M L
D M

Chapter 8
Ensemble Learning

Content

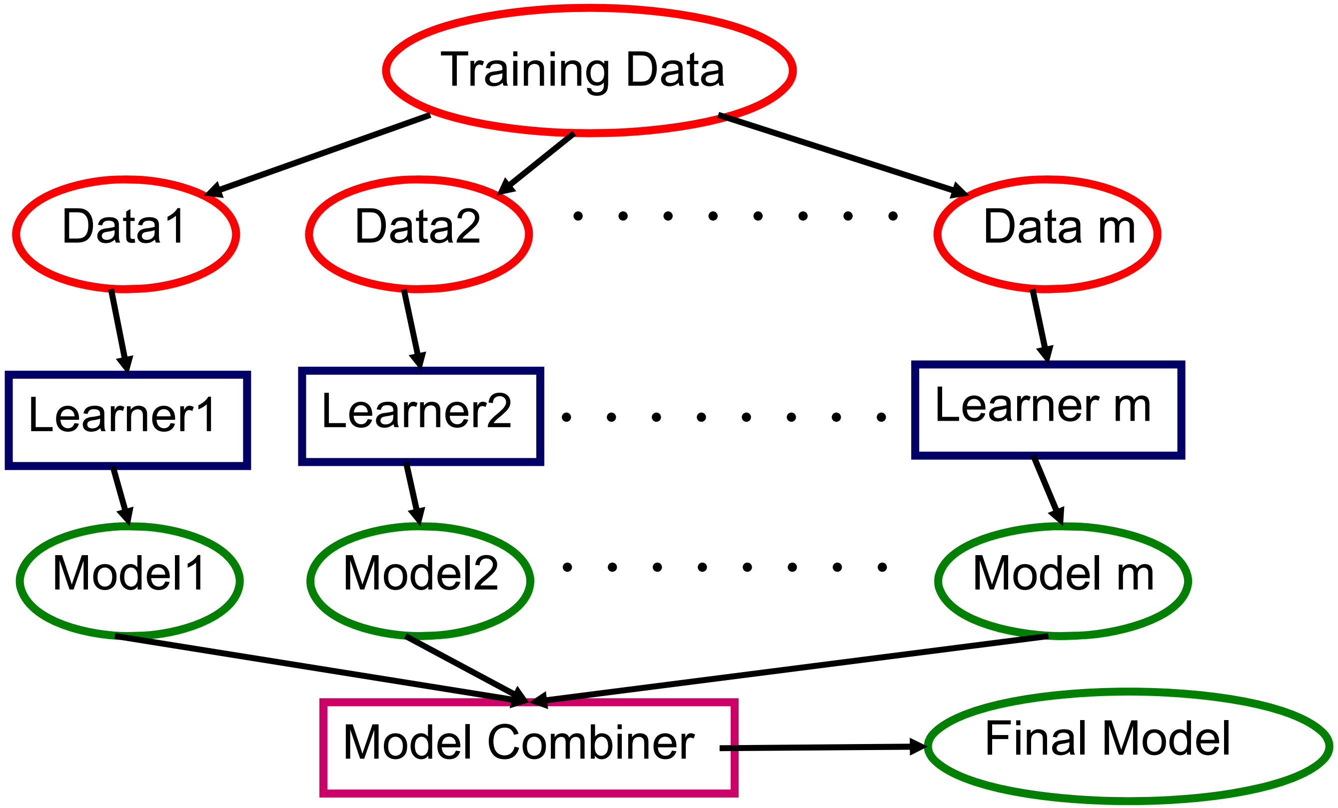
- Introduction
- Bagging
- Boosting
  - Adaboost

Introduction

MIMA

- What is ensemble learning
  - Learn multiple alternative definitions of a concept using different training data or different learning algorithms.
- Example 1
  - Generate 100 different decision trees from the same or different training set and have them vote on the best classification for a new example.

Introduction



Key motivation: reduce the error rate.

Different Learners

MIMA

- Different learning algorithms
- Algorithms with different choice for parameters
- Data set with different features
- Data set = different subsets

Homogenous Ensembles

MIMA

- Use a single, arbitrary learning algorithm but manipulate training data to make it learn multiple models.
  - Data1 ≠ Data2 ≠ … ≠ Data m
  - Learner1 = Learner2 = … = Learner m
- Different methods for changing training data:
  - Bagging: Resample training data
  - Boosting: Reweight training data

Bagging

MIMA

- Introduced by Breiman (1996)
  - *L. Breiman. Bagging predictors*. Machine Learning, 24(2): 123 – 140, 1996 (citations 16529)
  - Create ensembles by repeatedly randomly resampling the training data
- "Bagging" stands for "**b**ootstrap **agg**regat**ing**".
  - Given a training set of size *n*, create *m* samples of size *n* by drawing *n* examples from the original data, ***with replacement***.
  - Each ***bootstrap sample*** will on average contain 63.2% of the unique training examples, the rest are replicates.
  - Combine the *m* resulting models using simple majority vote.

Bootstrap

MIMA

- Example
  - What's the average price of house prices?
  - From F, get a sample $L=(x_1, x_2, \ldots, x_n)$, and calculate the average u.
  - Question: how reliable is u? What's the standard error of u? what's the confidence interval?

Bootstrap

MIMA

- One possibility: get several samples like F.
- Problem: it is impossible (or too expensive) to get multiple samples.
- Solution: bootstrap

Bootstrap

MIMA

Let the original samples be $L=(x_1,x_2,\ldots,x_n)$

- Repeat B time:
  - Generate a sample L_k of size n from L by sampling with replacement.
  - Compute $\hat{\theta}^*$ for x*.

  ➔ Now we end up with bootstrap values

$$\hat{\theta}^* = (\hat{\theta}_1^*, \ldots, \hat{\theta}_B^*)$$

- Use these values for calculating all the quantities of interest (e.g., standard deviation, confidence intervals)

Bootstrap

- An example

X=(3.12, 0, 1.57, 19.67, 0.22, 2.20)

Mean=4.46

X1=(1.57,0.22,19.67, 0,0,2.2,3.12)

Mean=4.13

X2=(0, 2.20, 2.20, 2.20, 19.67, 1.57)

Mean=4.64

X3=(0.22, 3.12,1.57, 3.12, 2.20, 0.22)

Mean=1.74

Bootstrap

MIMA

Original	1	2	3	4	5	6	7	8
Training set 1	2	7	8	3	7	6	3	1
Training set 2	7	8	5	6	4	2	7	1
Training set 3	3	6	2	7	5	6	2	2
Training set 4	4	5	1	4	6	4	3	8

Bootstrap

MIMA

- Cases where bootstrap does not apply
  - Small data sets: the original sample is not a good approximation of the population
  - Dirty data: outliers add variability in our estimates.
  - Dependence structures (e.g., time series, spatial problems): Bootstrap is based on the assumption of independence.

Bagging

MIMA

Let the original training data be L

- Repeat B times:
  - Get a <u>bootstrap sample</u> L_k from L.
  - Train a predictor using L_k.
- Combine B predictors by
  - Voting (for classification problem)
  - Averaging (for estimation problem)
- Bagging works well for “unstable” learning algorithms.
- Bagging can slightly degrade the performance of “stable” learning algorithms.

Bagging

- Unstable learning algorithms: small changes in the training set result in large changes in predictions.
  - Neural network
  - Decision tree
  - Regression tree
  - Subset selection in linear regression

- Stable learning algorithms:
  - K-nearest neighbors

Bagging-Summary

MIMA

- Bootstrap is a resampling method.
- Bagging is directly related to bootstrap.
  - It uses bootstrap samples to train multiple predictors.
  - Output of predictors are combined by voting or other methods.
- Experiment results:
  - It is effective for unstable learning methods.
  - It does not help stable learning methods.

Boosting

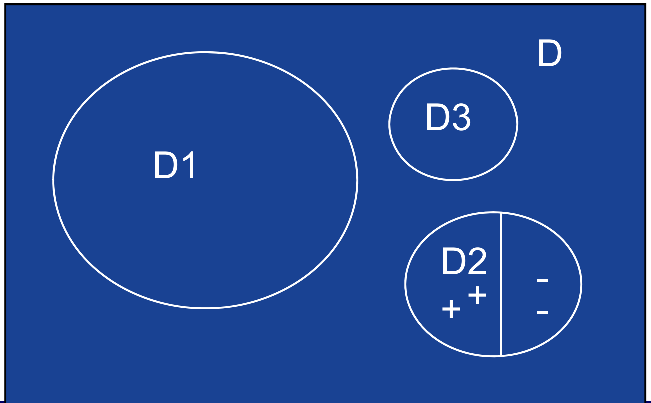
MIMA

- A family of methods
- Sequential production of classifiers
- Each classifier is dependent on the previous one, and focuses on the previous one's errors
- Examples that are incorrectly predicted in previous classifiers are chosen more often or weighted more heavily

Boosting

MIMA

- Robert E. Schapire, **The strength of weak learnability**. *Machine Learning*, 5(2):197-227, 1990. (citations 4600+)
- Consider creating three component classifiers for a two-category problem through boosting.
  - Randomly select $n_1 < n$ samples from D without replacement to obtain D_1
  - Train weak learner C_1
  - Select $n_2 < n$ samples from D with half of the samples misclassified by C_1 to obtain D_2
  - Train weak learner C_2
  - Select all remaining samples from D that C_1 and C_2 disagree on
  - Train weak learner C_3
  - Final classifier is vote of weak learners



AdaBoost

MIMA

- Yoav Freund, Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting, EuroCOLT '95 Proceedings of the Second European Conference on Computational Learning Theory (citations 15000+)
- Instead of resampling, uses training set re-weighting
  - Each training sample uses a weight to determine the probability of being selected for a training set.
- AdaBoost is an algorithm for constructing a “strong” classifier as linear combination of “simple” “weak” classifier
- Final classification based on weighted vote of weak classifiers

Introduction

MIMA

- What is AdaBoost?

AdaBoost

Adaptive Boosting

A learning algorithm

Building a strong classifier from a lot of weaker ones

Introduction

MIMA

$$h_1(x) \in \{-1, +1\}$$
$$h_2(x) \in \{-1, +1\}$$
$$\vdots$$
$$h_T(x) \in \{-1, +1\}$$

$$H_T(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

weak classifiers

strong classifier

slightly better than random

Introduction

MIMA

$$h_1(x) \in \{-1, +1\}$$
$$h_2(x) \in \{-1, +1\}$$
$$\vdots$$
$$h_T(x) \in \{-1, +1\}$$

weak classifiers

slightly better than random

- Each weak classifier learns by considering one simple feature
- T most beneficial features for classification should be selected
- How to
  - define features?
  - select beneficial features?
  - train weak classifiers?
  - manage (weight) training samples?
  - associate weight to each weak classifier?

Introduction

MIMA

How good the strong one will be?

$$h_1(x) \in \{-1, +1\}$$

$$h_2(x) \in \{-1, +1\}$$

$$\vdots$$

$$h_T(x) \in \{-1, +1\}$$

$$H_T(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

weak classifiers

strong classifier

slightly better than random

The AdaBoost Algorithm

MIMA

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X, y_i \in \{-1, +1\}$

Initialization: $D_1(i) = \frac{1}{m}, i = 1, \ldots, m$

$D_t(i)$:probability distribution of x_i's at time t

For $t = 1, \ldots, T$:

- Find classifier $h_t : X \to \{-1, +1\}$ which minimizes error wrt D_t ,i.e.,

$$h_t = \arg\min_{h_j} \varepsilon_j \quad \text{where } \varepsilon_j = \sum_{i=1}^{m} D_t(i)[y_i \neq h_j(x_i)]$$

minimize weighted error

- Weight classifier: $\alpha_t = \frac{1}{2} \ln \frac{1-\varepsilon_t}{\varepsilon_t}$

for minimize exponential loss

- Update distribution: $D_{t+1}(i) = \frac{D_t(i)\exp[-\alpha_t y_i h_t(x_i)]}{Z_t}$, Z_t is for normalization

Give error classified patterns more chance for learning.

The AdaBoost Algorithm

MIMA

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X, y_i \in \{-1, +1\}$

Initialization: $D_1(i) = \frac{1}{m}, i = 1, \ldots, m$

For $t = 1, \ldots, T$:

- Find classifier $h_t : X \to \{-1, +1\}$ which minimizes error wrt D_t ,i.e.,

$$h_t = \arg\min_{h_j} \varepsilon_j \quad \text{where } \varepsilon_j = \sum_{i=1}^{m} D_t(i)[y_i \neq h_j(x_i)]$$

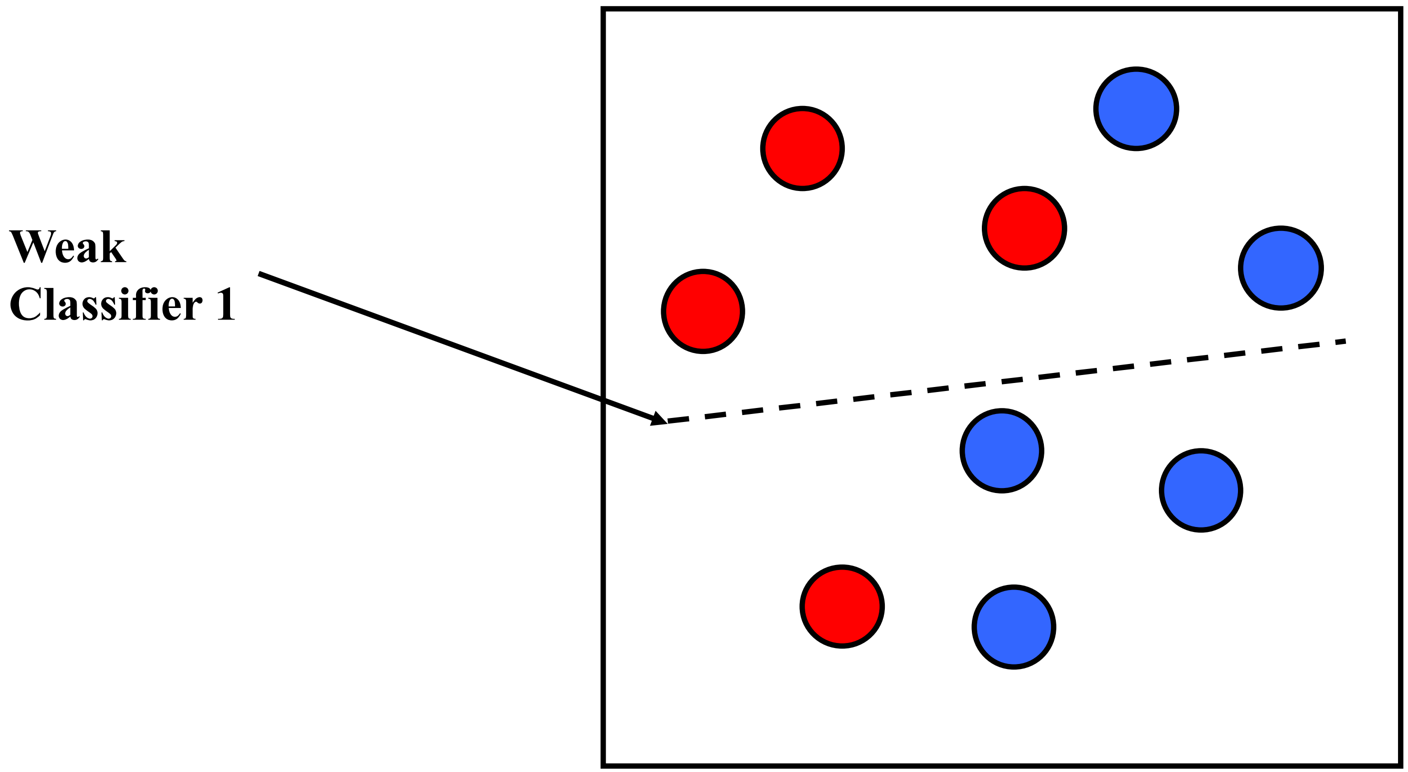
- Weight classifier: $\alpha_t = \frac{1}{2} \ln \frac{1 - \varepsilon_t}{\varepsilon_t}$

- Update distribution: $D_{t+1}(i) = \frac{D_t(i) \exp[-\alpha_t y_i h_t(x_i)]}{Z_t}$, Z_t is for normalization

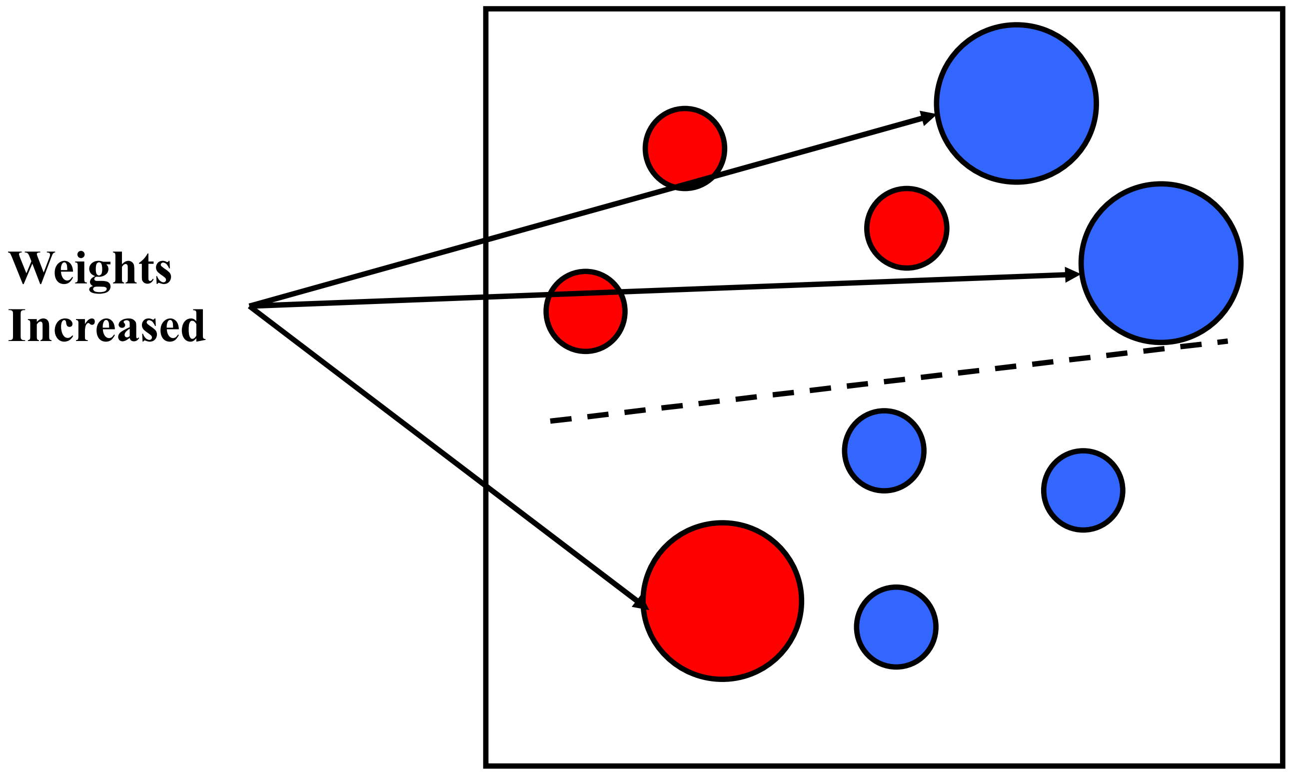
Output final classifier: $sign\left(H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \right)$

Illustration

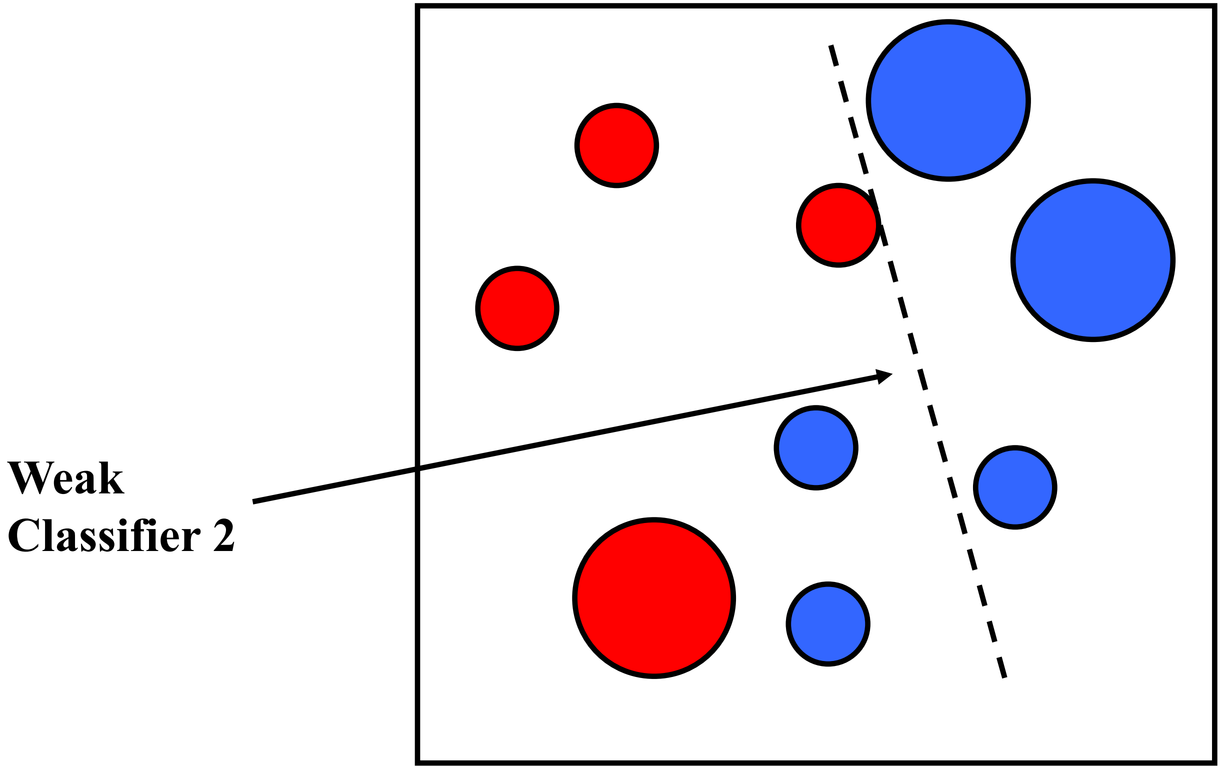
MIMA



Illustration

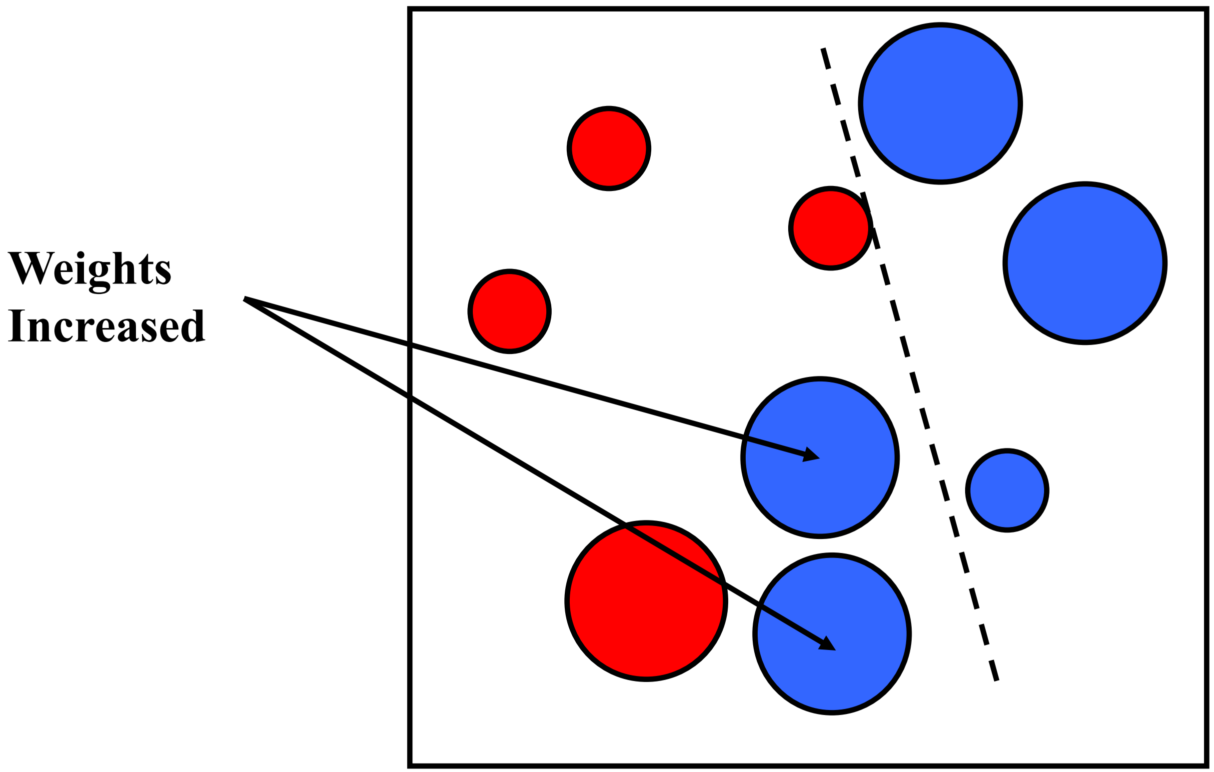


Illustration



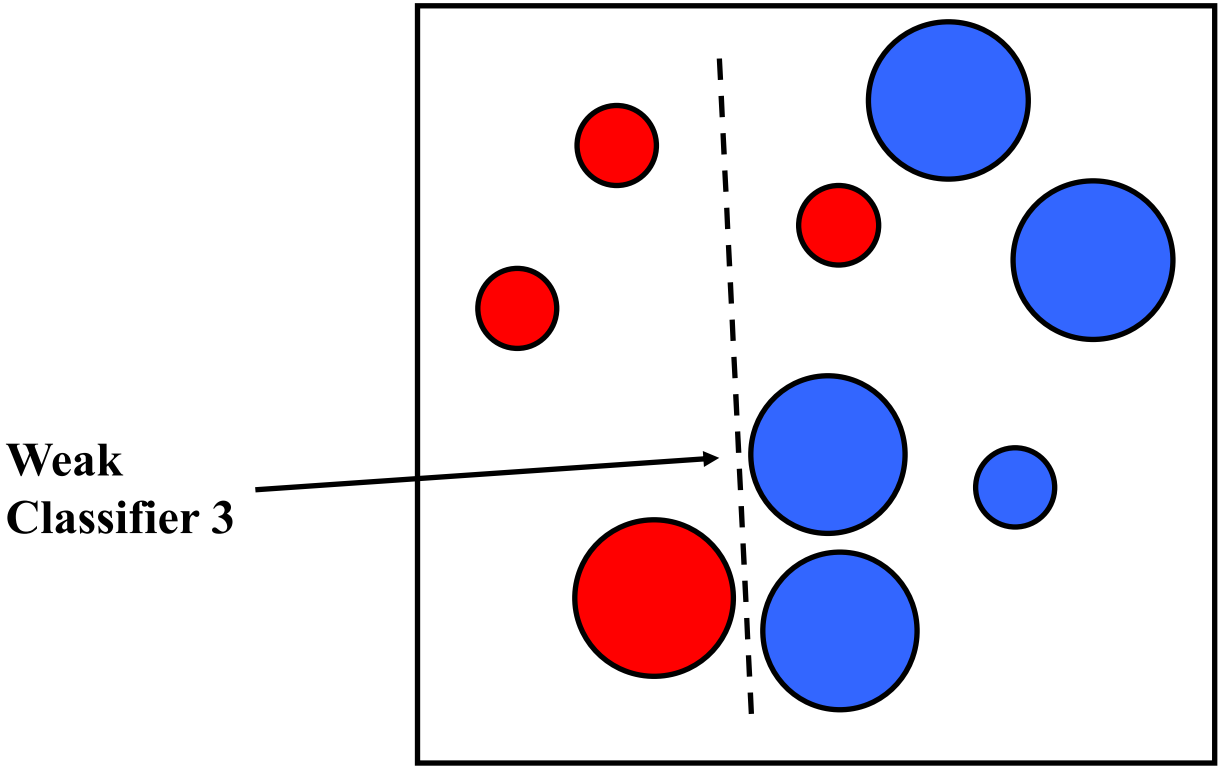
Illustration

MIMA



Illustration

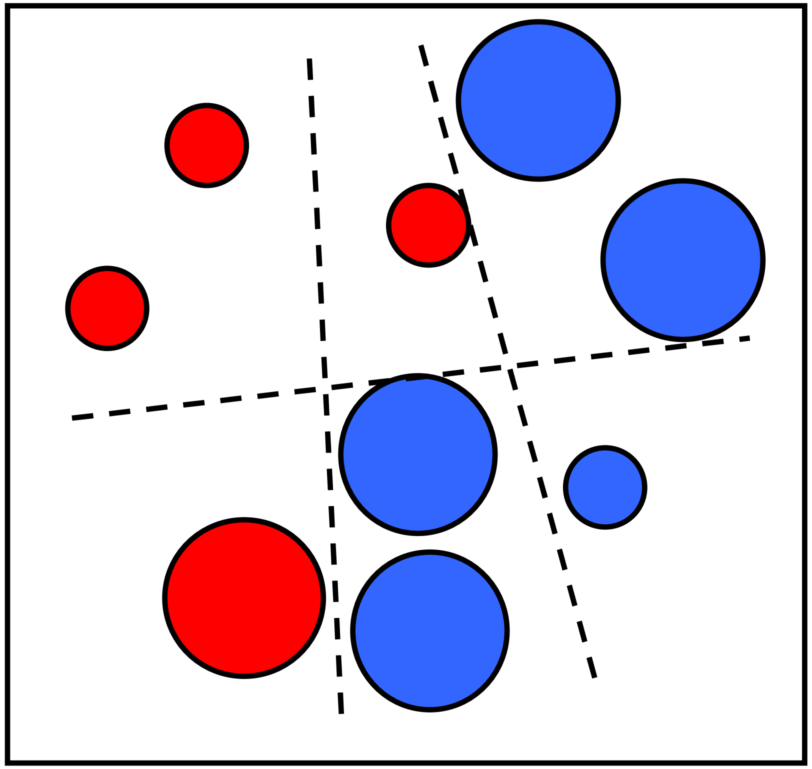
MIMA



Illustration

MIMA

Final classifier is a combination of weak classifiers



The AdaBoost Algorithm

MIMA

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X, y_i \in \{-1, +1\}$

Initialization: $D_1(i) = \frac{1}{m}, i = 1, \ldots, m$

For $t = 1, \ldots, T$:

How and why AdaBoost works?

- Weight classifier: $\alpha_t = \frac{1}{2} \ln \frac{1-\varepsilon_t}{\varepsilon_t}$

- Update distribution: $D_{t+1}(i) = \frac{D_t(i)\exp[-\alpha_t y_i h_t(x_i)]}{Z_t}$, Z_t is for normalization

Output final classifier: $sign\left(H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \right)$

The AdaBoost Algorithm

MIMA

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X, y_i \in \{-1, +1\}$

Initialization: $D_1(i) = \frac{1}{m}, i = 1, \ldots, m$

For $t = 1, \ldots, T$:

What goal the AdaBoost wants to reach?

- Weight classifier: $\alpha_t = \frac{1}{2} \ln \frac{1-\varepsilon_t}{\varepsilon_t}$
- Update distribution: $D_{t+1}(i) = \frac{D_t(i) \exp[-\alpha_t y_i h_t(x_i)]}{Z_t}$, Z_t is for normalization

Output final classifier: $sign\left(H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \right)$

The AdaBoost Algorithm

MIMA

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ wh

Initialization: $D_1(i) =$

For $t = 1, \ldots, T$:

They are goal dependent.

- Find classifier $h_t : X \to \{$ … i.e.,

$$h_t = \arg\min_{h_j} \varepsilon_j \quad \text{where } \varepsilon_j = \sum^{m} \ldots(i)[y_i \neq h_j(x_i)]$$

- Weight classifier: $\alpha_t = \frac{1}{2} \ln \frac{1-\varepsilon_t}{\varepsilon_t}$

- Update distribution: $D_{t+1}(i) = \frac{D_t(i)\exp[-\alpha_t y_i h_t(x_i)]}{Z_t}$, Z_t is for normalization

Output final classifier: $sign\left(H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \right)$

Goal

Final classifier: $sign\left(H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \right)$

Minimize exponential loss

$$loss_{\exp}\left[H(x)\right] = E_{x,y}\left[e^{-yH(x)}\right]$$

Maximize the margin $yH(x)$

Goal

MIMA

Final classifier: $sign\left(H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \right)$

Minimize $loss_{\exp}\left[H(x)\right] = E_{x,y}\left[e^{-yH(x)}\right]$

Define $H_t(x) = H_{t-1}(x) + \alpha_t h_t(x)$ with $H_0(x) = 0$

Then, $H(x) = H_T(x)$

$$E_{x,y}\left[e^{-yH_t(x)}\right] = E_x\left[E_y\left[e^{-yH_t(x)} \mid x\right]\right]$$

$$= E_x\left[E_y\left[e^{-y[H_{t-1}(x)+\alpha_t h_t(x)]} \mid x\right]\right]$$

$$= E_x\left[E_y\left[e^{-yH_{t-1}(x)} e^{-y\alpha_t h_t(x)} \mid x\right]\right]$$

$$= E_x\left[e^{-yH_{t-1}(x)}\left[e^{-\alpha_t} P(y = h_t(x)) + e^{\alpha_t} P(y \neq h_t(x))\right]\right]$$

$\alpha_t = ?$

Final classifier: $sign\left(H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \right)$

Minimize $loss_{\exp}\left[H(x)\right] = E_{x,y}\left[e^{-yH(x)}\right]$

Define $H_t(x) = H_{t-1}(x) + \alpha_t h_t(x)$ with $H_0(x) = 0$

Then, $H(x) = H_T(x)$

$$E_{x,y}\left[e^{-yH_t(x)}\right] = E_x\left[e^{-yH_{t-1}(x)}\left[e^{-\alpha_t}P(y = h_t(x)) + e^{\alpha_t}P(y \neq h_t(x))\right]\right]$$

Set $\dfrac{\partial}{\partial \alpha_t} E_{x,y}\left[e^{-yH_t(x)}\right] = 0$

$$\Rightarrow E_x\left[e^{-yH_{t-1}(x)}\underbrace{\left[-e^{-\alpha_t}P(y = h_t(x)) + e^{\alpha_t}P(y \neq h_t(x))\right]}_{0}\right] = 0$$

$\alpha_t = ?$

Final classifier: $sign\left(H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \right)$

Minimize $loss_{\exp}\left[H(x)\right] = E_{x,y}\left[e^{-yH(x)}\right]$

Define $H_t(x) = H_{t-1}(x) + \alpha_t h_t(x)$ with $H_0(x) = 0$

Then, $H(x) = H_T(x)$

$$\Rightarrow \alpha_t = \frac{1}{2}\ln\frac{P(y = h_t(x))}{P(y \neq h_t(x))} \Rightarrow \alpha_t = \frac{1}{2}\ln\frac{1-\varepsilon_t}{\varepsilon_t}$$

$P(x_i, y_i) = D_t(i)$

$$\varepsilon_t = P(\text{error}) \approx \sum_{i=1}^{m} D_t(i)[y_i \neq h_j(x_i)]$$

$$\Rightarrow E_x\left[e^{-yH_{t-1}(x)}\underbrace{\left[-e^{-\alpha_t}P(y = h_t(x)) + e^{\alpha_t}P(y \neq h_t(x))\right]}_{0}\right] = 0$$

$\alpha_t = ?$

Final classifier

Minimize los

Define $H_t(x) = $

Then, $H(x) = H$

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X, y_i \in \{-1, +1\}$

Initialization: $D_1(i) = \frac{1}{m}, i = 1, \ldots, m$

For $t = 1, \ldots, T$:

- Find classifier $h_t : X \to \{-1, +1\}$ which minimizes error wrt D_t, i.e.,

$$h_t = \arg\min_{h_j} \varepsilon_j \quad \text{where } \varepsilon_j = \sum_{i=1}^{m} D_t(i)[y_i \neq h_j(x_i)]$$

- Weight classifier: $\alpha_t = \frac{1}{2}\ln\frac{1-\varepsilon_t}{\varepsilon_t}$

- Update distribution: $D_{t+1}(i) = \frac{D_t(i)\exp[-\alpha_t y_i h_t(x_i)]}{Z_t}$, Z_t is for normalization

Output final classifier: $sign\left(H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)\right)$

$$\Rightarrow \alpha_t = \frac{1}{2}\ln\frac{P(y = h_t(x))}{P(y \neq h_t(x))} \Rightarrow \alpha_t = \frac{1}{2}\ln\frac{1-\varepsilon_t}{\varepsilon_t}$$

$P(x_i, y_i) = D_t(i)$

$$\varepsilon_t = P(\text{error}) \approx \sum_{i=1}^{m} D_t(i)[y_i \neq h_j(x_i)]$$

$$\Rightarrow E_x\left[e^{-yH_{t-1}(x)}\underbrace{\left[-e^{-\alpha_t}P(y = h_t(x)) + e^{\alpha_t}P(y \neq h_t(x))\right]}_{0}\right] = 0$$

$D_{t+1} = ?$

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X, y_i \in \{-1, +1\}$

Initialization: $D_1(i) = \frac{1}{m}, i = 1, \ldots, m$

For $t = 1, \ldots, T$:

- Find classifier $h_t : X \to \{-1, +1\}$ which minimizes error wrt D_t ,i.e.,

$$h_t = \arg\min_{h_j} \varepsilon_j \quad \text{where } \varepsilon_j = \sum_{i=1}^{m} D_t(i)[y_i \neq h_j(x_i)]$$

- Weight classifier: $\alpha_t = \frac{1}{2} \ln \frac{1-\varepsilon_t}{\varepsilon_t}$

- Update distribution: $D_{t+1}(i) = \frac{D_t(i)\exp[-\alpha_t y_i h_t(x_i)]}{Z_t}$, Z_t is for normalization

Output final classifier: $sign\left(H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)\right)$

Final classifier

Minimize los

Define $H_t(x) =$

Then, $H(x) = H$

$$\Rightarrow \alpha_t = \frac{1}{2} \ln \frac{P(y = h_t(x))}{P(y \neq h_t(x))} \Rightarrow \alpha_t = \frac{1}{2} \ln \frac{1-\varepsilon_t}{\varepsilon_t}$$

$$P(x_i, y_i) = D_t(i)$$

$$\varepsilon_t = P(\text{error}) \approx \sum_{i=1}^{m} D_t(i)[y_i \neq h_j(x_i)]$$

$$\Rightarrow E_x\left[e^{-yH_{t-1}(x)}\left[-e^{-\alpha_t} P(y = h_t(x)) + e^{\alpha_t} P(y \neq h_t(x))\right]\right] = 0$$

$$\underbrace{\left[-e^{-\alpha_t} P(y = h_t(x)) + e^{\alpha_t} P(y \neq h_t(x))\right]}_{0}$$

$D_{t+1} = ?$

Final classifier: $sign\left(H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \right)$

Minimize $loss_{\exp}\left[H(x)\right] = E_{x,y}\left[e^{-yH(x)}\right]$

Define $H_t(x) = H_{t-1}(x) + \alpha_t h_t(x)$ with $H_0(x) = 0$

Then, $H(x) = H_T(x)$

$$E_{x,y}\left[e^{-yH_t}\right] = E_{x,y}\left[e^{-yH_{t-1}} e^{-y\alpha_t h_t}\right] \approx E_{x,y}\left[e^{-yH_{t-1}}\left(1 - y\alpha_t h_t + \frac{1}{2}\alpha_t^2 y^2 h_t^2\right)\right]$$

$$\Rightarrow h_t = \arg\min_h E_{x,y}\left[e^{-yH_{t-1}}\left(1 - y\alpha_t h + \frac{1}{2}\alpha_t^2 y^2 h^2\right)\right]$$

$y^2 h^2 = 1$

$$\Rightarrow h_t = \arg\min_h E_{x,y}\left[e^{-yH_{t-1}}\left(1 - y\alpha_t h + \frac{1}{2}\alpha_t^2\right)\right]$$

$$\Rightarrow h_t = \arg\min_h E_x\left[E_y\left[e^{-yH_{t-1}}\left(1 - y\alpha_t h + \frac{1}{2}\alpha_t^2\right)\right] \mid x\right]$$

$D_{t+1} = ?$

MIMA

Final classifier: $sign\left(H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)\right)$

Minimize $loss_{\exp}\left[H(x)\right] = E_{x,y}\left[e^{-yH(x)}\right]$

Define $H_t(x) = H_{t-1}(x) + \alpha_t h_t(x)$ with $H_0(x) = 0$

Then, $H(x) = H_T(x)$

$$\Rightarrow h_t = \arg\max_h E_x\left[1 \cdot h(x) e^{-H_{t-1}(x)} \cdot P(y=1 \mid x) + (-1) \cdot h(x) e^{H_{t-1}(x)} \cdot P(y=-1 \mid x)\right]$$

$$\Rightarrow h_t = \arg\max_h E_x\left[E_y\left[e^{-yH_{t-1}}\left(yh\right)\right] \mid x\right]$$

$$\Rightarrow h_t = \arg\min_h E_x\left[E_y\left[e^{-yH_{t-1}}\left(-y\alpha_t h\right)\right] \mid x\right]$$

$$\Rightarrow h_t = \arg\min_h E_x\left[E_y\left[e^{-yH_{t-1}}\left(1 - y\alpha_t h + \frac{1}{2}\alpha_t^2\right)\right] \mid x\right]$$

$D_{t+1} = ?$

Final classifier: $sign\left(H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \right)$

Minimize $loss_{\exp}[H(x)] = E_{x,y}\left[e^{-yH(x)} \right]$

Define $H_t(x) = H_{t-1}(x) + \alpha_t h_t(x)$ with $H_0(x) = 0$

Then, $H(x) = H_T(x)$

$$\Rightarrow h_t = \arg\max_h E_x\left[1 \cdot h(x) e^{-H_{t-1}(x)} \cdot P(y=1 \mid x) + (-1) \cdot h(x) e^{H_{t-1}(x)} \cdot P(y=-1 \mid x) \right]$$

$$\Rightarrow h_t = \arg\max_h E_{x,y \sim e^{-yH_{t-1}(x)} P(y|x)} [yh(x)]$$ maximized when $y = h(x) \quad \forall x$

$$\Rightarrow h_t(x) = sign\left(E_{x,y \sim e^{-yH_{t-1}(x)} P(y|x)} [y \mid x] \right)$$

$$\Rightarrow h_t(x) = sign\left(P_{x,y \sim e^{-yH_{t-1}(x)} P(y|x)}(y = 1 \mid x) - P_{x,y \sim e^{-yH_{t-1}(x)} P(y|x)}(y = -1 \mid x) \right)$$

$D_{t+1} = ?$

MIMA

Final classifier: $sign\left(H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \right)$

Minimize $loss_{\exp}[H(x)] = E_{x,y}\left[e^{-yH(x)} \right]$

Define $H_t(x) = H_{t-1}(x) + \alpha_t h_t(x)$ with $H_0(x) = 0$

Then, $H(x) = H_T(x)$

At time t $x, y \sim e^{-yH_{t-1}(x)} P(y \mid x)$

$$\Rightarrow h_t(x) = sign\left(P_{x,y \sim e^{-yH_{t-1}(x)} P(y|x)}(y = 1 \mid x) - P_{x,y \sim e^{-yH_{t-1}(x)} P(y|x)}(y = -1 \mid x) \right)$$

$D_{t+1} = ?$

Final classifier

Minimize los

Define $H_t(x) = $

Then, $H(x) = H$

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X, y_i \in \{-1, +1\}$

Initialization: $D_1(i) = \frac{1}{m}, i = 1, \ldots, m$

For $t = 1, \ldots, T$:

- Find classifier $h_t : X \to \{-1, +1\}$ which minimizes error wrt D_t, i.e.,

$$h_t = \arg\min_{h_j} \varepsilon_j \text{ where } \varepsilon_j = \sum_{i=1}^{m} D_t(i)[y_i \neq h_j(x_i)]$$

- Weight classifier: $\alpha_t = \frac{1}{2} \ln \frac{1 - \varepsilon_t}{\varepsilon_t}$
- Update distribution: $D_{t+1}(i) = \frac{D_t(i) \exp[-\alpha_t y_i h_t(x_i)]}{Z_t}$, Z_t is for normalization

Output final classifier: $sign\left(H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \right)$

At time t $\quad x, y \sim e^{-yH_{t-1}(x)} P(y \mid x)$

At time 1 $\quad x, y \sim P(y \mid x) \qquad P(y_i \mid x_i) = 1 \Rightarrow D_1(1) = \frac{1}{Z_1} = \frac{1}{m}$

At time $t+1$ $\quad x, y \sim e^{-yH_t(x)} P(y \mid x) \equiv D_t e^{-\alpha_t y h_t(x)}$

$$\Rightarrow D_{t+1}(i) = \frac{D_t(i) \exp[-\alpha_t y_i h_t(x_i)]}{Z_t}, \; Z_t \text{ is for normalization}$$

Successful applications of ensembles

- Netflix prize

Users rate movies (1,2,3,4,5 stars);
Netflix makes suggestions to users based on previous rated movies.

Netflix prize

MIMA

"The Netflix Prize seeks to substantially improve the accuracy of predictions about how much someone is going to love a movie based on their movie preferences. Improve it enough and you win one (or more) Prizes. Winning the Netflix Prize improves our ability to connect people to the movies they love."

Netflix prize

MIMA

- Supervised learning task
  - Training data is a set of users and ratings (1,2,3,4,5 stars) those users have given to movies.
  - Construct a classifier that given a user and an unrated movie, correctly classifies that movie as either 1, 2, 3, 4, or 5 stars

$1 million prize for a 10% improvement over Netflix's current movie recommender/classifier

Netflix prize

MIMA

Netflix Prize

COMPLETED

Home | Rules | Leaderboard | Update

Progress Prize 2007

Description:	8.43% Improvement over Cinematch by team KorBell
Required Quiz RMSE:	0.9419
Required Test RMSE:	0.9430
Winning Quiz RMSE:	0.8712
Winning Test RMSE:	0.8723
Winning team:	KorBell
Date awarded:	2007-11-13 15:03:28
Algorithm description:	See the Forum announcement

Netflix prize

- The final solution consists of blending 107 individual results.

Netflix prize

Netflix Prize COMPLETED

Home | Rules | Leaderboard | Update

Grand Prize

Description:	10% improvement over Cinematch on the test subset
Required Quiz RMSE:	0.8563
Required Test RMSE:	0.8573
Winning Quiz RMSE:	0.8554
Winning Test RMSE:	0.8567
Winning team:	BellKor's Pragmatic Chaos
Date awarded:	2009-09-21 18:30:00
Algorithm description:	See the Forum announcement

Netflix prize

MIMA

Leaderboard

Showing Test Score. Click here to show quiz score

Display top 20 **leaders.**

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace_	0.8612	9.59	2009-07-24 17:18:43
9	Feeds2	0.8622	9.48	2009-07-12 13:11:51
10	BigChaos	0.8623	9.47	2009-04-07 12:33:59
11	Opera Solutions	0.8623	9.47	2009-07-24 00:34:07
12	BellKor	0.8624	9.46	2009-07-26 17:19:11

【 Thank You ! 】

Any Question?