Machine Learning

CSE 142

Xin (Eric) Wang

Friday, November 19, 2021

Today

- Model ensembles (Ch. 11)

Notes

- HW#4 out Friday, due on 12/1 (Wednesday)
  - Late submission deadline: 12/5 (Sunday)
  - Boosting (introduced today)

Model ensembles

Chapter 11 in the textbook

Model ensembles

- We've seen how combining features can be beneficial
- We can also combine models to increase performance
  - Combinations of models are known as model ensembles
  - Potential for better performance at the cost of increased complexity
- General approach to model ensembles:
  - Construct multiple different models from adapted versions of the training data (e.g., reweighted or resampled)
  - Combine the predictions of these models in some way (averaging, voting, weighted voting, etc.)
- Two of the best-known ensemble methods are bagging and boosting
- These are "meta" methods – i.e., they are independent of the particular learning method (linear classifier, SVM, etc.)

Bagging

- Bagging = "bootstrap aggregation"
  - Create T models on different random samples of the training data set
  - Each sample is a set of training data called a bootstrap sample
    - Could be any size – often $|D|$ is used, the size of the training set
- Bootstrap samples: Sample the data set with replacement (i.e., a data point can be chosen more than once in a bootstrap sample)
  - For $j = 1$ to $|D|$, choose with uniform probability from training data points $D = \{ d_1, d_2, \ldots, d_{|D|} \}$
  - Gather these into the bootstrap sample D_i
- Use $\{ D_1, D_2, \ldots, D_T \}$ to train models $\{ M_1, M_2, \ldots, M_T \}$, then combine the predictions of the T models
- Differences between the T bootstrap samples create diversity among the models in the ensemble

Bagging

Algorithm Bagging($D, T, \mathscr{A}$) – train an ensemble of models from bootstrap samples.

Input : data set D; ensemble size T; learning algorithm $\mathscr{A}$.

Output : ensemble of models whose predictions are to be combined by voting or averaging.

for $t = 1$ to T **do**

  build a bootstrap sample D_t from D by sampling $|D|$ data points with replacement;

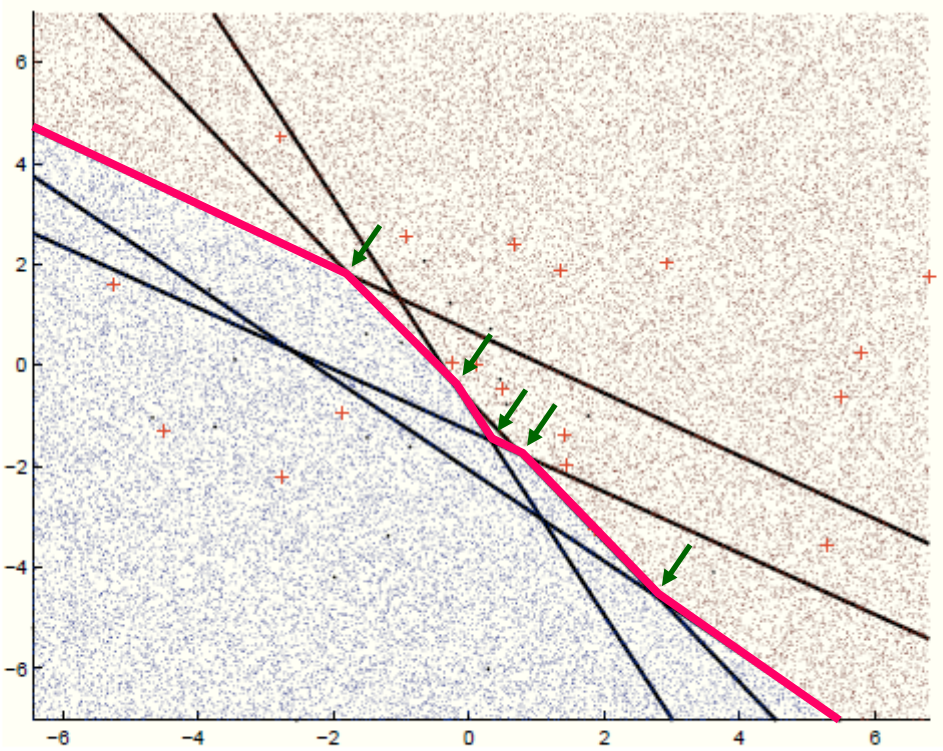
  run $\mathscr{A}$ on D_t to produce a model M_t;

end

return $\{M_t | 1 \leq t \leq T\}$

Bagging example 1

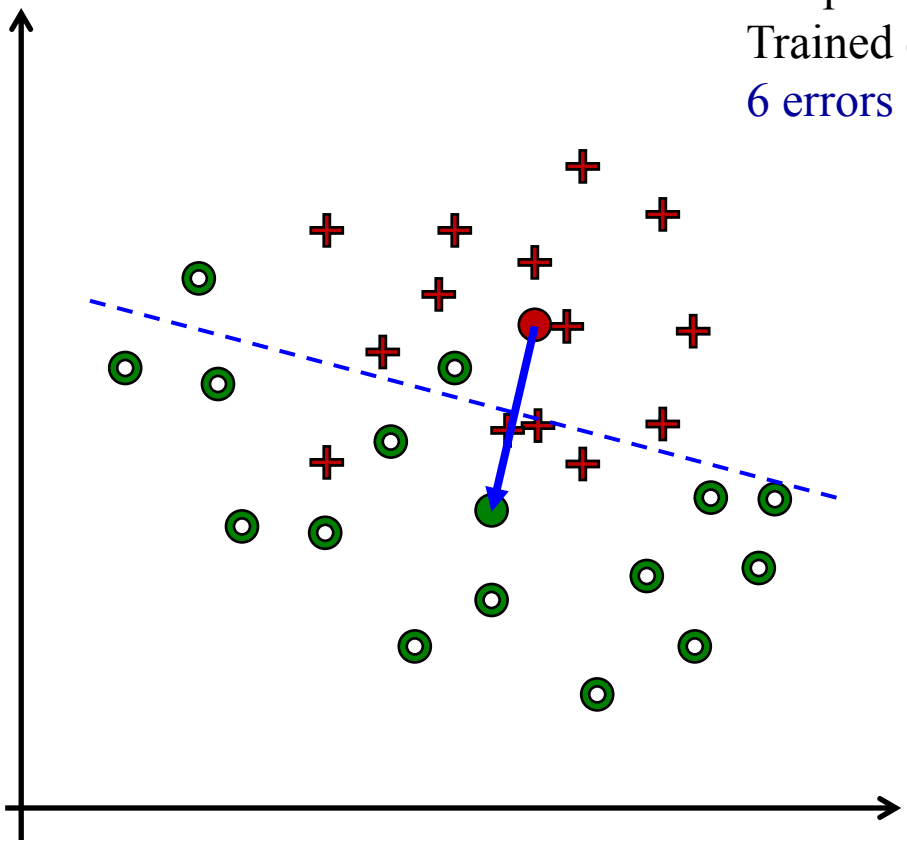
- Train 5 binary linear classifiers by bagging
- Use majority vote (for example) to determine classification output for a new $\boldsymbol{x}$
- The decision boundary is piecewise linear



Bagging example 2

Simple linear classifier
Trained on all data
6 errors

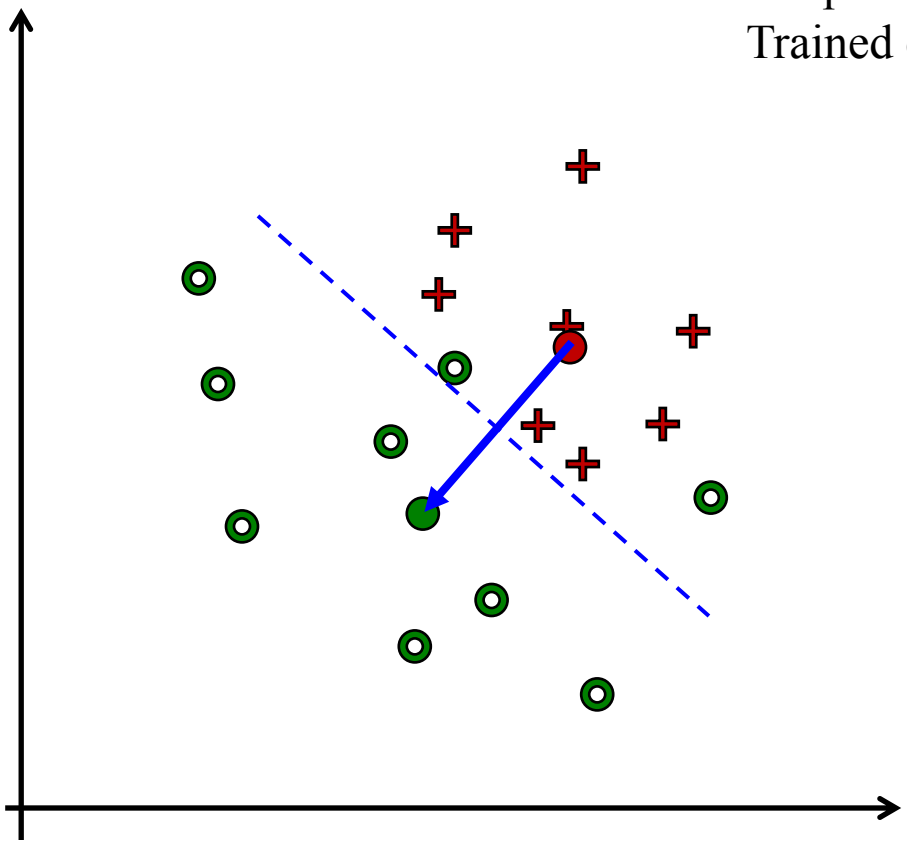
|D| = 29



Bagging example 2

Simple linear classifier
Trained on bootstrap sample set 1

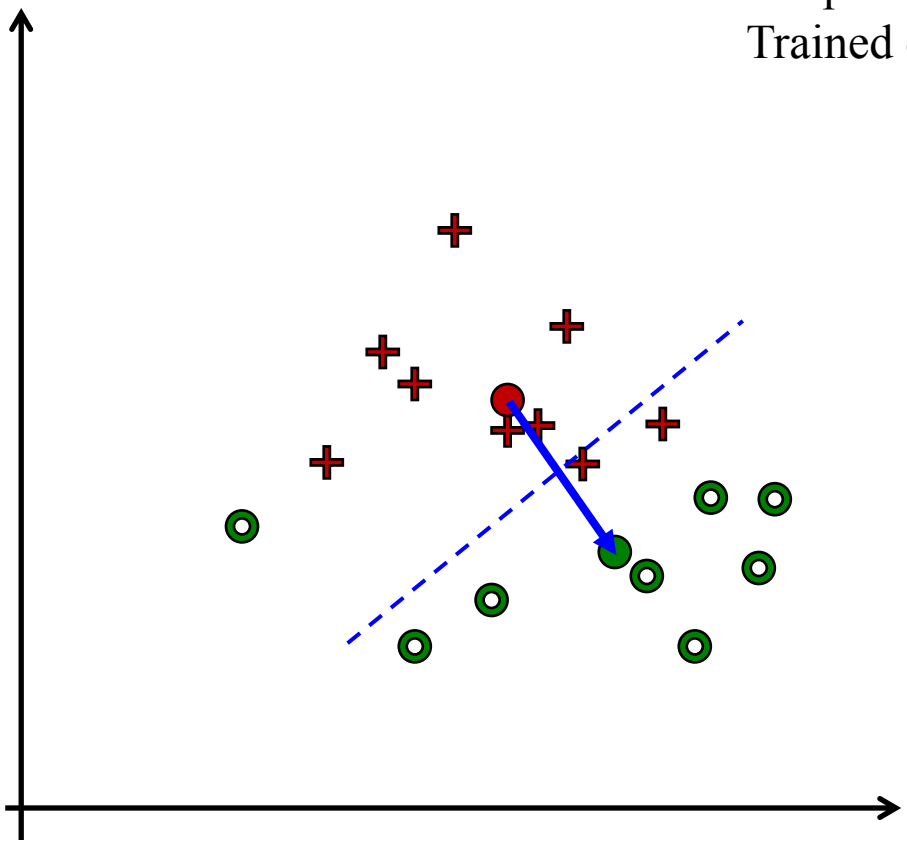
$|D_1| = 17$



Bagging example 2

Simple linear classifier
Trained on bootstrap sample set 2

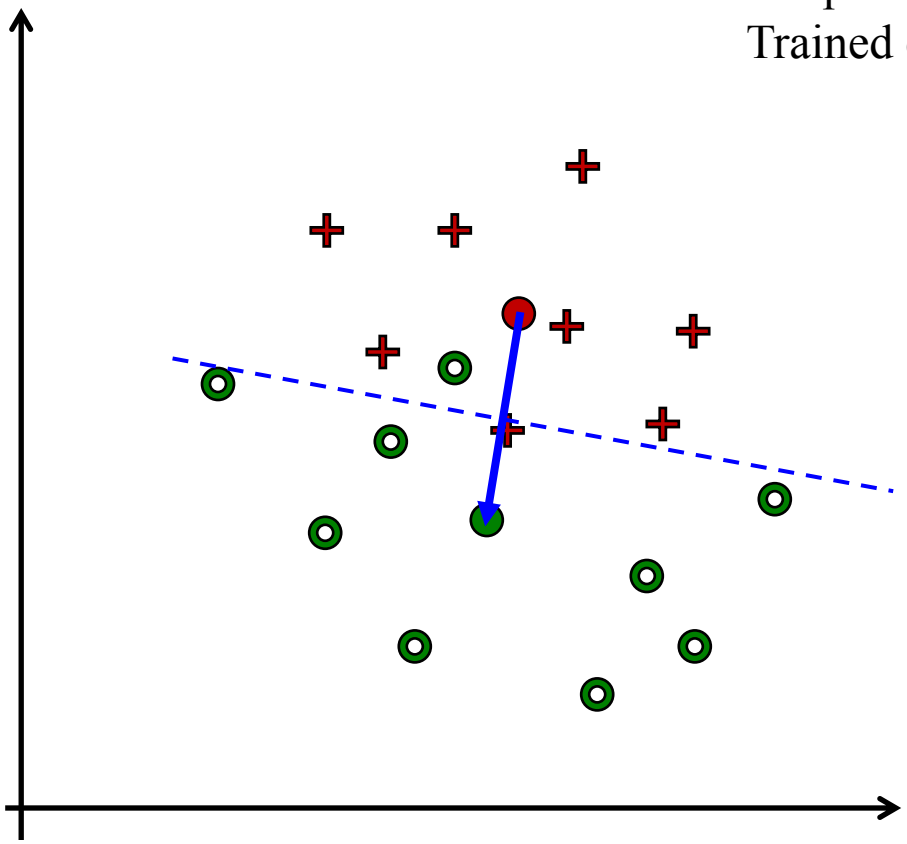
$|D_2| = 17$



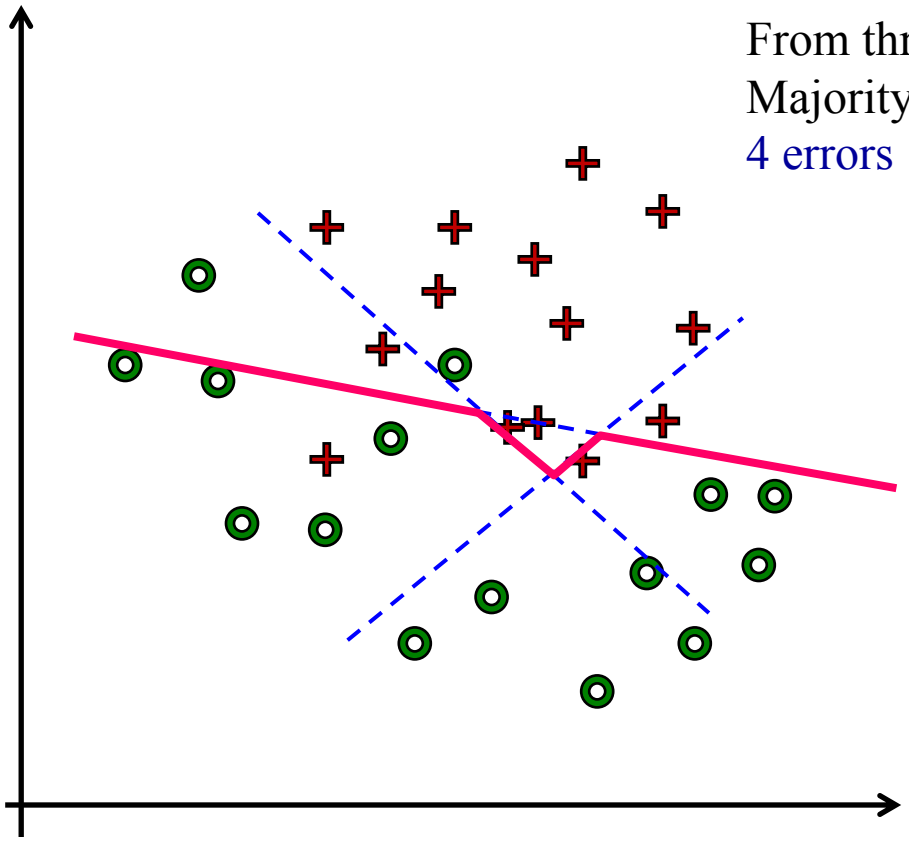
Bagging example 2

Simple linear classifier
Trained on bootstrap sample set 3

$|D_3| = 17$



Bagging example 2



Ensemble linear classifier
From three bootstrap data sets
Majority vote
4 errors

Bagging with tree models

- Bagging is particularly useful for tree models (e.g., decision trees), which can be very sensitive to variations in the training data
  - E.g., build *T* decision trees, produce *T* output labels for an input, then vote to determine the classifier output

- Subspace sampling: Build each decision tree from a different random subset of the **features**

A bunch of trees

Bagging + subspace sampling = *random forests* method

Random forest

Algorithm RandomForest(D, T, d) – train an ensemble of tree models from bootstrap samples and random subspaces.

Input : data set D; ensemble size T; subspace dimension d.

Output : ensemble of tree models whose predictions are to be combined by voting or averaging.

for $t = 1$ to T **do**

  build a bootstrap sample D_t from D by sampling $|D|$ data points with replacement;

  select d features at random and reduce dimensionality of D_t accordingly;

  train a tree model M_t on D_t without pruning;

end

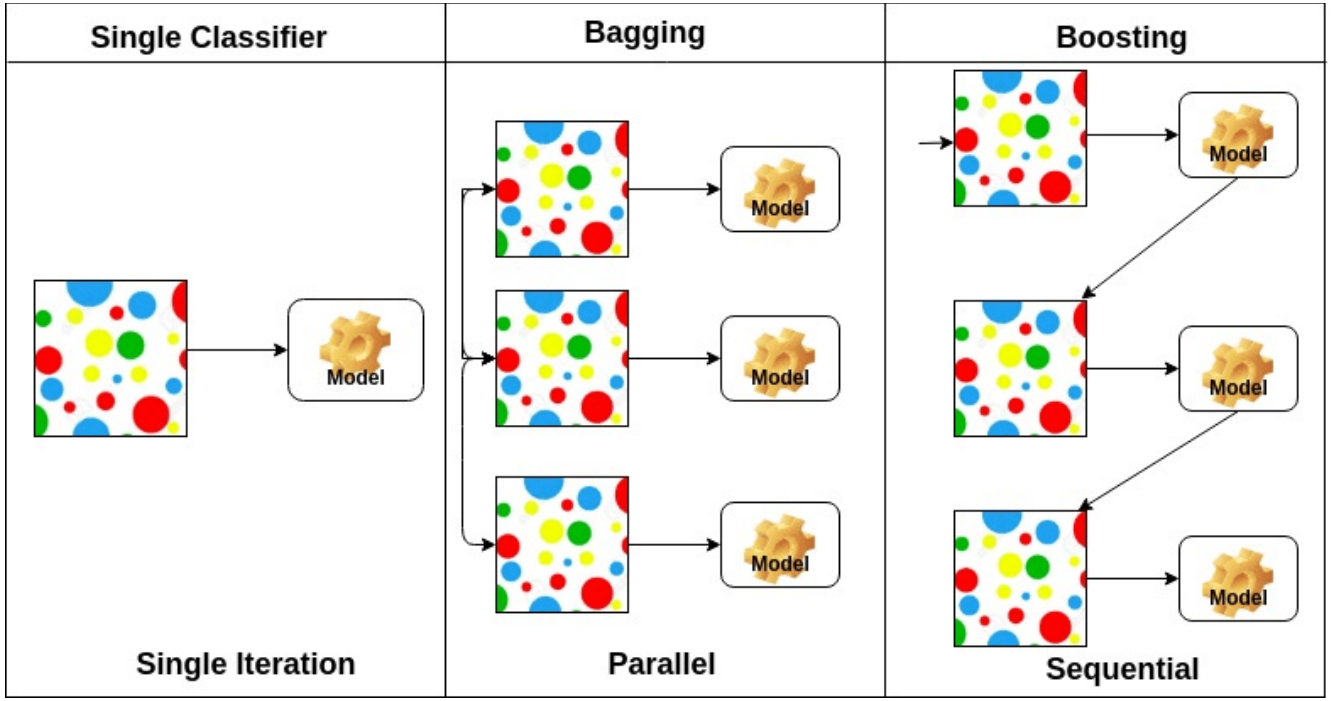
return $\{M_t | 1 \leq t \leq T\}$

Quiz: model ensemble

- See Canvas.

Boosting

- Boosting is similar to bagging, but it uses a more sophisticated technique to create diverse training sets
- The focus is on adding classifiers that do better on the misclassifications from earlier classifiers
  - By giving them a higher weight

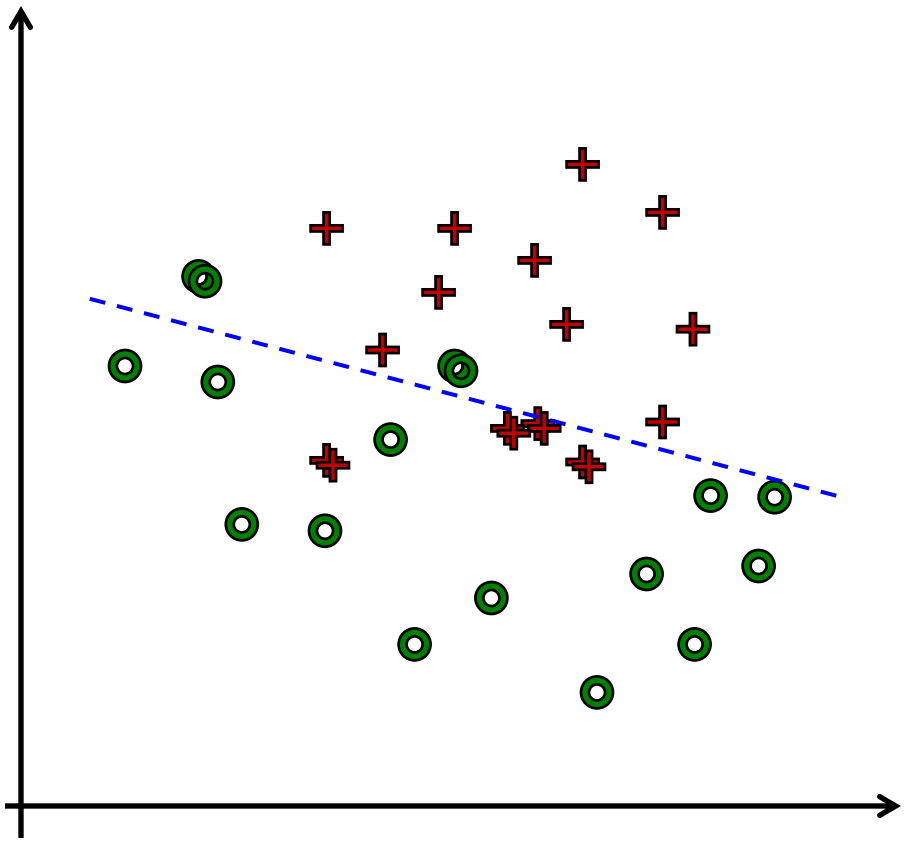


Boosting

- As with bagging, boosting attempts to create a *strong classifier* (*learner*) from a set of *weak classifiers* (*learners*)
  - The resulting ensemble model is less susceptible to overfitting
- Adaboost ("adaptive boosting")
  - An iterative ensemble method
  - As long as the performance of each binary classifier is better than chance ($\epsilon < 0.5$), it is guaranteed to converge to a **better** classifier
- Can be extended to multi-class classification

Boosting intuition

Training:

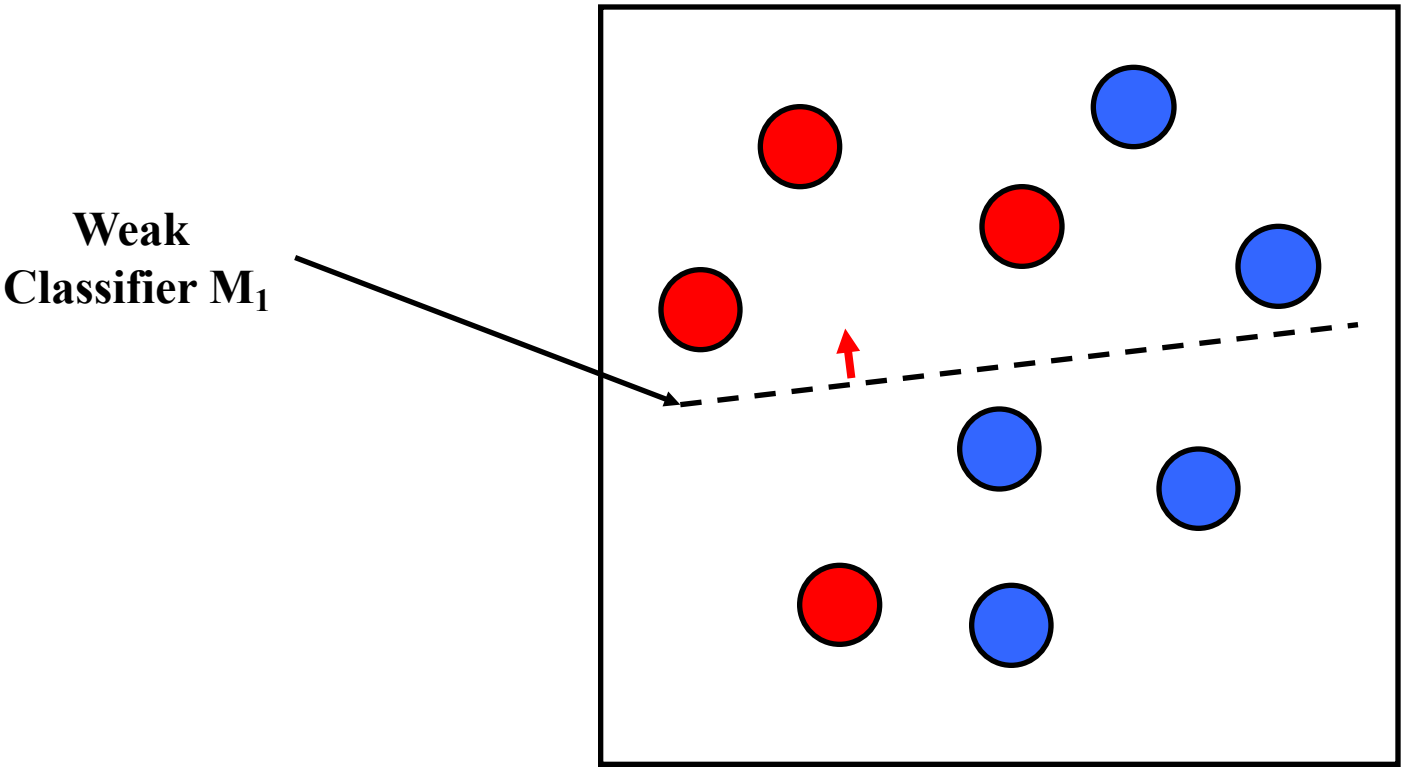


To increase the influence of the misclassified points, duplicate them (i.e., increase their relative weights by 100%)

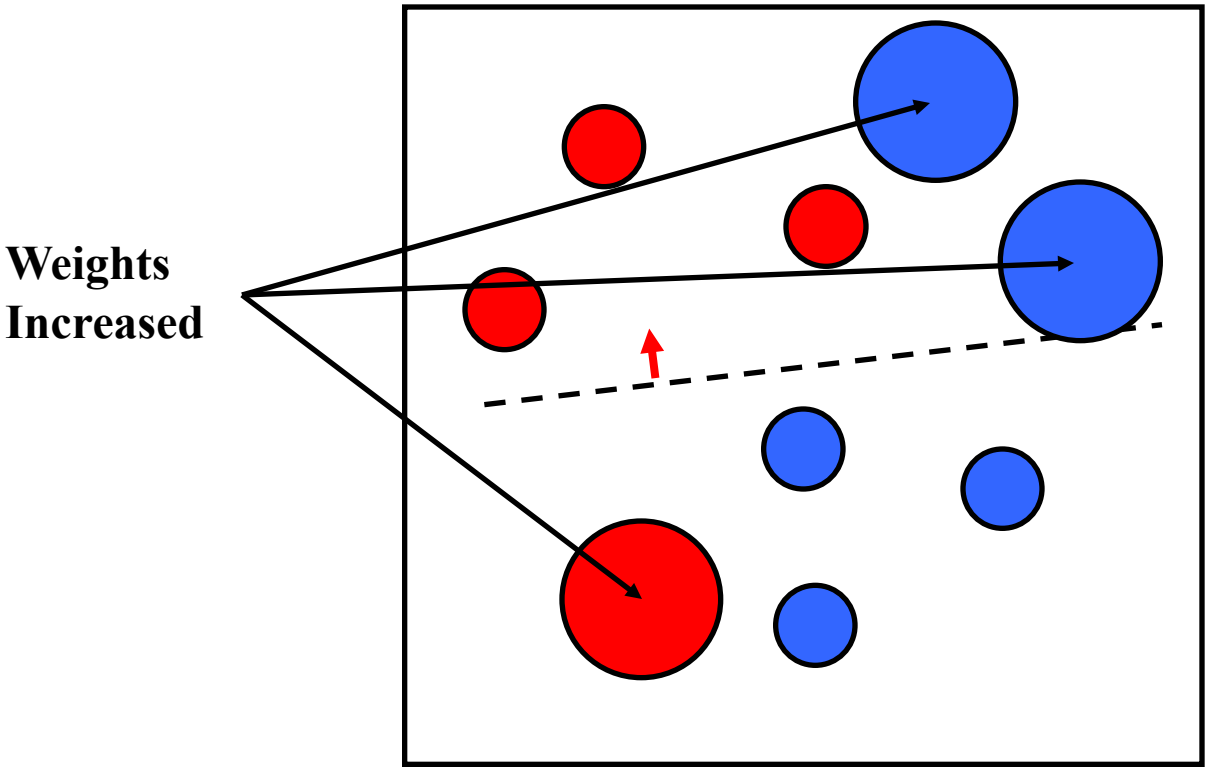
Or we could increase their weights by some other amount.

Assumption: we have a classifier model that takes into account the **weights** of data points

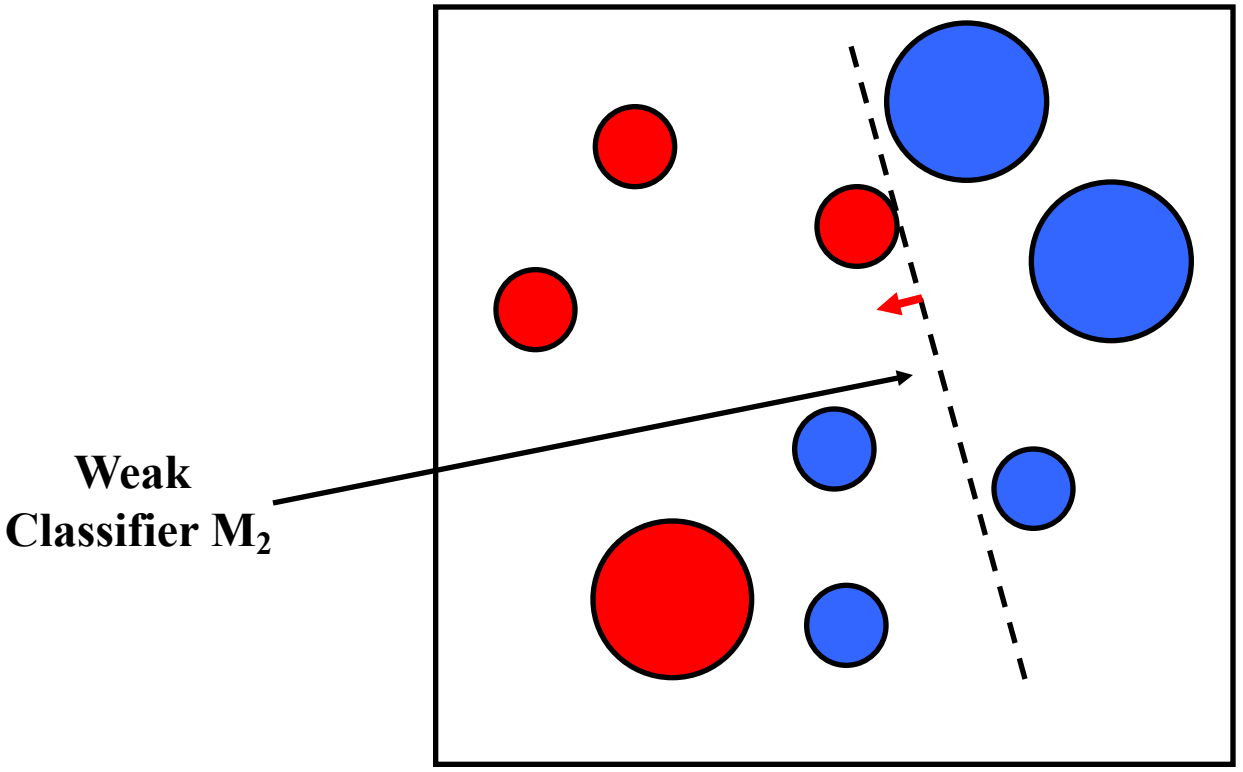
Boosting illustration



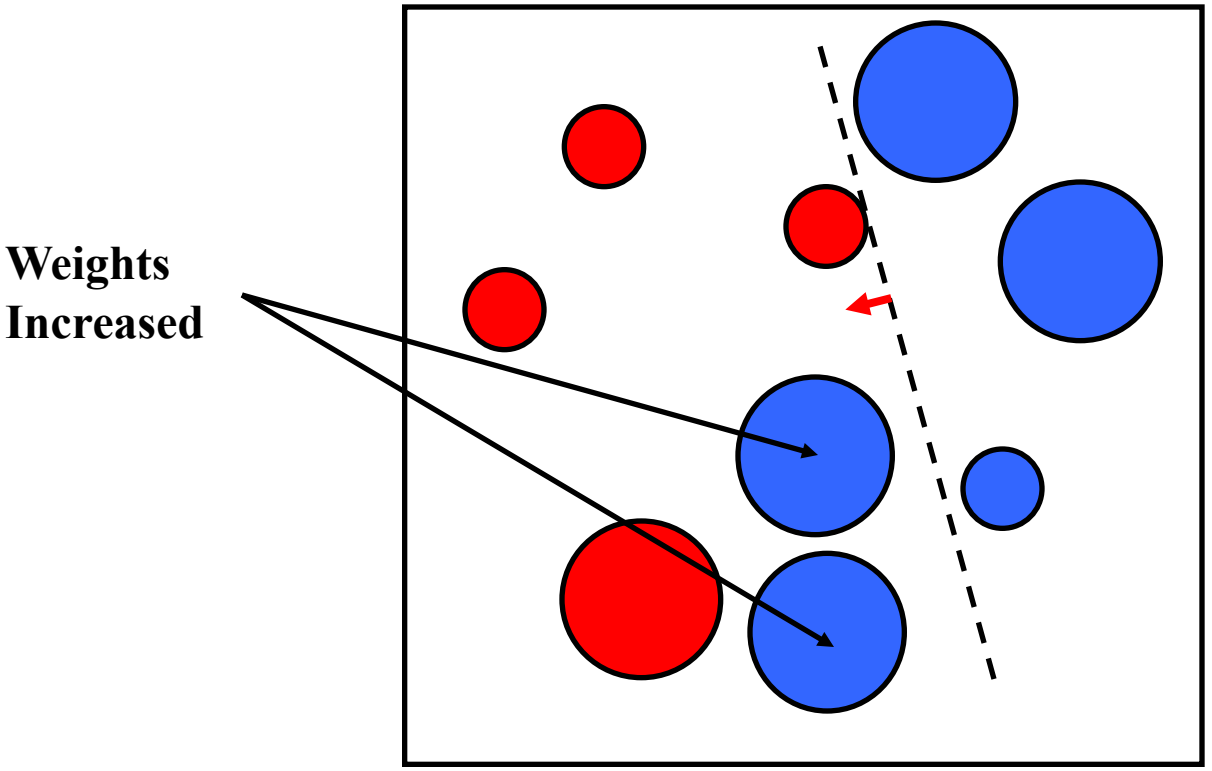
Boosting illustration



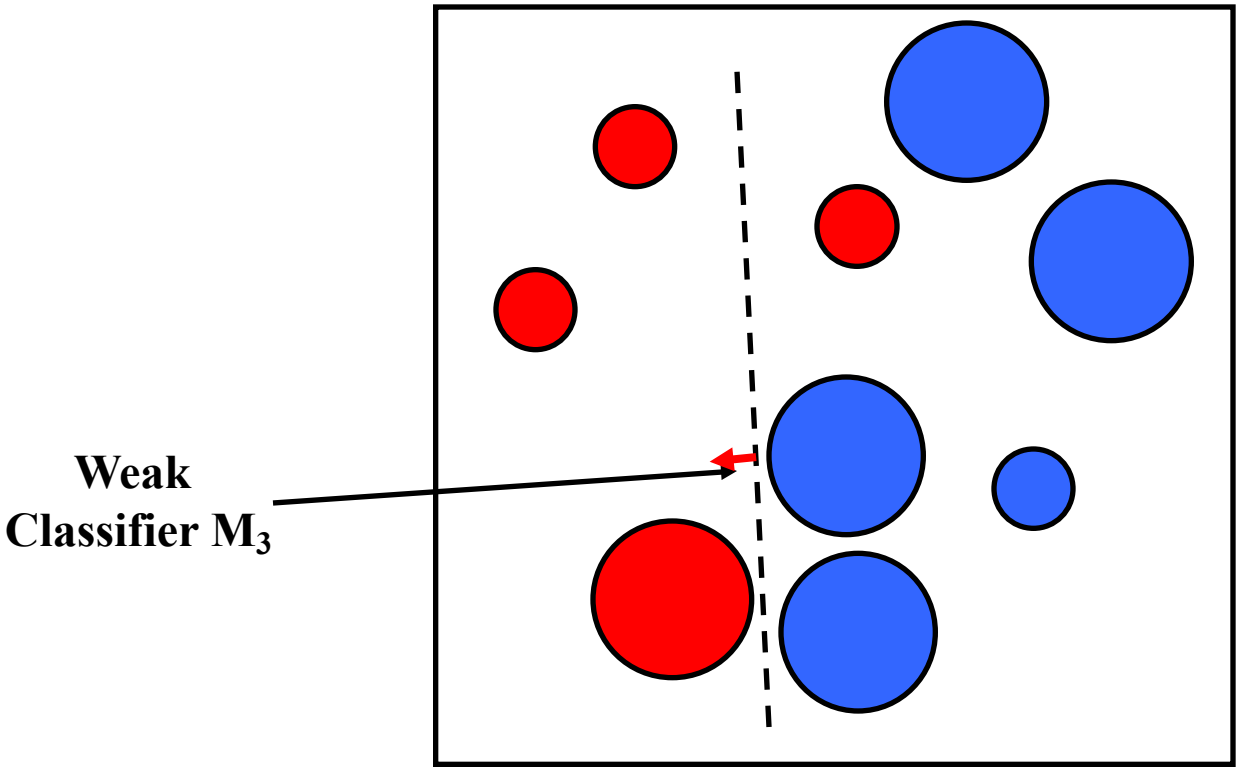
Boosting illustration



Boosting illustration

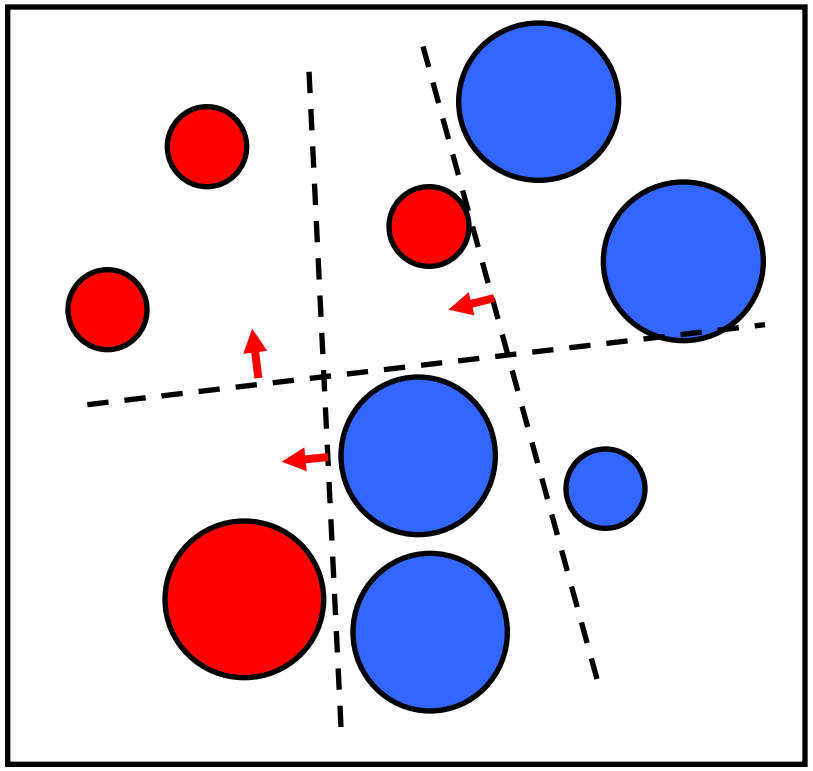


Boosting illustration



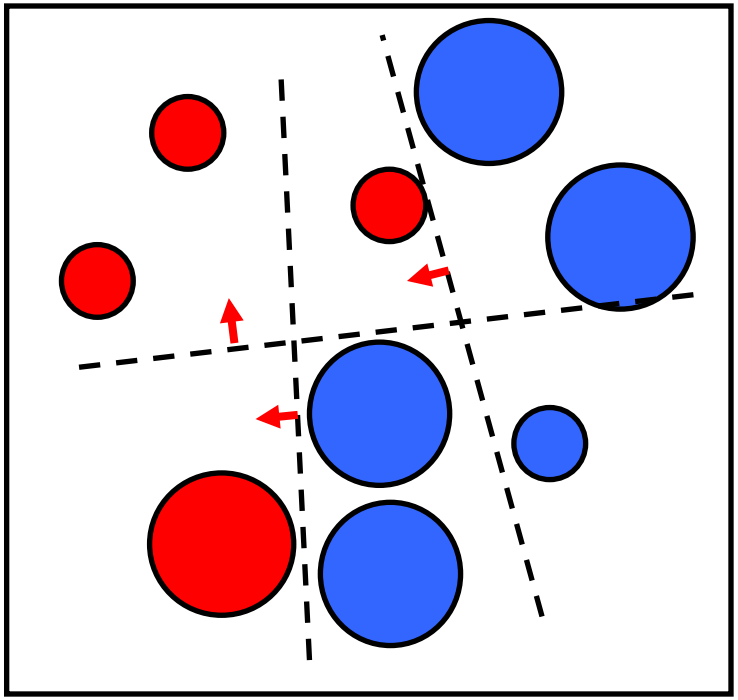
Boosting illustration

Final classifier is
a weighted combination
of weak classifiers



Boosting

- How much should the weights of training examples change?
  - Assign half of the total weights to the misclassified examples
- How to combine the individual models?
  - Confidence factor α_t for each model
  - High α_t to model with low error rate



Boosting

Assumption is complex problem, lots of data, no perfect classifier. I.e., $\epsilon_t > 0$

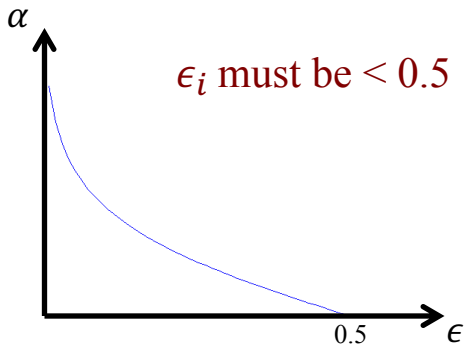
- Procedure:
  - Assign equal weights w_j to training data points
  - Train a classifier; assign it a confidence factor α_t based on the error rate ϵ_t

$$\alpha_t = \frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t}$$

  - Give misclassified instances a higher weight
    - Assign half of the total weight to the misclassified examples
  - Repeat for T classifiers or until $\epsilon_t \geq 0.5$
  - The ensemble predictor is a weighted average of the models (rather than majority vote)

$$M(x) = \sum_{t=1}^{T} \alpha_t M_t(x)$$

  - Threshold for binary output



Misclassified points

$$w' = \frac{w}{2\epsilon_t}$$

Correctly classified points

$$w' = \frac{w}{2(1-\epsilon_t)}$$

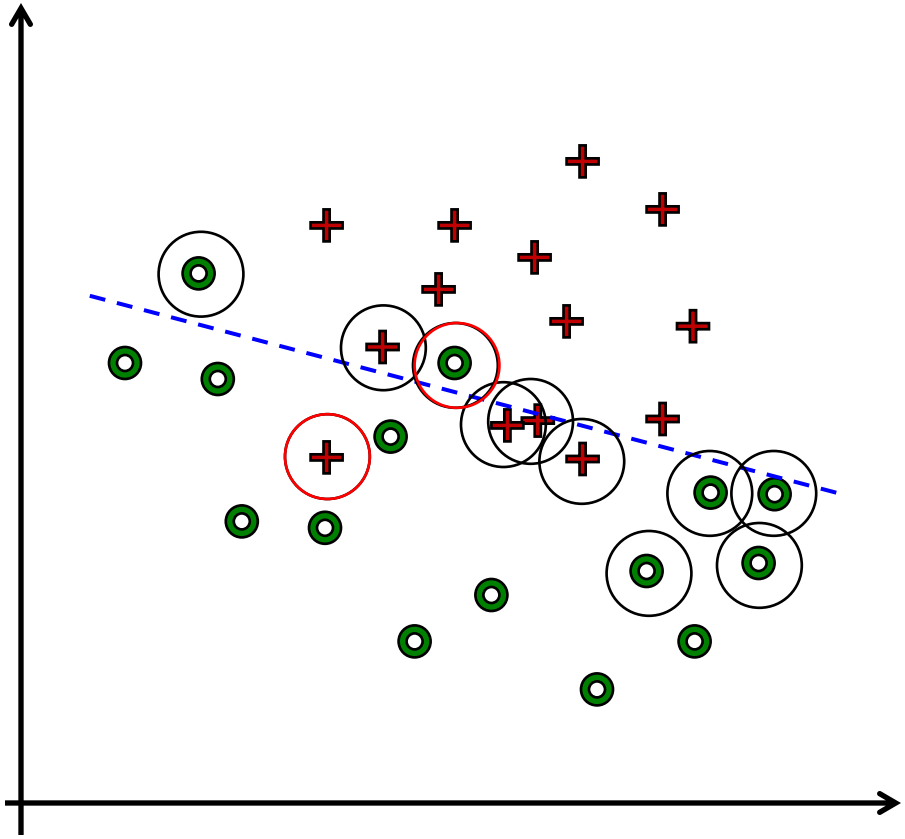
Weights will sum to 0.5+0.5=1

Boosting example

Misclassified points: $w' = \frac{w}{2\epsilon_t}$

Correctly classified points: $w' = \frac{w}{2(1-\epsilon_t)}$

29 points
Initial weights $w = 1/29 = 0.034$



Round 1:
6 errors

$$\epsilon_1 = \frac{6}{29} = 0.21$$

$$\alpha_1 = \frac{1}{2}\ln\frac{1-\epsilon_1}{\epsilon_1} = 0.67$$

$w' = \{ 2.42w, 0.63w \} = \{ 0.082, 0.021 \}$

Round 2:
7 errors => $\epsilon_2 = \frac{7}{29} = 0.24$?

$$\epsilon_2 = \frac{2 * 0.082 + 5 * 0.021}{6 * 0.082 + 23 * 0.021} = 0.28$$

$$\alpha_2 = \frac{1}{2}\ln\frac{1-\epsilon_2}{\epsilon_2} = 0.47$$

$w' = \{ 1.79w, 0.69w \}$

Continue for *T* iterations or until $\epsilon \geq 0.5$

Boosting

Computational complexity: $O(TDK)$
(ensemble size x data size x dimensionality)

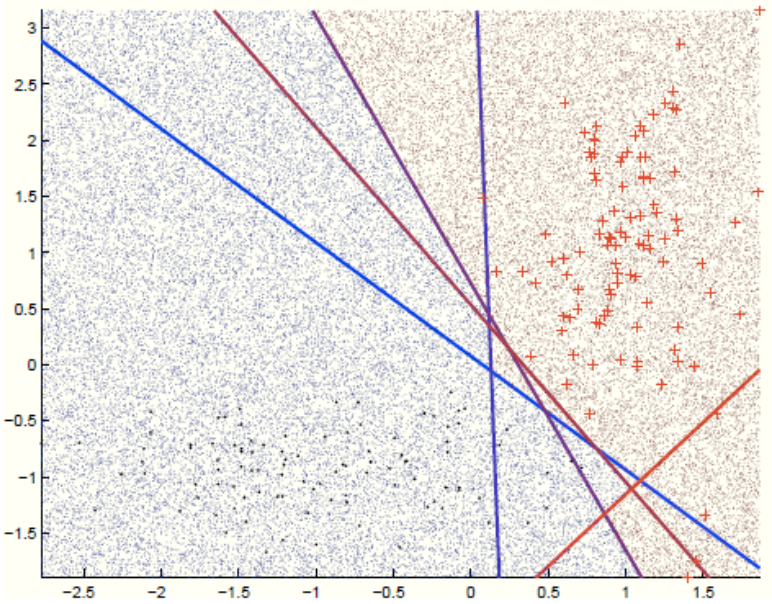
Algorithm Boosting$(D, T, \mathscr{A})$ – train an ensemble of binary classifiers from reweighted training sets.

Input : data set D; ensemble size T; learning algorithm $\mathscr{A}$.
Output : weighted ensemble of models.

$w_{1i} \leftarrow 1/|D|$ for all $x_i \in D$; // start with uniform weights
for $t = 1$ to T **do**
  run $\mathscr{A}$ on D with weights w_{ti} to produce a model M_t;
  calculate weighted error ϵ_t;
  if $\epsilon_t \geq 1/2$ **then**
    set $T \leftarrow t-1$ and break
  end
  $\alpha_t \leftarrow \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$; // confidence for this model
  $w_{(t+1)i} \leftarrow \frac{w_{ti}}{2\epsilon_t}$ for misclassified instances $x_i \in D$; // increase weight
  $w_{(t+1)j} \leftarrow \frac{w_{tj}}{2(1-\epsilon_t)}$ for correctly classified instances $x_j \in D$; // decrease
end
return $M(x) = \sum_{t=1}^{T} \alpha_t M_t(x)$

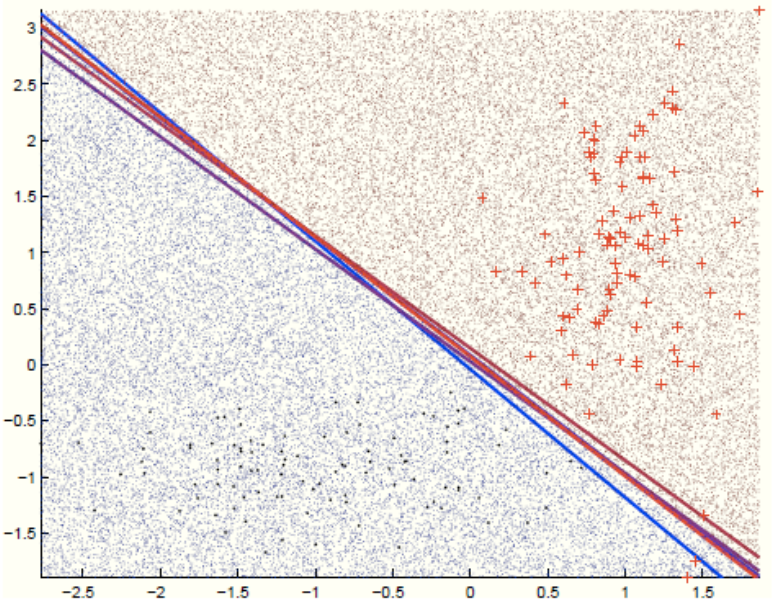
Boosting vs. bagging

Boosting



Can achieve zero training error by focusing on the misclassifications

Bagging



With relatively large bootstrap sample sets, there tends to be little diversity in the learned models

Ensemble methods: summary

- Model ensemble methods are meta-methods: they are ways to combine ML models, but they are agnostic to the kind of model being used
  - Simple linear classifier, Perceptron, SVM, neural network, etc. (weighted versions)
- They provide an opportunity to improve performance and (mostly) avoid over-fitting to the training data
  - Typically using randomization of some sort
- Bagging is essentially a variance reduction technique (increase consistency), while boosting is essentially a bias reduction technique (increase accuracy)
- There are many other ensemble approaches in machine learning

Next

- Machine learning experiments
- Neural network & deep learning