Boosting Decision Trees to Rank Static Analysis Reports

Athos Ribeiro
Advisor: Fabio Kon

Institute of Mathematics and Statistics
University of São Paulo

September 4, 2017

Outline

Outline

Continuous Source Code Static Analysis

- Monitors software repositories for new releases
- Run multiple security oriented static analysis tools on each new software release
- Universal static analysis report notation
- Filters potential flaws introduced in specific software versions
- Display warnings ordered according to their likelihood to point to actual software flaws

Ranking Static Analysis Reports

How to rank source code static analysis reports from multiple tools?

Outline

Ensemble Learning

- Supervised Learning
- Model Training
  - Select a good hypothesis through the space of possible hypotheses
- Ensemble Learning Method
  - Select a collection of hypotheses
  - Combine their predictions

Motivation

Majority-voting

Less likely that multiple hypotheses will misclassify a new example

Example

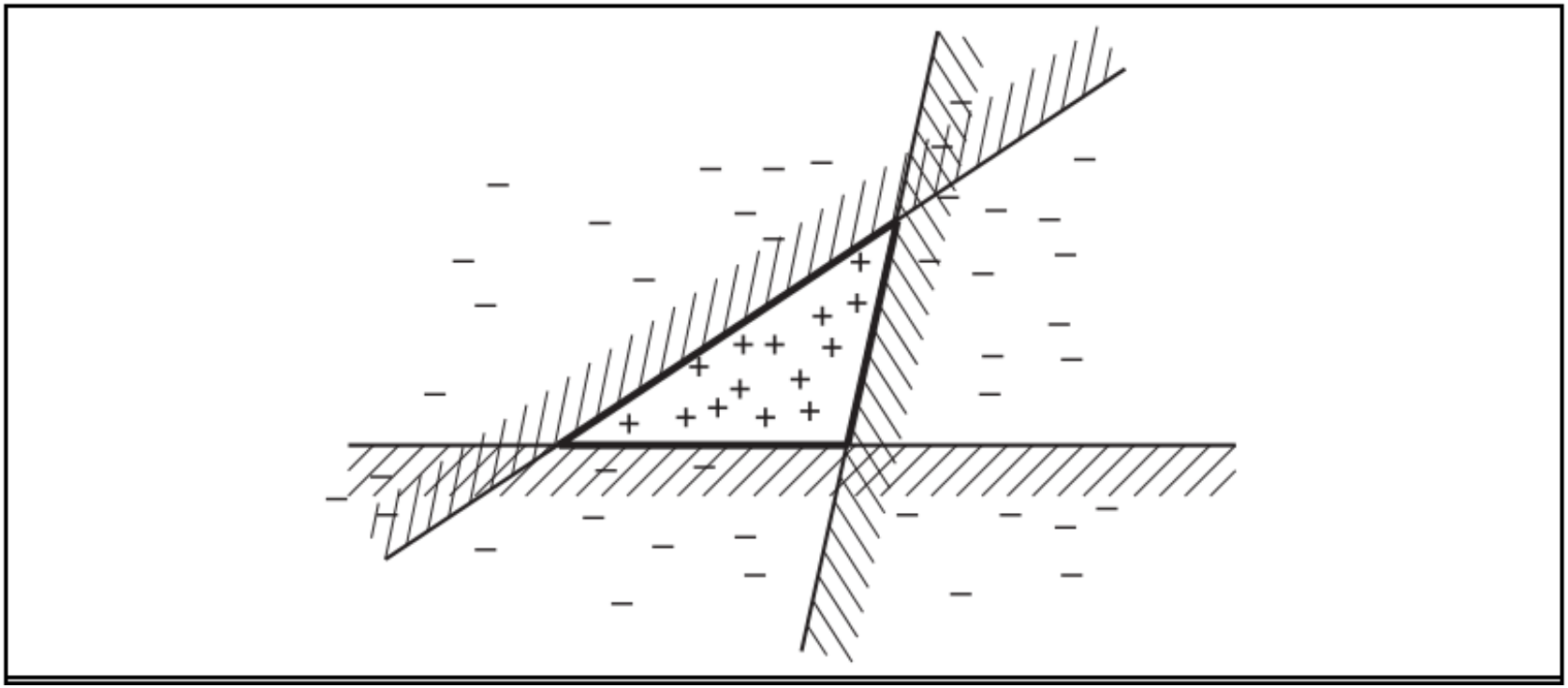

Figure: Ensemble Learning example (From [Russell and Norvig, 2002] p.749)

Outline

Boosting

- Widely used ensemble method
- Works with all sorts of classifiers
- Improve learners by focusing on misclassified examples
- Converts a series of **weak learners** into a strong learner
  - Hypotheses with accuracy slightly better than random guessing
- Applications:
  - Text categorization
  - Text filtering
  - **Ranking problems**
  - Classification problems in NLP

Boosting

- $h[-1, +1]$
- Weak classifier vs. Strong classifier

Boosting - Overview

- Runs a weak learning algorithm several times
- Each time in a different distribution of instances
- Generates several weak hypotheses
- Combine weak hypotheses into a single, more accurate hypothesis

Weighted Training Sets

- Each example in the training set has an associated weight $w_j \geq 0$
- The greater the example weight, the greater its importance during the learning of a hypothesis

Boosting idea

- All examples in the training set start with the same weight w_j
- From this first set, hypothesis h_1 is generated
  - h_1 misclassifies some examples. We Want the next hypothesis to perform better on those examples.
- The weight of the misclassified examples are increased and the weight of the examples classified correctly are decreased
- h_2 is generated from the new weighted training set
- Repeat until T hypotheses are generated
  - T is an input to the boosting algorithm
- The final ensemble hypothesis is a **weighted** majority combination of all T hypotheses, each weighted according to its performance on the training set

AdaBoost

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X$, $y_i \in Y = \{-1, +1\}$
Initialize $D_1(i) = 1/m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : X \to \{-1, +1\}$ with error

$$\epsilon_t = \Pr_{i \sim D_t} \left[h_t(x_i) \neq y_i \right].$$

- Choose $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$.
- Update:

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \\ &= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \end{aligned}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \text{sign} \left(\sum_{t=1}^{T} \alpha_t h_t(x) \right).$$

Figure: Adaboost Algorithm [Freund et al., 1999]

Outline

How to rank source code static analysis reports from multiple tools?

- Extract meaningful features from the static analysis reports
- Train decision tree based on such features. Review features until weak learner has error ≤ 0.5
- Boost decision tree
- Calibrate AdaBoost probabilities
- Use probabilities to rank warnings

Outline


Drucker, H. and Cortes, C. (1996).
Boosting decision trees.
In *Advances in neural information processing systems*, pages 479–485.


Freund, Y. (1995).
Boosting a weak learning algorithm by majority.
Information and computation, 121(2):256–285.


Freund, Y., Schapire, R., and Abe, N. (1999).
A short introduction to boosting.
Journal-Japanese Society For Artificial Intelligence, 14(771-780):1612.


Friedman, J., Hastie, T., and Tibshirani, R. (2001).
The elements of statistical learning, volume 1.
Springer series in statistics New York.


Russell, S. J. and Norvig, P. (2002).
Artificial intelligence: a modern approach.
{Pearson US Imports & PHIPEs}.