Notes on Boosting

Shasha Liao
Georgia Tech

November 3, 2020

Rationale: Combination of methods
General machine learning tasks, there is no algorithm that is always the most accurate.
"Can a set of weak learners create a single strong learner? " - Kearns and Valiant (1988, 1989)

1 Two main approaches

- Boosting
  - Run weak learner on weighted example set
  - Combine weak learners linearly
  - Require knowledge on the performance of weak learner
- Bagging
  - Run weak learners on bootstrap replicates of the training set
  - Average weak learners
  - Reduces variance

2 Boosting

Boosting: general methods of converting weak learners into a more powerful one

- Each learner is independent on the previous one and focuses on the previous one's errors
- Data that are incorrectly predicted in the previous rounds are weighted more heavily when deciding a new learner.

Questions:

- How to adjust weights for data?
- How to combine learners?

Boosting Setup

- Given a set of base classifiers $\{h_1, h_2, ...\}$, $h_i : X \to \{1, -1\}$.
- Training data:
$$(x^i, y^i), i = 1, ..., m, x^i \in X \text{ and } y^i \in \{1, -1\}$$
- Construct:
  - a sequence of distributions (weights on data sum up to 1) $D(i), i = 1, ..., m$.
  - a sequence $\{\alpha_k\}$ of nonnegative weights of base classifiers
  - final classifier performs significantly better than any base classifier

AdaBoost:

- Adaptive in the sense that subsequent weak learners are adjusted in favor of those instances misclassified by previous classifiers
- Distribution and weights are adjusted adaptively.

There are many variant of AdaBoost. Here is a basic one:

1. Construct $D_t : t = 1, ..., T$, where T is the number of iterations for updating distribution and weights.
2. Initialize $D_1(i)$ for $i = 1, 2, ..., m$.
3. Given D_t and weak learner h_t:
   Compute weighted misclassification rate:
$$\epsilon_t = \sum_{i=1}^{m} D_t(i)\mathbb{I}\{y^i \neq h_t(x^i)\}$$
   Compute weight for h_t:
$$\alpha_t = \frac{1}{2}\ln(\frac{1-\epsilon_t}{\epsilon_t})$$
4. Update weights for data:
$$D_{t+1}(i) = \frac{D_t(i)}{Z_t}e^{-\alpha_t y^i h_t(x^i)} = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if} \quad y^i = h_t(x^i) \\ e^{\alpha_t} & \text{otherwise.} \end{cases}$$
   Here Z_t = normalizing constant. $Z_t = \sum_{i=1}^{m} D_t(i)e^{-\alpha_t y^i h_t(x^i)}$.
5. Final classifier:
$$H(x) = \text{sign}(\sum_{t=1}^{T} \alpha_t h_t(x)).$$

Note that large ϵ_t gives small weight α_t and when $\epsilon_t > 0.5$, $\alpha_t < 0$ and the points classified correctly by h_t will have a higher weight in the next round. When $\epsilon_t < 0.5$, $\alpha_t > 0$ and the points classified incorrectly by h_t will have a higher weight in the next round. Check a toy example in the slides for better understanding.

Question: How are the weak learners h_t chosen? How are the updating rules in step 3 and 4 decided?

- Combined classifier:

$$f_t(x) = \alpha_1 h_1(x; \theta_1) + \cdots + \alpha_t h_t(x; \theta_t)$$

- Exponential Loss:

$$\hat{L}(f_t) = \frac{1}{m} \sum_{i=1}^{m} \exp(-y^i f_t(x^i)) = \frac{1}{m} \Pi_{s=1}^{t} Z_s \sum_{i=1}^{m} D_t(i) e^{-\alpha_t y^i h_t(x^i; \theta_t)}.$$

- We find h_t by finding θ_t that minimizes $\hat{L}(f_t)$, which also approximately minimizes ϵ_t.

- We find α_t by setting

$$\frac{\partial \hat{L}}{\partial \alpha_t} = 0,$$

  which gives the updating rule in step 3.

- Why do we update D_t the way in step 4?

See more detailed steps in the slides.

3 Extensions

- There are many other more recent algorithms such as:
  LPBoost, TotalBoost, BrownBoost, XGBoost, MadaBoost, LogitBoost

- Generalization: **Gradient boosting** is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically **decision trees**; allowing optimization of an arbitrary differentiable loss function.