CSE 473

Ensemble Learning

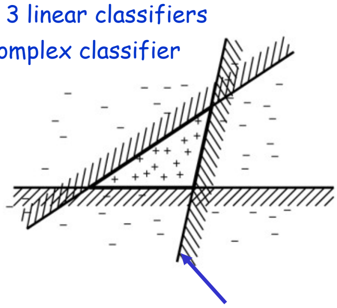
Ensemble Learning

- Sometimes each learning technique yields a different hypothesis (or function)
- But no perfect hypothesis…
- Could we combine several imperfect hypotheses to get a better hypothesis?

Example

Combining 3 linear classifiers

⇒ More complex classifier



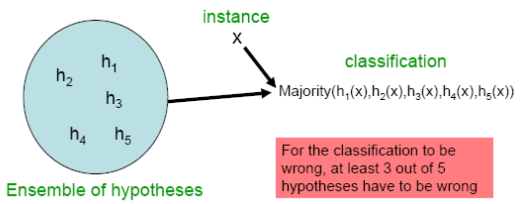
this line is one simple classifier saying that everything to the left is + and everything to the right is -

Ensemble Learning: Motivation

- Analogies:
  - Elections combine voters' choices to pick a good candidate (hopefully)
  - Committees combine experts' opinions to make better decisions
  - Students working together on Othello project
- Intuitions:
  - Individuals make mistakes but the "majority" may be less likely to
  - Individuals often have partial knowledge; a committee can pool expertise to make better decisions

Technique 1: Bagging

- Combine hypotheses via majority voting



Bagging: Analysis

- Assumptions:
  - Each h_i makes error with probability p
  - The hypotheses are independent

- Majority voting of n hypotheses:
  - k hypotheses make an error: $\binom{n}{k} p^k(1-p)^{n-k}$
  - Majority makes an error: $\Sigma_{k>n/2} \binom{n}{k} p^k(1-p)^{n-k}$
  - With n=5, p=0.1 ➔ err(majority) < 0.01

Error probability went down from 0.1 to 0.01!

Weighted Majority Voting

- In practice, hypotheses rarely independent

- Some hypotheses have less errors than others ⇒ all votes are not equal!

- Idea: Let's take a weighted majority

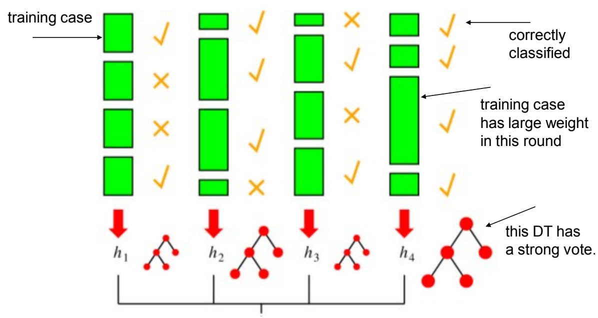
Technique 2: Boosting

- Most popular ensemble learning technique
  Computes a weighted majority of hypotheses
  Can "boost" performance of a "weak learner"

- Operates on a weighted training set
  Each training example (instance) has a "weight"
  Learning algorithm takes weight of input into account

- Idea: when an input is misclassified by a hypothesis, increase its weight so that the next hypothesis is more likely to classify it correctly

Boosting Example with Decision Trees (DTs)



Output of h_{final} is weighted majority of outputs of $h_1, \dots, h_4$

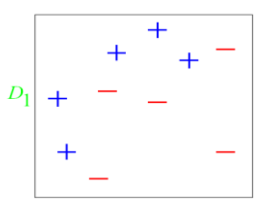
AdaBoost Algorithm

(Adaptive Boosting)

- $w_j \leftarrow 1/N \quad \forall_j$
- For m=1 to M do
  - $h_m \leftarrow$ learn(dataset,w)
  - err $\leftarrow$ 0
  - For each (x_j, y_j) in dataset do
    - If $h_m(x_j) \neq y_j$ then err $\leftarrow$ err + w_j
  - For each (x_j, y_j) in dataset do
    - If $h_m(x_j) = y_j$ then $w_j \leftarrow w_j$ err / (1-err)
  - w $\leftarrow$ normalize(w)
  - $z_m \leftarrow$ log [(1-err) / err]
- Return *weighted-majority(h,z)*

w: vector of N instance weights
z: vector of M hypoth. weights

AdaBoost Example

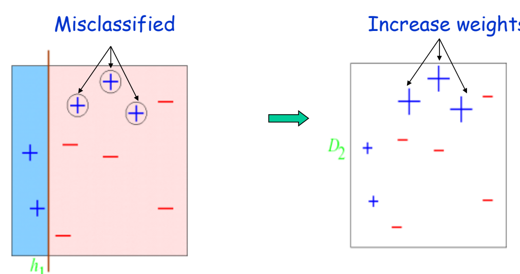


Original training set D_1 : Equal weights to all training inputs
Goal: In round t, learn classifier h_t that minimizes error with respect to weighted training set
h_t maps input to True (+1) or False (-1) $h_t : X \rightarrow \{-1, +1\}$

Taken from "**A Tutorial on Boosting**" by Yoav Freund and Rob Schapire

AdaBoost Example

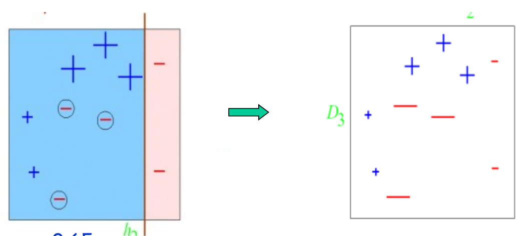
ROUND 1



z_1 = 0.42

AdaBoost Example

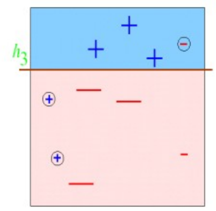
ROUND 2



$z_2 = 0.65$

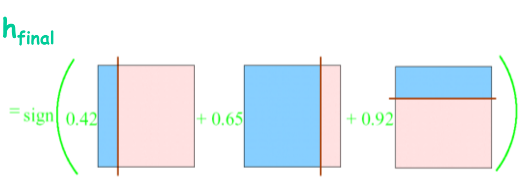
AdaBoost Example

ROUND 3



$z_3 = 0.92$

AdaBoost Example



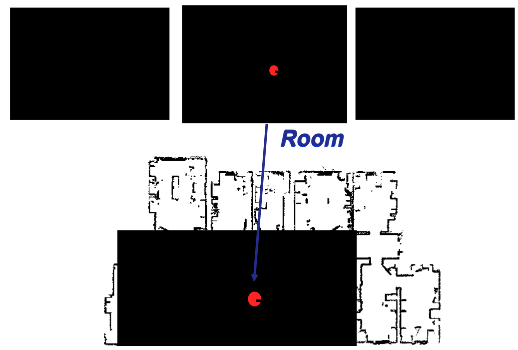
sign(x) = +1 if x > 0 and -1 otherwise

Example 1: Semantic Mapping

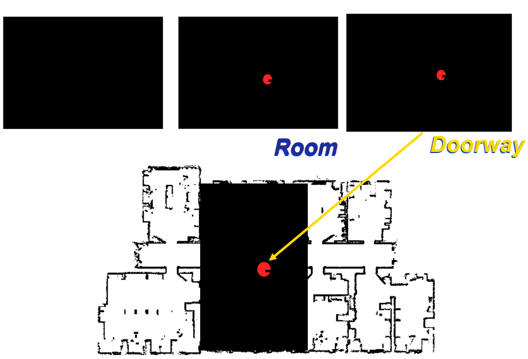
Motivation

- *Human-Robot interaction:*
  - ***User:*** *"Go to the corridor"*

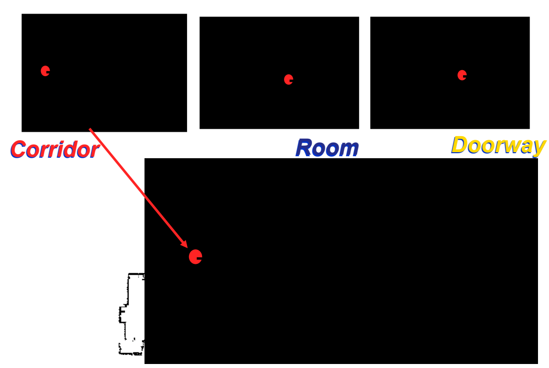
Shape



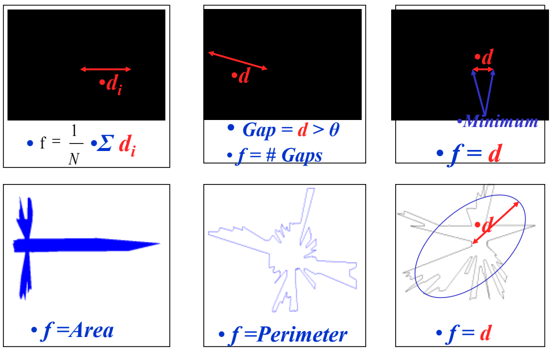
Observations



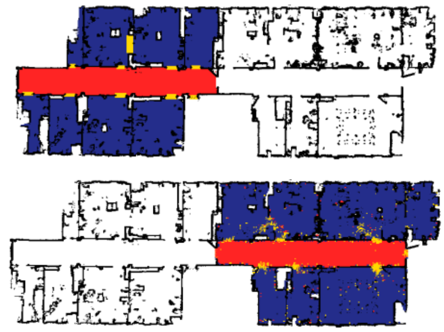
Observations



Simple Features



Experiments



Training (top)
examples: 16045

•Test (bottom)
•# examples:
•18726
•classification:
•93.94%

•Building 079 Uni. Freiburg

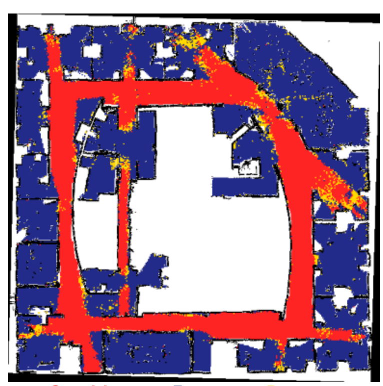
•Corridor •Room •Doorway

Application to a New Environment


Training map

Intel Research Lab in Seattle

Application to a New Environment

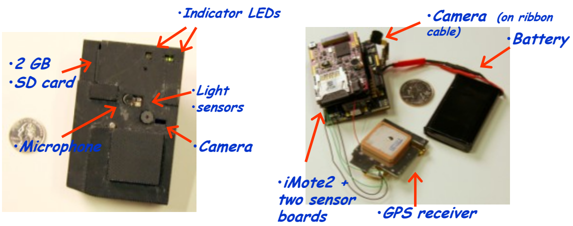



Training map

Intel Research Lab in Seattle

•Corridor •Room •Doorway

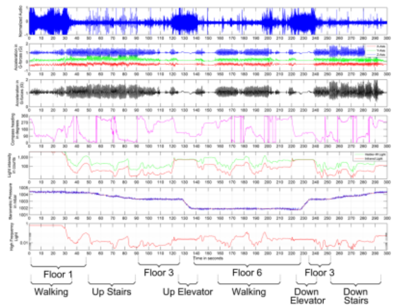
Example 2: Wearable Multi-Sensor Unit



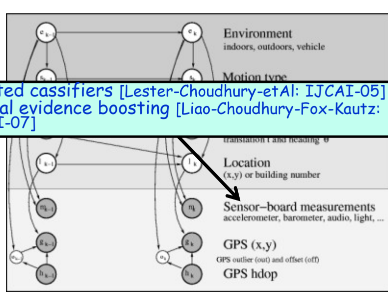
- Records 4 hours of audio, images (1/sec), GPS, and sensor data (accelerometer, barometric pressure, light intensity, gyroscope, magnetometer)

Data Stream

Courtesy of G. Borriello



Activity Recognition Model



- Boosted cassifiers [Lester-Choudhury-etAl: IJCAI-05]
- Virtual evidence boosting [Liao-Choudhury-Fox-Kautz: IJCAI-07]

•*Accuracy: 88% activities, 93% environment*

Boosting

- Extremely flexible framework
- Handles high-dimensional continuous data
- Easy to implement

- Limitation:

  Only models local classification problems