Lecture Slides for

INTRODUCTION TO

Machine Learning

ETHEM ALPAYDIN
© The MIT Press, 2004

Edited for CS 536 Fall 2005 – Rutgers University
Ahmed Elgammal

alpaydin@boun.edu.tr
http://www.cmpe.boun.edu.tr/~ethem/i2ml

CHAPTER 15:

Combining Multiple Learners

Rationale

- No Free Lunch thm: There is no algorithm that is always the most accurate
- Generate a group of base-learners which when combined has higher accuracy
- Different learners use different
  - Algorithms
  - Hyperparameters
  - Representations (Modalities)
  - Training sets
  - Subproblems

Lecture Notes for E Alpaydın 2004 Introduction to Machine Learning © The MIT Press (V1.1)

Voting

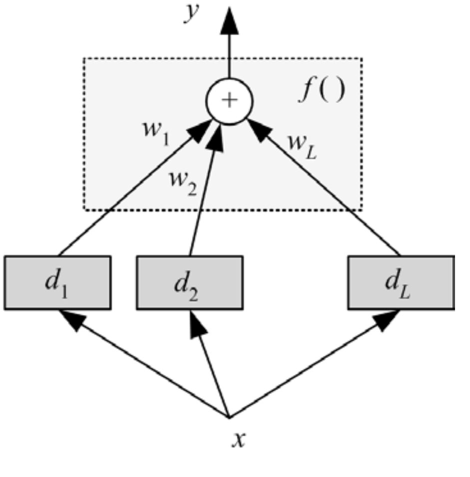
- Linear combination

$$y = \sum_{j=1}^{L} w_j d_j$$

$$w_j \geq 0 \text{ and } \sum_{j=1}^{L} w_j = 1$$

- Classification

$$y_i = \sum_{j=1}^{L} w_j d_{ji}$$



Lecture Notes for E Alpaydın 2004 Introduction to Machine Learning © The MIT Press (V1.1)

- Bayesian perspective:

$$P(C_i \mid x) = \sum_{\text{all models } \mathcal{M}_j} P(C_i \mid x, \mathcal{M}_j) P(\mathcal{M}_j)$$

- If d_j are iid

$$E[y] = E\left[\sum_j \frac{1}{L} d_j\right] = \frac{1}{L} L \cdot E[d_j] = E[d_j]$$

$$\text{Var}(y) = \text{Var}\left(\sum_j \frac{1}{L} d_j\right) = \frac{1}{L^2} \text{Var}\left(\sum_j d_j\right) = \frac{1}{L^2} L \cdot \text{Var}(d_j) = \frac{1}{L} \text{Var}(d_j)$$

Bias does not change, variance decreases by L

- Average over randomness

Lecture Notes for E Alpaydın 2004 Introduction to Machine Learning © The MIT Press (V1.1)

Bagging

- Use bootstrapping to generate L training sets and train one base-learner with each (Breiman, 1996)
- Draw L training sets at random with replacement.
- Use voting (Average or median with regression)
- Unstable algorithms profit from bagging
- Unstable algorithms: if small changes in the training set causes large difference in the generated learner: the algorithm has high variance. E.g., decision trees, multilayer perceptrons.

Lecture Notes for E Alpaydın 2004 Introduction to Machine Learning © The MIT Press (V1.1)

Boosting

- In bagging: generating complementary base-learner is left to chance and to the unstability of the learning methods
- In Boosting: actively try to generate complementary base-learner
- How: by training the next learner based on the mistakes of previous learners.
- Schapire 1990: combine three weak learners to generate a strong learner.
- Weak learner: error probability less than 1/2

AdaBoost

Adaptive Boosting:

Generate a sequence of base-learners each focusing on previous one's errors

(Freund and Schapire, 1996)

Training:

For all $\{x^t, r^t\}_{t=1}^N \in \mathcal{X}$, initialize $p_1^t = 1/N$

For all base-learners $j = 1, \ldots, L$

Randomly draw $\mathcal{X}_j$ from $\mathcal{X}$ with probabilities p_j^t

Train d_j using $\mathcal{X}_j$

For each (x^t, r^t), calculate $y_j^t \leftarrow d_j(x^t)$

Calculate error rate: $\epsilon_j \leftarrow \sum_t p_j^t \cdot 1(y_j^t \neq r^t)$

If $\epsilon_j > 1/2$, then $L \leftarrow j - 1$; stop

$\beta_j \leftarrow \epsilon_j / (1 - \epsilon_j)$

For each (x^t, r^t), decrease probabilities if correct:

If $y_j^t = r^t$ $p_{j+1}^t \leftarrow \beta_j p_j^t$ Else $p_{j+1}^t \leftarrow p_j^t$

Normalize probabilities:

$Z_j \leftarrow \sum_t p_{j+1}^t; \quad p_{j+1}^t \leftarrow p_{j+1}^t / Z_j$

Testing:

Given x, calculate $d_j(x), j = 1, \ldots, L$

Calculate class outputs, $i = 1, \ldots, K$:

$$y_i = \sum_{j=1}^L \left(\log \frac{1}{\beta_j}\right) d_{ji}(x)$$

AdaBoost

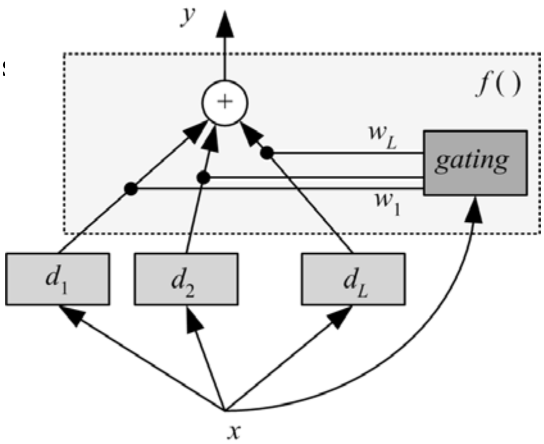
- AdaBoost works because it increases the margin at each step as the sample probabilities change
- Not all algorithms will benefit from Boosting
- Base-learner has to be simple and not accurate (high variance)

Mixture of Experts

Voting where weight:

$$y = \sum_{j=1}^{L} w_j d_j$$

(Jacobs et al., 1991)
Experts or gating can be nonlinear

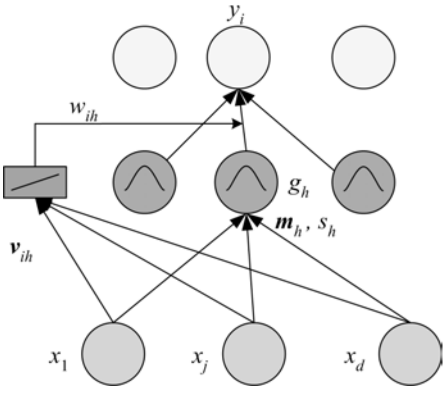


Mixture of Experts

- In RBF, each local fit is a constant, w_{ih}, second layer weight
- In MoE, each local fit is a linear function of x, a local expert:

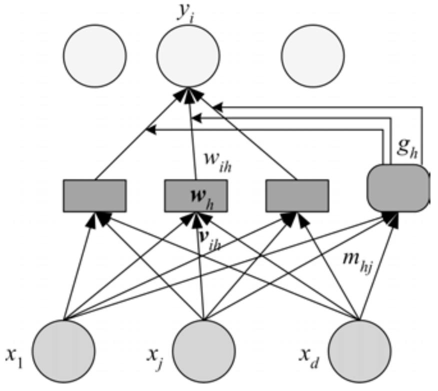
$$w_{ih}^t = \boldsymbol{v}_{ih}^t \boldsymbol{x}^t$$

(Jacobs et al., 1991)



Lecture Notes for E Alpaydın 2004 Introduction to Machine Learning © The MIT Press (V1.1)

MoE as Models Combined



- Radial gating:

$$g_h^t = \frac{\exp\left[-\left\|\boldsymbol{x}^t - \boldsymbol{m}_h\right\|^2 / 2s_h^2\right]}{\sum_l \exp\left[-\left\|\boldsymbol{x}^t - \boldsymbol{m}_l\right\|^2 / 2s_l^2\right]}$$

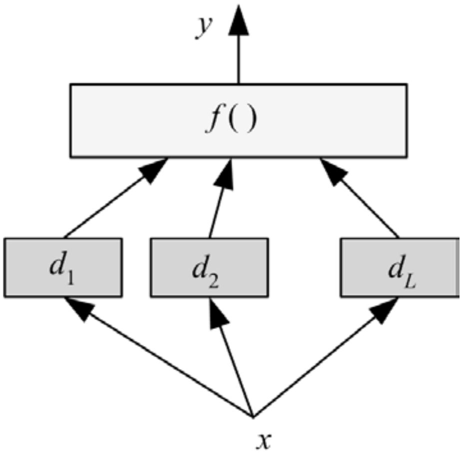
- Softmax gating:

$$g_h^t = \frac{\exp\left[\boldsymbol{m}_h^T \boldsymbol{x}^t\right]}{\sum_l \exp\left[\boldsymbol{m}_l^T \boldsymbol{x}^t\right]}$$

Lecture Notes for E Alpaydın 2004 Introduction to Machine Learning © The MIT Press (V1.1)

Stacking

- Combiner *f* () is another learner (Wolpert, 1992)

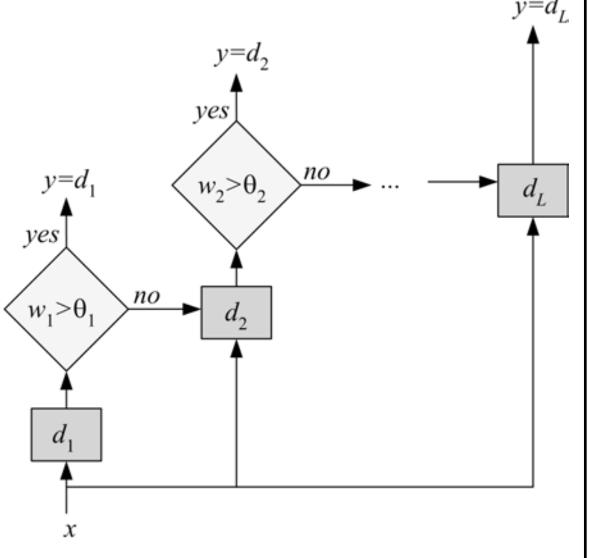


Lecture Notes for E Alpaydın 2004 Introduction to Machine Learning © The MIT Press (V1.1)

Cascading

Use d_j only if preceding ones are not confident

Cascade learners in order of complexity



Lecture Notes for E Alpaydın 2004 Introduction to Machine Learning © The MIT Press (V1.1)