Boost Picking: A Universal Method on Converting Supervised Classification to Semi-supervised Classification

FUQIANG LIU  
Beijing Institute of Technology, Beijing 100081 CHINA  
FQLIU@OUTLOOK.COM

FUKUN BI  
North China University of Technology, Beijing 100144 CHINA  
BIFUKUN@NCUT.EDU.CN

YIDING YANG  
Beijing Institute of Technology, Beijing 100081 CHINA  
YANGYDI9G3@BIT.EDU.CN

LIANG CHEN  
Beijing Institute of Technology, Beijing 100081 CHINA  
CHENL@BIT.EDU.CN

Abstract

This paper proposes a universal method, Boost Picking, to train supervised classification models mainly by unlabeled data. Boost Picking only adopts two weak classifiers to estimate and correct the error. It is theoretically proved that Boost Picking could train a supervised model mainly by unlabeled data as effectively as the same model trained by 100% labeled data, only if recalls of the two weak classifiers are all greater than zero and the sum of precisions is greater than one. Based on Boost Picking, we present "Test along with Training (TawT)" to improve the generalization of supervised models. Both Boost Picking and TawT are successfully tested in varied little data sets.

1. Introduction

Supervised learning is the machine learning task of inferring a function from labeled training data (Mohri et al., 2012). The achievements of supervised learning rely on labeled data bases, like LFW (Huang et al., 2007), ImageNet (Fei-Fei, 2010) and Microsoft COCO (Lin et al., 2014). Humans academically called "crowds" (Attenberg et al., 2011; Burton et al., 2012), who label the data in practice, greatly contribute to the development of machine learning. However, labeling data artificially is not a completely ideal strategy, especially when data erupt. There are two harsh facts:

• In practice, the scale of unlabeled data is far larger than that of labeled data, and the unlabeled data are substantially produced at any time.

• Labeling data by humans is too expensive when the scale of data increases.

Generally, using abundant data could improve the performance especially generalization of supervised learning models. Based on the above two harsh facts and the benefit of abundant data, we carried out the preliminary research on utilizing unlabeled data to train classical supervised classification models without any internal change, in both theory and practice.

We present a universal method named "Boost Picking" to train supervised classification models mainly by unlabeled data. Boost Picking just adds external modules rather than creates internal changes in the traditional models. Furthermore, the external modules are just two weak classifiers to estimate and correct the error of supervised models. We use two weak classifiers to train a strong classifier by unlabeled data (section 3.1) as well as a classifier trained by 100% labeled data. Boost Picking would work effectively only if the sum of the two weak classifiers precisions is greater than one (section 3.2). Precisions of the two weak classifiers determine whether Boost Picking works effectively. And the efficiency of Boost Picking is affected by both precisions and recalls of the two weak classifiers (section 3.3).

Based on Boost Picking, we present a new way named "Test along with Training, (TawT)" (section 4) to improve
the generalization of supervised classification models. Data to be classified are un-labeled, as a result, before classifying these data, train the supervised model using these unlabeled data by Boost Picking. The effect of TawT is theoretically analyzed in section 4.

The structure of Boost Picking refers to "P-N Learning" in "Tracking-Learning-Detection" (Kalal et al., 2012). Kalal’s work inspired Boost Picking (summarized in section 2). P-N Learning ignores the bias between the estimated labels and the true labels of examples in training data set. But the assumption of Boost Picking is looser and easier. Our contributions are listed as follows.

- Theoretically prove the establishment condition of Boost Picking.
- Present a theoretical guide on designing the weak classifiers to improve the efficiency of Boost Picking.
- Theoretically analyze the generalization improvement of the supervised models when they are used as TawT.

Boost Picking involves a small amount of labeled data and a large amount of un-labeled data in training the supervised learning models. This peculiarity is as same as that of Semi-supervised learning (Chapelle et al., 2006). Different from previous semi-supervised learning works like (Zhu, 2005; Zhu & Goldberg, 2009; Fazakis et al., 2015), Boost Picking, a universal method on training traditional supervised models using un-labeled data, adopts no internal change and has no limitation on the supervised models. Logistic regression (Hosmer Jr & Lemeshow, 2004), neural network (Haykin & Network, 2004), support vector machine (SVM) (Suykens & Vandewalle, 1999) and Adaboost (Rätsch et al., 2001) are all tested with varied data sets in machine (SVM) (Suykens & Vandewalle, 1999) and Adaboost (Haykin & Network, 2004), support vector machine (Hosmer Jr & Lemeshow, 2004), neural network (Haykin & Network, 2004), support vector machine (SVM) (Suykens & Vandewalle, 1999) and Adaboost (Rätsch et al., 2001) are all tested with varied data sets in experiments (section 5). Using Boost Picking, these models can be trained by 75% un-labeled data as effectively as same models trained by 100% labeled data. And TawT could improve generalizations of these models. Boost Picking can be regard as a universal method on converting supervised classification to semi-supervised classification.

2. Prior Work

This section introduces Kalal’s works. In TLD (Kalal et al., 2012), P-N Learning is an online learning method to learn the new form of the tracked object. It adopts two classifiers, which are named P-expert and N-expert respectively, to evaluate the error of the detector (a classifier) and correct it. P-expert estimates the false negatives from the negative outputs of the detector, while N-expert estimates the false positives from the positive outputs. Boost Picking refers to this thought.

Kalal theoretically proved that P-N Learning could train a detector whose error converges into zero (Kalal et al., 2010; 2012). Define \( \overrightarrow{d(k)} \) as the errors in kth iteration,

\[
\overrightarrow{d(k)} = [\alpha(k), \beta(k)]^T
\]

(1)

where \( \alpha(k) \) represents false positives and \( \beta(k) \) represents false negatives in outputs of the detector.

Define \( P^+, R^+ \) as the precision and the recall of P-expert respectively. Similarly, define \( P^-, R^- \) as the precision and the recall of N-expert. The recursive equations are as equation (2) (Kalal et al., 2012).

\[
\begin{align}
\alpha(k + 1) &= \alpha(k) - R^- \cdot \alpha(k) + \frac{1 - P^+}{P^+} \cdot R^+ \cdot \beta(k) \\
\beta(k + 1) &= \beta(k) - R^+ \cdot \beta(k) + \frac{1 - P^-}{P^-} \cdot R^- \cdot \alpha(k)
\end{align}
\]

(2)

Reconstruct equation (1) and (2),

\[
\overrightarrow{d}(k + 1) = M \overrightarrow{d}(k)
\]

(3)

where \( M = \left[ \begin{array}{ccc} 1 - R^- & \frac{1 - P^+}{P^+} \cdot R^+ \\ \frac{1 - P^-}{P^-} \cdot R^- & 1 - R^+ \end{array} \right] \).

Kalal concludes that \( d \) will converge to zero only if eigenvalues of the transformation matrix \( M \) are all smaller than one (Kalal et al., 2010). It should be noted that P-N Learning try training a classifier without error, \( d \rightarrow 0 \). In practice, it is difficult for the error to converge into zero, because the classifier almost cannot be trained to be errorless even through P-N experts could find out all false positives and false negatives from the outputs. The conclusion in Kalal’s works is based on an assumption that the trained classifier can perfectly classify the data in training set. P-N Learning ignores the bias of the classifier in training set.

3. Boost Picking

This section details Boost Picking, drives its establishment condition and analyzes its efficiency in theory.

3.1. Description of Boost Picking

The structure of Boost Picking is showed as Fig 1. Supposing \( x \) is an example from a feature-space \( X \), and \( y \) is its corresponding label from a label space \( Y = \{1, -1\} \), the training set of Boost Picking is composed by a labeled set \( X_l \) with its label \( L_l \) and an unlabeled set \( X_u \). Supposing that \( L_u \) are the true labels of \( X_u \) (do not exist in fact), define \( F : x \rightarrow y \) as an supervised learning model trained by 100% labeled data set \( X_{(100\%)} = \{X_l, X_u\} \) with \( \{L_l, L_u\} \). The goal of Boost Picking is to train an effective model \( f \rightarrow F \) by \( X_l, L_l \) and \( X_u \).
First, the supervised model is trained by labeled data $X_l$, which is named "initialization" (① in Fig 1). Through this step, the model is initialized as $f_0$.

Second, the un-labeled data $X_u$ are classified by the trained model (② in Fig 1).

$$y^k = f_{k-1}(x), x \in X_u,$$  \hspace{1cm} (4)

where $y^k$ is the estimated label in kth iteration and $f_{k-1}$ is the result of the k-1th iteration.

Then use two classifiers, FP Finder and FN Finder, to estimate the error in the outputs. Here, functions of FP Finder and FN Finder are similar with N-expert and P-expert. FP Finder detects the false positives in positive outputs and FN Finder detects the false negatives in negative outputs.

$$\left\{ \begin{array}{l} X^f = FP(X^+) \\ X^f = FN(X^-) \end{array} \right.$$  \hspace{1cm} (5)

And then, the supervised learning model is re-trained by a new data set. The new data set is composed by randomly selected labeled data and the results of FP Finder and FN Finder. $X_{new} = X_{new} \cup \{X^f^+, X^f^-\}$. In the new data set, labels of FP Finder results are set as 1 and labels of FN Finder results are set as 1. This strategy is similar with supervised bootstrapping (Sung & Poggio, 1995), recognizing misclassified examples and adding them to the training set with correct labels.

The first step "initialization" (① in Fig 1) is executed only one time. And the second step (② in Fig 1) is circularly executed until satisfying the convergence condition. The establishment condition of Boost Picking is theoretically proved in section 3.2.

Algorithm 1 Boost Picking

Input: data $X_l$ with label $L_l$, data $X_u$

Initialize $trainset = X_l$, $Y_{now} = 0$

repeat

Train $f$ with $trainset$

$Y_{pre} = Y_{now}$

$Y_{now} = f(X_u)$

$X^f = FP(X^+), X^f^- = FN(X^-)$

$trainset = trainset \cup \{X^f^+, X^f^-\}$

$con1 = \sum(|Y_{now} - Y_{pre}|)/2N_u < \theta_{diff}$

$X_{ran} = ranselect(X_l)$

$con2 = \sum(|L_{ran} - f(X_{ran})|)/2N_{ran} < \theta_{bias}$

until $con1 + con2 == 2$

where $N_u$ is the size of $X_u$. Another convergence condition is that the classification results of part of labeled data has a relatively small bias.

$$\sum(|L_{ran} - f(X_{ran})|)/2N_{ran} < \theta_{bias}$$  \hspace{1cm} (7)

where $x_{ran} \in X_{ran}$ and $N_{ran}$ is the scale of $X_{ran}$. It should be noted that each $X_{ran}$ is randomly selected from $X_l$ anew.

Unlike the equation (6), equation (7) is an empirical solution for the problem that FP Finder and FN Finder might be designed so poor that the establishment condition cannot be always satisfied just like our experiments (section 5.2). In section 3.2.3, why this condition is needed is detailed. Algorithm 1 is the format pseudocode of Boost Picking in practice.

3.2. Establishment Condition

This subsection drives the establishment condition of Boost Picking. It includes:

- Prove that Boost Picking could train supervised models mainly by un-labeled data as effectively as same models trained by all labeled data, only if the eigenvalues of the transformation matrix (defined in section 2) are all smaller than one.

- Derive that Boost Picking would work effectively only if sum of two weak classifiers precisions is greater than one and recalls are all greater than zero.

- Discuss the instantaneous situation.

3.2.1. CONVERGENCE DERIVATION

The conclusion in prior work (section 2) is established based on an assumption that the trained model could perfectly classify the data in training set. Under most conditions, this assumption cannot be established. However,
when Boost Picking aims to train $f \rightarrow F$ by $X_l, L_l$ and $X_u$, the assumption can be looser. Here, the conclusion that $f \rightarrow F$ only if the eigenvalues of the transformation matrix $M$ are smaller than one is derived.

What should be paid attention is that both $f$ and $F$ are a same model and all their parameters are same. The only difference between $f$ and $F$ is the training data set (section 3.1). Define $X^T$ is the data set where examples in $X_{100\%}$ can be correctly classified by $F$, and $X^F$ is the data set where examples cannot be correctly classified by $F$.

\[
\begin{align*}
X^T &= \{ x \in X_{100\%} | F(x) = l \} \\
X^F &= \{ x \in X_{100\%} | F(x) \neq l \}
\end{align*}
\]

(8)

where $l$ is the true label of $x$. Define $\Delta e$ as the difference between $f$ and $F$, and $\Delta e$ is valued by the number of examples whose classification results by $f$ and $F$ are different.

\[
\Delta e = \sum \Delta e_i / 2N
\]

(9)

where $N$ is the size of $X_{100\%}$.

$\Delta e$ can be reconstructed as

\[
\Delta e = d + b
\]

(10)

where $d_i = |f(x_i) - F(x_i)|$, $x_i \in X^T$, $d = \sum d_i / 2N$ and $b_i = |f(x_i) - F(x_i)|$, $x_i \in X^F$, $b = \sum b_i / 2N$. $d$ means the number of examples that can be correctly classified by $F$ but cannot be correctly classified by $f$. $b$ means the number of examples that can be correctly classified by $f$ but cannot be correctly classified by $F$.

Define $bias_i = |l_i - F(x_i)|$, $x_i \in X_{100\%}$ and $bias = \sum bias_i / 2N$. $bias$ represents the bias between the labels estimated by $F$ and the true labels. When $x_i \in X^T, bias_i = 0$. In contrast, when $x_i \in X^F, bias_i = 2$.

\[
b_i = 0 \text{ when } x_i \in X^F \cap f(x_i) = l, \text{ and } b_i = 2 \text{ when } x_i \in X^F \cap f(x_i) = l_i, \text{ as a result, } b = \sum |l - F(x)| / 2N, x \in X^F \cap f(x) = l.
\]

Based on the above conclusion, $bias$ can be reconstructed as equation (11).

\[
bias = b + n
\]

(11)

where $n_i = |l - F(x_i)|$, $x_i \in X^F \cap f(x_i) \neq l$ and $n = \sum n_i / 2N$.

Generally, bias can be reduced by increasing the complexity of models, while it will be almost constant if the model is determined (including the model’s parameters). Define $e_i = |l |i - f(x_i)|$, $x_i \in X_{100\%}$ and $e = \sum e_i / 2N$. $e$ represents the "error" between the labels estimated by $f$ and the true labels.

\[
e = bias + \Delta e - 2b = d + bias - b \leq d + bias
\]

(12)

Table 1. Relationship among $\Delta e_i, d_i, b_i, bias_i, n_i$ and $e_i$

| CONDITION | $F(x_i) = l_i$ | $F(x_i) \neq l_i$ |
|-----------|----------------|-----------------|
| $f(x_i) = l_i$ | $\Delta e_i, d_i, b_i, bias_i$ | $\Delta e_i, b_i, bias_i$ |
| $f(x_i) \neq l_i$ | $\Delta e_i$ | $bias_i$ |
| $n_i$ | $e_i$ |

Table 1 shows the relationship among $\Delta e_i, d_i, b_i, bias_i, n_i$ and $e_i$. Take $b_i$ for instance, $b_i$ appears in $(f(x_i) = l_i, F(x_i) \neq l_i)$, and this means that $\forall x_i \in X_{100\%}$, if $f(x_i) = l_i$ & $F(x_i) \neq l_i$, $b_i = 2$. As for the quadrants where $b_i$ does not exit, $b_i$ is not defined or equal to zero. $\Delta e_i, b_i, bias_i, n_i$ and $e_i$ are similar. From Table 1, equation (10), (11) and (12) can be easily derived.

As above mentioned, $b$ represents the improvement between the performance of $f$ and the performance of $F$ in training set $X_{100\%}$. $b$ might be the cost to make the model $F$ avoid being over-fitting. However, because $f$ and $F$ are a same model ($b$ is relatively small) and $f$ is un-fitting in early iteration, we temporarily ignore the negative impact of $b$ on the generalization and regard $b$ is helpful for model improvement. Furthermore, even if $f$ harms the generalization, section 4 details how to improve the generalization of the supervised learning models using Boost Picking.

As for $d$, it can be proved that $d \rightarrow 0$ as section 2, because $x_i$ can be perfectly classified ($F(x_i) = l_i$) when $x_i \in X^T$.

\[
\bar{d}(k - 1) = M \bar{d}(k)
\]

(13)

d $\rightarrow 0$ when the eigenvalues of transformation matrix $M$ are smaller than 1.

As a result, if $b = 0$, $d \rightarrow 0$, $\Delta e = 0$, and $f \rightarrow F$; if $b \neq 0$, $d \rightarrow 0$, $\Delta e = b$, and $f$ is better than $F$, as least in training set. This part gets the conclusion that $f$ could be trained by $X_l, L_l$ and $X_u$ as effectively as $F$ trained by $X_{100\%}$ and $L_{100\%}$, only if eigenvalues of $M$ are smaller than one.

3.2.2. FP FINDER AND FN FINDER DESIGN

This part theoretically analyzes how to design FN Finder and FP Finder to make sure eigenvalues of $M$ are all smaller than one and then $f \rightarrow F$. Assuming $\lambda_1, \lambda_2$ are eigenvalues of transformation matrix $M$, the expression of the two eigenvalues are as equation (14).

\[
\lambda_1, \lambda_2 = \frac{(2 - R^- - R^+) \pm \sqrt{\Delta}}{2}
\]

(14)

where $P^+, R^+, P^-, R^- \in [0, 1]$ (defined in section 2), and

\[
\Delta = (R^+ - R^-)^2 + 4 \frac{(1 - P^+)(1 - P^-)}{P^+P^-} R^+ R^-
\]

(15)

$\Delta \geq 0$ is consistently satisfied because $(R^+ - R^-)^2 \geq 0$ and $(1 - P^+)(1 - P^-) R^+ R^- \geq 0$. $\Delta = 0$ under the condition
that $R^+ = R^- \& \& (P^+ = 1 \mid P^- = 1)$ or $R^+ = R^- = 0$.

When the above condition is established, $\lambda_1 = \lambda_2$.

Assuming $\lambda_2 \leq \lambda_1$, because $2 - R^- - R^+ \geq 0$, $|\lambda_2| \leq |\lambda_1|$. As section 3.2.1 referred, $(|\lambda_1|, |\lambda_2|) < 1$ leads to $d \rightarrow 0$. $\lambda_1 \geq 0$ is consistently since $\sqrt{\Delta} \geq 0$. If $\lambda_1 < 1$, $|\lambda_2| \leq |\lambda_1| < 1$, and $d \rightarrow 0$.

If $\lambda_1 < 1$, $(\frac{2}{2} - R^- + R^+ \lambda^2 + \sqrt{\Delta}) < 1$, and derive

$$\sqrt{\Delta} < R^- + R^+ \quad (16)$$

Simplify equation (16) and obtain

$$\frac{1 - P^+ - P^-}{P^+ - P^-} R^+ R^- < 0 \quad (17)$$

Based on equation (17), FP Finder and FN Finder should be carefully designed as equation (18). Then eigenvalues of $M$ are all smaller than one and $d \rightarrow 0$ is guaranteed.

$$\left\{ \begin{array}{ll}
R^+ R^- \neq 0 \\
\quad P^+ + P^- > 1 
\end{array} \right. \quad (18)$$

Fig 2 shows the relationship between the maximal eigenvalue of $M$ and $(P^+, P^-)$ when $R^+ = R^-$. The red line is $P^+ + P^- = 1$, and the maximal eigenvalue in the red line is definitely equal to one. When $P^+ + P^- > 1$, the max eigenvalue is smaller than one. Fig 3 shows that no matter what the recalls of the two classifiers are, the maximal eigenvalue is definitely equal to one when $P^+ + P^- = 1$. And the maximal eigenvalue is smaller than one when $P^+ + P^- > 1$.

As a result, the sum of precisions of FP Finder and FN Finder determines whether Boost Picking can train $f$ as effectively as $F$. Under the condition that recalls of the two weak classifiers are neither equal to zero, the differences between $f$ and $F$ will drop to zero in the iteration process only if $P^+ + P^- > 1$.

Here, based on the above theoretical analysis and Fig 2, 3, it is concluded that if recalls of the two weak classifiers are both greater than zero and sum of precisions is greater than one, Boost Picking could train an effective supervised classification models mainly by unlabeled data as effectively as the same model trained by all labeled data.

3.2.3. instantaneous situation

In section 3.2.2, $P^+, P^-, R^+, R^-$ are constant, while the precisions and recalls are usually varied with the iteration processing in practice. Assuming that $P^+(k), P^-(k), R^+(k), R^-(k)$ are the instantaneous values of the precisions and recalls, expression of $P^+$ is written as equation (19).

$$P^+ = \frac{1}{T} \sum_{k=1}^{T} P^+(k) \quad (19)$$

where $T$ is the generation of the iteration process. $P^-, R^+, R^-$ are defined similarly. The ideal condition is that FP Finder and FN Finder satisfy equation (19) in every generation.

$$\left\{ \begin{array}{ll}
R^+(k) R^-(k) \neq 0 \\
P^+(k) + P^-(k) > 1 
\end{array} \right. \quad (20)$$

Equation (20) is relatively strict compared to equation (18). In practice, $P^+(k), P^-(k), R^+(k), R^-(k)$ are likely to be choppy, and $d$ is choppy decreasing. If $k \in [K, T]$, $P^+(k) + P^-(k) > 1$ and $T - K$ is relatively long enough, it
would achieve \( d \to 0 \) and \( f \to F \). Because of the shock of \( P^+ + P^- \), equation (7) is another convergence condition to guarantee that \( d \) converges into zero rather than other values. Adopting random selection in equation (7) aims to avoid being over-fitting.

### 3.3. Performance Analysis

This section discusses how to design FP Finder and FN Finder to ensure Boost Picking works efficiently. Two aspects are involved in the discussion. The first one is focused on the whole iteration process and discusses the shock. And the second one is focused on single iteration and states how to reduce \( d \) as much as possible in one generation.

#### 3.3.1. Efficiency Improvement from Global Aspect

According to section 3.2.3, \( d \) would be choppy because \( P^+(k) + P^-(k) \) might not be always greater than one. Define \( K^+ = \{ k | P^+(k) + P^-(k) \leq 1 \} \), and \( K^- = \{ k | P^+(k) + P^-(k) > 1 \} \). Assuming \( \Delta d(k) \) is the increment of \( d \) in kth iteration, if \( d \to 0 \), \( \sum_{k \in K^-} | \Delta d(k) | - \sum_{k \in K^+} | \Delta d(k) | \geq \frac{d}{0} \). As a result, large \( K^+ \) has an obvious negative impact on reducing the generation of convergence. Thus, FP Finder and FN Finder should satisfy equation (20) as much as they could.

#### 3.3.2. Efficiency Improvement from One Generation

Based on the theory of dynamical system (Ogata, 2001), \( d \) converges to zero more quickly if absolute values of the eigenvalues are smaller. There is a negative correlation between the speed of error dropping and \( (|\lambda_1|, |\lambda_2|) \). It can be written as equation (21).

\[
\Delta d \propto (-|\lambda_1|, -|\lambda_2|) \quad (21)
\]

The way to make \( d \) converge to zero quickly is to design \( |\lambda_1|, |\lambda_2| \) as small as possible. Supposing \( |\lambda_1| + |\lambda_2| \) as the cost function, the optimal FN Finder and FP Finder should be designed as equation (22).

\[
R^+, R^-, P^+, P^- = \arg \min_{R^+, R^-, P^+, P^-} |\lambda_1| + |\lambda_2|
\]
\[s.t. P^+ + P^- > 1, 0 < R^+, R^-, P^+, P^- \leq 1 \quad (22)
\]

Supposing \( \lambda_2 \leq \lambda_1, \lambda_1 \geq 0 \) based on equation (14). If \( \lambda_1 \lambda_2 < 0, \lambda_2 < 0 \), and the determinate of \( M \) is smaller than 0.

\[
|M| = 1 - (R^+ + R^-) + \frac{(P^+ + P^-) - 1}{P^+ P^-} R^+ R^- < 0 \quad (23)
\]

Assuming \( C(R^+, R^-) = |M|, C < 0 \to \lambda_2 < 0 \) and \( C \geq 0 \to (\lambda_1, \lambda_2 \geq 0) \) (shown in Fig 4(a)). When \( (R^+, R^-) = (1,0) \) or \( (0,1) \), \( C(R^+, R^-) \) is definitely equal to zero. No matter what \( P^+, P^- \) is, all curves \( C = 0 \) go through \( (0,1) \) and \( (1,0) \) (shown in Fig 4(b-d)). Larger \( P^+ \) or \( P^- \) makes \( \lambda_2 \) more likely to be greater than zero, because \( C = 0 \) is closer to \( (1,1) \) when \( P^+ \) or \( P^- \) is larger and that means more area where \( C > 0 \) (shown in Fig 4(b,c)); the smaller \( |P^+ - P^-| \) is, the less likely \( (\lambda_1 \lambda_2 > 0) \) (shown in Fig 4(d)).

**Figure 4.** Figures about relationships between minimum eigenvalue and \( C(R^+, R^-) \). (a) shows the minimum eigenvalue decreasing with the increasing of \( (R^+, R^-) \), and (a) marks where \( C > 0, C = 0 \) and \( C < 0 \); (b)(c) show \( C = 0 \) shifts to \( (1,1) \) with the increasing of \( (P^+, P^-) \); (d) shows when \( P^+ + P^- \) is constant, smaller \( |P^+ - P^-| \) leads to \( C = 0 \) more away from \( (1,1) \).

Based on equation (14) and relationship between \( \lambda_1 \lambda_2 \) and \( C(R^+, R^-) \),

\[
|\lambda_1| + |\lambda_2| = \begin{cases}
2 - R^+ - R^- . C(R^+, R^-) \geq 0 \\
\sqrt{\lambda_1 \lambda_2} . C(R^+, R^-) < 0
\end{cases} \quad (24)
\]

When \( C(R^+, R^-) \geq 0, |\lambda_1| + |\lambda_2| \) will decrease if \( R^+ + R^- \) increases. And if \( R^+ + R^- \) is constant, the efficiency of Boost Picking is similar (In Fig 5, contour lines are in planes \( R^+ + R^- = P \), when \( C(R^+, R^-) > 0 \). When \( C(R^+, R^-) < 0, |\lambda_1| + |\lambda_2| = \sqrt{\lambda_1 \lambda_2} \) that is an increasing function for both \( R^+ \) and \( R^- \). And if \( R^+ + R^- \) is constant, the smaller \( |R^+ - R^-| \) is, the smaller \( |\lambda_1| + |\lambda_2| \) is (In Fig 5, the contours bulge to \( (1,1) \)). Consequently, if \( P^+ + P^- > 1 \), the optimal design of FN Finder and FP Finder is \( R^+ = R^- \cap C(R^+, R^-) = 0 \). Especially, when \( P^+ = P^- \), the best design is \( R^+ = R^- = P^+ = P^- \) and the efficiency of Boost Picking is positively correlation to the precisions.
It should be noted that for certain $P^+$ and $P^-$, $R^+$ and $R^-$ are not as larger as better. If $P^+$, $P^-$ are larger, the $C(R^+, R^-)$ (purple line in Fig5) would shift close to (1,1), which means $R^+$, $R^-$ can be larger to reduce $|\lambda_1| + |\lambda_2|$. 

Combining section(3.2.2), the above theoretical analysis and Fig 5, we concluded that

- Precisions of FP Finder and FN Finder determine the upper boundary of the performance of Boost Picking, including effectiveness and efficiency.
- If recalls of FP Finder and FN Finder are relatively high, it is better to adjust the balance of the two Finders or improve the precisions to make Boost Picking more efficient.
- If the recalls are not ideal, the best way to improve Boost Picking performance is to improve the recalls, rather than keep balance or improve the precisions.

4. Test along with Training

This section states a method to improve the generalization of the supervised classification models by Boost Picking. Because boost picking can use un-labeled data to train the supervised classification models, before classifying the un-labeled data in test set, use these un-labeled data to train the model. In fact, these data are classified during the Boost Picking process. As a result, this methods is named "Test along with Training (TawT)".

Fig 6 shows the concept of TawT. In traditional process, supervised learning models are trained by labeled data in the training set and classify the un-labeled data in test set. As for Boost Picking, the training set involves un-labeled data. Based on Boost Picking, TawT cuts across the boundary between training set and test set to improve the generalization.

Based on equation (12), when $d \to 0$, relationship among "bias", "variance" and "error" (Geman et al., 1992; James, 2003) could be simplified as Fig 7. In Fig 7, $e$ (defined as equation (12)) represents the "error" between the labels estimated by $f$ and the true labels. It should be noted that left "bias" and right "bias" are some different. The "new training set" used to compute right "bias" is an assumed set. This set includes the training set used to compute left "bias" and the test set with its assumed true label set. In theory, TawT could eliminate the influence of variance and improve the performance of supervised models (equation (12) and Fig 7), which is tested in section 5.

5. Experiment

This section details experiments to verify functions of Boost Picking and TawT. Logistic regression(lr), bp-neural network(nn), support vector machine(svm) and adaboost(ada) are involved in these experiments. For certain model, parameters are same among tradition supervised classification, Boost Picking and TawT. Two data sets, heart scale and wine, are used. For one data set $X$ with its label set $L$, it is separated into two part, $X_{rest}$ and $X_{test}$, $X_{rest}/X_{test}$ means the ratio of number of examples and its results is 3/1. $X_{test}$ is the constant test data set for all experiments. And for all experiments, the measurement of model performance is $f1 = \frac{2*Precision*Recall}{Precision+Recall}$. 

---

**Figure 5.** $|\lambda_1| + |\lambda_2|$ varied with $R^+$ and $R^-$ under $P^+ = P^- = 0.6$

**Figure 6.** Concept of TawT

**Figure 7.** Error Simplification
5.1. Baseline

This subsection illustrates that only small amount of labeled data cannot train an effective supervised model. Classical models are trained by different training sets $X_{\text{train}}$ with different scales and the trained models are test in $X_{\text{test}}$. $X_{\text{train}} \subset X_{\text{rest}}$ and $X_{\text{train}}/X_{\text{rest}}$ ranges from 0.1 to 1. Fig 9 shows the experiment. The horizontal axis represents $X_{\text{train}}/X_{\text{rest}}$. Furthermore, this experiment also determines the baseline or goal for Boost Picking. According to Fig 9, generally, larger $X_{\text{train}}/X_{\text{rest}}$ leads to larger $f1$ in $X_{\text{test}}$. We choose the models trained by whole $X_{\text{rest}}$ as baselines.

5.2. FP Finder and FN Finder in Experiments

Here, a very poor FP Finder and FN Finder are designed to verify equation (18). Compute the center of positive outputs and the examples that are the farthest 20% from the center are regarded as false positives. FP is designed as the above statement and FN is designed similarly. This two classifiers are very weak and their performances wave. In these experiments, the average sum of precisions is 1.06 and the average sum of recalls is 0.12.

5.3. Verification

This subsection tests the functions of Boost Picking and TawT in practice, whose results are shown as Fig 8. For Boost Picking experiments, $X_{\text{rest}}$ are all used to train models but only examples from $X_{\text{label}}$ have labels. $X_{\text{label}} \subset X_{\text{rest}}$ and $X_{\text{label}}/X_{\text{rest}}$ ranges from 5% to 50%. As for TawT experiments, all $X$ including $X_{\text{test}}$ is used to train models and only labels of $X_{\text{label}}$ can be used. Corresponding Boost Picking experiments, the index is also $X_{\text{label}}/X_{\text{rest}}$.

Generally, when $X_{\text{label}}/X_{\text{rest}}$ is more than 0.25, models can be trained close to or super than the corresponding baseline (shown in Fig 8). As a result, using Boost Picking can train a supervised classification model by 75% unlabeled data as well as the same model trained by 100% labeled data. When using supervised classification models as TawT, performances of these models are improved. In Fig 8, the green line (TawT result) is usually higher than the blue line (Boost Picking without test set).

In these experiments, the convergence generation is long because of the two poorly designed classifiers. To improve the efficiency of Boost Picking, the two classifiers should be designed as the guide in section 3.3.
References

Attenberg, Josh, Ipeirotis, Panagiotis G, and Provost, Foster J. Beat the machine: Challenging workers to find the unknown unknowns. Human Computation, 11:11, 2011.

Burton, Michele A, Brady, Erin, Brewer, Robin, Neylan, Callie, Bigham, Jeffrey P, and Hurst, Amy. Crowdsourcing subjective fashion advice using vizwiz: challenges and opportunities. In Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility, pp. 135–142. ACM, 2012.

Chapelle, Olivier, Schölkopf, Bernhard, Zien, Alexander, et al. Semi-supervised learning. 2006.

Fazakis, Nikos, Karlos, Stamatis, Kotsiantis, Sotiris, and Sgarbas, Kyriakos. Self-trained lmt for semisupervised learning. Computational Intelligence and Neuroscience, 2016, 2015.

Fei-Fei, L. Imagenet: crowdsourcing, benchmarking & other cool things. In CMU VASC Seminar, 2010.

Geman, Stuart, Bienenstock, Elie, and Doursat, René. Neural networks and the bias/variance dilemma. Neural computation, 4(1):1–58, 1992.

Haykin, Simon and Network, Neural. A comprehensive foundation. Neural Networks, 2(2004), 2004.

Hosmer Jr, David W and Lemeshow, Stanley. Applied logistic regression. John Wiley & Sons, 2004.

Huang, Gary B., Ramesh, Manu, Berg, Tamara, and Learned-Miller, Erik. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

James, Gareth M. Variance and bias for general loss functions. Machine Learning, 51(2):115–135, 2003.

Kalal, Zdenek, Matas, Jiri, and Mikolajczyk, Krystian. Pn learning: Bootstrapping binary classifiers by structural constraints. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 49–56. IEEE, 2010.

Kalal, Zdenek, Mikolajczyk, Krystian, and Matas, Jiri. Tracking-learning-detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 34(7):1409–1422, 2012.

Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Dollár, Piotr, and Zitnick, C Lawrence. Microsoft coco: Common objects in context. In Computer Vision–ECCV 2014, pp. 740–755. Springer, 2014.

Mohri, Mehryar, Rostamizadeh, Afshin, and Talwalkar, Ameet. Foundations of machine learning. MIT press, 2012.

Ogata, Katsuhiko. Modern control engineering. Prentice Hall PTR, 2001.

Rätsch, Gunnar, Onoda, Takashi, and Müller, K-R. Soft margins for adaboost. Machine learning, 42(3):287–320, 2001.

Sung, Kah Kay and Poggio, Tomaso. Example based learning for view-based human face detection. Pattern Analysis and Machine Intelligence IEEE Transactions on, 20(1):39–51, 1995.

Suykens, Johan AK and Vandewalle, Joos. Least squares support vector machine classifiers. Neural Processing Letters, 9(3):293–300, 1999.

Zhu, Xiaojin. Semi-supervised learning literature survey. 2005.

Zhu, Xiaojin and Goldberg, Andrew B. Introduction to semi-supervised learning. Synthesis lectures on artificial intelligence and machine learning, 3(1):1–130, 2009.