Seminar 2. Gradient Boosting

Alexander Romanenko

MIPT, SRR

16/09/2015
Moscow

Table of content

Composition

$X^l = (x_i, y_i)_{i=1}^l \in X \times Y$ — обучающая выборка, $y_i = y^*(x_i)$;

$a(x) = C(b(x))$ — алгоритм, где

- $b : X \to R$ — базовые алгоритмы,
- $C : R \to Y$ — решающее правило,
- R — пространство оценок

Определение

Композиция базовых алгоритмов $b_1, \ldots, b_T$

$$a(x) = C(F(b_1(x), \ldots, b_T(x))),$$

где $F : R^T \to R$ — корректирующий оператор

Boosting for binary classification

$Y = \{-1, +1\}$, $b_t : X \to \{-1, 0, +1\}$, $C(b) = \text{sign}(b)$
Weighted voting

$$a(x) = \text{sign}(\sum_{t=1}^{T} \alpha_t b_t(x)), x \in X.$$

Number of error on X^l:

$$Q_T = \sum_{i=1}^{l} \left[y_i \sum_{t=1}^{T} \alpha_t b_t(x_i) < 0 \right]$$

Heuristics:

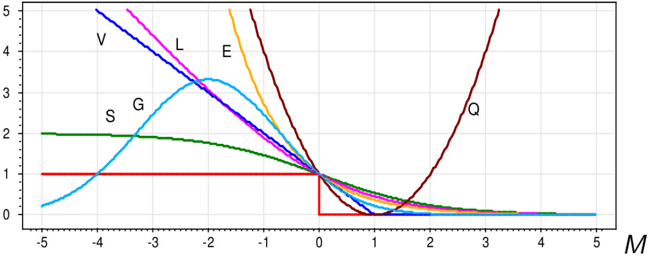
- Fix $\alpha_1 b_1(x), \dots, \alpha_{t-1} b_{t-1}(x)$, then find $\alpha_t b_t(x)$
- Plain approximation of $[M \leqslant 0]$

Variety of boosting

Vast variety of boosting algorithms. We can vary:

- Loss functions (type of solving task)
- Base algorithms
- Optimization methods
- Way of composition construction
- Methods of noise features control

Loss functions



$E(M) = e^{-M}$ — экспоненциальная (AdaBoost);
$L(M) = \log_2(1 + e^{-M})$ — логарифмическая (LogitBoost);
$Q(M) = (1 - M)^2$ — квадратичная (GentleBoost);
$G(M) = \exp(-cM(M + s))$ — гауссовская (BrownBoost);
$S(M) = 2(1 + e^{M})^{-1}$ — сигмоидная;
$V(M) = (1 - M)_+$ — кусочно-линейная (из SVM);

Anyboost

Вход: обучающая выборка X^ℓ; параметр T;

Выход: базовые алгоритмы и их веса $\alpha_t b_t$, $t = 1, \dots, T$;

1: инициализировать отступы: $M_i := 0$, $i = 1, \dots, \ell$;

2: **для всех** $t = 1, \dots, T$

3: вычислить веса объектов:
$w_i = -\mathscr{L}'(M_i)$, $i = 1, \dots, \ell$;

4: обучить базовый алгоритм согласно принципу DOOM:
$$b_t := \arg\max_b \sum_{i=1}^{\ell} w_i y_i b(x_i);$$

5: решить задачу одномерной минимизации:
$$\alpha_t := \arg\min_{\alpha > 0} \sum_{i=1}^{\ell} \mathscr{L}\big(M_i + \alpha b_t(x_i) y_i\big);$$

6: обновить значения отступов:
$M_i := M_i + \alpha_t b_t(x_i) y_i$; $i = 1, \dots, \ell$;

Task of Gradient Boosting Machine

Task of regression: $Y = \mathbb{R}$, training set X^l,
Goal: to construct composition $F_M(x)$:

$$F_M(x) = c + \sum_{m=1}^{M} \alpha_m h_m(x),$$

$$\mathbb{E}_{x,y} L(y, F(y)) \to \min$$

where

$L(y, F(x))$ — differentiable loss function;
$h_m \in \mathcal{H} = \{h : \mathbb{X} \to \mathbb{R}\}, m = 1, \ldots, M;$
$c \equiv const$, $\mathcal{H}$ — assemblage of base functions.

GBM

GMB as boosting solves task with greedy and stepwise procedure.
Initially:

$$F_0(x) \equiv c = \operatorname*{arg\,min}_{\alpha} \sum_{i=1}^{l} L(y_i, \alpha),$$

On step t :

$$(h_t, \alpha_t) = \operatorname*{arg\,min}_{h_t, \alpha_t} \sum_{i=1}^{l} L(y_i, F_{t-1}(x_i) + \alpha_t h_t(x_i)),$$

$$F_t(x) = F_{t-1}(x) + \alpha_t h_t(x)$$

Idea of GBM

$L(y, F(x))$ is function with l arguments on X^l:

$$F(x_1), \ldots, F(x_l).$$

So we can consider not functions $F(x)$ but l-dimension space of values in on training set.

Task on iteration t is to make step $\boldsymbol{\delta}_t$:

$$\boldsymbol{f}_t = \boldsymbol{f}_{t-1} + \boldsymbol{\delta}_t = (F_{t-1}(x_1) + \boldsymbol{\delta}_t^1, \ldots, F_{t-1}(x_l) + \boldsymbol{\delta}_t^l)$$

to minimize value $\mathcal{L}(\boldsymbol{f}_t)$:

$$\mathcal{L}(\boldsymbol{f}) = \sum_{i=1}^{l} L(y_i, \boldsymbol{f}^i) = \sum_{i=1}^{l} L(y_i, F(x_i)).$$

Idea of GBM

We can find $\boldsymbol{g}_t = \nabla L(\boldsymbol{f}_{t-1})$ and make step $\boldsymbol{\delta}_t = -\alpha_t \boldsymbol{g}_t$, $\alpha_t > 0$.
Evidently

$$\boldsymbol{g}_t^i = \left.\frac{\partial L(y_i, z)}{\partial z}\right|_{z=F_{t-1}(x_i)}.$$

Stride parameter α_t we can find as solution of 1d minimization task:

$$\alpha_t = \arg\min_{\alpha} \sum_{i=1}^{l} L(y_i, \boldsymbol{f}_{t-1}^i - \alpha \boldsymbol{g}_t^i).$$

Finally,

$$\boldsymbol{f}_t = \boldsymbol{f}_{t-1} - \alpha_t \boldsymbol{g}_t.$$

Idea of GBM

BUT we just can make steps along definite base functions h.

Solution: let's find the most «co-directional» h to anti-gradient:

$$h_t = \operatorname*{arg\,min}_{\beta,h} \sum_{i=1}^{l} \left(-\boldsymbol{g}_t^i - \beta h(x_i)\right)^2$$

Weight of base function h_t :

$$\alpha_t = \operatorname*{arg\,min}_{\alpha} \sum_{i=1}^{l} L(y_i, F_{t-1}(x_i) + \alpha h_t(x_i)),$$

Algorithm of GBM

1. Initialization: $F_0(x) = \underset{\alpha}{\arg\min} \sum_{i=1}^{l} L(y_i, \alpha)$
2. For $t = 1, \ldots, M$:
    1. For $l = 1, \ldots, l$ calculate $\boldsymbol{g}_t^i = \left.\frac{\partial L(y_i, z)}{\partial z}\right|_{z = F_{t-1}(x_i)}$
    2. Finding next base function h_t:
    $$h_t = \underset{\beta, h}{\arg\min} \sum_{i=1}^{l} \left(-\boldsymbol{g}_t^i - \beta h(x_i)\right)^2$$
    3. Finding weight α of base function h_t :
    $$\alpha_t = \underset{\alpha}{\arg\min} \sum_{i=1}^{l} L(y_i, F_{t-1}(x_i) + \alpha h_t(x_i))$$
    4. Update composition: $F_t(x) = F_{t-1}(x) + \alpha_t h_t(x)$

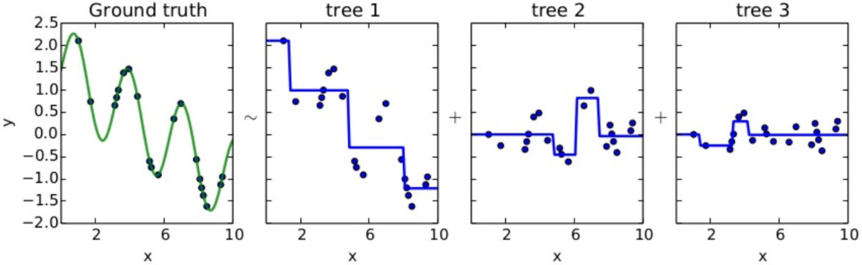
Question

$L(y, F(x)) = (y - F(x))^2$. Find $\boldsymbol{g}_t$?

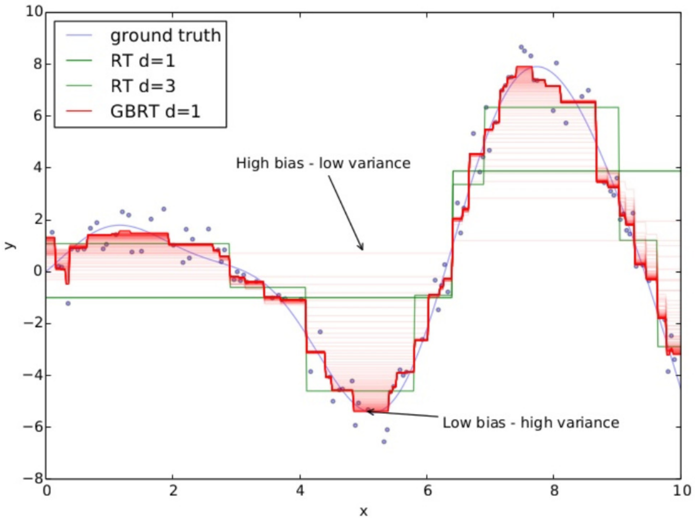
Question

$$L(y, F(x)) = (y - F(x))^2. \text{ Find } \boldsymbol{g}_t?$$

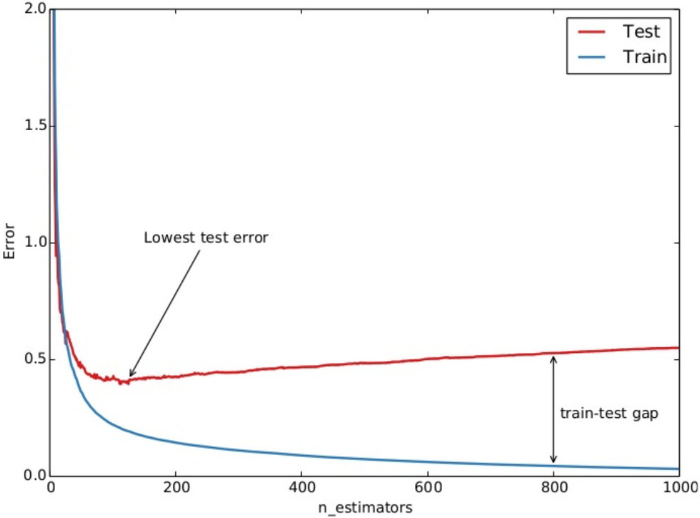
It's example of residual fitting



Comparison with random trees



Complexity and overfitting



Stochastic GB

Let's find h_t on random subset $\tilde{X}_t$ of training set X^l.
So finding next base function h_t will be:

$$h_t = \underset{\beta,h}{\arg\min} \sum_{\tilde{x} \in \tilde{X}_t} \left(-\boldsymbol{g}_t^i - \beta h(\tilde{x})\right)^2$$

Properties

- Decreasing overfitting
- Speed up solving of minimization task
- Good performance gain on practice
- Recommended value is 0.5

Shrinkage

Another regularization strategy is to scale the contribution of each weak learner by a factor ν :

$$F_t(x) = F_{t-1}(x) + \nu \alpha_t h_t(x)$$

Properties

- Decreasing overfitting
- Usually slowing convergence
- Good performance gain on practice
- Recommended value is 0.1

Shrinkage and Stochastic GBM