2 PAC-обучаемость и сжатие

2.1 PAC-learning

Определение 2. *Класс гипотез $\mathcal{H}$ является PAC-обучаемым над множеством объектов $Z = z_1, \cdots z_d$ для функции потерь $l : \mathcal{H} \times Z \to \mathbb{R}_+$, если существует функция $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ и алгоритм $\mathcal{A}$ со свойством: для всех $\varepsilon, \delta \in (0,1)$, для любого распределения $\mathcal{D}$ над множеством объектов Z алгоритм $\mathcal{A}$ возвращает такое $h \in \mathcal{H}$, что с вероятностью $1-\delta$ выполняется:*

$$\mathsf{E}_{z\sim\mathcal{D}}\left[l\big(h,z\big)\right] \leq \min_{h'\in\mathcal{H}} \mathsf{E}_{z\sim\mathcal{D}}\left[l\big(h',z\big)\right] + \varepsilon. \tag{2.1}$$

Пример для задачи классификации выражение (2.1) можно переписать в виде:

$$P\big[p\big(\left\{h\big(z\big) \neq f\big(z\big)\right\}\big) > \varepsilon\big] \leq \delta$$

2.2 Sample Compression scheme

Схема сжатия данных с параметром k состоит из двух отображений (κ, ρ) :

1. κ получает на вход выборку S, а на выходе получаем пару (S', I) , где $|S'| = k$;
2. ρ получает на вход пару (S', I) на выходе выдает гипотезу h.

Причем выполняется следующее условие:

1. $\kappa\big(Y, y\big) = \left((Z, z), I\right)$;
2. $\rho\big(\kappa\big(Y, y\big)\big)|_Y = y$.

2.3 Compression implais learning

Любую схему сжатия с параметром k можно рассматривать как алгоритм обучения $A = \rho \circ \kappa$. То что данный алгоритм PAC-обучаем доказывает следующая теорема.

Теорема 2. *Алгоритм обучения $A = \rho \circ \kappa$ является PAC-обучаемым, то есть*

$$P\big[p\big(\left\{h\big(z\big) \neq f\big(z\big)\right\}\big) > \varepsilon\big] \leq |I| \sum_{j=1}^{k} \binom{d}{j} (1-\varepsilon)^{m-j},$$

где p распределение над Z.

Доказательство. Сначала заметим, что всего существует

$$\sum_{j=1}^{k} \binom{d}{j}$$

подмножеств T множества Z размера не более k. С другой стороны всего есть $|I|$ вариантов выбрать информацию сжатия $i \in I$. Из выше описанного получаем, что каждой паре (T, i) соответствует своя функция

$$h_{T,i} = \rho\big(\left(T, i\right), i\big).$$

С построения $h_{T,i}$ следует, что $h_{T,i}$ не зависит от $Z \setminus T$, тогда получаем, что если

$$p\big(\left\{h_{T,i}\big(x\big) \neq f(x)\right\}\big) \geq \varepsilon,$$

то для всех $m - |T|$ выполняется получаем, что

$$\prod_{t=1}^{m-|T|} p\big(\left\{h_{T,i}\big(x\big) \neq f(x)\right\}\big) \leq (1-\varepsilon)^{m-|T|}. \tag{2.2}$$

Получаем, что для любого $h_{T,i}$ выполняется неравенство (2.2).

И того получаем, что для произвольной $h_{T,i}$ выполняется неравенство:

$$P\big[p\big(\big\{h_{T,i}(z) \neq f(z)\big\}\big) > \varepsilon\big] \leq (1-\varepsilon)^{m-|T|},$$

Рассмотрим множество функций при фиксированном $i \in I$:

$$\mathcal{H}_i = \{h_{T,i} : |T| \leq k\}, \tag{2.3}$$

тогда для алгоритма A для подмножества функций, которые получены при помощи сжатой информации i получаем:

$$P\big[p\big(\big\{h_{T,i}(z) \neq f(z)\big\}\big) > \varepsilon\big] \leq \sum_{j=1}^{k} \binom{d}{j} (1-\varepsilon)^{m-j}, \tag{2.4}$$

где $h_{T,i}$ это лучший алгоритм из множества $\mathcal{H}_i$.

Теперь заметим, что финальная функция h принадлежит множеству:

$$\mathcal{H}_{\kappa,\rho} = \{h_{T,i} : |T| \leq k, i \in I\}. \tag{2.5}$$

Вспомним, что для каждого T таких функций $|I|$, из чего уже для произвольного h используя выражение (2.4) имеем следующее неравенство:

$$P\big[p\big(\big\{h(z) \neq f(z)\big\}\big) > \varepsilon\big] \leq |I| \sum_{j=1}^{k} \binom{d}{j} (1-\varepsilon)^{m-j},$$

что и доказывает исходную теорему. □

Список литературы

[1] *Floyd, S., Warmuth, M.* (1995) Sample Compression, Learnability, and the Vapnik-Chervonenkis Dimension. // Machine Learning 21, 269–304. https://doi.org/10.1023/A:1022660318680

[2] *В.В.Вьюгин* КОЛМОГОРОВСКАЯ СЛОЖНОСТЬ И АЛГОРИТМИЧЕСКАЯ СЛУЧАЙНОСТЬ (2012) // МФТИ.

[3] *Shay Moran, Amir Yehudayoff* Sample Compression Schemes for VC Classes (2015) // https://www.cs.bgu.ac.il/~adsmb182/wiki.files/meni-lecture.pdf