Boosting

Yi Zhang

1

How to Make a Good Decision?

- Ask expert for rule
- Collect set of examples where rule fails (hard cases)
- Ask expert again for selected set of hard cases
- And so on…

- Take weighted majority vote of rules

2

Boosting Classifiers

- Learn a classifier from a single training set
- Reweighting examples to emphasize misclasified
- Relearn a classifier from reweighted examples
- And so on….

- Final classifer = weighted combination of classifiers

3

Boosting

- Definition of Boosting
- Advantages of Boosting
- More deeper understanding of Boosting
  - Intuitive view
  - Maximizes margins – relation to SVM
  - Statistics view
  - Multiple classes
- Conclusion

4

More formal definition of Boosting

- Training data: $\{(x_1, y_1), \ldots (x_m, y_{m1})\}$ $\{1, -1\}$
- For t=1 to T:
  - Get a distribution D_t on training data
  - Find weak classifier:

    $h_t : X \rightarrow \{-1, 1\}$

    With error $\varepsilon_t = \Pr(h_t(x_i) \neq y_i \mid D_t)$
- Output final classifier

AdaBoost (97)

- Initialize $D_1 = 1/m$
- Given D_t & h_t

$$D_{t+1}(i) = \frac{D_t \exp(-\alpha_t y_i h_t(x_i))}{Z_t} = \frac{D_t}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

Z_t : *nomalization factor* $\alpha_t = \frac{1}{2} \ln(\frac{1-\varepsilon_t}{\varepsilon_t})$

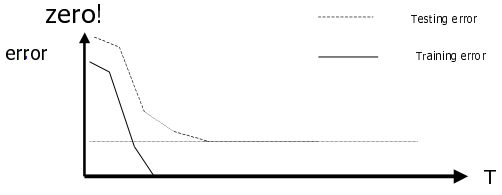
- Output final classifier:

Advantage of Boosting☺

- Simple and easy to program
- No parameter to tune except T (number of weak learners)
- Can be combined with any classifier only if those classifiers work better than random: things turn out to be so easy
- Find outliers – maybe a disadvantage: susceptible to noise

Some thing even more nice ☺

- No over fitting(in most case) :Test error continue to drop after training error is zero!



Why?

- **Why it works so well ?**

9

Boosting(Intuitively:Training error is small)

- Weights of incorrectly classified examples are increased so that the weak learner is forced to focus on the hard case
- If x_i is incorrectly classified by H (final classifier), then it is incorrectly classified by weighted majority of h_t , thus x_i must have a large weight in final distributuion D_{T+1}
- Total weight is 1 => number of incorrectly classified examples is small

10

Boosting(Intuitively: Testing error is also small)

- Variance is too high for a single rule from the weak learner, thus error observed on training data could vary greatly on new test data
- Greatly reduce the variance by inducing multiple rules from the same data.

11

Margine

- Define : $\mathbf{margine\ of\ (x,y)=y*f(x)}$
- Boosting is minimize

$$\sum_i Exp(-y_i f(x_i)) = \sum_i Exp(-y_i \sum_t \alpha_t h_t(x_i)))$$

- Thus maximize margine
- Testing error continues to drop after training error is zero

12

Margine(2): relation to SVM

- Both maximize margins:

$$\theta = \max_{w} \min_{i} \left(\frac{(\vec{\alpha} * h(x_i)) y_i}{\| \vec{\alpha} \|} \right)$$

- SVM: Euclidean norm for $\vec{\alpha}$
- AdaBoost : Manhattan norm $\vec{\alpha}$

13

Statistics View (Too theoretical☹)

- Structural Model: additive on the logistic scale with the weak learner as basis element $F(x) = \sum_{j=1}^{T} f_j(x_j)$
- The AdaBoost algorithm produces adaptive Newton updates for minimizing

$$E(e^{-yF(x)})$$

(Details can be found in Jermome's paper: Additive Logistic Regression: a Statistical View of Boosting)

14

Why minimizing $E(e^{-yF(x)})$

- Maximizing Margine (Previous slides)
- Equivalent to maximizing the log likelihood of binormial distribution of y:

$$P(y=1 \mid x) = p(x) = \frac{e^{F(x)}}{e^{F(x)} + e^{-F(x)}}$$

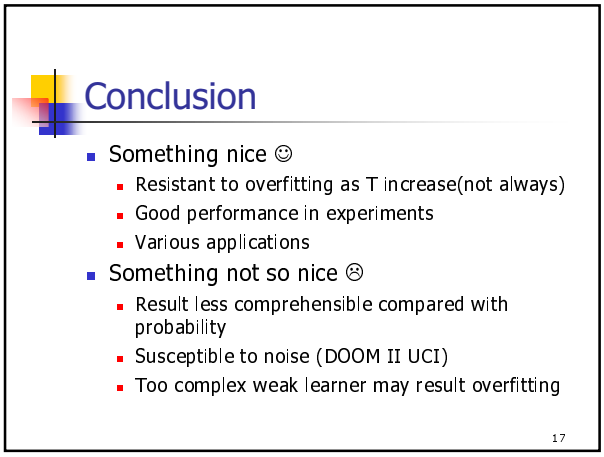
$$E\ell(y, p(x)) = E(\log(P(y \mid p(x)))) = -E \log(1 + e^{-yF(x)})$$

15

Multiclass Problems

- Reduce to binary problem by creating several questions for each case:
  - x belongs to class 1?
  - x belongs to class 2?
  - x belongs to class 3?....
- More details (Jerom's paper P21)

16


Conclusion
Something nice ☺
Resistant to overfitting as T increase(not always)
Good performance in experiments
Various applications
Something not so nice ☹
Result less comprehensible compared with probability
Susceptible to noise (DOOM II UCI)
Too complex weak learner may result overfitting
17