

Теория и практика машинного обучения

• Лекция 3 •

Комбинаторная теория переобучения

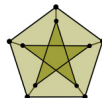
Воронцов Константин Вячеславович

МФТИ • МГУ • ВШЭ • ВЦ РАН • Яндекс • FORECSYS



Комбинаторика и алгоритмы
для школьников

• Летняя школа — 2014 •
24 августа 2014

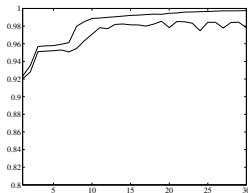


Содержание

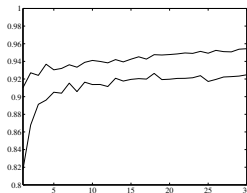
- 1 Задача оценивания переобучения**
 - Явление переобучения
 - Понятие вероятности переобучения
 - Зачем нужны оценки переобучения
- 2 Эксперименты с переобучением**
 - Эксперимент с монотонной цепью
 - Эксперимент с пороговыми классификаторами
 - Эксперимент с парой классификаторов
- 3 Комбинаторная теория переобучения**
 - Гипергеометрическое распределение
 - Простые модельные семейства
 - Оценка расслоения-связности

Пример переобучения: задача диагностики по ЭКГ

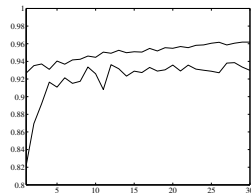
Зависимости AUC от числа используемых признаков K



некроз ГБК



хронический гастрит



зоб щитовидной железы

Тонкая (верхняя) линия — на обучающей выборке

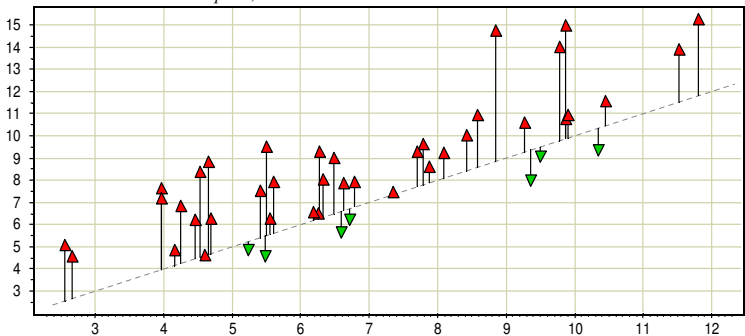
Толстая (нижняя) линия — на тестовой выборке

AUC на контроле систематически хуже, чем на обучении

Пример переобучения: ещё одна медицинская задача

Задача предсказания отдалённого результата хирургического лечения атеросклероза. Точки — различные алгоритмы.

Частота ошибок на контроле, %



Частота ошибок на обучении, %

Ошибки на контроле систематически чаще, чем на обучении

Пример переобучения: полиномиальная регрессия

Зависимость $y(x) = \frac{1}{1 + 25x^2}$ на отрезке $x \in [-2, 2]$.

Признаковое описание $x \mapsto (1, x^1, x^2, \dots, x^n)$.

Модель полиномиальной регрессии

$a(x, w) = w_0 + w_1x + \dots + w_nx^n$ — полином степени n .

Обучение методом наименьших квадратов:

$$Q(w, X) = \sum_{i=1}^{\ell} (w_0 + w_1x_i + \dots + w_nx_i^n - y_i)^2 \rightarrow \min_{w_0, \dots, w_n} .$$

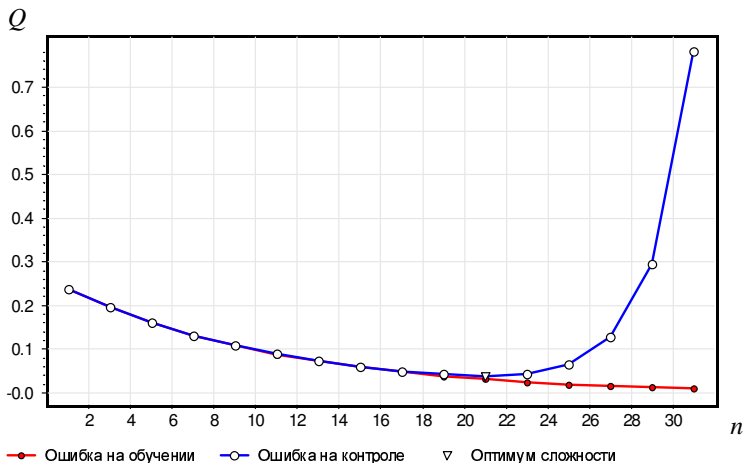
Обучающая выборка: $X = \{x_i = 4\frac{i-1}{\ell-1} - 2 \mid i = 1, \dots, \ell\}$.

Контрольная выборка: $\bar{X} = \{x_i = 4\frac{i-0.5}{\ell-1} - 2 \mid i = 1, \dots, \ell - 1\}$.

Что происходит с $Q(w^*, X)$ и $Q(w^*, \bar{X})$ с ростом степени n ?

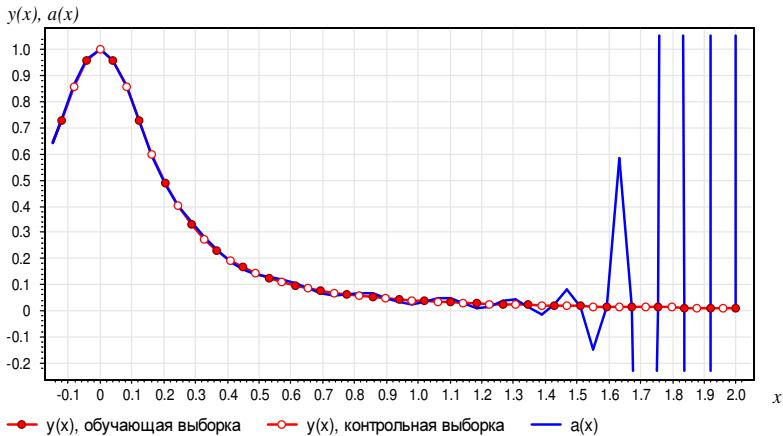
Пример переобучения: эксперимент при $\ell = 50$, $n = 1..31$

Переобучение — это когда $Q(w^*, \bar{X}) \gg Q(w^*, X)$:



Пример переобучения: эксперимент при $\ell = 50$

$$y(x) = \frac{1}{1 + 25x^2}; \quad a(x) \text{ — полином степени } n = 38$$



Наша цель — лучше понять явление переобучения

- Почему возникает переобучение?
- Всегда ли оно возникает?
- Как оценить величину возможного переобучения?
- Как её уменьшить?

Бинарная функция потерь. Матрица ошибок

$X^L = \{x_1, \dots, x_L\}$ — конечное генеральное множество объектов

$A = \{a_1, \dots, a_D\}$ — конечное семейство алгоритмов

$I(a, x) = [\text{алгоритм } a \text{ ошибается на объекте } x]$

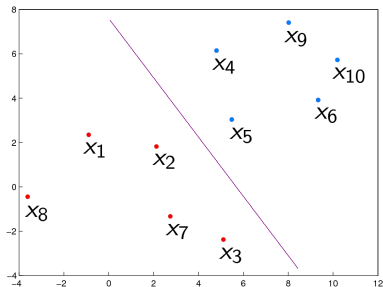
$L \times D$ -матрица ошибок с попарно различными столбцами:

	a_1	a_2	a_3	a_4	a_5	a_6	\dots	a_D	
x_1	1	1	0	0	0	1	\dots	1	X — наблюдаемая (обучающая) выборка длины l
\dots	0	0	0	0	1	1	\dots	1	
x_ℓ	0	0	1	0	0	0	\dots	0	
$x_{\ell+1}$	0	0	0	1	1	1	\dots	0	\bar{X} — скрытая (контрольная) выборка длины $k = L - l$
\dots	0	0	0	1	0	0	\dots	1	
x_L	0	1	1	1	1	1	\dots	0	

$n(a, U) = \sum_{x \in U} I(a, x)$ — число ошибок $a \in A$ на выборке $U \subset X^L$

$\nu(a, U) = \frac{1}{|U|} n(a, U)$ — частота ошибок a на выборке U

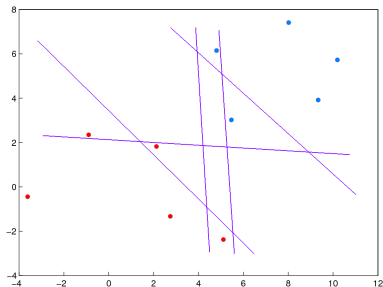
Пример. Матрица ошибок линейных классификаторов



1 вектор с 0 ошибками

x_1	0
x_2	0
x_3	0
x_4	0
x_5	0
x_6	0
x_7	0
x_8	0
x_9	0
x_{10}	0

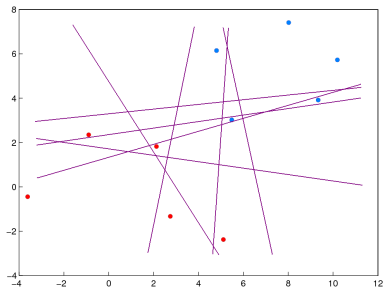
Пример. Матрица ошибок линейных классификаторов



1 вектор с 0 ошибками
5 векторов с 1 ошибкой

x_1	0	1	0	0	0	0
x_2	0	0	1	0	0	0
x_3	0	0	0	1	0	0
x_4	0	0	0	0	1	0
x_5	0	0	0	0	0	1
x_6	0	0	0	0	0	0
x_7	0	0	0	0	0	0
x_8	0	0	0	0	0	0
x_9	0	0	0	0	0	0
x_{10}	0	0	0	0	0	0

Пример. Матрица ошибок линейных классификаторов



1 вектор с 0 ошибками
 5 векторов с 1 ошибкой
 8 векторов с 2 ошибками
 и т. д...

x_1	0	1	0	0	0	0	1	0	0	0	0	1	1	0	...
x_2	0	0	1	0	0	0	1	1	0	0	0	0	0	0	...
x_3	0	0	0	1	0	0	0	1	1	0	0	0	0	1	...
x_4	0	0	0	0	1	0	0	0	1	1	0	0	0	0	...
x_5	0	0	0	0	0	1	0	0	0	1	1	1	0	0	...
x_6	0	0	0	0	0	0	0	0	0	0	1	0	1	0	...
x_7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	...
x_8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
x_9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
x_{10}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...

Задача оценивания вероятности переобучения

Метод обучения μ по выборке X строит алгоритм $a^* = \mu(X) \in A$
Пример — метод минимизации эмпирического риска (МЭР):

$$a^* = \mu(X) = \arg \min_{a \in A} \nu(a, X).$$

Переобученность $\delta(\mu, X) = \nu(a^*, \bar{X}) - \nu(a^*, X)$
Переобучение — это событие $\delta(\mu, X) \geq \varepsilon$

Основное вероятностное предположение:
все разбиения $X \sqcup \bar{X} = X^L$ равновероятны

Основная задача — оценить **вероятность** переобучения:

$$Q_\varepsilon(\mu, X^L) = \frac{1}{C_L^L} \sum_{X \subset X^L} [\delta(\mu, X) \geq \varepsilon] = \mathbf{P}[\delta(\mu, X) \geq \varepsilon]$$

Способ применения оценок переобучения

Допустим, получена оценка сверху:

$$Q_\varepsilon(\mu, X^L) = P[\nu(a^*, \bar{X}) - \nu(a^*, X) \geq \varepsilon] \leq \eta(\varepsilon, \mu).$$

Тогда с вероятностью не менее $(1 - \eta)$

$$\nu(a^*, \bar{X}) \leq \nu(a^*, X) + \varepsilon(\eta, \mu),$$

где $\varepsilon(\eta, \mu)$ — функция, обратная к $\eta(\varepsilon, \mu)$.

Получаем критерий для выбора метода обучения μ :

$$\underbrace{\nu(\mu(X), X)}_{\text{эмпирический риск}} + \underbrace{\varepsilon(\eta, \mu)}_{\text{регуляризатор}} \rightarrow \max_{\mu}$$

(в частности, для отбора признаков или выбора модели).

Наводящие соображения

Обучение $a^* = \mu(X)$ — это выбор по неполной информации.
Чем больше вариантов выбора, тем сильнее можно ошибиться.
Переобучение должно увеличиваться с ростом числа $|A|$.

Но что значит «число различных алгоритмов»?

- Если два алгоритма различаются только на одном объекте, то это почти один и тот же алгоритм.
- Если из двух алгоритмов один очень хороший, а второй явно очень плохой, то даже неполной информации хватит, чтобы почти никогда не выбирать второй.
Значит, и в этом случае мы имеем, скорее, один алгоритм.

Так сколько же у нас реальных вариантов выбора?

Эксперименты с модельными семействами алгоритмов

Физика — экспериментальная, естественная наука, часть естествознания. Математика — это та часть физики, в которой эксперименты дешёвы [академик В.И.Арнольд]

Хотим экспериментально обнаружить, какие свойства матрицы ошибок влияют на вероятность переобучения.

- 1 Будем изучать *модельные семейства алгоритмов*, задавая их непосредственно своими матрицами ошибок.
- 2 Будем оценивать **вероятность** методом Монте–Карло — как долю разбиений выборки из случайного подмножества N разбиений, $|N|$ порядка 10^3 – 10^4 :

$$\hat{Q}_\varepsilon(\mu, X^L) = \frac{1}{|N|} \sum_{X \in N} \left[\nu(\mu(X), \bar{X}) - \nu(\mu(X), X) \geq \varepsilon \right].$$

Монотонная цепь классификаторов

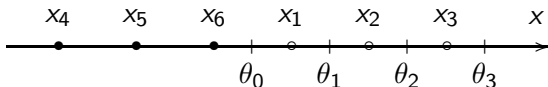
Одномерное пороговое решающее правило:

$$a_d(x) = [x \geq \theta_d], \quad d = 0, \dots, D$$

Пример:

2 класса {●, ○}

6 объектов


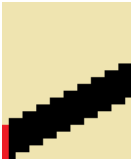




Матрица потерь:

	a_0	a_1	a_2	a_3
x_1	0	1	1	1
x_2	0	0	1	1
x_3	0	0	0	1
x_4	0	0	0	0
x_5	0	0	0	0
x_6	0	0	0	0

Эксперимент с монотонной цепью и ещё тремя семействами

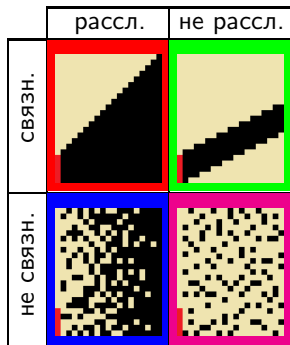
Четыре матрицы ошибок с одинаковым лучшим алгоритмом:

	с расслоением по числу ошибок	без расслоения по числу ошибок
<p>со связностью: соседние алгоритмы отличаются в 1 объекте (образуется <i>цепь</i>)</p>		
<p>без связности: соседние алгоритмы существенно различны (<i>цепь</i> не образуется)</p>		

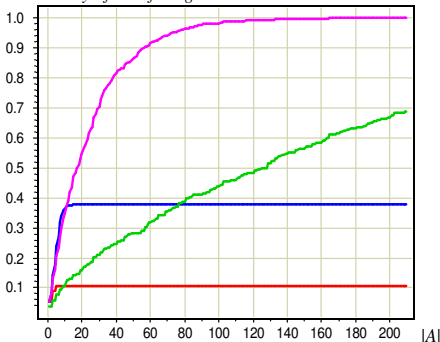
Постепенно добавляя алгоритмы в $\{a_1, \dots, a_D\}$, построим зависимости вероятности переобучения Q_ϵ от числа D .

Эксперимент с монотонной цепью и ещё тремя семействами

$\ell = k = 100$, $m = 10$, $\varepsilon = 0.05$, $|N| = 10^4$ разбиений Монте-Карло.



Probability of overfitting

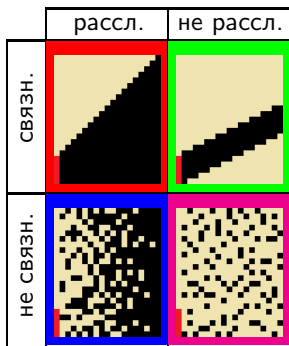


- Огромные семейства с R&S могут почти не переобучаться
- Без R&S уже 30 алгоритмов могут сильно переобучаться

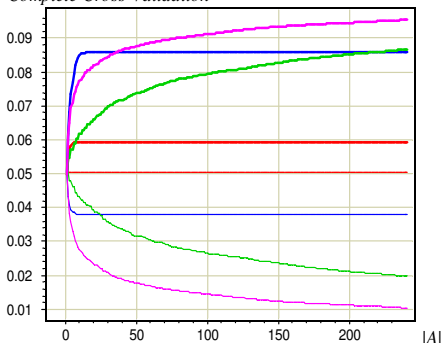
Эксперимент с монотонной цепью и ещё тремя семействами

Оценка кросс-валидации (Complete Cross-Validation, CCV)

$$CCV(\mu, X^L) = \frac{1}{C_L} \sum_{X \subset X^L} \nu(\mu(X), \bar{X})$$

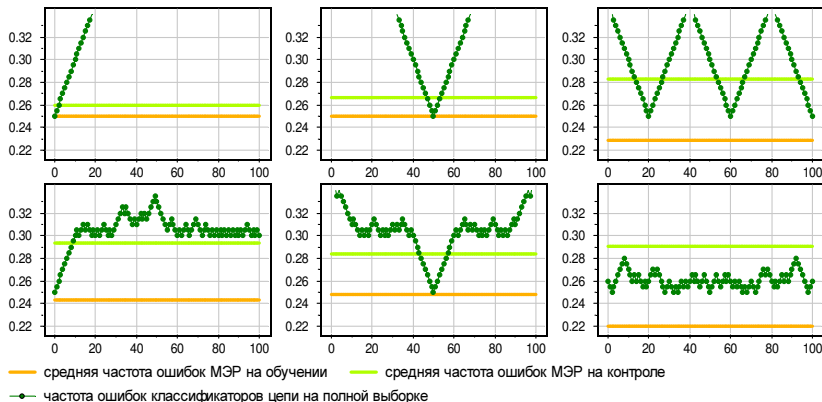


Complete Cross-Validation



Эксперимент. Переобучение цепей с различным расслоением

Условия эксперимента: $\ell = k = 50$, $m = 25$, $\varepsilon = 0.05$,
метод Монте-Карло по $|N| = 10^4$ случайных разбиений.



Семейство из двух алгоритмов $A = \{a_1, a_2\}$

Пусть для алгоритмов a_1, a_2 известны m_0, m_1, m_2, m_3 :

$$a_1 = (1, \dots, 1, 1, \dots, 1, 0, \dots, 0, 0, \dots, 0);$$

$$a_2 = (\underbrace{1, \dots, 1}_{m_0}, \underbrace{0, \dots, 0}_{m_1}, \underbrace{1, \dots, 1}_{m_2}, \underbrace{0, \dots, 0}_{m_3}).$$

Расстояние Хэмминга между векторами ошибок:

$$r(a_1, a_2) = \sum_{i=1}^L |l(a_1, x_i) - l(a_2, x_i)| = m_1 + m_2$$

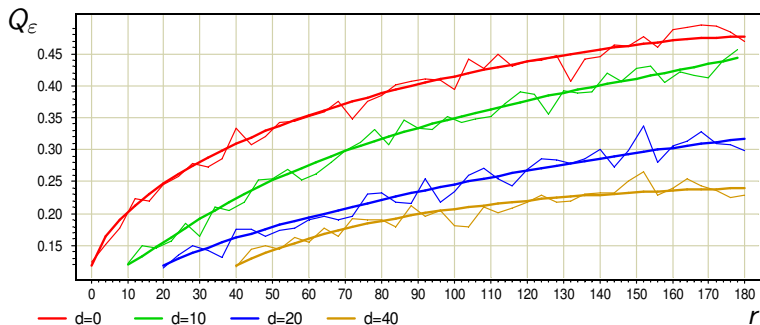
Расслоение измеряется разностью числа ошибок:

$$d(a_1, a_2) = |n(a_1, X^L) - n(a_2, X^L)| = |m_1 - m_2|$$

Условия эксперимента: $\ell = k = 100$, $m_0 = 20$, $\varepsilon = 0.05$,
 метод Монте-Карло по $|N| = 10^4$ случайных разбиений.

Эффекты сходства и расслоения для пары алгоритмов

Зависимость вероятности переобучения Q_ϵ
от расстояния Хэмминга r и расслоения d :



- переобучение возникает даже при выборе из двух алгоритмов
- чем более они схожи, тем меньше переобучение
- чем больше расслоение, тем меньше переобучение

Тривиальный частный случай $A = \{a\}$

Пусть $A = \{a\}$ — одноэлементное множество, $m = n(a, X^L)$.

Тогда вероятность переобучения есть вероятность большого отклонения частот ошибок в двух подвыборках:

$$Q_\varepsilon(a, X^L) = P[\nu(a, \bar{X}) - \nu(a, X) \geq \varepsilon].$$

Теорема

Для любого X^L , любого $\varepsilon \in [0, 1]$

$$Q_\varepsilon(a, X^L) = \mathcal{H}_L^{\ell, m} \left(\frac{\ell}{L}(m - \varepsilon k) \right),$$

где $\mathcal{H}_L^{\ell, m}(z) = \sum_{s=0}^{\lfloor z \rfloor} \frac{C_m^s C_{L-m}^{\ell-s}}{C_L^\ell}$ — функция гипергеометрического распределения.

Доказательство

1. Обозначим $s = n(a, X)$.
2. Школьная задача по теории вероятностей:
 в урне L шаров, m из них чёрные; извлекаем ℓ шаров наугад.
 Какова вероятность того, что s из них чёрные?

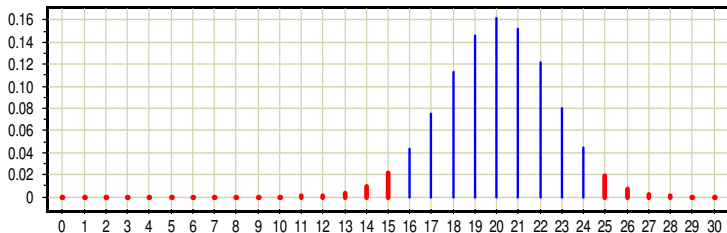
$$P[n(a, X) = s] = C_m^s C_{L-m}^{\ell-s} / C_L^\ell.$$

3. Распишем Q_ε , подставив $\nu(a, \bar{X}) = \frac{m-s}{k}$, $\nu(a, X) = \frac{s}{\ell}$:

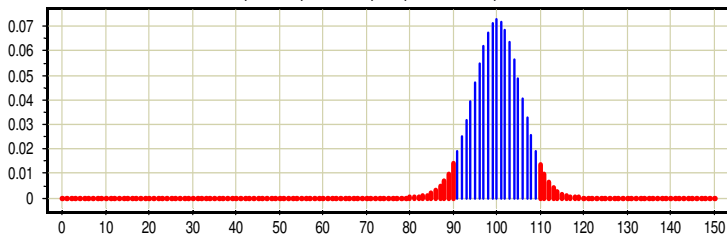
$$\begin{aligned} Q_\varepsilon(a, X^L) &= P[\nu(a, \bar{X}) - \nu(a, X) \geq \varepsilon] = \\ &= \sum_{s=0}^{\ell} \underbrace{\left[\frac{m-s}{k} - \frac{s}{\ell} \geq \varepsilon \right]}_{s \leq \frac{\ell}{L}(m - \varepsilon k)} \underbrace{P[n(a, X) = s]}_{C_m^s C_{L-m}^{\ell-s} / C_L^\ell} = \\ &= \mathcal{H}_L^{\ell, m} \left(\frac{\ell}{L}(m - \varepsilon k) \right). \quad \blacksquare \end{aligned}$$

Гипергеометрическое распределение $h(s|m) = C_m^s C_{L-m}^{\ell-s} / C_L^\ell$

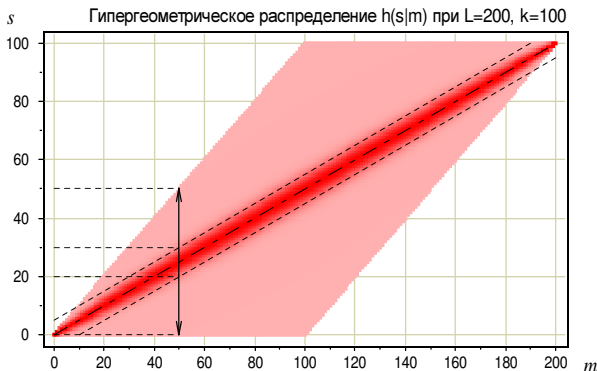
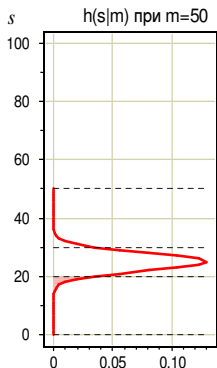
H Гипергеометрическое распределение при $L=300, k=100, m=30, \eta=0.05$



H Гипергеометрическое распределение при $L=1500, k=500, m=150, \eta=0.05$



Гипергеометрическое распределение $h(s|m) = C_m^s C_{L-m}^{\ell-s} / C_L^\ell$



Предсказание числа $m = n(a, X^L)$ по числу $s = n(a, X)$ возможно благодаря узости гипергеометрического пика, причём при $\ell, k \rightarrow \infty$ он сужается, и $\nu(a, X) \rightarrow \nu(a, \bar{X})$ (явление концентрации вероятности, закон больших чисел).

Семейство из двух алгоритмов $A = \{a_1, a_2\}$

Пусть для алгоритмов a_1, a_2 известны m_0, m_1, m_2, m_3 :

$$a_1 = (1, \dots, 1, 1, \dots, 1, 0, \dots, 0, 0, \dots, 0);$$

$$a_2 = (\underbrace{1, \dots, 1}_{m_0}, \underbrace{0, \dots, 0}_{m_1}, \underbrace{1, \dots, 1}_{m_2}, \underbrace{0, \dots, 0}_{m_3}).$$

Теорема

Если метод μ минимизирует эмпирический риск, $A = \{a_1, a_2\}$, то для любого $\varepsilon \in [0, 1]$

$$Q_\varepsilon(\mu, X^L) = \sum_{s_0=0}^{m_0} \sum_{s_1=0}^{m_1} \sum_{s_2=0}^{m_2} \frac{C_{m_0}^{s_0} C_{m_1}^{s_1} C_{m_2}^{s_2} C_{L-m_0-m_1-m_2}^{\ell-s_0-s_1-s_2}}{C_L^\ell} \times$$

$$\times \left([s_1 < s_2] [s_0 + s_1 \leq \frac{\ell}{L}(m_0 + m_1 - \varepsilon k)] + \right.$$

$$\left. + [s_1 \geq s_2] [s_0 + s_2 \leq \frac{\ell}{L}(m_0 + m_2 - \varepsilon k)] \right).$$

Монотонная цепь алгоритмов $A = \{a_0, a_1, \dots, a_D\}$

При каких условиях a_d окажется результатом обучения?

	a_0	a_1	a_2	a_3	...	a_{D-1}	a_D
x_1	0	1	1	1	...	1	1
x_2	0	0	1	1	...	1	1
x_3	0	0	0	1	...	1	1
...
x_D	0	0	0	0	...	0	1
$L-D-m$	0	0	0	0	...	0	0

	0	0	0	0	...	0	0
m	1	1	1	1	...	1	1

	1	1	1	1	...	1	1

$$X_d = \{x_{d+1}\}$$

$$X'_d = \{x_1, \dots, x_d\}$$

$$[\mu(X) = a_d] = \begin{cases} [X_d \subseteq X] [X'_d \subseteq \bar{X}], & d \leq k, d < D; \\ [X'_d \subseteq \bar{X}], & d \leq k, d = D; \\ 0, & d > k. \end{cases}$$

Порождающие и запрещающие множества

Опр. Если для алгоритма a найдутся множества объектов X_a и X'_a такие, что для всех обучающих выборок X , $|X| = \ell$

$$[\mu(X)=a] = [X_a \subseteq X] [X'_a \subseteq \bar{X}],$$

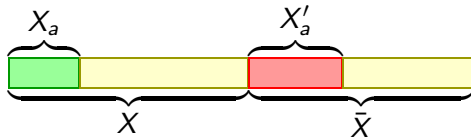
то X_a называется *порождающим*, X'_a — *запрещающим*.

Лемма

Вероятность получить в результате обучения алгоритм a

$$P_a = P[\mu(X)=a] = P[X_a \subseteq X] [X'_a \subseteq \bar{X}] = \frac{C_{L_a}^{\ell_a}}{C_L^\ell},$$

где $L_a = L - |X_a| - |X'_a|$, $\ell_a = \ell - |X_a|$.



Основная теорема

Теорема

Если у каждого алгоритма есть множества X_a и X'_a , то

$$Q_\varepsilon(\mu, X^L) = \sum_{a \in A} P_a \mathcal{H}_{L_a}^{\ell_a, m_a}(s_a(\varepsilon)),$$

где $m_a = n(a, X^L \setminus X_a \setminus X'_a)$, $s_a(\varepsilon) = \frac{\ell_a}{L}(n(a, X^L) - \varepsilon k) - n(a, X_a)$.

Доказательство.

$$\begin{aligned} Q_\varepsilon &= \mathbb{P}[\nu(\mu(X), \bar{X}) - \nu(\mu(X), X) \geq \varepsilon] = \\ &= \mathbb{P} \sum_{a \in A} [\mu(X) = a] \sum_{s=0}^{\ell_a} [n(a, X \setminus X_a) = s] \underbrace{[\nu(a, \bar{X}) - \nu(a, X) \geq \varepsilon]}_{s \leq s_a(\varepsilon)} \\ &= \sum_{a \in A} \sum_{s=0}^{s_a(\varepsilon)} \mathbb{P}[X_a \subseteq X] [X'_a \subseteq \bar{X}] [n(a, X \setminus X_a) = s] \dots \end{aligned}$$

Завершение доказательства

Итак, нам надо оценить $P[X_a \subseteq X][X'_a \subseteq \bar{X}][n(a, X \setminus X_a) = s]$, это доля вот таких разбиений выборки:

$$a = \left(\underbrace{X_a; \overbrace{1, \dots, 1}^s; 0, \dots, 0}_{X \setminus X_a}; \underbrace{X'_a; \overbrace{1, \dots, 1}^{m_a - s}; 0, \dots, 0}_{\bar{X} \setminus X'_a} \right)$$

$\underbrace{\hspace{15em}}_X \qquad \underbrace{\hspace{15em}}_{\bar{X}}$

Число таких разбиений равно $C_{m_a}^s C_{L_a - m_a}^{l_a - s}$, подставляем в Q_ε :

$$\begin{aligned} Q_\varepsilon &= \sum_{a \in A} \sum_{s=0}^{s_a(\varepsilon)} \frac{C_{m_a}^s C_{L_a - m_a}^{l_a - s}}{C_L^l} = \sum_{a \in A} \left(\frac{C_{L_a}^{l_a}}{C_L^l} \right) \left(\sum_{s=0}^{s_a(\varepsilon)} \frac{C_{m_a}^s C_{L_a - m_a}^{l_a - s}}{C_{L_a}^{l_a}} \right) = \\ &= \sum_{a \in A} P_a \mathcal{H}_{L_a}^{l_a, m_a}(s_a(\varepsilon)). \quad \blacksquare \end{aligned}$$

Вернёмся к монотонной цепи

Опр. Метод μ — *пессимистичный МЭР*, если он из множества алгоритмов, лучших на обучении, выбирает худший на контроле:

$$A(X) = \text{Arg min}_{a \in A} n(a, X); \quad \mu(X) = \arg \max_{a \in A(X)} n(a, \bar{X}).$$

Теорема

Пусть $A = \{a_0, \dots, a_D\}$ — монотонная цепь алгоритмов, $m = n(a_0, X^L)$, метод μ — *пессимистичный МЭР*.

Тогда вероятность переобучения равна

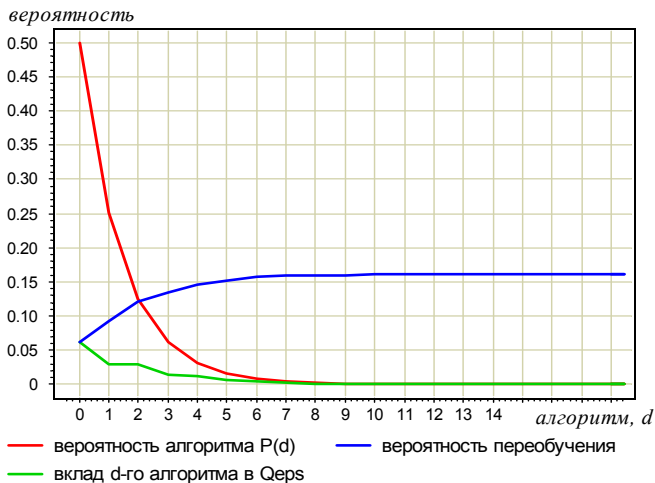
$$Q_\varepsilon = \sum_{d=0}^{\min\{k, D\}} \frac{C_{L-d-\delta}^{\ell-\delta}}{C_L^\ell} \mathcal{H}_{L-d-\delta}^{\ell-\delta, m} \left(\frac{\ell}{L} (m + d - \varepsilon k) \right),$$

где $\delta = [d < D]$.

Доказательство: подставить уже найденные X_a, X'_a в теорему.

Расчёт по выведенной формуле

Условия эксперимента: $\ell = k = 100$, $m = 20$, $\varepsilon = 0.05$



Граф расслоения–связности множества алгоритмов

Определим бинарные отношения на множестве алгоритмов A :
частичный порядок $a \leq b$: $I(a, x) \leq I(b, x)$ для всех $x \in X^L$;
предшествование $a \prec b$: $a \leq b$ и $r(a, b) = 1$.

Опр. Граф расслоения–связности $\langle A, E \rangle$:

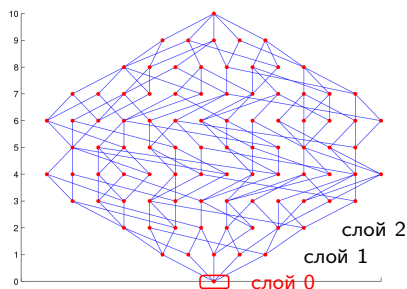
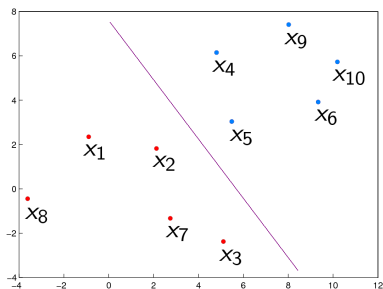
A — множество попарно различных векторов ошибок;

$E = \{(a, b) : a \prec b\}$.

Свойства графа расслоения–связности:

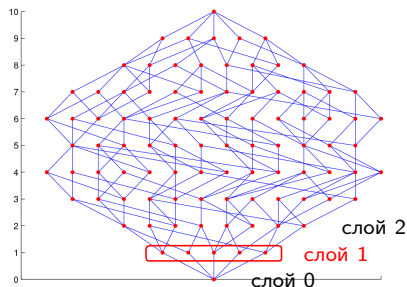
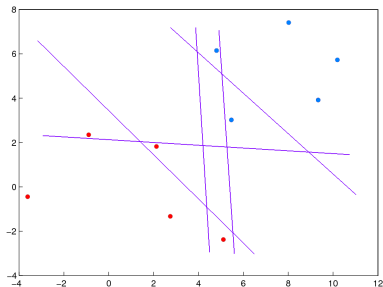
- это подграф графа Хассе отношения порядка \leq на A ;
- каждому ребру (a, b) соответствует объект $x_{ab} \in X^L$, такой, что $I(a, x_{ab}) = 0$, $I(b, x_{ab}) = 1$;
- граф является многодольным со слоями
 $A_m = \{a \in A : m(a, X^L) = m\}$, $m = 0, \dots, L$;

Пример. Семейство линейных алгоритмов классификации



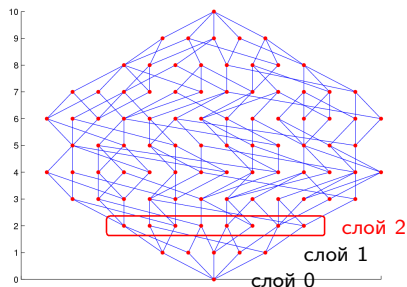
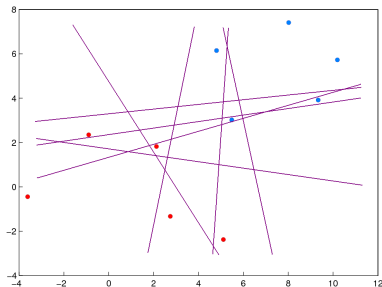
	слой 0
x ₁	0
x ₂	0
x ₃	0
x ₄	0
x ₅	0
x ₆	0
x ₇	0
x ₈	0
x ₉	0
x ₁₀	0

Пример. Семейство линейных алгоритмов классификации



	слой 0	слой 1				
X ₁	0	1	0	0	0	0
X ₂	0	0	1	0	0	0
X ₃	0	0	0	1	0	0
X ₄	0	0	0	0	1	0
X ₅	0	0	0	0	0	1
X ₆	0	0	0	0	0	0
X ₇	0	0	0	0	0	0
X ₈	0	0	0	0	0	0
X ₉	0	0	0	0	0	0
X ₁₀	0	0	0	0	0	0

Пример. Семейство линейных алгоритмов классификации



	слой 0	слой 1						слой 2								
X ₁	0	1	0	0	0	0	0	1	0	0	0	0	1	1	0	...
X ₂	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	...
X ₃	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	...
X ₄	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	...
X ₅	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	...
X ₆	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	...
X ₇	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	...
X ₈	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
X ₉	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
X ₁₀	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...

Характеристики расслоения и связности алгоритма

Теорема

Если μ — пессимистичный минимизатор эмпирического риска, то для любого алгоритма a

$$[\mu(X)=a] \leq [X_a \subseteq X] [X'_a \subseteq \bar{X}], \quad \forall X \in X^L,$$

если взять порождающее и запрещающее множества

$$X_a = \{x_{ab} \in X^L \mid a \prec b\};$$

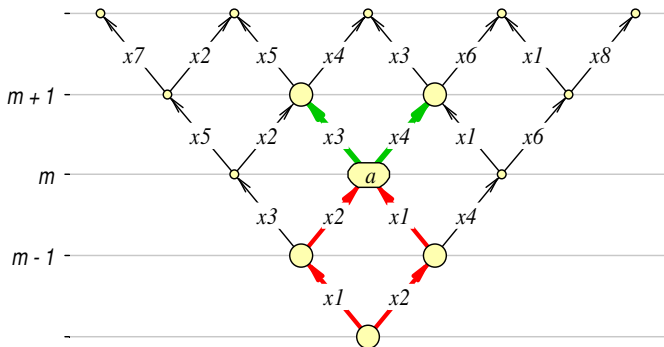
$$X'_a = \{x \in X^L \mid \exists b \in A: b \prec a, l(b, x) < l(a, x)\};$$

Опр. *Верхняя связность* $u(a) = |X_a|$ алгоритма a
— число всех рёбер, исходящих из вершины a .

Опр. *Неполноценность* $q(a) = |X'_a|$ алгоритма a
— число различных объектов на рёбрах путей, ведущих в a .

Пример: двумерная сеть алгоритмов

Верхняя связность $u(a) = 2$, порождающее $X_a = \{x_3, x_4\}$
 Неполноценность $q(a) = 2$, запрещающее $X'_a = \{x_1, x_2\}$



Верхняя оценка расслоения-связности

Теорема

Для пессимистичного метода МЭР μ , любых X^L , A и $\varepsilon \in (0, 1)$

$$Q_\varepsilon(\mu, X^L) \leq \sum_{a \in A} \frac{C_{L-u-q}^{\ell-u}}{C_L^\ell} \mathcal{H}_{L-u-q}^{\ell-u, m-q} \left(\frac{\ell}{L} (m - \varepsilon k) \right),$$

где $u = |X_a|$ — верхняя связность алгоритма a ,
 $q = |X'_a|$ — неполноценность алгоритма a ,
 $m = n(a, X^L)$ — число ошибок алгоритма a .

Следствие: $P[\mu(X) = a] \leq \frac{C_{L-u-q}^{\ell-u}}{C_L^\ell}$.

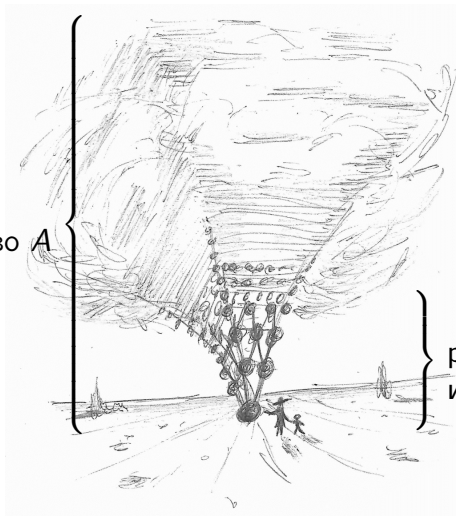
Свойства оценки расслоения–связности

$$Q_\varepsilon \leq \sum_{a \in A} \frac{C_{L-u-q}^{\ell-u}}{C_L^\ell} \mathcal{H}_{L-u-q}^{\ell-u, m-q} \left(\frac{\ell}{L} (m - \varepsilon k) \right)$$

- 1 Вклад алгоритма $a \in A$ убывает экспоненциально по $u(a) \Rightarrow$ **связные семейства меньше переобучаются**;
по $q(a) \Rightarrow$ **только нижние слои вносят вклад в Q_ε** .
- 2 Оценка обращается в равенство в нетривиальных случаях (монотонные цепи и многомерные сети алгоритмов).
- 3 Если $q(a) > k$, то $P_a = 0$ и вклад алгоритма a равен 0 \Rightarrow при малых k оценка вырождается.
- 4 При $|A| = 1$ совпадает с оценкой для одного алгоритма.

Идея использования оценок расслоения-связности

всё семейство A



реально
используемая часть A

Резюме

- Свойства расслоения и связности уменьшают переобучение.
- На практике семейства, как правило, ими обладают. Иначе вероятность переобучения была бы близка к 1 уже при $|A|$ порядка нескольких десятков.
- Практическое применение комбинаторных оценок:
 - 1) оценить $\eta = Q_\varepsilon(\mu, X^L)$ по нескольким нижним слоям;
 - 2) применив обращение, оценить ε через η ;
 - 3) использовать оценку $\nu(X) + \varepsilon$ как критерий выбора модели, метода или отбора признаков.

Цель — лучше понять явление переобучения

- Почему возникает переобучение?
из-за выбора в условиях неполной информации —
оптимизируем алгоритм по конечной обучающей выборке
- Всегда ли оно возникает?
да
- Как оценить величину возможного переобучения?
 - 1) эмпирически — по кросс-валидации,
 - 2) теоретические оценки завышены или трудно вычислимы
- Как её уменьшить?
 - 1) упрощать модель → уменьшать $|A|$
 - 2) уточнять модель → увеличивать расслоение
 - 3) использовать регуляризации → увеличивать расслоение
 - 4) строить непрерывные модели → увеличивать связность

Алгоритм исследователя

Вход: задача;

Выход: решение;

1 **повторять**

2 | размышлять над задачей самому;

3 | обсуждать задачу с коллегами;

4 | делать вычислительные эксперименты;

5 | пробовать решить задачу в простых частных случаях;

6 | читать современную научную литературу по этой теме;

7 **пока** задача не будет решена;

Воронцов Константин Вячеславович

voron@forecsys.ru

www.MachineLearning.ru • Участник:Vokov

Если что-то было не понятно,
не стесняйтесь подходить и спрашивать :)