	Training error	Testing error	Speedup
WNS-AdaBoost	3	13396	8.85	135s	13%	20%	2.23
WNS-AdaBoost	2.7	18278	10.05	236.10s	11%	17%	1.3
AdaBoost	–	25010	–	320.19s	10.3%	16%	–

- Texture segmentation

TABLE II
TRAINING TIME FOR THE "TEXTURE SEGMENTATION" EXPERIMENT.

Method	δ	Number of training samples	Time for applying WNS (s)	Time for training	Speedup
WNS-PBT	1	1808	108.62	5.82s	38351
WNS-PBT	0.7	27206	11186.96	502.64s	444
PBT	–	524288	–	62hours	–

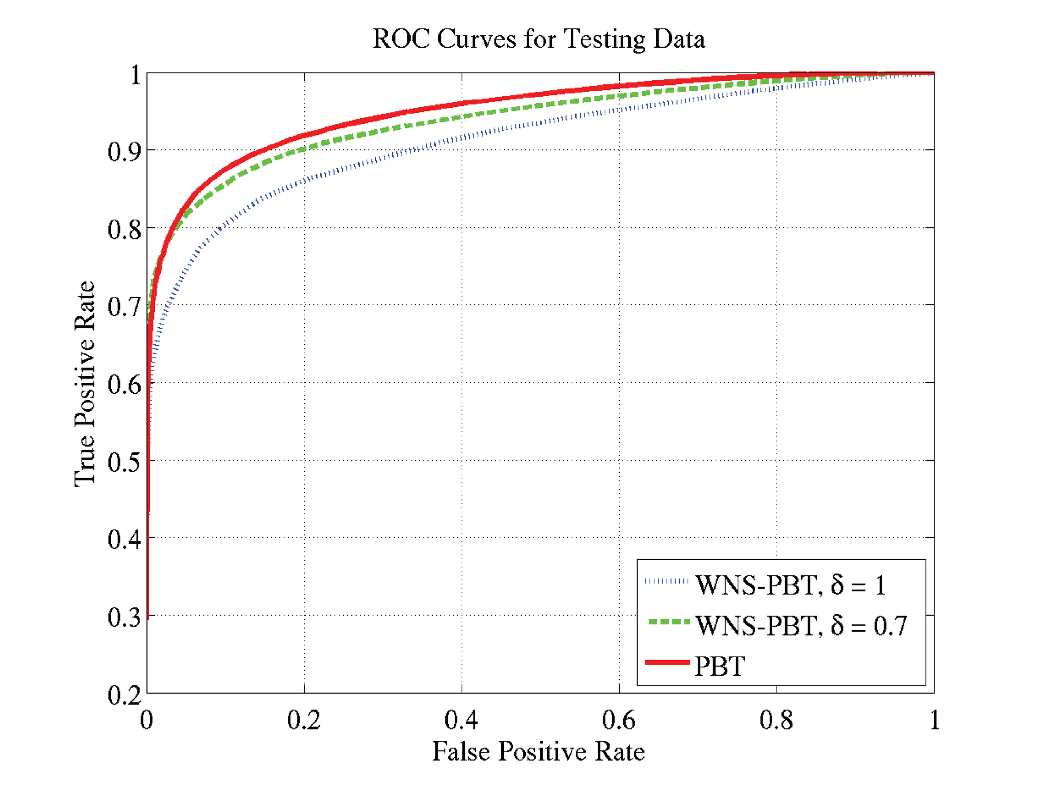

Fig 3: ROC curves for texture segmentation.

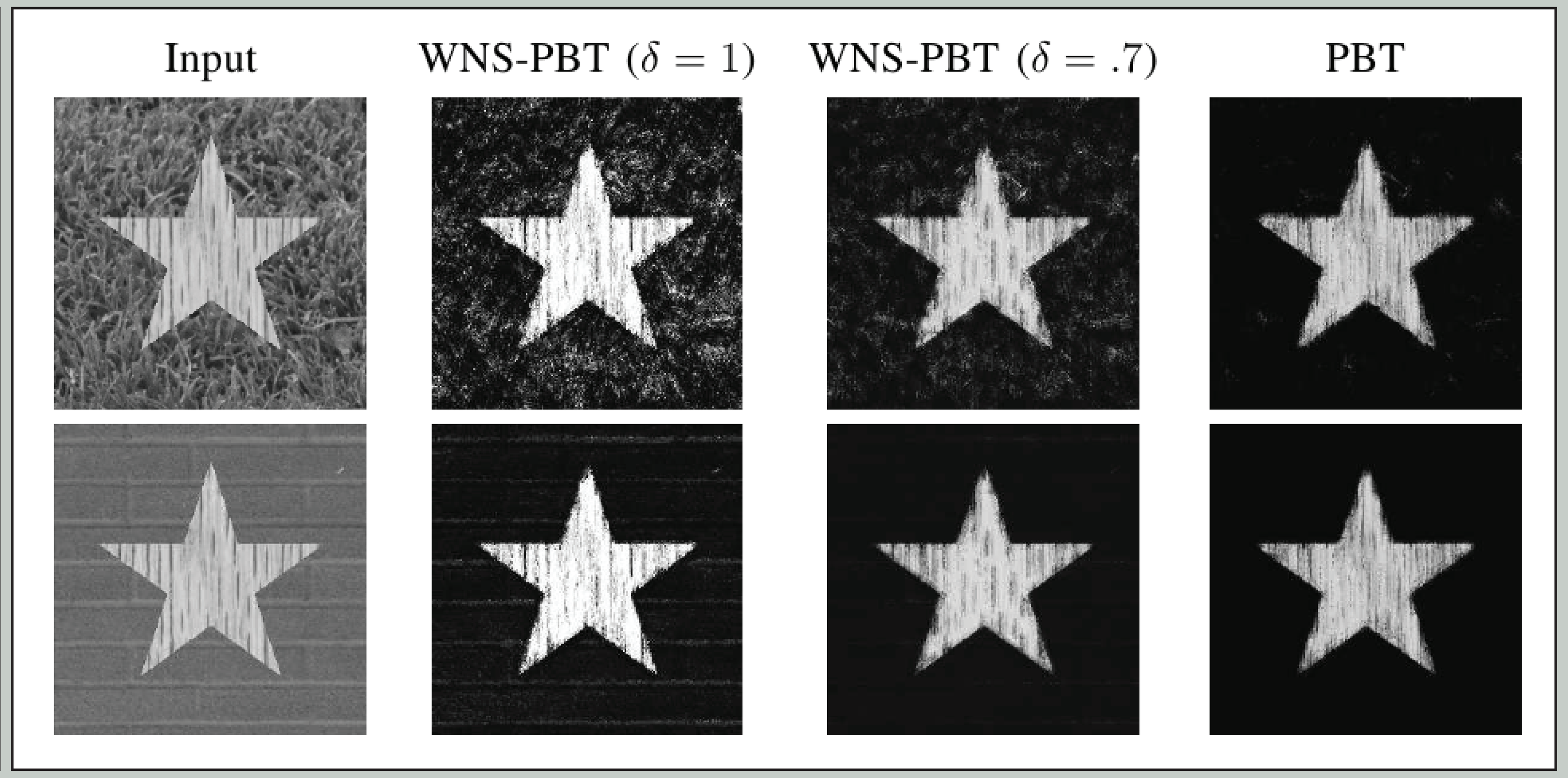

Fig 4: Test results for the texture segmentation experiment.

CONCLUSION

- WNS provides a compact representation of the distribution of the training data in a way that is naturally amenable to AdaBoost, or any other AdaBoost-based classifier.
- WNS-AdaBoost reduces the overall computational complexity and increases the speed of the training process.
- The improvement in training speed is achieved potentially at the expense of a small reduction in accuracy.