THE ANDREW & ERNA VITERBI
FACULTY OF
ELECTRICAL
ENGINEERING

SIPL
Signal and Image Processing Lab

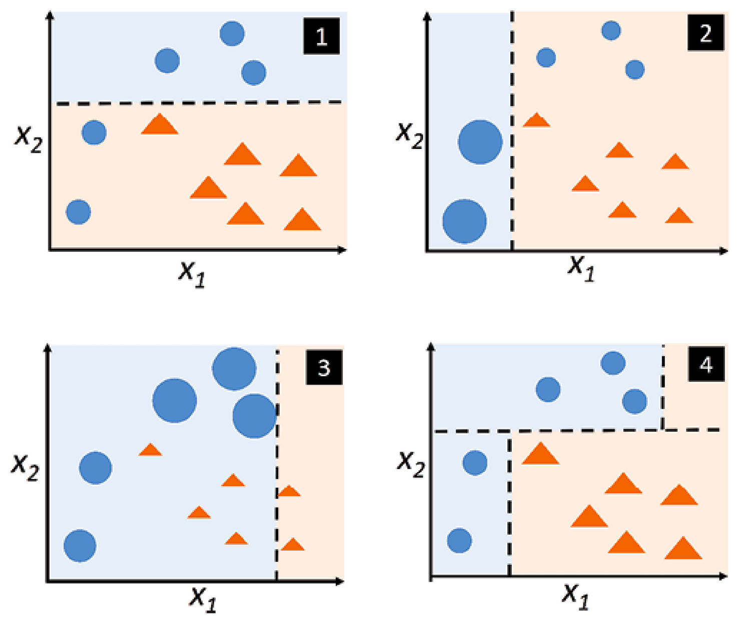
TECHNION
Israel Institute
of Technology

Error Resilient Real AdaBoost

Asaf Goren and Da-El Klang, Supervised by Yuval Ben-Hur

Introduction

- AdaBoost algorithms combine outputs from multiple weak classifiers to a final classification decision.
- The algorithms are sensitive to noise between the weak-classifiers and main computer.
- Previous work dealt with Discrete AdaBoost in noisy environment.



Example of weak-classifiers (1-3)
and the output of AdaBoost Algorithm (4)

Goals

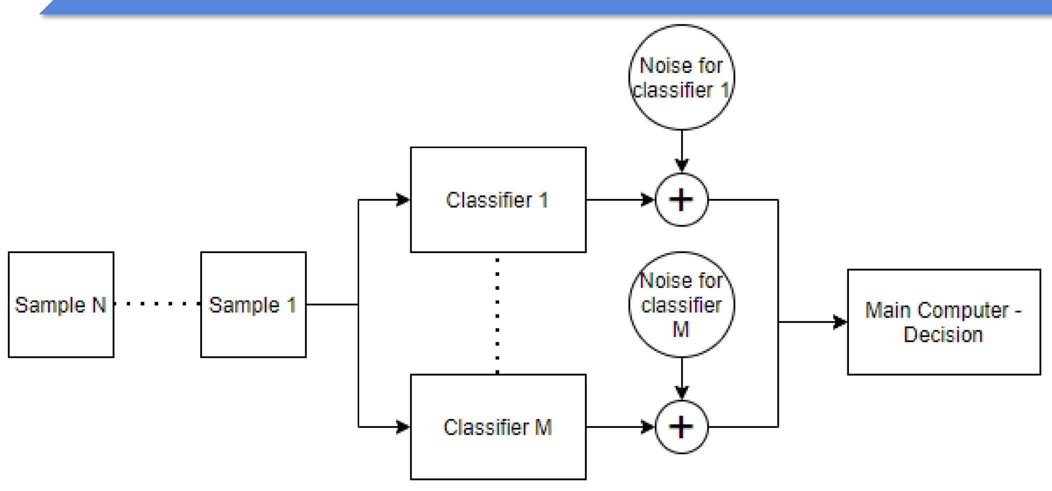
- In this project, we focus on Real AdaBoost and aim to:
  - Introduce a noisy channel framework and investigate algorithm's sensitivity in this framework.
  - Develop resilient modifications of the algorithm and test performance on real-world data.

Real AdaBoost Algorithm

- Initialize weights: $w_{1,1} = w_{2,1} = \cdots = w_{M,1} = \frac{1}{N}$
- For m in 1 to M:
  - Find a weak classifier, $f_m(x)$, that outputs a class probability estimate
    $f_m(x_i) = \Pr[y_i = 1|x_i]$, and minimizes
    $\epsilon_m = \sum_{i=1}^{N} w_{i,m} \cdot \exp\{-y_i H_m(x_i)\}$

    Where $H_m = \frac{1}{2} \cdot \ln\left\{\frac{1-f_m(x_i)}{f_m(x_i)}\right\}$
  - Update the weights
    $$w_{i,m+1} = w_{i,m} \exp\{-y_i H_m(x_i)\}$$
  - Renormalize the weights such that $\sum_{i=1}^{N} w_{i,m+1} = 1$
- The final classifier:
  $f(x_i) = sign\,\{\sum_{m=1}^{M} H_m(x_i)\}$

General Noise Model



Noise is added between the weak classifiers and the main computer

Real AdaBoost in AWGN

- Real AdaBoost original final classifier:
  $$f(x) = sign\{\sum_{m=1}^{M} H_m(x)\}$$
- Assuming AWGN channel between every weak classifier to the main computer:
  $$\widehat{H}_m(x_i) = H_m(x_i) + n_m, \qquad n_m \sim N(0, \sigma_m^2)$$
  $$\hat{f}(x) = sign\left\{\sum_{m=1}^{M} H_m(x_i) + n_m\right\}$$

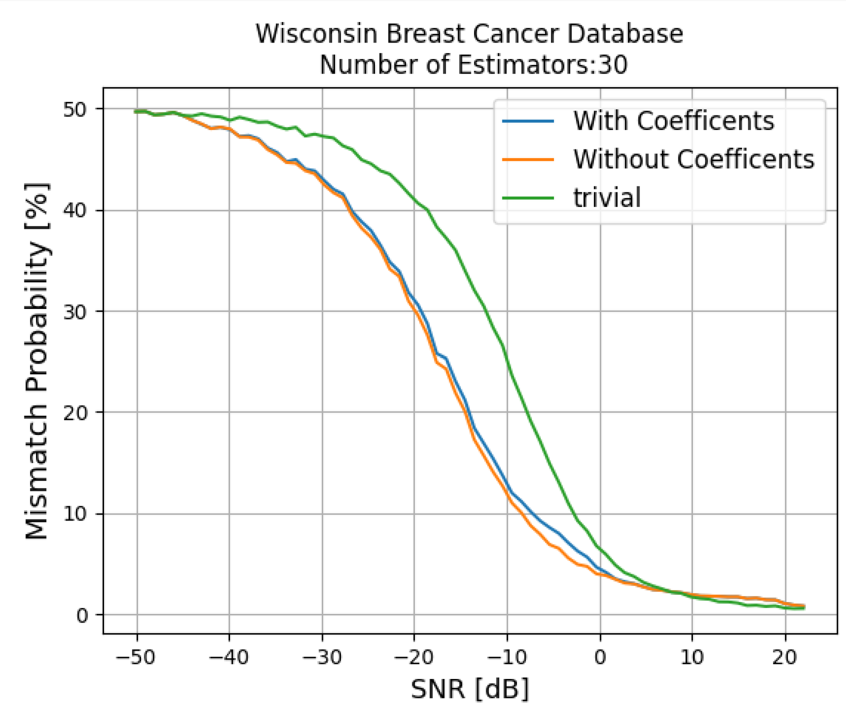
Adding Coefficients

- We suggest a ***resilient* Real AdaBoost** algorithm, by intervening in the final decision:
  $$\hat{f}_\alpha(x) = sign\left\{\sum_{m=1}^{M} \alpha_m \widehat{H}_m(x_i)\right\}$$
  $$= sign\left\{\sum_{m=1}^{M} \alpha_m H_m(x_i) + \alpha_m n_m\right\}$$
- The coefficients will depend both on the data and the noise characteristics.

Optimizing Coefficients

- Method 1: minimize the mismatch probably between the noisy decision and the noiseless decision:
  $$\min_{\alpha_m} \frac{1}{N} \sum_{i=1}^{N} \Pr\left[\hat{f}_\alpha(x_i) \neq f_\alpha(x_i)\right] =$$
  $$\min_{\alpha_m} \frac{1}{N} \sum_{i=1}^{N} Q\left(\frac{sign\{\sum_{m=1}^{M} \alpha_m H_m(x_i)\} \sum_{m=1}^{M} \alpha_m H_m(x_i)}{\sqrt{\sum_{m=1}^{M} \alpha_m^2 \sigma_m^2}}\right)$$
- Method 2: minimize the mismatch probability between the noisy decision and the **original** decision (without coefficients):
  $$\min_{\alpha_m} \frac{1}{N} \sum_{i=1}^{N} \Pr\left[\hat{f}_\alpha(x_i) \neq f(x_i)\right] =$$
  $$\min_{\alpha_m} \frac{1}{N} \sum_{i=1}^{N} Q\left(\frac{sign\{\sum_{m=1}^{M} H_m(x_i)\} \sum_{m=1}^{M} \alpha_m H_m(x_i)}{\sqrt{\sum_{m=1}^{M} \alpha_m^2 \sigma_m^2}}\right)$$

Results - AWGN



Comparisons of our suggested algorithms and the original algorithm on Wisconsin Breast Cancer database

AWGN - Conclusions

- We successfully improved the performance of Real AdaBoost in a noisy environment by adding coefficients to the weak classifiers.
- Minimizing the mismatch probability w.r.t the original decision resulted in slightly better performance than minimizing the mismatch probability w.r.t the weighted decision.

Quantized Real AdaBoost

- We tried two different approaches:
  - Find optimal number of bits for each classifier
  - Find optimal joint quantizer levels for all classifiers
- We assume that the total number of bits is constrained.

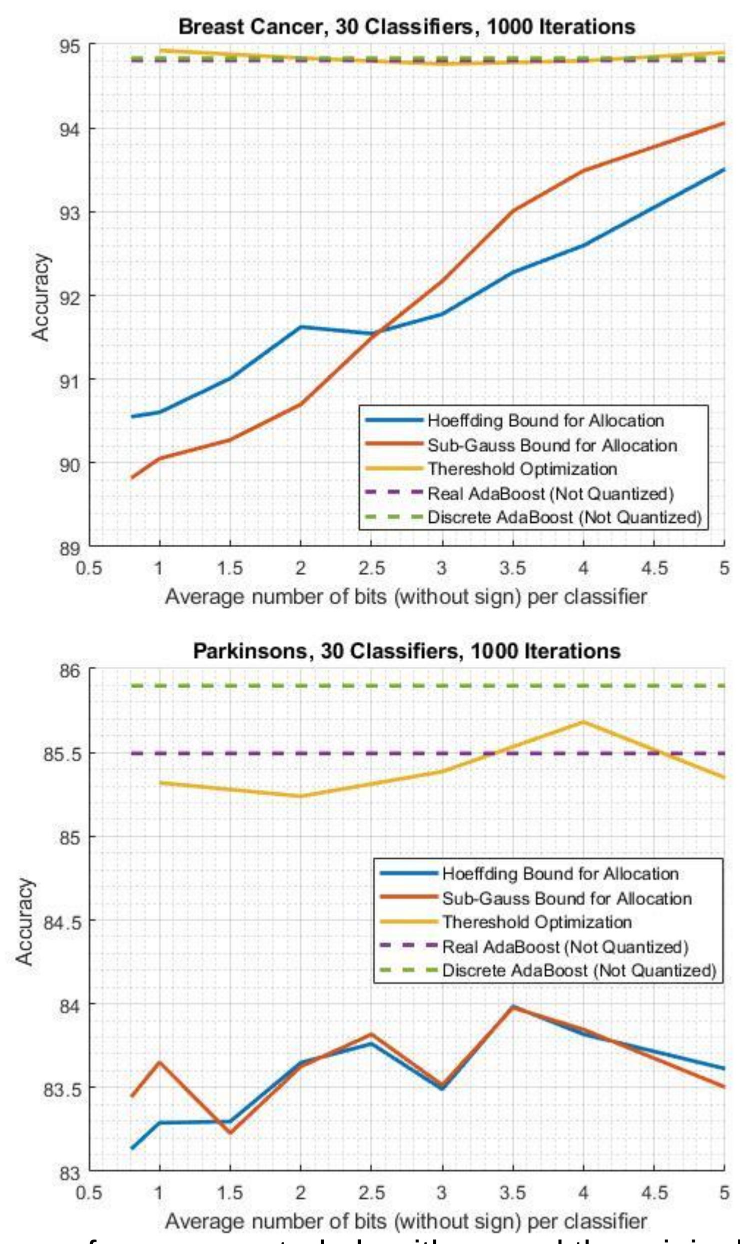
Bits Allocation

- Quantization noise can be simulated as additive uniform noise, that decreases as we allocate more bits.
- Using Hoeffding inequality and Sub-Gaussian proxy, we bound the mismatch probability.
- We minimize the proxies by continuous optimization methods, and round the solution to the closest integer.

Optimal Threshold Allocation

- In this approach, we control the quantizer levels, and the weak classifiers coefficients.
- The optimal initialization of the levels is given by K-Means algorithm.
- We solve the problem numerically and find the optimal thresholds and coefficients jointly.

Results - Quantization



Comparisons of our suggested algorithms and the original algorithms on Wisconsin Breast Cancer and Parkinson Disease database

Quantization - Conclusions

- Thresholds Optimization approach achieved:
  - Better results than the trivial solution.
  - Similar performance to Real-AdaBoost with only 2 bits per classifier.
- The bound-based solutions performance weren't as good.

April 2021