Стохастический градиентный бустинг (SGB)

Известно, что рандомизации могут повышать качество композиции за счёт повышения различности базовых алгоритмов (на этом основаны bagging, RF, RSM)

Идея:
на шагах 3–5 использовать не всю выборку X^ℓ,
а случайную подвыборку с повторениями, как в бэггинге.

Преимущества:

- улучшается качество
- улучшается сходимость
- уменьшается время обучения

Friedman G. Stochastic Gradient Boosting. 1999.

Частный случай — алгоритм AdaBoost

Исторически первый вариант бустинга (1995).
Задача классификации на два класса, $Y = \{-1, +1\}$,

$\mathscr{L}(b(x_i), y_i) = e^{-b(x_i)y_i}$ — экспоненциальная функция потерь, убывающая функция отступа $M_i = b(x_i)y_i$

Преимущества:

- для обучения b_t на каждом шаге t решается стандартная задача минимизации взвешенного эмпирического риска
- задача оптимизации α_t решается аналитически

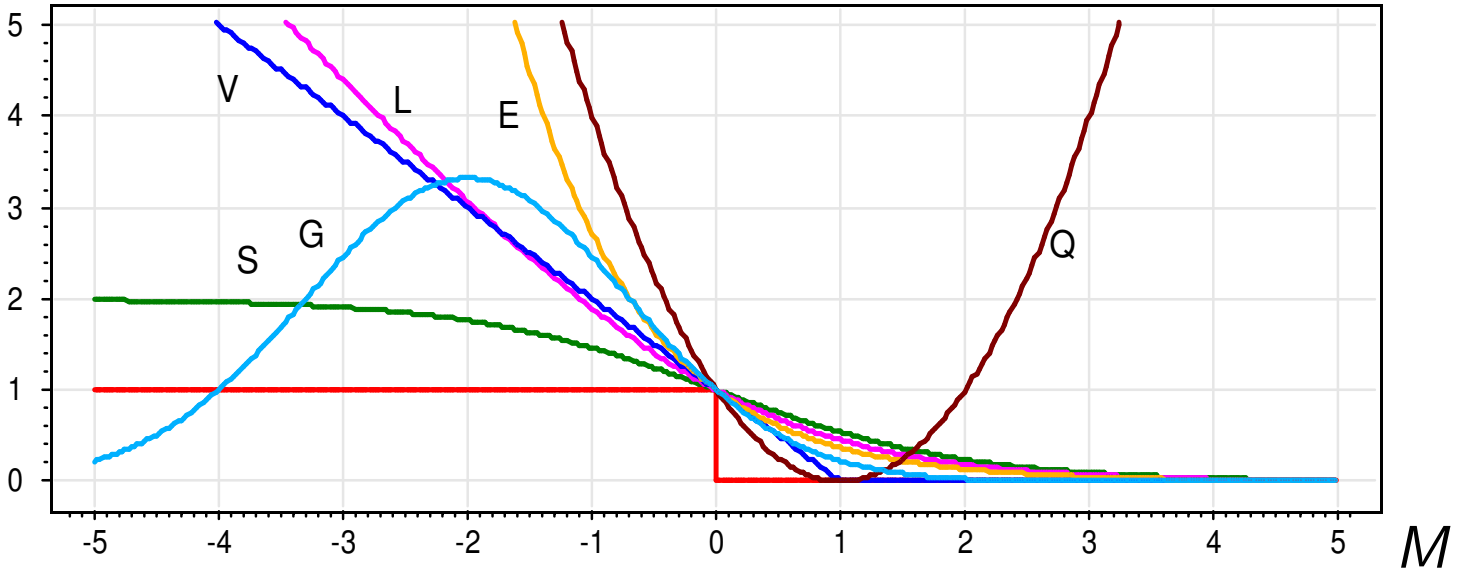
Недостаток:

- AdaBoost слишком чувствителен к выбросам из-за экспоненциального роста функции потерь при $M_i < 0$

Freund Y., Schapire R. E. A decision-theoretic generalization of on-line learning and an application to boosting. 1995.

Частные случаи при различных функциях потерь $\mathscr{L}$

Функции потерь $\mathscr{L}(M)$ в задачах классификации на два класса



$E(M) = e^{-M}$ — экспоненциальная (AdaBoost);

$L(M) = \log_2(1 + e^{-M})$ — логарифмическая (LogitBoost);

$G(M) = \exp\big(-cM(M + s)\big)$ — гауссовская (BrownBoost);

$Q(M) = (1 - M)^2$ — квадратичная;

$S(M) = 2(1 + e^{M})^{-1}$ — сигмоидная;

$V(M) = (1 - M)_+$ — кусочно-линейная (SVM);

Градиентный бустинг над деревьями

Решающее дерево — это кусочно-постоянная функция:

$$b(x) = \sum_{t=1}^{T} \alpha_t [x \in \Omega_t],$$

где T — число листьев,
Ω_t — область t-го листа,
α_t — прогноз в t-м листе.

Идея: каждый лист — базовый алгоритм $b_t(x) = [x \in \Omega_t]$; градиентным шагом определяется прогноз α_t в t-м листе:

$$\alpha_t = \arg\min_{\alpha > 0} \underbrace{\sum_{x_i \in \Omega_t} \mathscr{L}\left(u_{t-1,i} + \alpha, y_i\right)}_{\text{суммарная потеря в } t\text{-м листе}}.$$

После определения всех α_t можно добавить в композицию следующее дерево, оптимизировав его структуру по MSE.

Оптимизация прогнозов в листьях:

$$\alpha_t = \arg\min_{\alpha>0} \sum_{x_i \in \Omega_t} \mathscr{L}\left(u_{t-1,i} + \alpha, y_i\right).$$

Для некоторых функций потерь решение находится аналитически:

- средний квадрат ошибок, MSE, $\mathscr{L}(b, y) = (b - y)^2$:

$$\alpha_t = \frac{1}{|\Omega_t|} \sum_{x_i \in \Omega_t} \left(y_i - u_{t-1,i}\right).$$

- средняя абсолютная ошибка, MAE, $\mathscr{L}(b, y) = |b - y|$:

$$\alpha_t = \underset{x_i \in \Omega_t}{\text{median}} \left\{y_i - u_{t-1,i}\right\}.$$

В общем случае аналитического решения нет.

Резюме

- Градиентный бустинг — наиболее общий из всех бустингов:
  — произвольная функция потерь
  — произвольное пространство оценок R
  — подходит для регрессии, классификации, ранжирования
- Важное открытие середины 90-х: обобщающая способность бустинга не ухудшается с ростом сложности T
- Стохастический вариант SGB — лучше и быстрее
- Градиентный бустинг над решающими деревьями часто работает лучше, чем случайный лес
- Технология Yandex.MatrixNet — это градиентный бустинг над «небрежными» решающими деревьями ODT
  (ODT — oblivious decision tree)