Машинное обучение. Boosting

Алексей Колесов

Белорусский государственный университет

27 сентября 2017 г.

Содержание

Probably approximately correct learnability

Класс гипотез H называют **вероятно приблизительно верно изучаемым** (probably approximately correct learnable) если существует такая функция $m_H : (0,1)^2 \to \mathbb{N}$ и алгоритм, такой что

- для любых $\epsilon, \delta \in (0,1)$
- для любого распределения D над X
- для любой функции $f : X \to \{0,1\}$

если выполняется предположение о реализуемости, то если мы выполним алгоритм на выборке из $m \geqslant m_H(\epsilon, \delta)$ независимых одинаково распределённых элементов из D и размеченных f, то алгоритм вернёт гипотезу $h \in H$ такую, что с вероятностью как минимум $1 - \delta$, выполняется $L_{D,f}(h) \leqslant \epsilon$

No free lunch theorem

No free lunch theorem

Пусть A — любой алгоритм машинного обучения для задачи бинарной классификации и 0-1 функции потерь над пространством X. Пусть m — число, меньшее, чем $|X|/2$. Тогда при размере выборки m будет существовать распределение D, такое что:

- найдётся функция $f : X \to \{0, 1\}$, такая что $L_D(f) = 0$
- с вероятностью не меньшей $\frac{1}{7}$ выполняется, что $L_D(A(S)) \geqslant \frac{1}{8}$

Bias-Complexity tradeoff

$$L_D(h_S) = \epsilon_{\text{app}} + \epsilon_{\text{est}} + \epsilon_{\text{bayes}}$$

- ϵ_{bayes} — ошибка оптимального байесовского классификатора
- $\epsilon_{\text{app}} = \min\limits_{h \in H} L_D(h) - \epsilon_{\text{bayes}}$ — ошибка аппроксимации (насколько H подходит задаче)
- $\epsilon_{\text{est}} = L_D(h_S) - \min\limits_{h \in H} L_D(h)$ — упущенное качество на данном H (насколько хорошо решили задачу при данном H)

Фундаментальная теорема PAC-изучаемости

Фундаментальная теорема PAC-изучаемости

Пусть H — семейство гипотез из X в $\{0,1\}$ и мы используем $0-1$ функцию потерь. Тогда следующие утверждения эквивалентны:

1. H обладает свойством равномерной сходимости
2. H агностически PAC-изучаемый с ERM-алгоритмом
3. H агностически PAC-изучаемый
4. H PAC-изучаемый
5. H PAC-изучаемый с ERM-алгоритмом
6. $\mathrm{VCdim}(H) < \infty$

Мотивация бустинга

Проблемы:

- решение bias-complexity tradeoff
- ERM-алгоритмы часто сложны или вычислительно затратны

Содержание

Probably approximately correct learnability

Класс гипотез H называют **вероятно приблизительно верно изучаемым** (probably approximately correct learnable) если существует такая функция $m_H : (0,1)^2 \to \mathbb{N}$ и алгоритм, такой что

- для любых $\epsilon, \delta \in (0,1)$
- для любого распределения D над X
- для любой функции $f : X \to \{0,1\}$

если выполняется предположение о реализуемости, то если мы выполним алгоритм на выборке из $m \geqslant m_H(\epsilon, \delta)$ независимых одинаково распределённых элементов из D и размеченных f, то алгоритм вернёт гипотезу $h \in H$ такую, что с вероятностью как минимум $1-\delta$, выполняется $L_{D,f}(h) \leqslant \epsilon$

Weak learner

Алгоритм A называется γ-**weak leaner** для класса H, если существует функция $m_H : (0,1) \to \mathbb{N}$, такая что для любых

- $\delta \in (0,1)$
- распределения D над X
- для любой $f : X \to \{-1;1\}$

и выполненном условии реализуемости (относительно H, D, f) для гипотезы $h \in H$, полученной с помощью A над выборкой из $m > m_H(\delta)$ элементов из D и размеченных f с вероятностью как минимум $1-\delta$ выполняется, что $L_D(h) \leqslant 1/2 - \gamma$
Класс H тогда называется γ-**weak learnable** (γ-слабо изучаемым)

О weak-leaners

- из фундаментальной теоремы знаем, что $m_H(\epsilon, \delta) \geqslant C_1 \dfrac{\mathrm{VCdim}(H) + \log(1/\delta)}{\epsilon}$, а значит, если $\mathrm{VCdim}(H) = \infty$, то H — не γ-слабо изучаемый
- статистически PAC-learnability = weak-learnability

План получения преимущества

- возьмём «богатый» класс H (с потенциально сложным ERM-алгоритмом)
- попытаемся найти «бедный» класс B, такой что:
  - ERM_B — эффективно реализуем
  - для любой выборки S, размеченной гипотезой из H, ERM_B имеет ошибку не более $1/2 - \gamma$

Вопрос: можно ли имея слабый алгоритм ERM_B построить сильный для H?

Пример weak-leaner

Пусть $X = \mathbb{R}$ и $H = \{h_{\theta_0,\theta_1,b} : \theta_0, \theta_1 \in \mathbb{R}, b \in \{-1; +1\}\}$

$$\begin{cases} b & \text{если } x < \theta_0 \vee x > \theta_1 \\ -b & \text{иначе} \end{cases}$$

Возьмём $B = \{x \mapsto \operatorname{sign}(x - \theta) \cdot b : \theta \in R, b \in \{-1; +1\}\}$

Доказательство

Докажем, что ERM_B — γ-weak learner для H с $\gamma = 1/12$

- $b \in B$ может точно разметить как минимум два «куска» для любой $h \in H$
- как минимум один «кусок» p имеет $D(p) \leqslant 1/3 \Rightarrow \min_{b \in B} L_D(b) \leqslant 1/3$
- $\mathrm{VCdim}(B) = 2 \Rightarrow$ для $m > C \log(1/\delta)/\epsilon^2$ выполняется $\mathbb{P}[L_D(\mathrm{ERM}_B) > 1/3 + \epsilon] < \delta$
- при $\epsilon = 1/12$, получаем, что ошибка не больше, чем $1/3 + 1/12 = 1/2 - 1/12$

Быстрый ERM-алгоритм для decision stump

Пусть $X = \mathbb{R}^d$,
$H_{DS} = \{x \mapsto \mathrm{sign}(\theta - x_i) \cdot b : \theta \in \mathbb{R}; i \in [d], b \in \{-1, +1\}\}$
Построим $ERM_{H_{DS}}$ относительно W:

$$\min_{h \in H_{DS}} \sum_{i=1}^{m} W_i 1_{[h(x_i) \neq y_i]} =$$

$$\min_{j \in [d]} \min_{\theta \in \mathbb{R}} \left(\sum_{i:y_i=1} W_i 1_{[x_{i,j} > \theta]} + \sum_{i:y_i=-1} W_i 1_{[x_{i,j} \leqslant \theta]} \right)$$

- наивный алгоритм за $\mathcal{O}(dm^2)$
- если отсортировать по каждой из координат $\mathcal{O}(dm \log m)$ и перебирать θ в порядке возрастания, то легко за $\mathcal{O}(dm)$

Содержание

Идея boosting-a

- week-learner работает для **любого** D
- меняя D, получаем разные слабые гипотезы
- комбинируя несколько слабых гипотез, получаем сильную гипотезу

Краткое содержание

- работаем над классом $Y = \{-1; +1\}$
- итеративный алгоритм (T раундов):
  - поддерживаем распределение важности объектов D_t
  - строим **линейную** комбинацию слабых классификаторов $f_t = \sum_{i=1}^{t} \alpha_t h_t$, $\alpha_t > 0$

Алгоритм

Алгоритм 1 AdaBoost

Вход: $S = ((x_1, y_1), \ldots, (x_m, y_m))$; $y_i \in \{-1; +1\}$

for $i = 1, \ldots, m$ **do**
    $D_1 = \frac{1}{m}$
end for
for $t = 1, \ldots, T$ **do**
    $h_t =$ базовая гипотеза с ошибкой $\epsilon_t = \mathbb{P}_{D_t}[h_t(x_i) \neq y_i]$
    $\alpha_t = \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$
    $Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$
    for $i = 1, \ldots, m$ **do**
        $D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$
    end for
end for
return $A = \mathrm{sign}(\sum_{i=1}^{t} \alpha_t h_t)$

Замечания

- Распределение D_t над объектами выборки:
  - изначально равномерное
  - в каждом раунде вес неправильно классифицируемых объектов увеличивается
  - $D_{t+1}(i) = \frac{e^{-y_i f_t(x_i)}}{m\prod_{s=1}^{t} Z_s}$, так как

$$D_{t+1}(i) = \frac{D_t(i)e^{-\alpha_t y_i h_t(x_i)}}{Z_t} =$$

$$\frac{D_{t-1}(i)e^{-\alpha_t y_i h_t(x_i)}e^{-\alpha_{t-1} y_i h_{t-1}(x_i)}}{Z_t Z_{t-1}} = \frac{1}{m}\frac{e^{-y_i \sum_{s=1}^{t}\alpha_s h_s(x_i)}}{\prod_{s=1}^{t} Z_s}$$

- вес α_t зависит от точности классификатора на раунде t

Иллюстрация 1

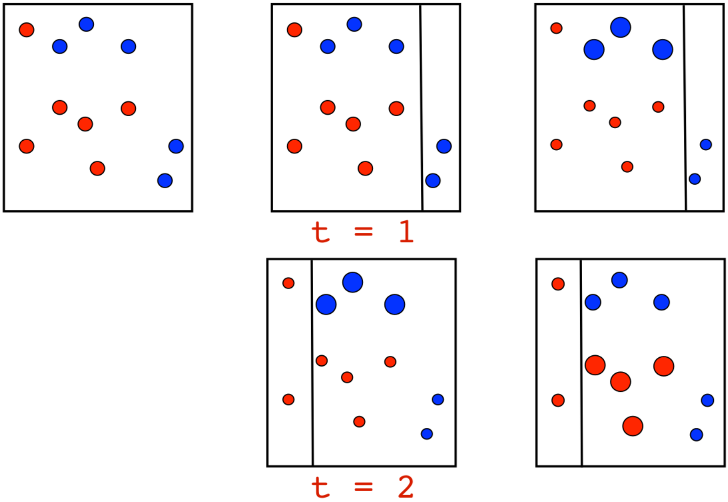


Иллюстрация 2

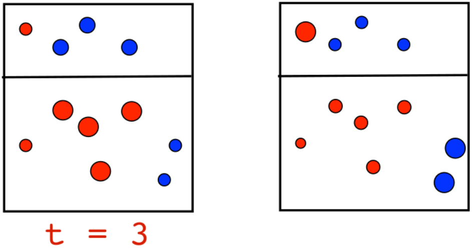
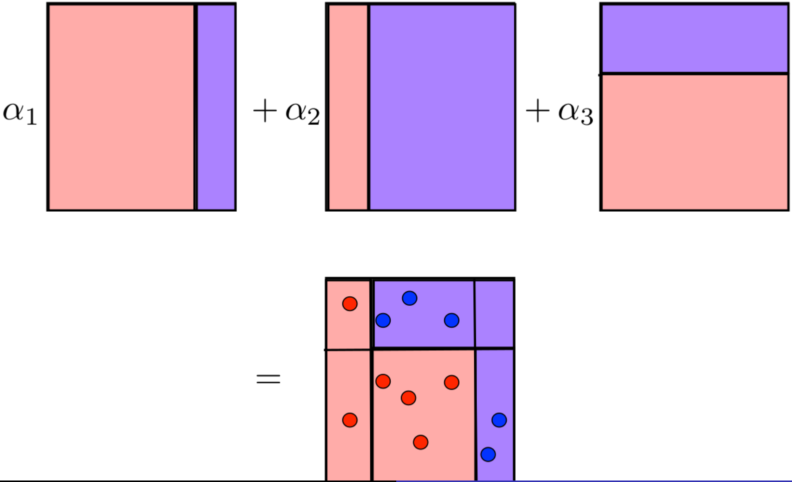


Иллюстрация 3



Минимизация эмпирического риска

Теорема о эмпирическом риске для AdaBoost

Эмпирический риск для классификатора, полученного AdaBoost удовлетворяет

$$L_S(h) \leqslant exp\left[-2\sum_{t=1}^{T}(\tfrac{1}{2}-\epsilon_t)^2\right]$$

Если $\gamma \leqslant (\frac{1}{2}-\epsilon_t)$, то

$$L_S(h) \leqslant exp(-2\gamma^2 T)$$

Замечание: γ необязательно знать заранее!

Доказательство

Хотим доказать:

$$L_S(h) \leqslant exp\left[-2\sum_{t=1}^{T}(\tfrac{1}{2}-\epsilon_t)^2\right]$$

Имеем:

$$D_{t+1}(i) = \frac{e^{-y_i f_t(x_i)}}{m\prod_{s=1}^{t} Z_s}$$

Распишем:

$$L_S(h) = \frac{1}{m}\sum_{i=1}^{m} 1_{y_i f(x_i)<0} \leqslant \frac{1}{m}\sum_{i=1}^{m} exp(-y_i f(x_i))$$
$$\leqslant \frac{1}{m}\sum_{i=1}^{m}\left[m\prod_{t=1}^{T} Z_t\right] D_{T+1}(i) = \prod_{t=1}^{T} Z_t$$

Доказательство

Так как Z_t — это коэффициент нормализации:

$$\begin{aligned}
Z_t &= \sum_{i=1}^{m} D_t(i) e^{-\alpha_t y_i h_t(x_i)} && (1) \\
&= \sum_{i: y_i h_t(x_i) \geqslant 0} D_t(i) e^{-\alpha_t} + \sum_{i: y_i h_t(x_i) < 0} D_t(i) e^{\alpha_t} && (2) \\
&= (1 - \epsilon_t) e^{-\alpha_t} + \epsilon_t e^{\alpha_t} && (3) \\
&= (1 - \epsilon_t) \sqrt{\frac{\epsilon_t}{1 - \epsilon_t}} + \epsilon_t \sqrt{\frac{1 - \epsilon_t}{\epsilon_t}} && (4) \\
&= 2\sqrt{\epsilon_t (1 - \epsilon_t)} && (5)
\end{aligned}$$

Доказательство

Хотим доказать:

$$L_S(h) \leqslant exp\left[-2\sum_{t=1}^{T}(\tfrac{1}{2}-\epsilon_t)^2\right]$$

Имеем:

$$L_S(h) \leqslant \prod_{s=1}^{T} Z_t$$

Распишем:

$$\prod_{t=1}^{T} Z_t = \prod_{t=1}^{T} 2\sqrt{\epsilon_t\left(1-\epsilon_t\right)} = \prod_{t=1}^{T}\sqrt{1-4\left(\frac{1}{2}-\epsilon_t\right)^2} \tag{6}$$

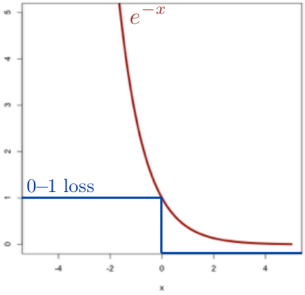
$$\leqslant \prod_{y=1}^{T}\exp\left[-2\left(\frac{1}{2}-\epsilon_t\right)^2\right] = \exp\left[-2\sum_{t=1}^{T}\left(\frac{1}{2}-\epsilon_t\right)^2\right] \tag{7}$$

Замечания по доказательству

- α_t минимизирует $(1-\epsilon_t)e^{-\alpha_t} + \epsilon_t e^{\alpha_t}$
- из такой же логики можно выбрать α_t, когда базовые классификаторы возвращают не только $\{-1; 1\}$
- $(1-\epsilon_t)e^{-\alpha_t} = \epsilon_t e^{\alpha_t}$, масса «правильных» и «неправильных» объектов равна

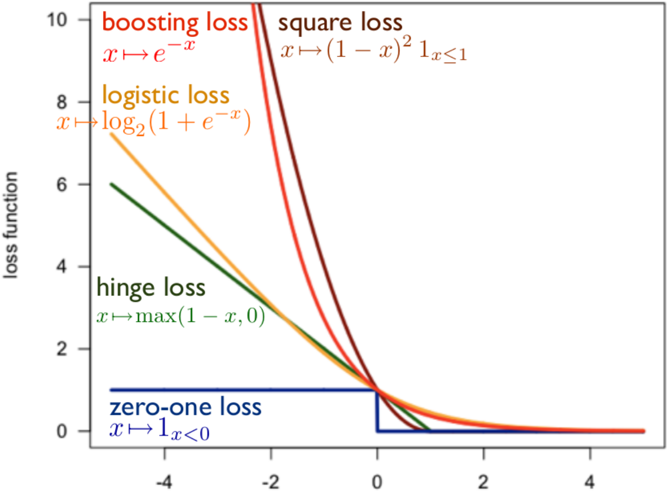
AdaBoost = метод координатного спуска

Если выбрать $F(\alpha) = \sum_{i=1}^{m} e^{-y_i f(x_i)} = \sum_{i=1}^{m} e^{-y_i \sum_{t=1}^{T} \alpha_t h_t(x_i)}$, то AdaBoost совпадает с методом координатного спуска!



AnyBoost — общее название семейства алгоритмов с другими функциями потерь.

Другие функции потерь



VCdim(H) для AdaBoost

- AdaBoost строит линейную комбинацию базовых гипотез (например, для decision stumps получается линейная комбинация полуплоскостей)
- рассмотрим «богатство» такого класса

$$L(B, T) = \left\{ x \mapsto \operatorname{sign}\left(\sum_{t=1}^{T} w_t h_t(x) \right) : w \in \mathbb{R}^T, h_t \in B \right\}$$

T управляет bias-complexity tradeoff.

Пример

Рассмотрим базовый класс:

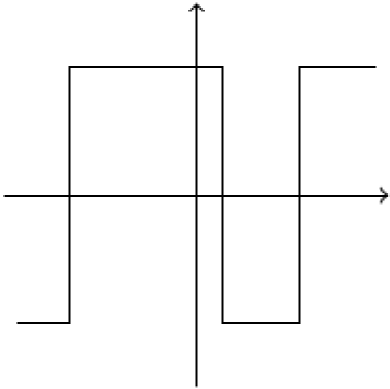
$$H_{DS1} = \{x \mapsto \operatorname{sign}(x - \theta) \cdot b : \theta \in \mathbb{R}, b \in \{-1; 1\}\}$$

Легко видеть, что $\mathrm{VCdim}(H_{DS1}) = 2$
Рассмотрим также класс G_r кусочно-постоянных функций на r «кусках»

$$g_r(x) = \sum_{i=1}^{r} \alpha_i 1_{[x \in (\theta_{i-1}, \theta_i]]},$$
$$\alpha_i \in \{-1; 1\}, -\infty = \theta_0 < \theta_1 < \ldots < \theta_r = \infty$$

Очевидно, что $\mathrm{VCdim}(G_r) \geqslant r$

Пример



Можно показать, что $G_r \subseteq L(H_{DS1}, T)$

Теорема о VCdim($L(B, T)$)

Теорема о VCdim($L(B, T)$)

Пусть B — базовый класс и

$$L(B, T) = \left\{ x \mapsto \operatorname{sign}\left(\sum_{t=1}^{T} w_t h_t(x) \right) : w \in \mathbb{R}^T, h_t \in B \right\}$$

Тогда, если $T \geqslant 3$ и $\mathrm{VCdim}(B) \geqslant 3$, то

$$\mathrm{VCdim}(L(B, T)) \leqslant T(\mathrm{VCdim}(B) + 1)(3 \log(T(\mathrm{VCdim}(B)+1))+2)$$

Доказательство

- пусть $C = (x_1, \ldots, x_m)$ выборка, которая может быть разукрашена $L(B, T)$, $\mathrm{VCdim}(B) = d$
- каждая раскраска C — это выбор $h_1, \ldots, h_T \in B$, а затем применение линейной функции к $(h_1(x), \ldots, h_T(x))$
- каждая $h_i \in B$ раскрашивает C не более, чем $(em/d)^d$ способами (Sauer's lemma) $\Rightarrow$ T гипотез — не более, чем $(em/d)^{Td}$ способами
- линейная функция в T-мерном пространстве порождает не более $(em/T)^T$ раскрасок

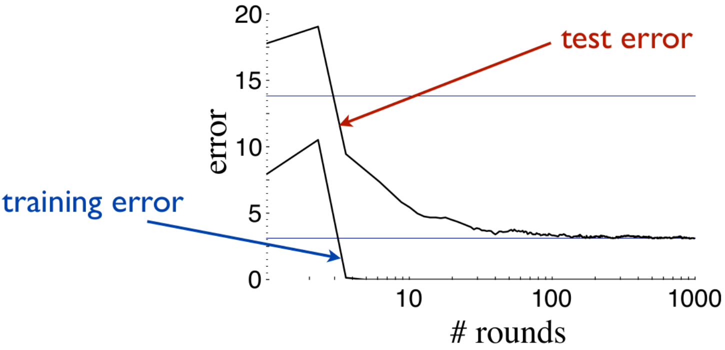
Всего раскрасок $L(B, T)$ может породить не более:

$$(em/d)^{dT}(em/T)^T \leqslant m^{(d+1)T}$$

А значит, $2^m \leqslant m^{(d+1)T}$

Содержание

Минимизация true risk



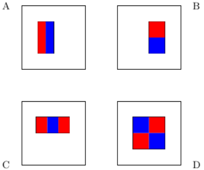
C4.5 decision trees (Schapire et al., 1998).

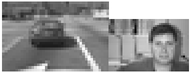
Выбросы

- **выброс** (outlier) — объект, «непохожий» на остальную выборку (например, ошибки измерения)
- AdaBoost по построению плохо работает на выборке с выбросами
- можно использовать другую функцию (см. BrownBoost)
- можно его использовать для определения выбросов

AdaBoost в задаче распознавания лиц

- AdaBoost показывает себя хорошо во многих практических задачах
- в домашнем задании вам предложено решить задачу классификации изображения на два класса: лицо или машина
- предполагаемый базовый классификатор — decision stumps над гипотезами Виола-Джонса





Итоги

- ввели понятие weak-learner
- показали, что можно использовать комбинацию weak-learners, чтоб получить более богатый класс гипотез
- рассмотрели алгоритм AdaBoost

Литература

- Shai Shalev-Shwartz and Shai Ben-David — Understanding Machine Learning: From theory to algorithms (глава 10)
- Mehryar Mohri — Foundations of Machine Learning (Lection Boosting) — `http://www.cs.nyu.edu/~mohri/mls/ml_boosting.pdf`
- `http://www.machinelearning.ru/wiki/index.php?title=AnyBoost`