CSE 571
Probabilistic Robotics

Boosting

Some slides taken from: Wolfram Burgard, Hongbo Deng, Antonio Torralba

Why boosting?

- A simple algorithm for learning robust classifiers
  - Freund & Shapire, 1995
  - Friedman, Hastie, Tibshhirani, 1998
- Provides efficient algorithm for sparse feature selection
  - *Tieu & Viola, 2000*
  - *Viola & Jones, 2003*
- Easy to implement, doesn't requires external optimization tools.

Motivation

- Indoor mapping is an important task in mobile robotics.

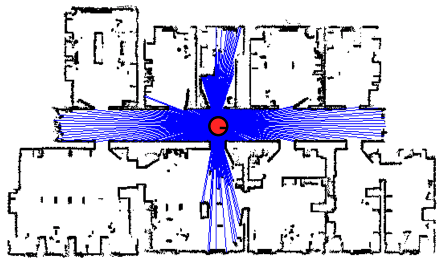
Motivation

- Semantic mapping:

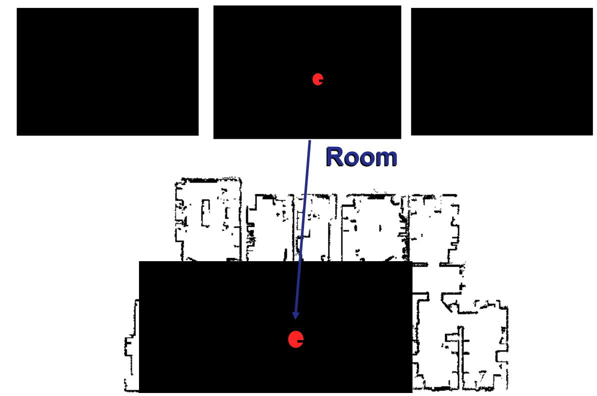
- Human-Robot interaction:
  User: "Go to the corridor"

Goal

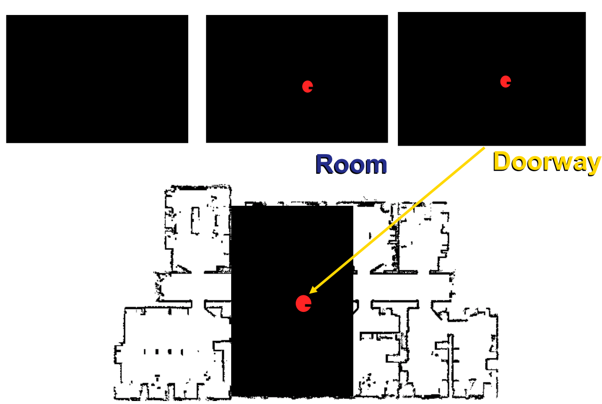
- Classification of the position of the robot using one single observation: a 360° laser range data.



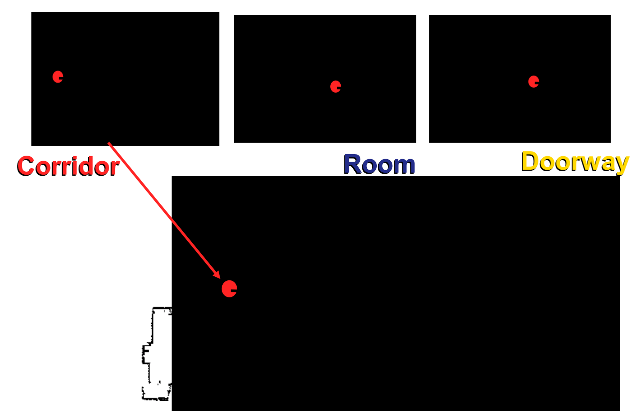
Observations



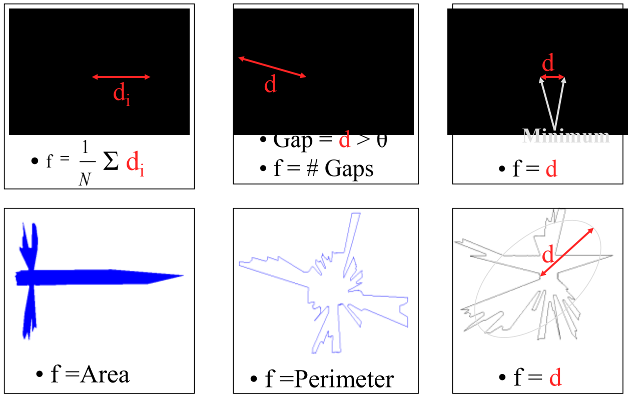
Observations



Observations



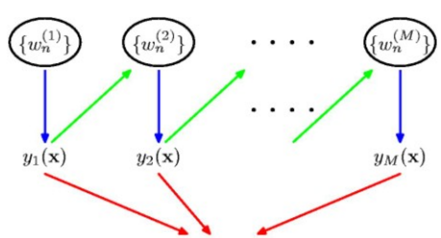
Simple Features



Combining Features

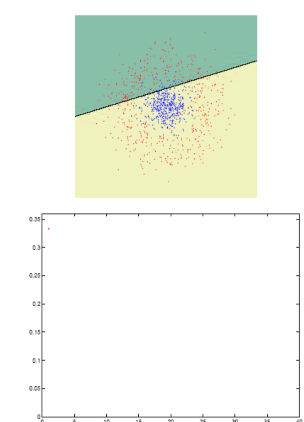
- Observation:
  There are many simple features f_i.
- Problem:
  Each single feature f_i gives poor classification rates.
- Solution:
  Combine multiple simple features to form a final classifier using AdaBoost.

Idea



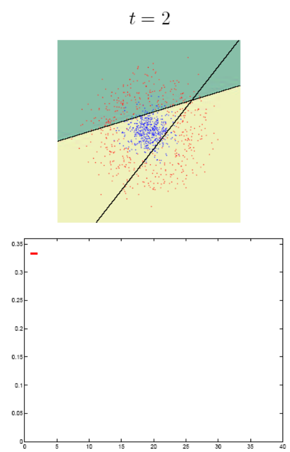
$$Y_M(\mathbf{x}) = \operatorname{sign}\left(\sum_m^M \alpha_m y_m(\mathbf{x})\right)$$

Example



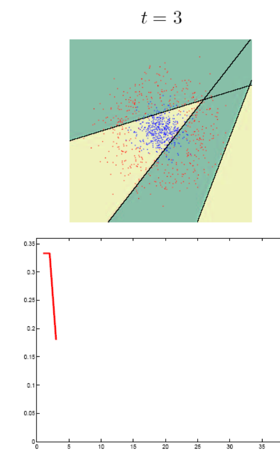
12

Example



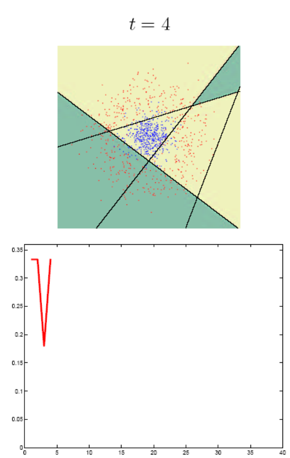
13

Example



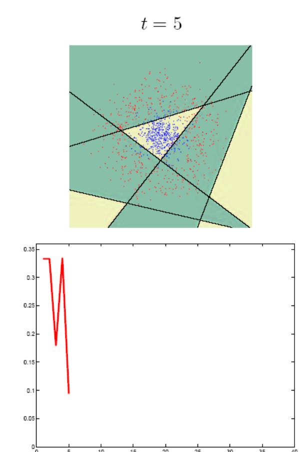
14

Example



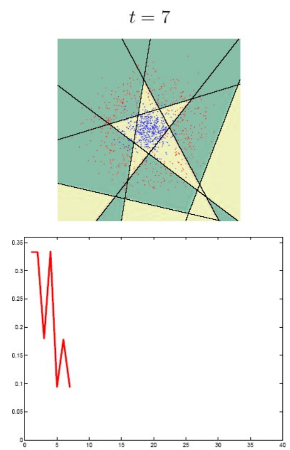
15

Example



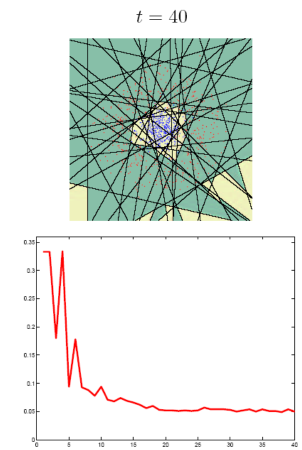
16

Example



17

Example



18

AdaBoost Algorithm

1. Initialize data weights $w_n^{(1)} = 1/N$
2. For $m = 1, \ldots, M$,
   2.1 Fit a classifier $y_m(x)$ by minimizing
   $J_m = \sum w_n^{(m)} I(y_m(\mathbf{x}_n \neq t_n))$
   2.2 Evaluate quantity ϵ_m (ratio of misclassified)
   $\epsilon_m = \frac{\sum_n w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n)}{\sum_n w_n^{(m)}}$
   Evaluate quantity α_m (weight for classfier m)
   $\alpha_m = \log(\frac{1-\epsilon_m}{\epsilon_m})$
   2.3 Update the data weighting coefficients
   $w_n^{(m+1)} = w_n^{(m)} e^{\alpha_m I(y_m(\mathbf{x}_n) \neq t_n)}$
3. Make prediction by $Y_M(\mathbf{x}) = \mathrm{sign}(\sum_{m=1}^{M} \alpha_m y_m(\mathbf{x}))$

Sequential Optimization

- Original motivation in learning theory
- Can be derived as sequential optimization of exponential error function

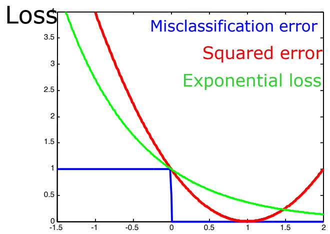
$E = \sum_{n=}$

with

$f_m(x) = \frac{1}{2} \sum_{l=}$

- Fixing first m-1 classifiers and minimizing wrt m-th classifier gives AdaBoost equations!

Exponential loss



Squared error

$$J = \sum_{t=1}^{N} [y_t - F(x_t)]^2$$

Exponential loss

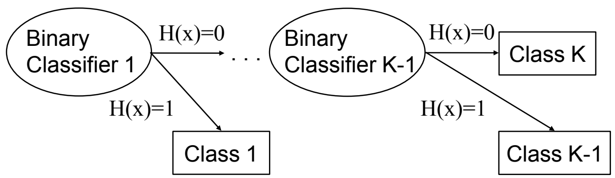
$$J = \sum_{t=1}^{N} e^{-y_t F(x_t)}$$

Further Observations

- Test error often decreases even when training error is already zero
- Many other boosting algorithms possible by changing error function
- Regression
- Multiclass

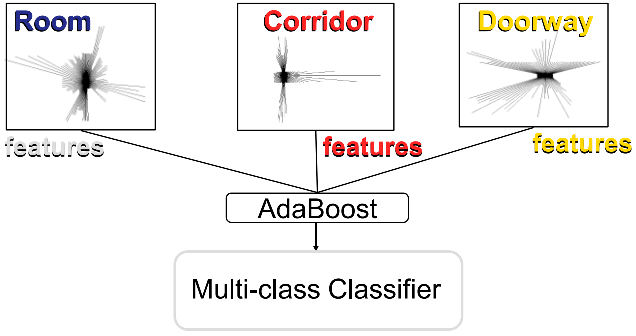
Multiple Classes

- Sequential AdaBoost for K different classes.



Summary of the Approach (1)

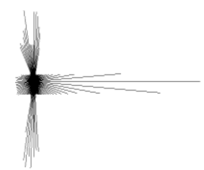
- Training: learn a strong classifier using previously labeled observations:



Summary of the Approach (2)

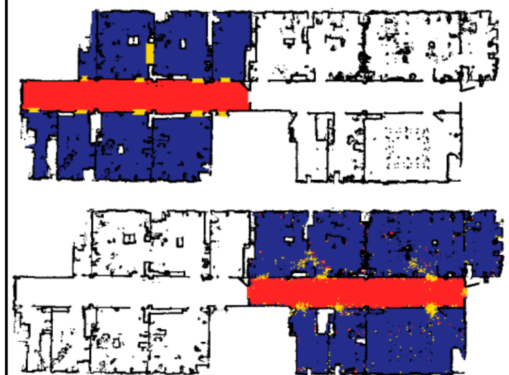
- Test: using the classifier to classify new observations:

New observation



features → Multi-class Classifier → **Corridor**

Experiments



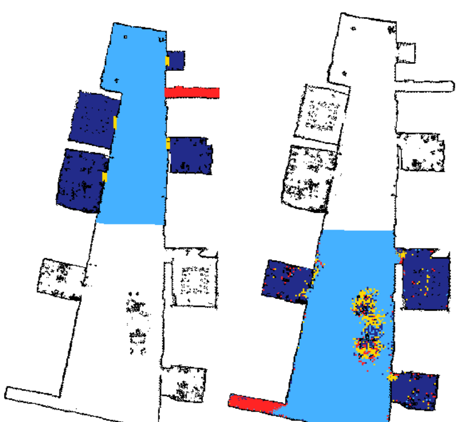
Training (top)
examples:
16045

Test (bottom)
examples:
18726
classification:
93.94%

Building 079
Uni. Freiburg

Corridor Room Doorway

Experiments



Training (left)
examples:
13906

Test (right)
examples:
10445
classification:
89.52%

Building 101
Uni. Freiburg

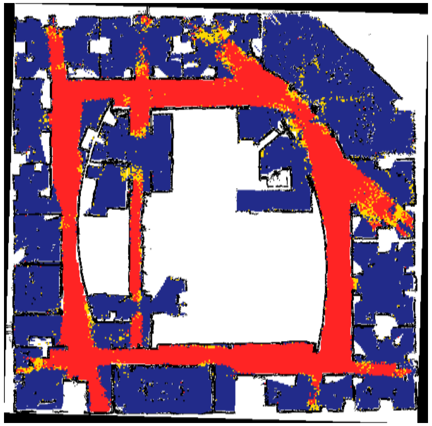
Corridor Room Doorway Hallway

Application to a New Environment

Training map

Intel Research Lab in Seattle

Application to a New Environment



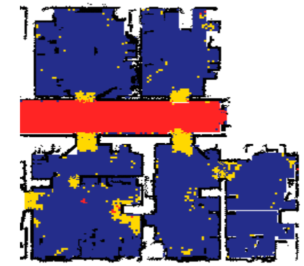
Training map

Intel Research Lab in Seattle

Corridor Room Doorway

Sequential AdaBoost vs AdaBoost.M2

Sequential AdaBoost classification: 92.10 %



AdaBoost.M2 classification: 91.83 %

Corridor Room Doorway

Boosting

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

- Limitation:
  - Only models local classification problems