

# ShareBoost: Boosting for Multi-View Learning with Performance Guarantees

Jing Peng  
Computer Science Department  
Montclair State University

Costin Barbu  
Air and Missile Defense Technology  
MIT Lincoln Laboratory

Guna Seetharaman  
Information Directorate  
AFMC AFRL/RITB

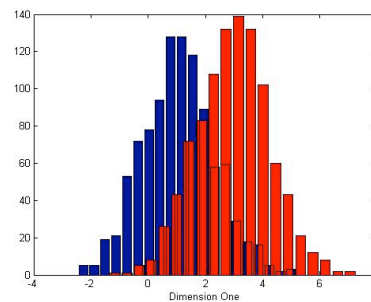
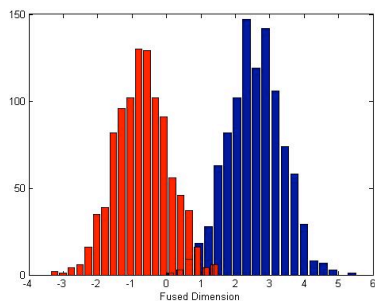
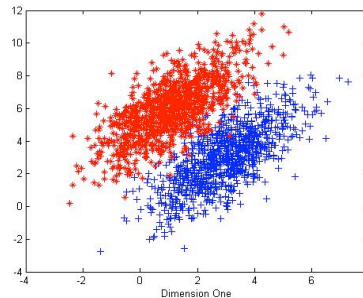
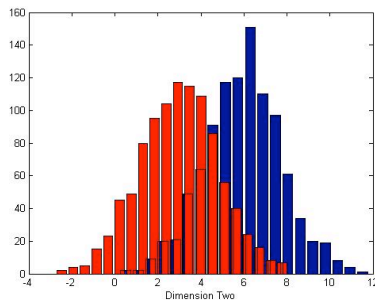
Wei Fan & Xian WU  
T. J. Watson Research  
IBM

Kannappan Palaniappan  
Computer Science Department  
University of Missouri-Columbia

European Conference on Machine Learning  
Athens, Greece September 2011

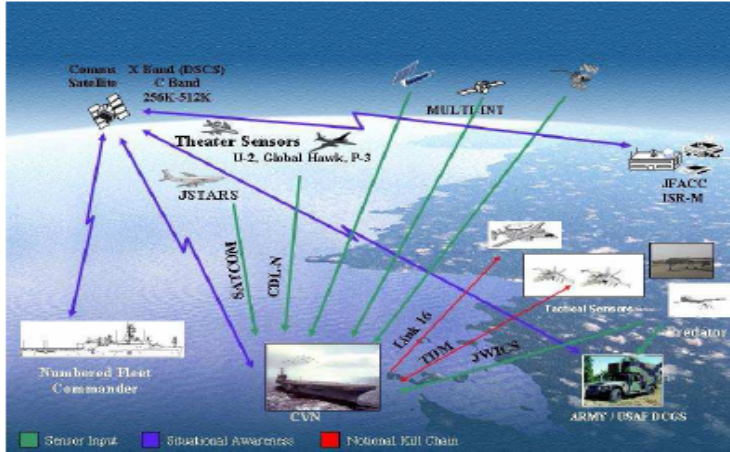
# Motivation

- ❑ **Capture information from multi-sources**
  - Exponentially quicken identification and classification
- ❑ **Mine most relevant information sources**
  - Robust against noise
  - Reduced computation



- ❑ 2D example: relying on either dimension one (lower right) or dimension two (upper left) results in significant overlapping.
- ❑ Fused dimension (lower left) of dimension one and dimension two results in less overlap between two classes.
- ❑ Goal is to find right combination of source information.

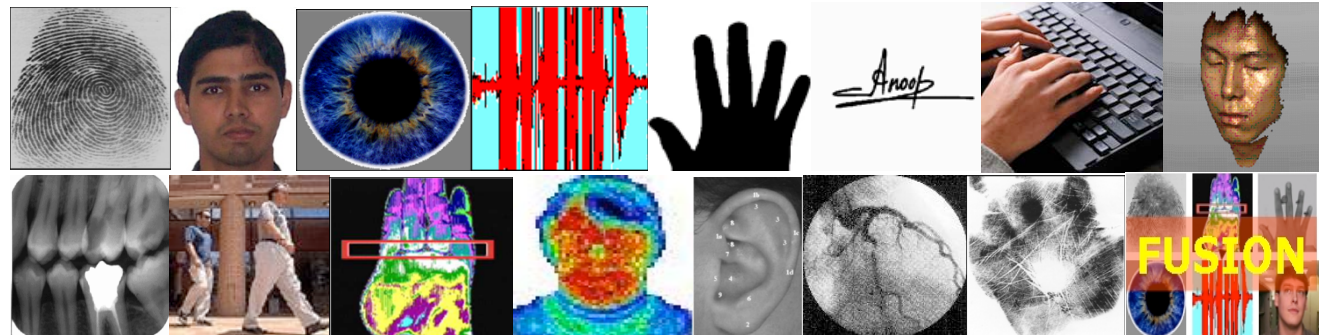
# Applications



- Multi-sensor data fusion for target-clutter discrimination
  - Infrared
  - SAR
  - Digital images

- Multi-modal biometrics data fusion for user identification

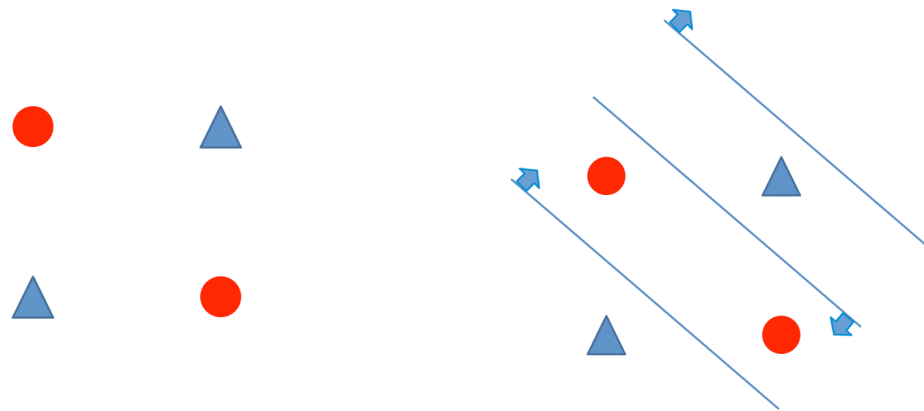
- Fingerprints
- Face
- Iris
- Voice



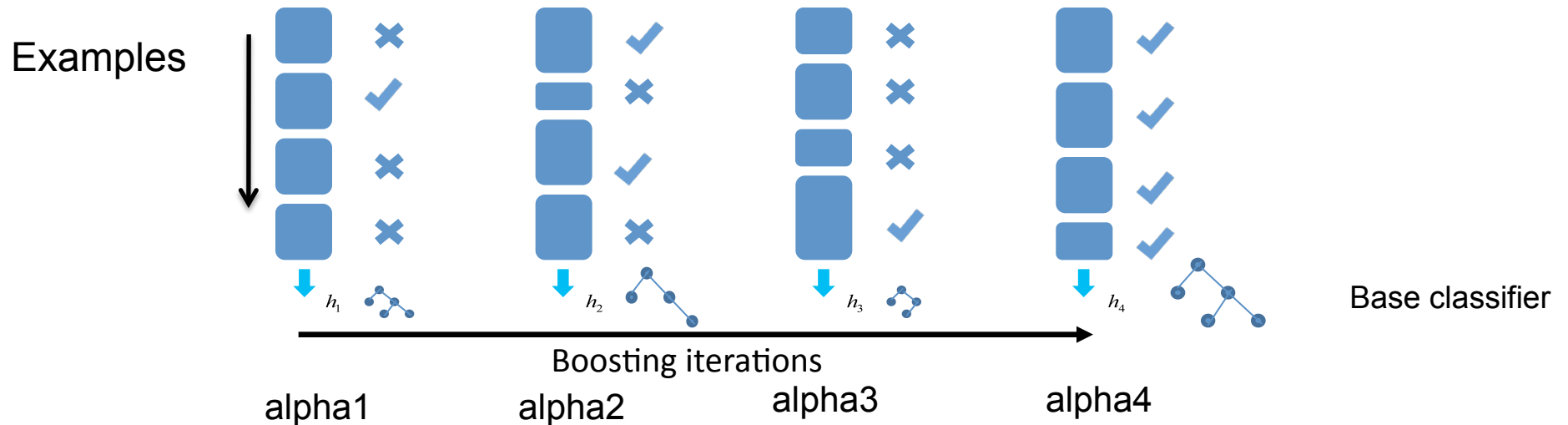
# New Algorithm for Multi-View Learning

## ShareBoost: Boosting based learning algorithm

- ❑ Idea: combine multiple simple classifiers to generate a powerful one.
- ❑ Well known example:
  - no single linear classifier can classify the example (lower left) correctly.
  - a combination of linear classifiers can make correct prediction (lower right). I.e., two red circles are on the positive side (indicated by arrow) of at least two classifiers, while each blue triangle is always on the negative side of at least two classifiers.



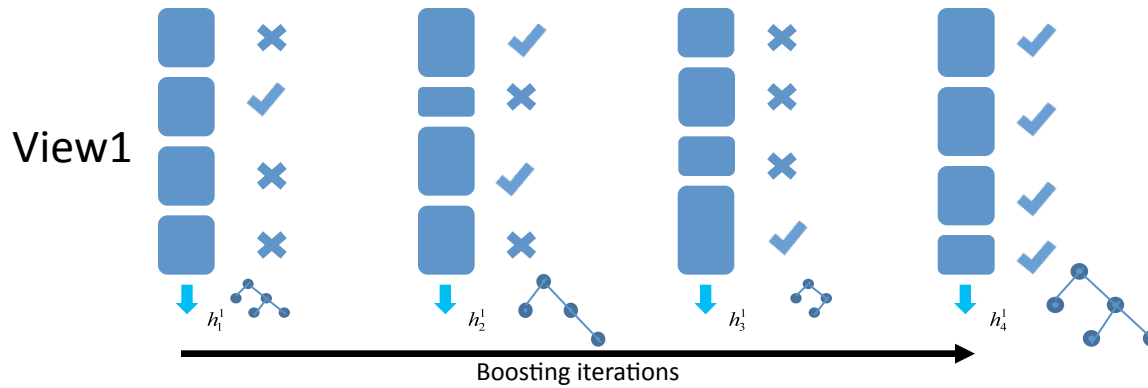
# AdaBoost



- ❑ Each rectangle represents an example
- ❑ Size of rectangles represents weight assigned to the example
- ❑ ✓ denotes the example is classified correctly
- ❑ ✗ denotes the example is classified incorrectly
- ❑ When an example is misclassified, its weight will be increased at the next iteration, while weight of a correctly classified example will be decreased. Thus, difficult or misclassified examples receive more attention
- ❑ Size of a tree indicates its contribution (correctness) to final classification
- ❑ Final classifier consists of a weighted combination of base classifiers  $h$ , where weight  $\alpha$  reflects the size/quality of the trees

$$H = \alpha_1 h_1 + \alpha_2 h_2 + \alpha_3 h_3 + \alpha_4 h_4$$

# AdaBoost Applied Independently



$$H^1 = \alpha_1^1 h_1^1 + \alpha_2^1 h_2^1 + \alpha_3^1 h_3^1 + \alpha_4^1 h_4^1$$

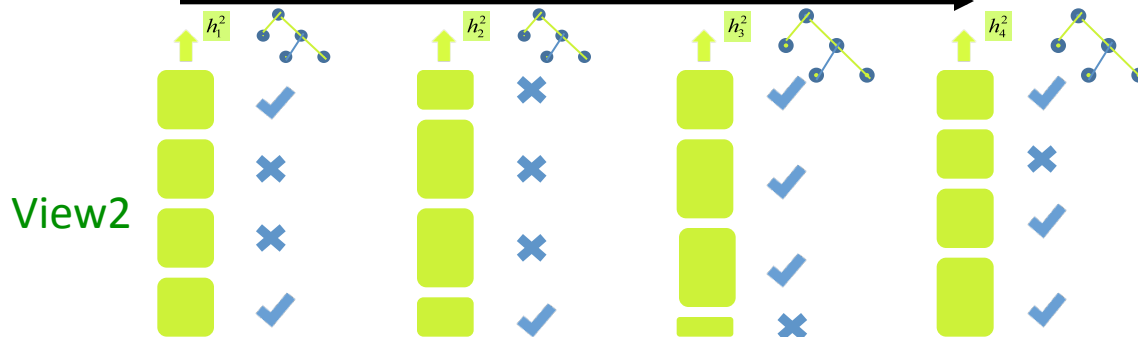
$$H^2 = \alpha_1^2 h_1^2 + \alpha_2^2 h_2^2 + \alpha_3^2 h_3^2 + \alpha_4^2 h_4^2$$

- AdaBoost can be applied to each view independently
- Final classifiers from each view are combined

$$H = H^1 + H^2$$

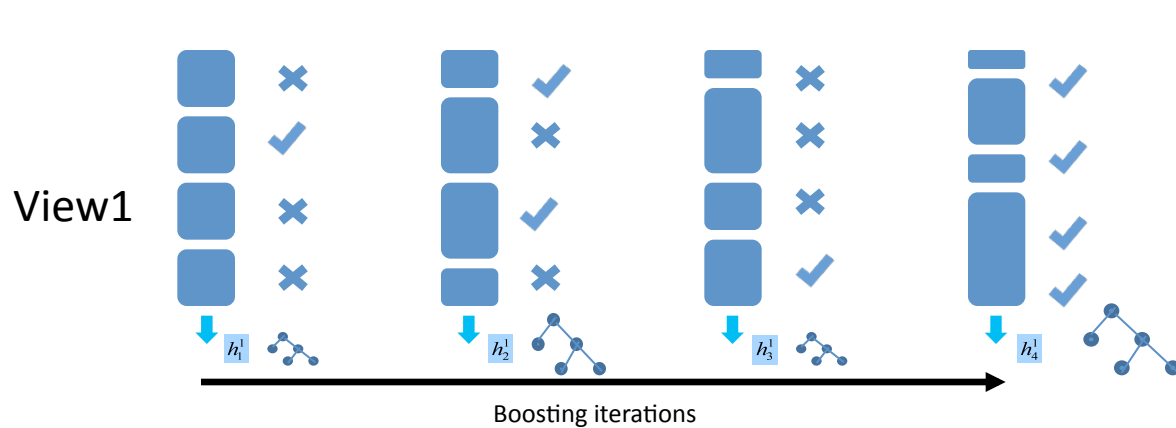
- Weight associated with each example is computed independently for each view

**Drawback:** it cannot exploit interplay between views



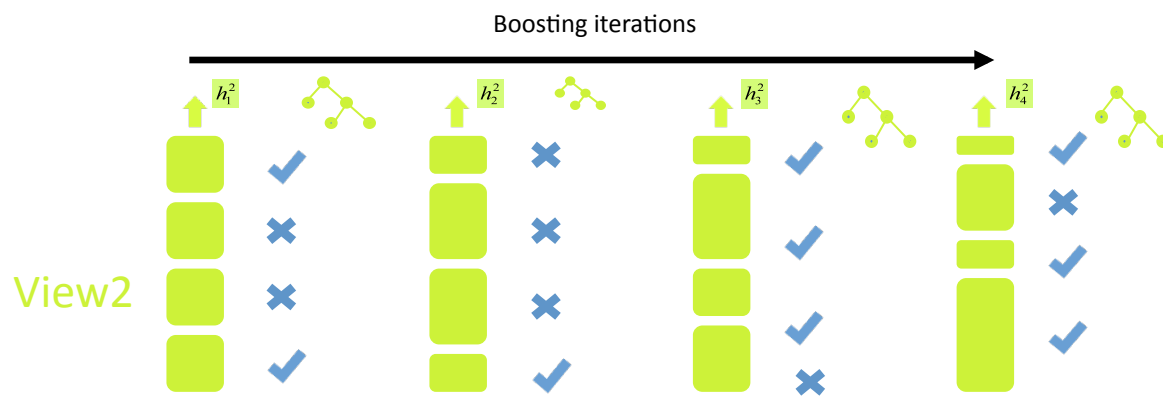
Each view has 4 base classifiers built independently. No linking between views. Total of 8 classifiers for the final H (multiple ways the classifiers can be combined).

# ShareBoost: Exploit Interplay between Views



- Similar to AdaBoost, boosting is applied to each view
- However, weight associated with each example is shared by all views
- This share mechanism allows ShareBoost to exploit interplay between views

$$H = \alpha_1^2 h_1^2 + \alpha_2^1 h_2^1 + \alpha_3^2 h_3^2 + \alpha_4^1 h_4^1$$



- Final classifier is assembled flexibly from base classifiers that provide best classification
- ShareBoost requires fewer base classifiers, thus less complex, providing better generalization

# Randomized ShareBoost (rShareBoost)

## ShareBoost

- Robust against noise as a result of shared weight mechanism
- Rather complex:  $MT O(\max\{q, \log(n)\} n \log(n))$ —trees as base classifiers
  - $n$ : number of training examples
  - $q$ : number of dimensions
  - $T$ : number of boosting iterations
  - $M$ : number of views
- Each base classifier (decision tree) has time complexity  $O(\max\{q, \log(n)\} n \log(n))$ .  
 $T$  iterations result in complexity  $T O(\max\{q, \log(n)\} n \log(n))$ . This must be done along  $M$  views, resulting in  $MT O(\max\{q, \log(n)\} n \log(n))$

## rShareBoost

- Probabilistically chooses a view to perform boosting
- Convergence can be proven theoretically using armed bandit approach
  - Efficient in reducing training error
  - Reduced complexity:  $T O(\max\{q, \log(n)\} n \log(n))$



# Randomized ShareBoost (rShareBoost)

- Edge of a base classifier

$$\beta_t = \sum_i w_i(t) y_i h_t(x_i) = E_{i \sim W(t)}[y_i h_t(x_i)]$$

- Training error (over T pulls)  $\leq \prod_{t=1}^T \sqrt{1 - \beta_t^2}$

- Reward function

$$r_t(j) = 1 - \sqrt{1 - \beta_t^2(V_j)}$$

*Theorem:* Let  $V = \{V_1, \dots, V_M\}$  be  $M$  views

There exist  $V_i \in V$  and  $0 < \rho \leq 1$ , s.t. for any distribution over training data, base learner returns a classifier from  $V_{i+}$  with edge  $\beta_{V_{i+}} \geq \rho$ . Then, with probability at least  $1 - \delta$ , training error of rShareBoost will become 0 after at most in time polynomial in

$$(\log(M/\delta), 1/\rho, \log n)$$

# Randomized ShareBoost (rShareBoost)

- ❑ Use armed bandit approach to prove convergence
- ❑ How to measure performance?
  - Regret
- ❑ Weak regret

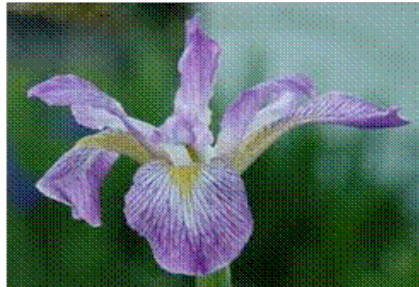
$Reg_A(T) = G_{max}(T) - G_A(T)$  Regret in selecting AlgoA

$$G_{max}(T) = \max_i \sum_{t=1}^T r_t(i)$$

$$G_A(T) = \sum_{t=1}^T r(j_t) \quad \text{Total reward Algo A receives over T pulls}$$

- ❑ Reward obtained by algorithm A is compared to best fixed arm
- ❑ Use bounds on weak regret (Auer,Cesa-Bianchi,Freund&Schapire,SIAM Comp.'02) to establish convergence for rShareBoost

# Simple Illustration – Iris Flowers



virginica



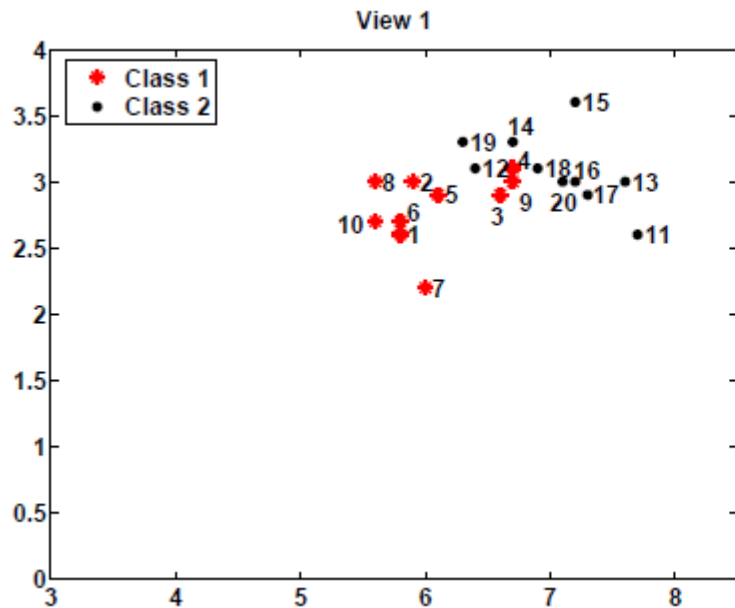
sibirica



- ❑ Each flower is represented by two views:
  - Sepal and Petal
- ❑ Each view has two features:
  - Length
  - Width

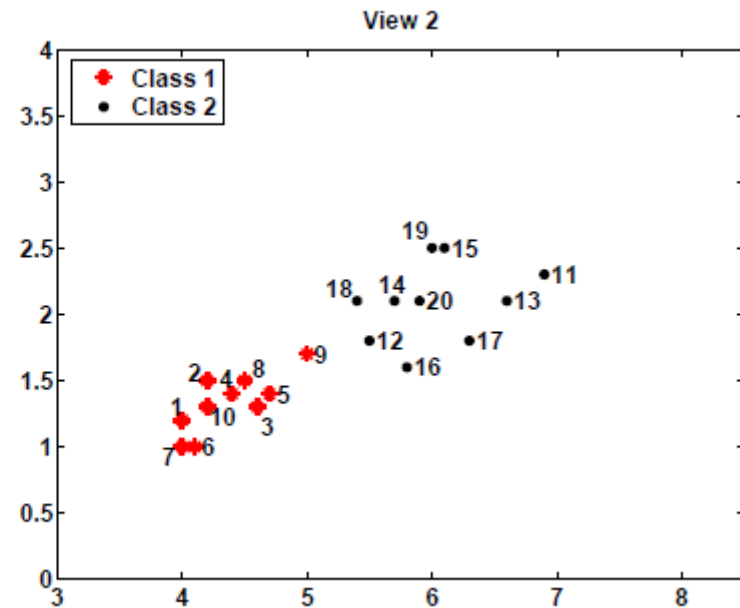
# Simple Illustration

Two views of Iris data



Sepal

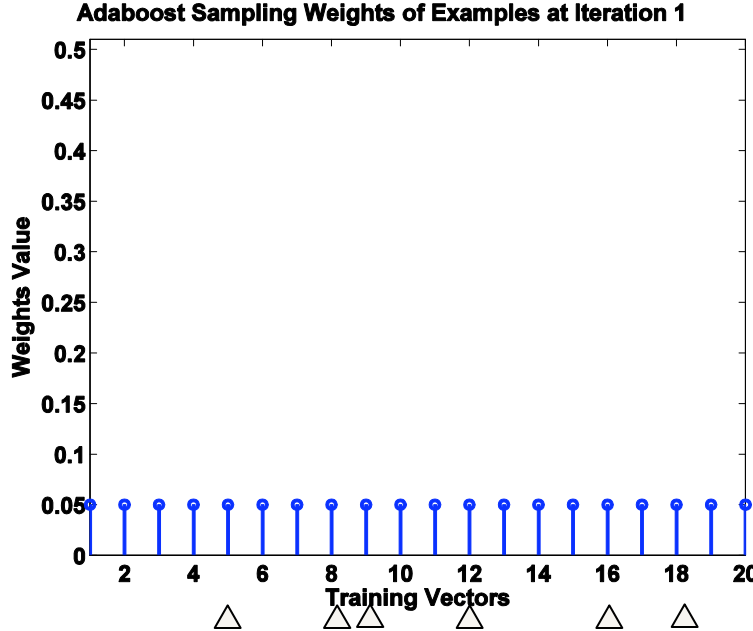
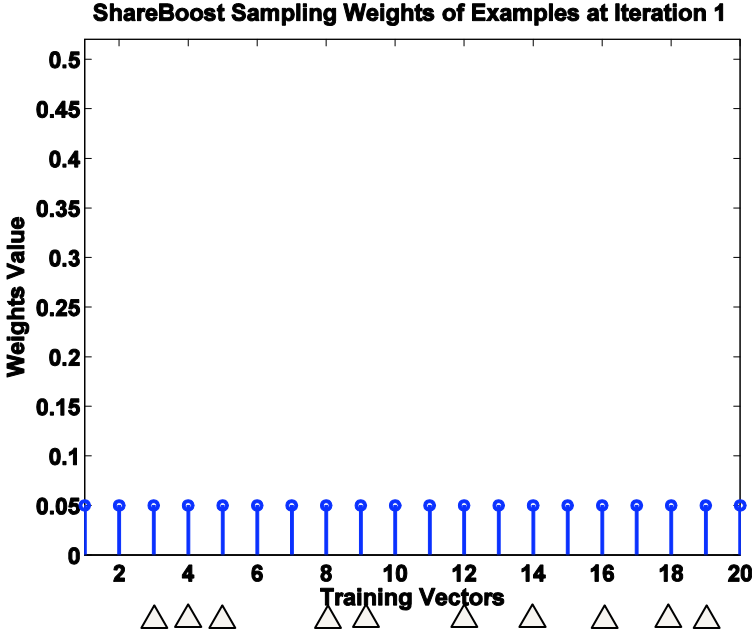
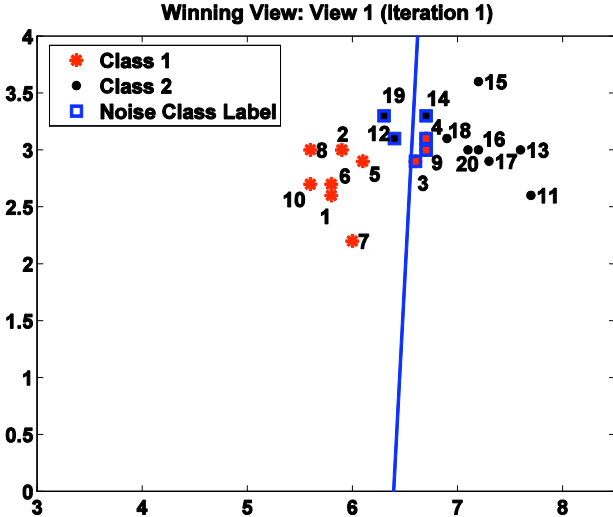
Noisy examples: 3, 4, 9, 12, 14, 19



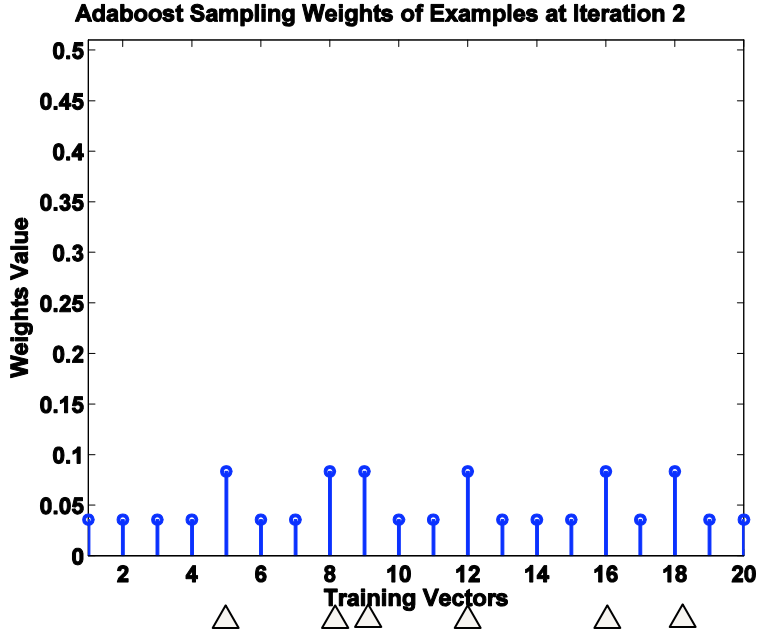
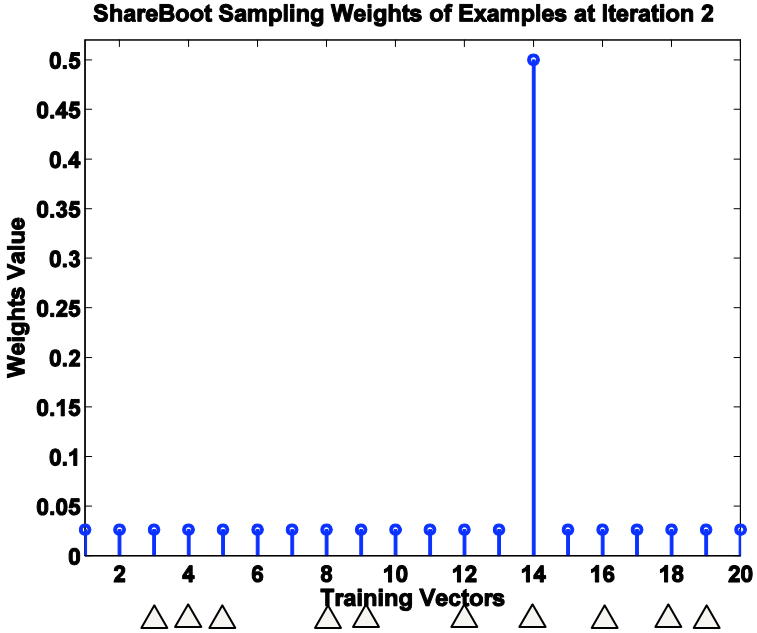
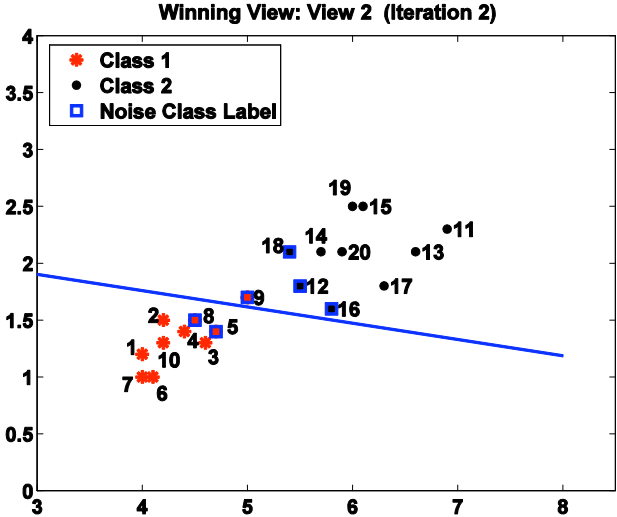
Petal

Noisy examples: 5, 8, 9, 12, 16, 18

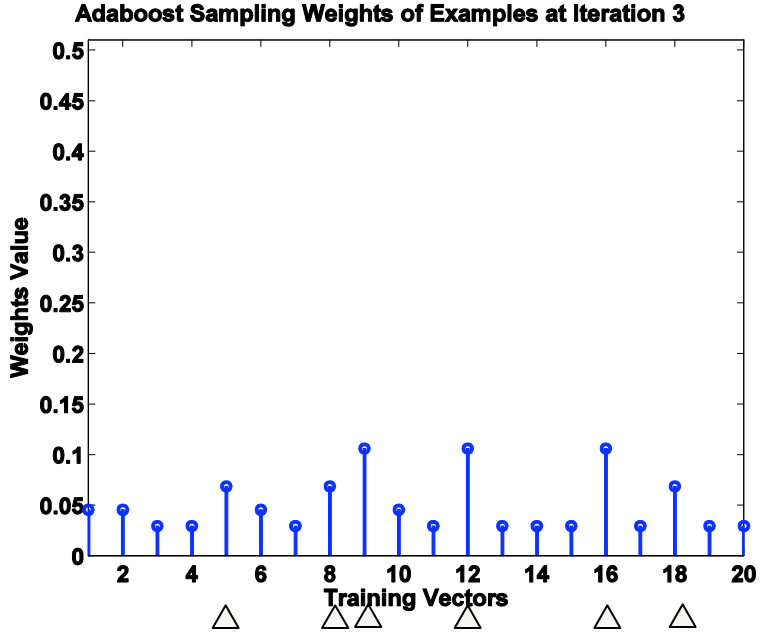
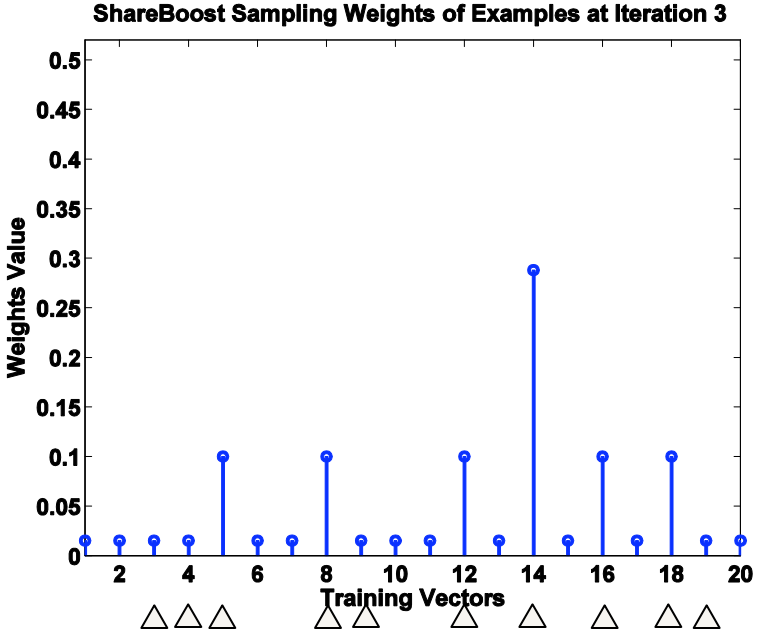
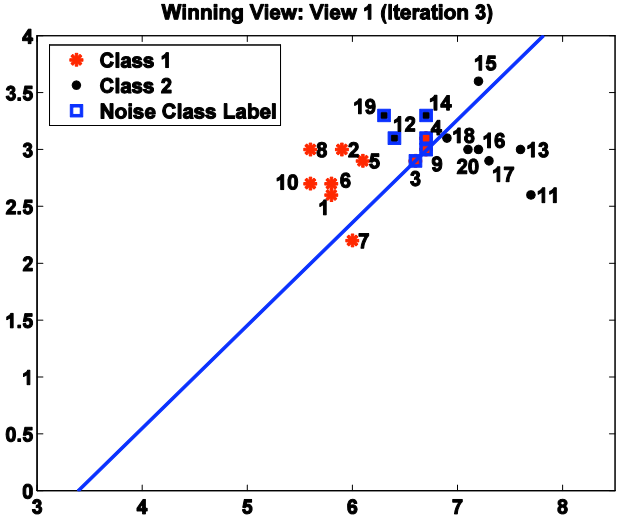
# Simple Illustration



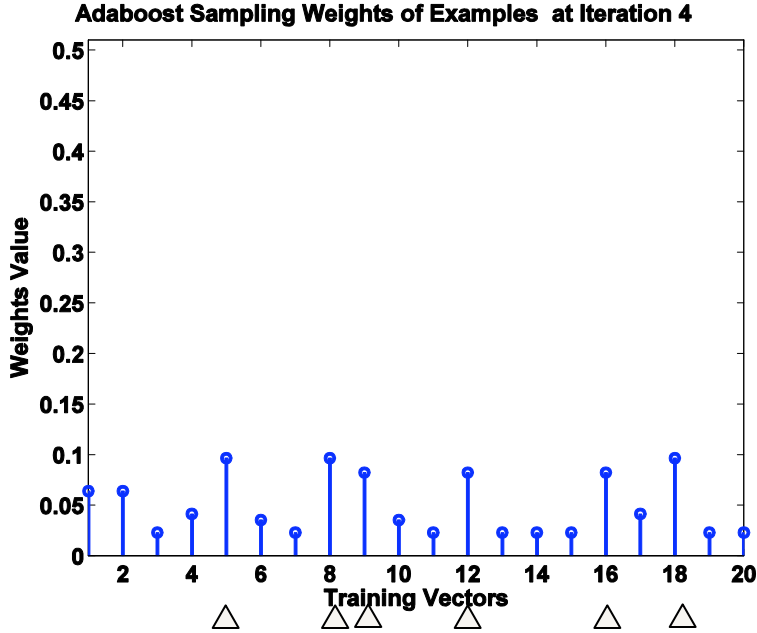
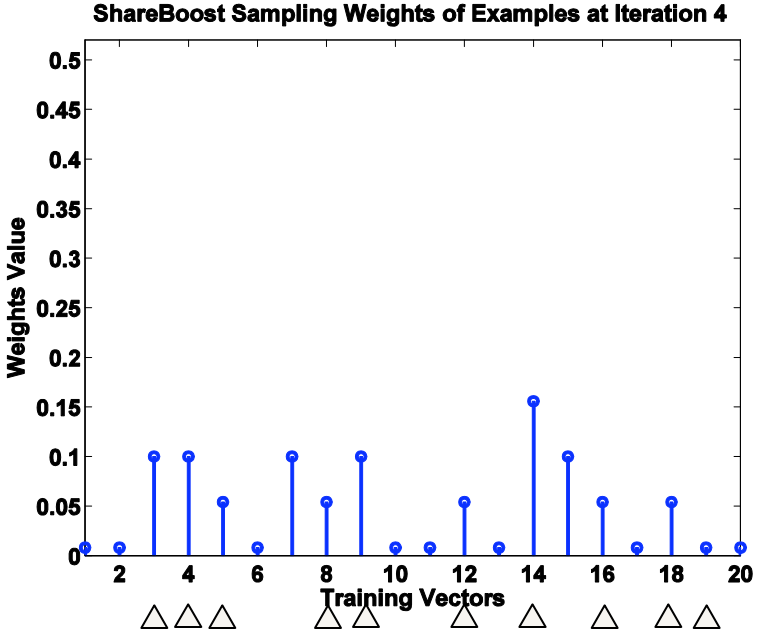
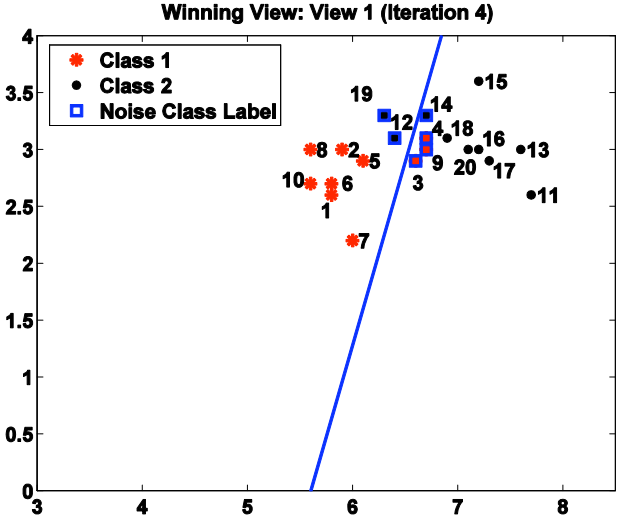
# Simple Illustration



# Simple Illustration

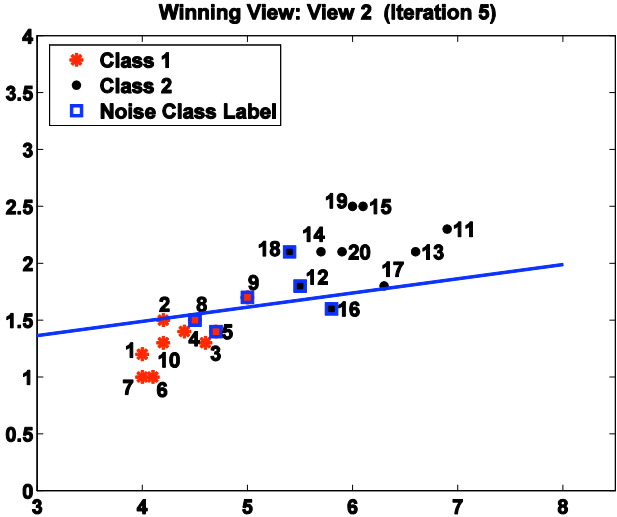


# Simple Illustration

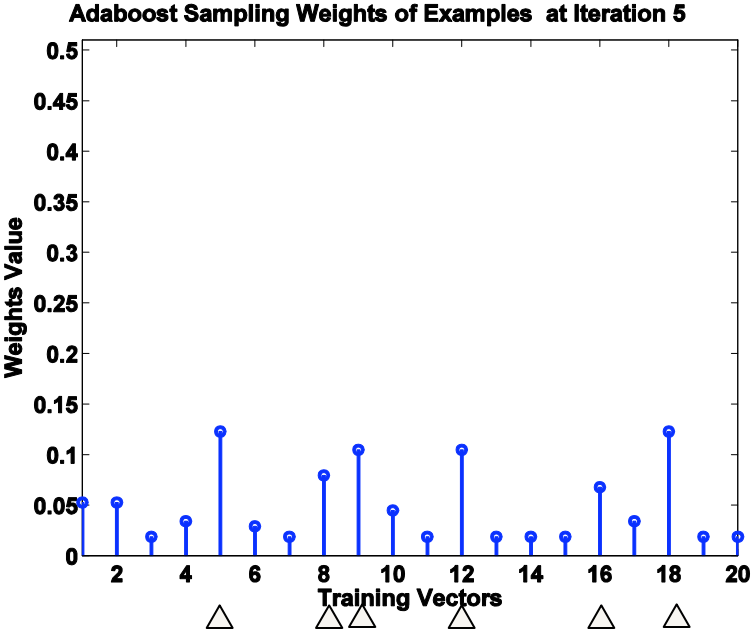
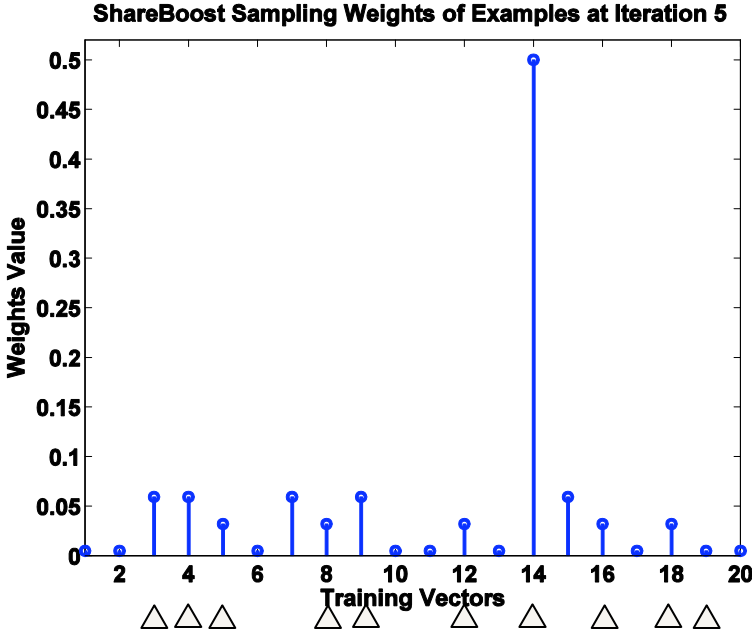




# Simple Illustration

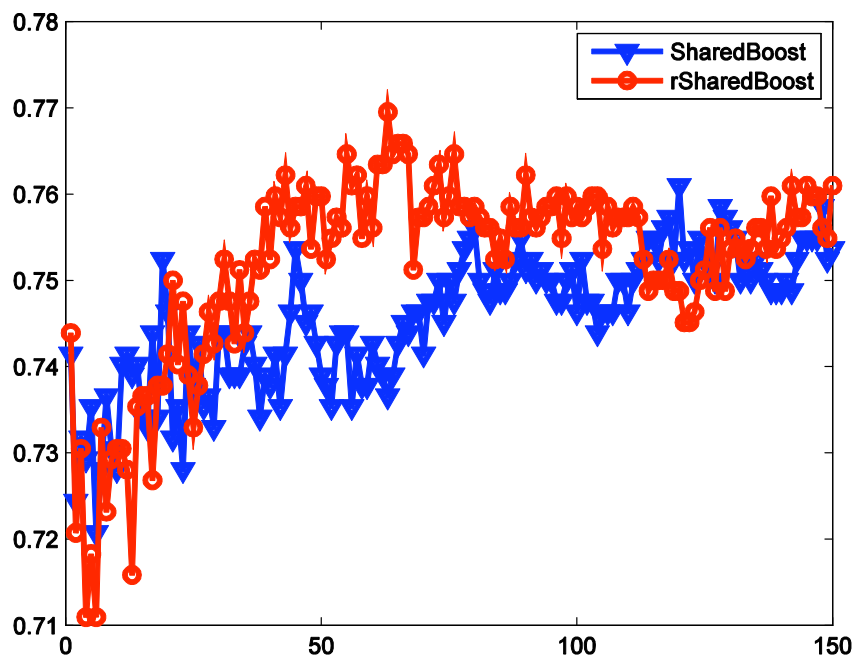


Number of noisy examples that receive a high weight In AdaBoost is more.

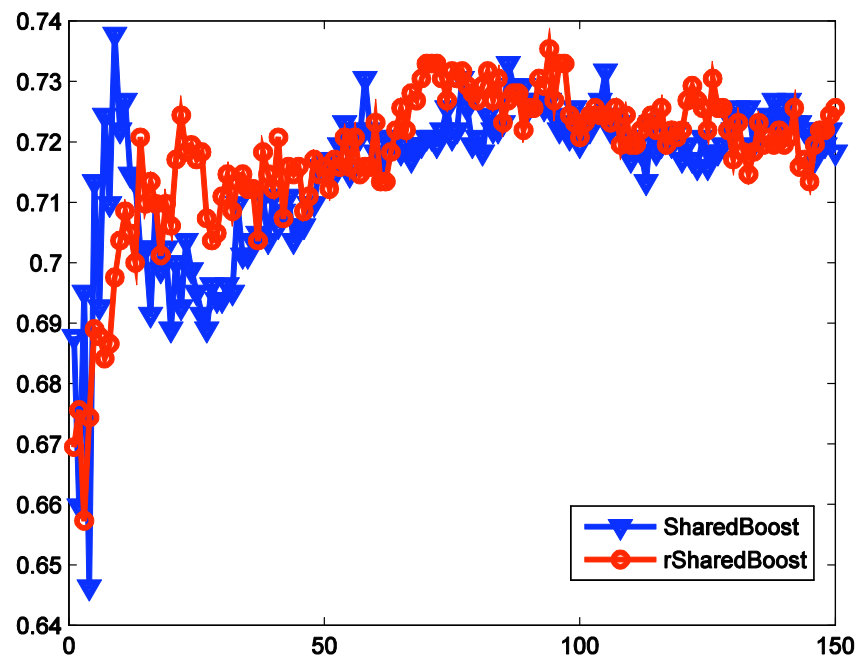




# Performance of ShareBoost and rShareBoost over 150 Base Classifiers



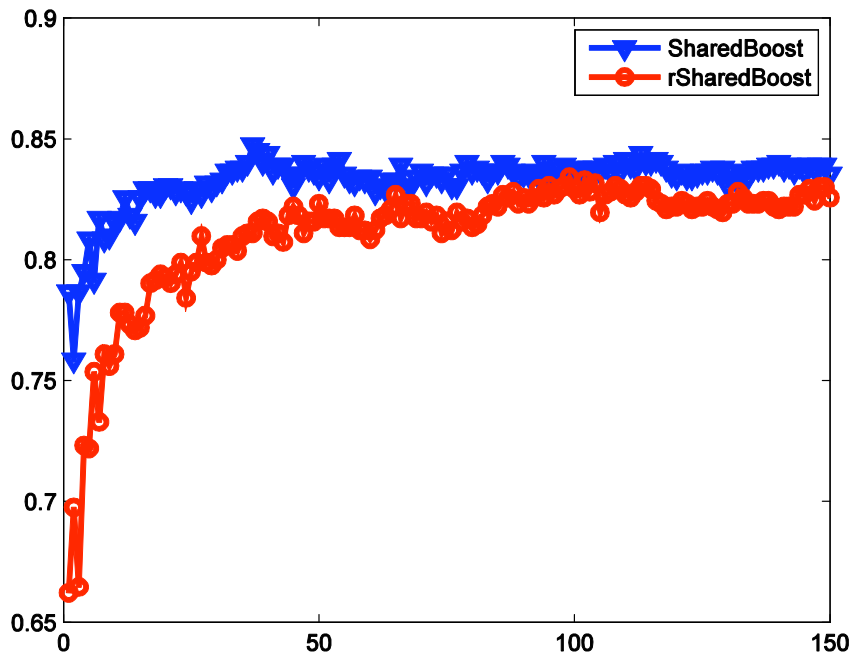
Noise free



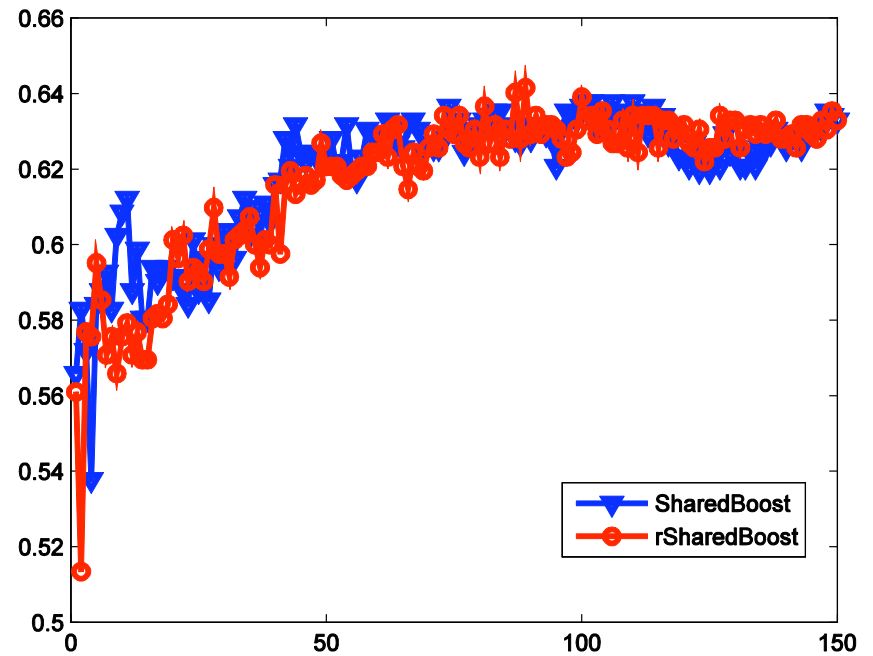
30% noise over 3 views

Face Data

# Performance of ShareBoost and rShareBoost over 150 Base Classifiers



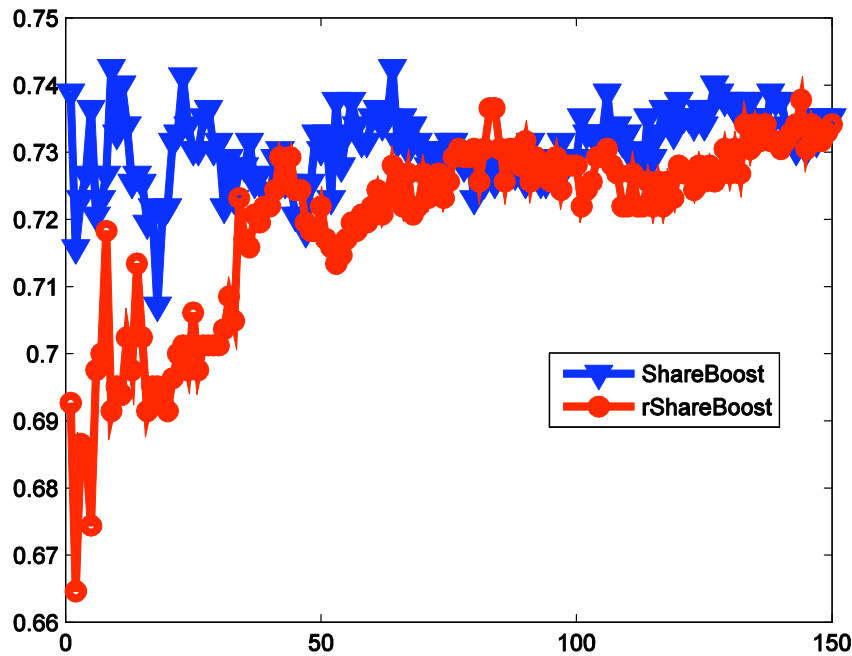
Noise free



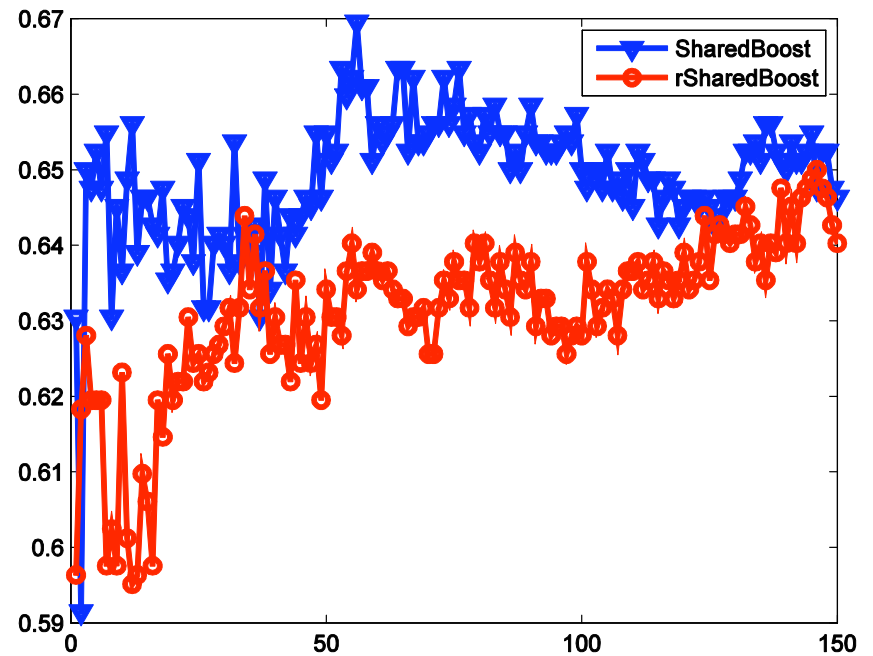
30% noise over 3 views

Gender Data

# Performance of ShareBoost and rShareBoost over 150 Base Classifiers



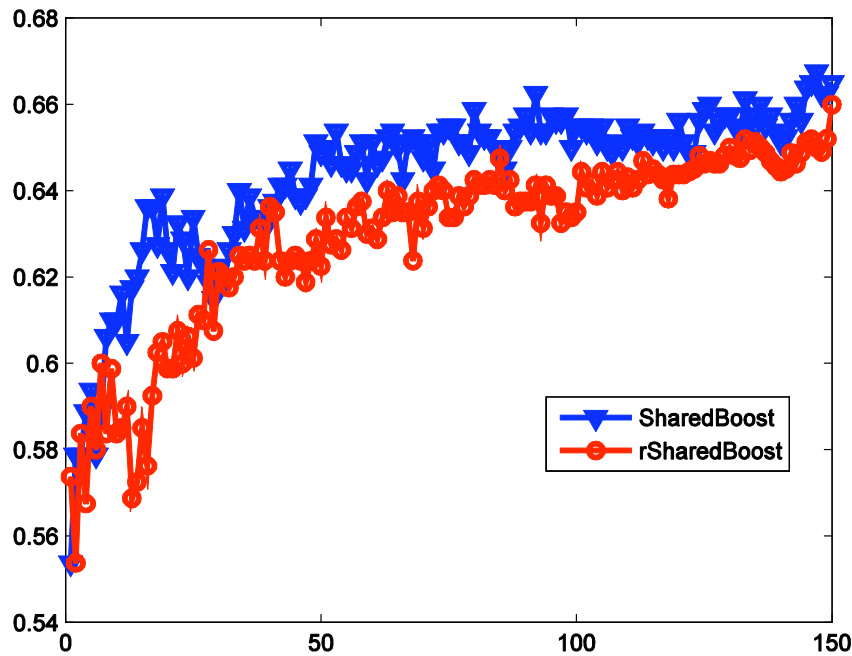
Noise free



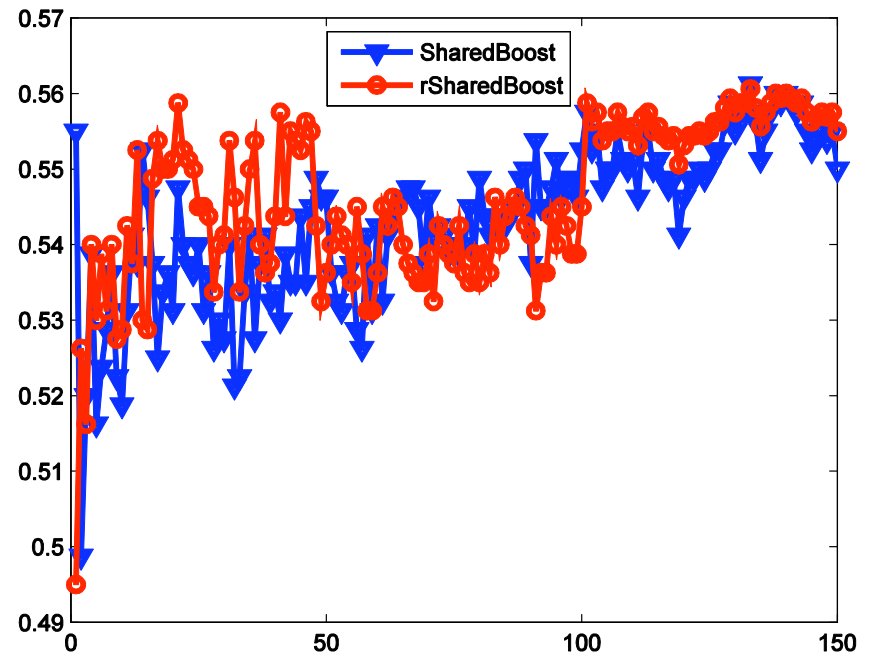
30% noise over 3 views

Glass Data

# Performance of ShareBoost and rShareBoost over 150 Base Classifiers



Noise free



30% noise over 3 views

Gene Data

# Summary

- ❑ ShareBoost is a novel boosting technique for multi-view learning
  - It applies the weight to all views
  - It is capable of exploiting interplay between views
  - It is robust against noise
  
- ❑ rShareBoost is randomized version of ShareBoost
  - It achieves the same performance as ShareBoost
  - It can be shown that rShareBoost efficient in reducing training error by applying armed bandit approach
  - It is less complex than ShareBoost
  
- ❑ Both ShareBoost and rShareBoost outperform competing methods on examples we have experimented with