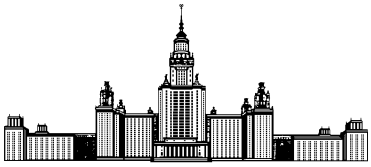
Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной Математики и Кибернетики

Кафедра Математических Методов Прогнозирования

ДИПЛОМНАЯ РАБОТА СТУДЕНТА 517 ГРУППЫ

«Оценки обобщающей способности и применение логических алгоритмов классификации в задаче распознавания вторичной структуры белка»

Выполнил:

студент 5 курса 517 группы

Толстихин Илья Олегович

Научный руководитель:

д.ф-м.н., доцент

Воронцов Константин Вячеславович

Заведующий кафедрой

Математических Методов

Прогнозирования, академик РАН _____ Ю. И. Журавлёв

К защите допускаю

«_____» _____ 2010 г.

К защите рекомендую

«_____» _____ 2010 г.

Москва, 2010

Содержание

Аннотация

Данная работа выполнена в рамках комбинаторной теории надежности обучения по прецедентам [4, 3, 5]. Основной целью данной работы является исследование влияния эффектов сходства и расслоения на вероятность переобучения и расширение класса семейств алгоритмов, для которых возможно получение численно точных оценок вероятности переобучения.

В работе предлагается обобщение теоретико-группового подхода [7], упрощающего получение точных оценок вероятности переобучения в случаях, когда семейство алгоритмов обладает определённой симметрией. Выводятся точные оценки вероятности переобучения для трех модельных семейств — шара алгоритмов, нижних слоев шара и пересечения шара со слоем алгоритмов. Экспериментально показывается, что учёт сходства алгоритмов существенно улучшает точность оценок. Показывается возможность аппроксимации вероятности переобучения расслоенных семейств несколькими их нижними слоями.

В практической части работы рассматривается задача распознавания вторичной структуры белка по его первичной структуре. Предлагаются алгоритмы поиска локальных закономерностей. Исследуются эффекты расслоения и сходства в семействах закономерностей. Показывается, что, благодаря эффекту расслоения, выбор информативных закономерностей базе белков PDB (Protein Data Bank) практически не подвержен переобучению.

1 Введение

С явлением переобучения часто приходится сталкиваться при решении задач классификации, регрессии, прогнозирования. Оно заключается в том, что алгоритм, найденный по критерию минимума частоты ошибок на обучающей выборке, может значительно чаще ошибаться на независимой контрольной выборке. Знание точного значения вероятности переобучения позволило бы управлять процессом обучения и строить более надежные алгоритмы.

Классические оценки вероятности переобучения, предложенные В. Н. Вапником и А. Я. Червоненкисом [1, 2], сильно завышены. Несмотря на многочисленные попытки улучшения точности оценок, предпринятые с тех пор, проблема завышенности остается открытой. Серия экспериментов [5, 4] показала, что для получения численно точных оценок вероятности переобучения необходим одновременный учет двух свойств семейства алгоритмов — степени *сходства* алгоритмов внутри семейства и *расслоения* семейства по уровням частоты ошибок.

В [3] предложена комбинаторная теория надежности обучения по прецедентам, позволяющая получать точные оценки вероятности переобучения. Основной целью данной работы является исследование влияния эффектов сходства и расслоения на вероятность переобучения, а также расширение класса семейств алгоритмов, для которых возможно получение точных комбинаторных оценок вероятности переобучения.

В разделе 1 определяются основные понятия и приводятся классические оценки.

В разделе 2 предлагается обобщение теоретико-группового подхода [7].

В разделах 3 и 4 выводятся точные оценки вероятности переобучения для трёх модельных семейств алгоритмов, и на их примере исследуется влияние расслоения и сходства алгоритмов на вероятность переобучения.

В разделе 5 исследуется переобучение логических закономерностей в прикладной задаче распознавания вторичной структуры белка.

1.1 Определения и обозначения

Пусть заданы множество объектов $\mathbb{X} = \{x_1, x_2, \ldots, x_L\}$, множество алгоритмов $A = \{a_1, a_2, \ldots a_D\}$ и *индикатор ошибки* — бинарная функция $I : \mathbb{X} \times A \to \{0, 1\}$:

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

Алгоритмы задаются своими бинарными *векторами ошибок* $a \equiv \big(I(a,x_i)\big)_{i=1}^{L}$ длины L. Таким образом множество A является элементом системы всех возможных множеств алгоритмов $2^{\mathbb{A}}$ ($\mathbb{A} = \{0,1\}^L$ — множество всех бинарных векторов длины L). *Матрицей ошибок* называется $D\times L$-матрица, строками которой являются векторы ошибок алгоритмов из множества A. *Число ошибок* алгоритма a на выборке $X \subseteq \mathbb{X}$ вычисляется следующим образом:

$$n(a,X) = \sum_{x\in X} I(a,x).$$

Частотой ошибок алгоритма a на выборке $X \subseteq \mathbb{X}$ (его *эмпирическим риском*) называется величина

$$\nu(a,X) = \frac{n(a,X)}{|X|}.$$

Для фиксированного целого числа ℓ обозначим $[\mathbb{X}]^\ell$ множество всевозможных выборок $X \subseteq \mathbb{X}$.

Методом обучения называется отображение $\mu\colon [\mathbb{X}]^\ell \to A$, ставящее в соответствие произвольной выборке X длины ℓ некоторый алгоритм a из семейства A.

Метод обучения μ, минимизирующий число ошибок на обучающей выборке $X \subseteq \mathbb{X}$, называется *методом минимизации эмпирического риска (МЭР)*:

$$\mu X = \arg\min_{a\in A} n(a,X). \tag{1.1}$$

Уклонением частот ошибок алгоритма a на разбиении X называется величина $\delta(a,X) = \nu\left(a,\bar{X}\right) - \nu(a,X)$. Величину $\delta_\mu(X) = \nu\left(\mu X, \mathbb{X}\backslash X\right) - \nu\left(\mu X, X\right)$ будем называть *переобученностью* метода μ на выборке X. В случае большого уклонения частот $\delta_\mu(X) \geqslant \varepsilon$, где ε — фиксированный *порог переобучения*, будем говорить, что *метод μ переобучен на выборке X*.

Основной задачей является вычисление *вероятности переобучения* — вероятности большого уклонения частот ошибок выбираемых методом μ алгоритмов на *контрольной выборке* $\bar{X} = \mathbb{X}\backslash X$ по сравнению с обучающей выборкой X.

Следуя комбинаторному подходу [3], будем полагать, что при фиксированном ℓ все C_L^ℓ разбиений *генеральной выборки* $\mathbb{X}$ на наблюдаемую обучающую X длины ℓ и контрольную $\bar{X}$ длины k, равновероятны, $k+\ell = L$. Тогда вероятность большого уклонения частот ошибок может быть записана в виде:

$$Q_\varepsilon(\mu,\mathbb{X}) = P\left[\delta_\mu(X) \geqslant \varepsilon\right] = \frac{1}{C_L^\ell} \sum_{X\subseteq[\mathbb{X}]^\ell} \left[\delta_\mu(X) \geqslant \varepsilon\right],$$

а вероятность получения алгоритма $a \in A$ в результате обучения — в виде

$$P\left[\mu X = a\right] = \frac{1}{C_L^\ell} \sum_{X \subseteq [\mathbb{X}]^\ell} \left[\mu X = a\right].$$

1.2 Постановка задачи

Классические оценки Вапника-Червоненкиса [2] (далее VC-оценки) в рамках комбинаторного подхода могут быть записаны следующим образом [5]:

$$Q_\varepsilon(\mu, \mathbb{X}) \leqslant \sum_{m=\lceil \varepsilon k \rceil}^{L} \Delta_m H_L^{l,m}\left(\tfrac{l}{L}(m - \varepsilon k)\right) \leqslant$$

$$\leqslant |A| \max_{m=0,\ldots,L} H_L^{l,m}\left(\tfrac{l}{L}(m - \varepsilon k)\right), \tag{1.2}$$

где Δ_m — *коэфициент разнообразия* m-го слоя (множества алгоритмов A_m, допускающих ровно m ошибок на генеральной выборке $\mathbb{X}$), равный числу различных алгоритмов m-го слоя; $H_L^{\ell,m}(z)$ — *гипергеометрическая функция распределения:*

$$H_L^{\ell,m}(z) = \sum_{s=\max(0,m-k)}^{\lfloor z \rfloor} \frac{C_m^s C_{L-m}^{\ell-s}}{C_L^\ell}.$$

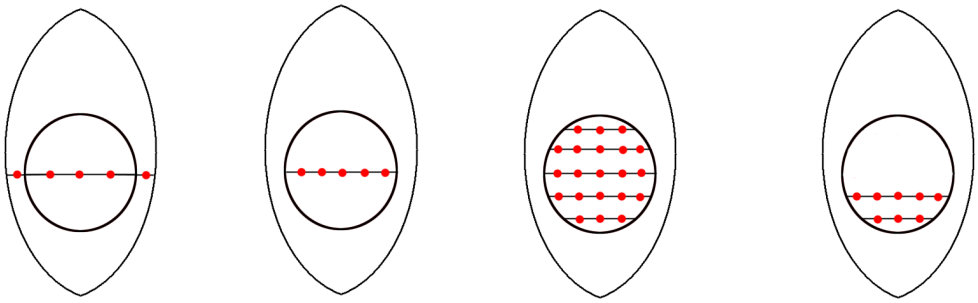
Известно, что VC-оценки сильно завышены, а в наиболее интересных для практики случаях малых выборок и сложных семейств алгоритмов превосходят единицу.

В [4, 3] экспериментально установлены основные причины завышенности VC-оценок. Дело в том, что VC-оценки учитывают лишь размерность матрицы ошибок, но не учитывают её содержимое. Для получения точных оценок необходимо одновременно учитывать эффекты *сходства* и *расслоения* в семействе алгоритмов.

- **Расслоение:** семейства, используемые для решения практических задач, обычно содержат относительно мало «хороших» алгоритмов в нижних слоях и огромное число «неподходящих» алгоритмов в верхних слоях. Алгоритм, допускающий бóльшее число ошибок, имеет меньшую вероятность быть выбранным на этапе обучения. Это означает, что алгоритмы из верхних слоёв фактически не вносят вклады в вероятность переобучения.

- **Сходство:** в реальных семействах для каждого алгоритма, как правило, находится некоторое количество других алгоритмов, похожих на него в смысле метрики Хэмминга. Чем «плотнее» семейство алгоритмов, то есть чем больше в нём похожих алгоритмов, тем сильнее завышено неравенство Буля, используемое при выводе VC-оценок, и, как следствие, сами VC-оценки.

В данной работе влияние эффектов сходства и расслоения на вероятность переобучения исследуется на примере четырех модельных семейств:



1) слой 2) слой шара 3) шар 4) нижние слои шара

1. Семейство, состоящее из случайных алгоритмов m-го слоя, не обладает ни расслоением, ни связностью.

2. Семейство, состоящее из всех алгоритмов m-го слоя хэммингова шара радиуса r с центром в некотором алгоритме a_0 из m-го слоя, $n(a_0, \mathbb{X}) = m$. Это семейство не обладает эффектом расслоения, но обладает эффектом сходства, причём является в некотором смысле «наиболее плотным».

3. Хэммингов шар — это семейство со сходством и расслоением.

4. Наконец, t нижних слоёв хэммингова шара — это семейство, позволяющее исследовать эффект расслоение внутри хэммингова шара.

2 Симметрии семейств алгоритмов

Метод минимизации эмпирического риска, введенный в первой главе, связан со следующей неоднозначностью: в семействе $A \subseteq \mathbb{A}$ может оказаться несколько алгоритмов, минимизирующих (1.1): $|A(X)| > 1$, где

$$A(X) = \operatorname{Arg}\min_{a \in A} n(a, X).$$

В [4] вводится *метод пессимистической минимизации эмпирического риска* (ПМЭР), который в этом случае выбирает алгоритм из $A(X)$ с максимальным числом ошибок на контрольной выборке $\bar{X}$:

$$\mu X \in \operatorname{Arg}\max_{a \in A(X)} n(a, \bar{X}).$$

Если и таких алгоритмов оказывается несколько, то выбирается алгоритм с большим порядковым номером, в соответствии с некоторым фиксированным линейным порядком на множестве алгоритмов A.

Метод пессимистической минимизации эмпирического риска представляет, главным образом, теоретический интерес, поскольку он даёт точные (достигаемые) верхние оценки вероятности переобучения. Однако он не реализуем на практике, поскольку требует знания скрытой контрольной выборки $\bar{X}$. Кроме того, введение линейного порядка на алгоритмах является искусственным приемом, не имеющим адекватных аналогов среди практических методов обучения.

В работе [7] предложен *рандомизированный метод минимизации эмпирического риска*, не требующий введения линейного порядка на алгоритмах. Он выбирает произвольный алгоритм из $A(X)$ случайно и равновероятно.

Для рандомизированного метода вводится понятие *вклада алгоритма* a в вероятность переобучения:

$$Q_\varepsilon(\mu, \mathbb{X}, a) = \frac{1}{C_L^l} \sum_{X \in [\mathbb{X}]^\ell :\, a \in A(X)} \frac{[\delta(a, X) \geqslant \varepsilon]}{|A(X)|},$$

а сама вероятность переобучения определяется следующим образом:

$$Q_\varepsilon(\mu, \mathbb{X}) = \sum_{a \in A} Q_\varepsilon\left(\mu, X, a\right)$$

Далее будет использоваться только рандомизированный метод МЭР.

Симметрическая группа S_L очевидным образом действует на множестве объектов $S_L : \mathbb{X} \to \mathbb{X}$. Определим действие элемента π группы S_L на произвольную выборку $X \in [\mathbb{X}]^\ell$ поэлементным действием π на объекты выборки X: $\pi(X) = \{\pi(x) : x \in X\}$. Очевидно, действие элемента группы на выборку не меняет числа объектов в ней $|X| = |\pi(X)|$.

Действие группы S_L на множестве всех алгоритмов $\mathbb{A}$ определим перестановкой координат векторов ошибок алгоритмов: $\pi(a) = \left(a\left(\pi^{-1}(x_i)\right)\right)_{i=1}^L$. При таком определении выполняется естественное равенство $n(a, X) = n\left(\pi(a), \pi(X)\right)$. Заметим, что это отображение биективно.

Наконец, действие элемента π группы S_L на системе всех подмножеств алгоритмов $2^{\mathbb{A}}$ определяется по правилу $\pi(A) = \{\pi(a) : a \in A\}$.

Группой симметрий $S(A)$ множества алгоритмов $A \subseteq \mathbb{A}$ будем называть его стационарную подгруппу:

$$S(A) = \left\{\pi \in S_L : \pi(A) = A\right\}.$$

Орбитой действия группы симметрий A на алгоритм $a \in A$ будем называть множество $\{\pi(a) : \pi \in S(A)\}$. Из теории групп известно, что орбиты разных элементов множества либо не пересекаются, либо совпадают. Таким образом, множество алгоритмов A разбивается на классы эквивалентности — непересекающиеся орбиты действия группы $S(A)$. Алгоритмы из одной орбиты договоримся называть *идентичными* (понятие эквивалентных алгоритмов вводилось в [2] совершенно в другом смысле).

В [7] показано, что идентичные алгоритмы вносят равные вклады в вероятность переобучения. А именно, справедливы две теоремы.

Теорема 2.1. *Вероятность получения идентичных алгоритмов в результате обучения одинакова:* $\forall \pi \in S(A)$ *выполнено*

$$P\left[\mu X = a\right] = P\left[\mu X = \pi(a)\right].$$

Теорема 2.2. *Идентичные алгоритмы дают равный вклад в вероятность переобучения:* $\forall \pi \in S(A)$ *выполнено*

$$Q_\varepsilon(\mu, \mathbb{X}, a) = Q_\varepsilon\left(\mu, \mathbb{X}, \pi(a)\right).$$

Эти результаты существенно упрощают получение точных оценок вероятности переобучения в тех случаях, когда семейство алгоритмов обладает определённой симметрией.

Замечание. При использовании метода пессимистической минимизации эмпирического риска вероятности получения идентичных алгоритмов могут не совпадать. Приведём в качестве примера матрицу ошибок для двух алгоритмов и четырёх объектов:

	x_1	x_2	x_3	x_4
a_1	1	1	0	0
a_2	0	0	1	1

Положим $\ell = k = 2$. Очевидно, что при обучении ПМЭР алгоритм a_1 будет выбран лишь на разбиении $X = \{x_3, x_4\}, \bar{X} = \{x_1, x_2\}$. Таким образом, вероятности получения алгоритмов a_1 и a_2 в результате обучения ПМЭР не совпадают. В то же время, алгоритмы являются идентичными, и вероятности их получения при обучении рандомизированным методом равны.

2.1 Лемма о подгруппе симметрий

Множество всех орбит действия группы симметрий $S(A)$ на A будем обозначать $\Omega(A)$. Описанные результаты дают возможность представить вероятность переобучения в виде:

$$Q_\varepsilon(\mu, \mathbb{X}) = \sum_{\omega \in \Omega(A)} |\omega| Q_\varepsilon\left(\mu, \mathbb{X}, a_\omega\right) = \\ = \sum_{\omega \in \Omega(A)} \frac{|\omega|}{C_L^l} \sum_{X \in [\mathbb{X}]^\ell \colon a_\omega \in A(X)} \frac{[\delta(a_\omega, X) \geqslant \varepsilon]}{|A(X)|}, \tag{2.1}$$

где a_ω— произвольный представитель орбиты $\omega \in \Omega(A)$.

На практике нахождение группы симметрий $S(A)$ может оказаться слишком сложной задачей. Поэтому в данной работе предлагается ограничиться поиском подгруппы $\hat{S}$ группы симметрий. Для найденной подгруппы формула (2.1), очевидно, остается справедливой. В этом случае $\Omega(A)$ будем понимать как множество орбит действия подгруппы $\hat{S} \leqslant S(A)$ на A.

Лемма 2.1. *Алгоритмы из одной орбиты действия подгруппы группы симметрий $S(A)$ на множестве алгоритмов A дают равные вклады в вероятность переобучения: $\forall \pi \in \hat{S} \leqslant S(A)$ выполнено*

$$Q_\varepsilon(\mu, \mathbb{X}, a) = Q_\varepsilon\left(\mu, \mathbb{X}, \pi(a)\right).$$

Доказательство. Орбита действия подгруппы $\hat{S}$ является подмножеством соответствующей орбиты действия группы $S\left(a\right)$, таким образом, лемма очевидным образом следует из теоремы 2.2. ■

2.2 Симметрии и орбиты разбиений

Мы определили действие группы симметрий $S(A)$ на множестве всевозможных разбиений $[\mathbb{X}]^\ell$. Орбитой разбиения X назовём множество $\{\pi(X) \colon \pi \in S(A)\}$. Разбиения из одной орбиты действия симметрической группы $S(A)$ на множестве разбиений $[\mathbb{X}]^\ell$ будем называть *идентичными*.

В дальнейшем нам понадобится следующая лемма [7].

Лемма 2.2. *Для любой перестановки π*

$$n(a, X) = n\left(\pi(a), \pi(X)\right).$$

Теорема 2.3. *Для любых двух идентичных разбиений $X_1, X_2 \subseteq \mathbb{X}$ верно*

$$|A(X_1)| = |A(X_2)|.$$

Доказательство. Докажем, что если $a_0 \in A(X)$ и $\pi \in S(A)$, то $\pi(a_0) \in A(\pi(X))$.

$$n(a_0, X) = \min_{a \in A} n(a, X) = \min_{a \in A} n(\pi(a), \pi(X)).$$

Поскольку $\pi(A) = A$, то

$$\min_{a \in A} n(\pi(a), \pi(X)) = \min_{a \in A} n(a, \pi(X)).$$

А так как

$$n(\pi(a_0), \pi(X)) = n(a_0, X) = \min_{a \in A} n(a, \pi(X)),$$

то $\pi(a_0) \in A(\pi(X))$. ■

Теорема 2.4. *Если $X_1, X_2 \subseteq \mathbb{X}$ — идентичные разбиения и существует алгоритм $a \in A(X_1) \cap A(X_2)$, то верно:*

$$\delta(a, X_1) = \delta(a, X_2).$$

Доказательство. Пусть $X_2 = \pi(X_1), \pi \in S(A)$. Поскольку $a \in A(X_1)$ то

$$\pi(a) \in A(\pi(X_1)) = A(X_2).$$

Но алгоритм a также принадлежит множеству $A(X_2)$. Следовательно

$$n(a, X_2) = n(\pi(a), X_2) = n(\pi(a), \pi(X_1)) = n(a, X_1).$$

Отсюда следует утверждение теоремы. ■

Множество всех орбит действия группы симметрий $S(A)$ на $[\mathbb{X}]^\ell$ будем обозначать $\Omega(\mathbb{X})$. Теоремы 2.3 и 2.4 дают возможность переписать (2.1) в следующем виде:

$$Q_\varepsilon(\mu, \mathbb{X}) = \sum_{\omega \in \Omega(A)} \frac{|\omega|}{C_L^l} \sum_{\tau \in \Omega(\mathbb{X})} |X \in \tau : a_\omega \in A(X)| \frac{[\delta(a_\omega, X_\tau) \geqslant \varepsilon]}{|A(X_\tau)|}. \qquad (2.2)$$

Замечание Этот результат справедлив и для подгруппы $\hat{S} \leqslant S(A)$, поскольку орбита действия $\hat{S}$ — подмножество соответствующей орбиты действия $S(A)$.

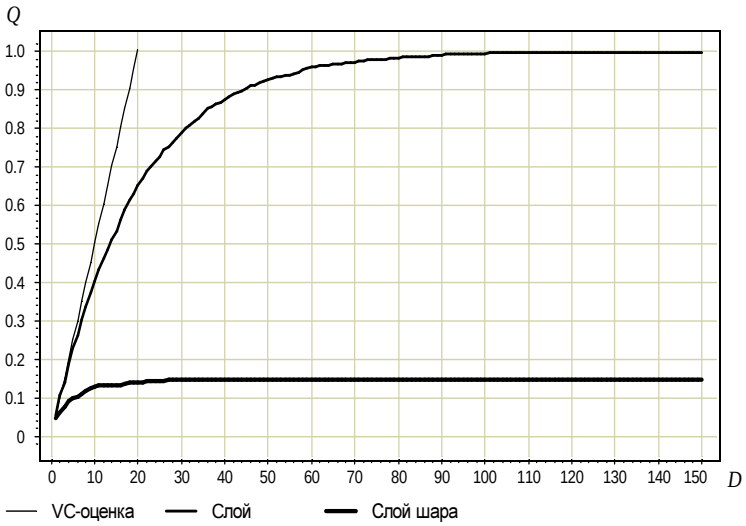


Рис. 1: Оценки для множеств случайных алгоритмов слоя и алгоритмов слоя шара.

3 Эффект сходства

Начнём изучение влияния эффекта сходства на вероятность переобучения с простого эксперимента.

Для произвольной заданной матрицы ошибок вероятность переобучения Q_ε нетрудно оценить эмпирически методом Монте-Карло. Для этого достаточно заменить вероятность, определяемую как долю всех разбиений выборки, на долю разбиений из заданного случайного подмножества разбиений. В данном эксперименте бралась тысяча случайных разбиений. Сравнивались два семейства, лежащих в m-м слое, $m = 10$. Первое семейство состояло из D случайных представителей m-го слоя. Второе семейство состояло из D случайных представителей пересечения m-го слоя и хэмминова шара радиуса $r = 2$ с центром в том же m-м слое. Алгоритмы из второго семейства мощностью 1901 брались в порядке увеличения хэммингова расстояния от центра шара. Длина выборки $l = k = 100$, порог переобученности $\varepsilon = 0.05$. Кроме того, вычислялась VC-оценка (1.2), очевидно, одинаковая для обоих семейств. Зависимости эмпирических оценок вероятности переобучения Q_ε от числа алгоритмов D в семействе представлены на рисунке 1.

По графику можно сделать следующие выводы.

1. Вероятность переобучения для более «плотного» второго семейства проходит значительно ниже, чем для «разреженного» первого семейства. Получение точных оценок вероятности переобучения для хэммингова шара и его слоёв представляет значительный теоретический интерес, поскольку хэммингов шар является «макси-

мально плотным» множеством булевых векторов.

2. Для первого семейства вероятность переобучения Q_ε достигает единицы уже при $D \approx 100$. Это означает, что для оценивания вероятности переобучения «плотных» семейств не обязательно брать все алгоритмы семейства. Для получения достаточно точных нижних оценок можно брать относительно небольшое «разреженное» подмножество, состоящее из несхожих алгоритмов. В частности, можно отбирать их случайным образом.

Ниже будут получены точные оценки вероятности переобучения для слоя шара, полного хэммингова шара и заданного числа его нижних слоёв. Поскольку хэмминговы шары обладают определёнными симметриями, вполне естественно использовать теоретико-групповой подход и предполагать, что применяется рандомизированный метод обучения.

3.1 Точная оценка вероятности переобучения для слоя шара

Зададим семейство алгоритмов A как пересечение хэммингова шара $B_r(a_0)$ с m-м слоем алгоритмов A_m, где $m = n(a_0, \mathbb{X})$. Положим без ограничения общности, что алгоритм a_0 ошибается на m первых объектах генеральной выборки $\mathbb{X}$.

В дальнейшем множество, состоящее из первых m объектов, будем обозначать X^m. Множество, состоящее из последних $L - m$ объектов, будем обозначать X^{L-m}. Расстояние Хэмминга между алгоритмами $a_1, a_2 \in A$ будем обозначать $d(a_1, a_2)$. Введём обозначение $b = m - \lfloor r/2 \rfloor$.

Определим стандартным образом декартово произведение $S_m \times S_{L-m}$, где S_m и S_{L-m} — симметрические группы перестановок, действующих на множествах X^m и X^{L-m} соответственно. Действие элемента (π, φ) группы $S_m \times S_{L-m}$, где $\pi \in S_m$ и $\varphi \in S_{L-m}$, на алгоритмы определяется последовательным действием перестановок π и φ. При этом, очевидно, $\pi\varphi = \varphi\pi$.

Лемма 3.1. *Группа $S_m \times S_{L-m}$ является подгруппой группы симметрии семейства алгоритмов A.*

Доказательство. Нам достаточно показать, что если $a \in B_r(a_0) \cap A_m$, то и $\pi(a) \in \in B_r(a_0) \cap A_m$, где перестановка $\pi \in S_m \times S_{L-m}$.

Поскольку действие элементов симметрической группы S_L на алгоритмы не меняет числа ошибок алгоритмов, то $n(\pi(a), \mathbb{X}) = n(a, \mathbb{X}) = m$. Очевидно, для

$\pi \in S_m \times S_{L-m}$ справедливы следующие равенства: $n(a, X^m) = n(\pi(a), X^m)$ и $n\left(a, X^{L-m}\right) = n\left(\pi(a), X^{L-m}\right)$. Из них получаем:

$$d(a, a_0) = m - n(a, X^m) + n(a, X^{L-m}) = \\ = m - n(\pi(a), X^m) + n(\pi(a), X^{L-m}) = d(\pi(a), a_0).$$

Итак, алгоритм $\pi(a)$ допускает m ошибок на генеральной выборке $\mathbb{X}$, и расстояние Хэмминга $d(\pi(a), a_0) = d(a, a_0) \leqslant r$. ■

Лемма 3.2. *Орбитами действия группы $S_m \times S_{L-m}$ на множестве алгоритмов A являются пересечения m-го слоя булева куба со сферами радиусов $2k$, $k = 0 \ldots \lfloor r/2 \rfloor$ с центрами в a_0.*

Доказательство. Пусть алгоритм a допускает m ошибок на генеральной выборке и принадлежит сфере радиуса r_1: $d(a, a_0) = r_1$. В доказательстве предыдущей леммы мы установили, что если $\pi \in S_m \times S_{L-m}$, то $d(\pi(a), a_0) = r_1$ и $\pi(a)$ также принадлежит m-му слою. Очевидно также, что расстояние Хэмминга между двумя алгоритмами одного слоя — величина четная.

Осталось доказать, что для любых алгоритмов $a_1, a_2 \in A_m$ таких, что $d(a_1, a_0) = d(a_2, a_0) = r_1$, найдется перестановка $\pi \in S_m \times S_{L-m}$ для которой $\pi(a_1) = a_2$. Но это следует из того, что $n(a_1, X^m) = n(a_2, X^m)$ и $n\left(a_1, X^{L-m}\right) = n\left(a_2, X^{L-m}\right)$. Последний факт легко установить, выразив число ошибок, допускаемых алгоритмами a_1 и a_2 на множестве X^m, через m и r_1. ■

Таким образом, всего имеется $\lfloor r/2 \rfloor + 1$ орбит. Пронумеруем их индексами от b до m по числу ошибок, допускаемых алгоритмами из них на множестве X^m. Тогда в h-й орбите, которую мы будем обозначать $\mathrm{Orb}(h)$, ровно $C_m^h C_{L-m}^{m-h}$ алгоритмов. В дальнейших рассуждениях может помочь рис. 2, на котором показано по одному алгоритму из каждой орбиты рассматриваемого семейства.

В процессе доказательства мы будем пользоваться формулой (2.2). Для этого нам необходимо исследовать условия попадания алгоритмов во множество $A(X)$.

Лемма 3.3. $\forall X, \forall a \in A \setminus \mathrm{Orb}(m - \lfloor r/2 \rfloor)$,

$$a \in A(X) \Leftrightarrow n(a, X) = 0.$$

Доказательство. Достаточность очевидна. Докажем необходимость.

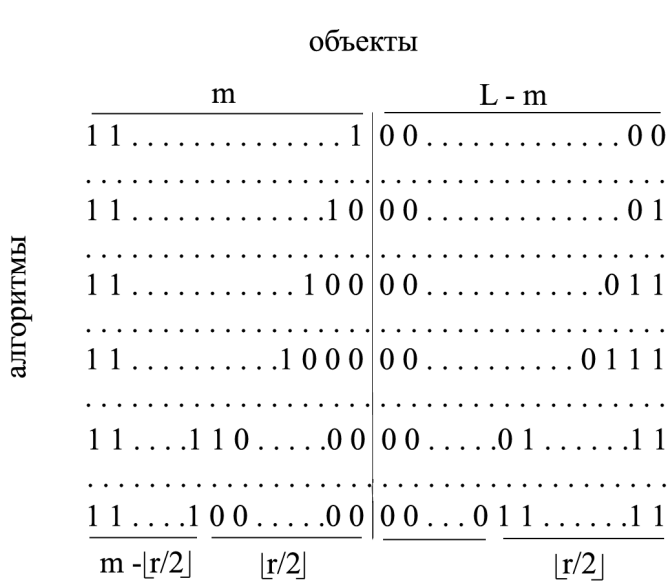


Рис. 2: Алгоритмы из орбит семейства A.

Пусть алгоритм $a^h \in \mathrm{Orb}(h),\ h \neq b$, попал во множество $A(X)$. Нам требуется доказать, что он не допускает ошибок на выборке X.

Пусть $X^{l_1} = |X^m \cap X|,\ X^{l_2} = |X^{L-m} \cap X|;\ l_1 = |X^{l_1}|,\ l_2 = |X^{l_2}|;\ l = |X|,$ $l = l_1 + l_2;\ X = X^{l_1} \cup X^{l_2}$. В дальнейших рассуждениях может помочь рис. 2, на котором представлены алгоритмы из разных орбит семейства A.

Допустим, $l_1 \leqslant \lfloor r/2 \rfloor$. Покажем, что для любого X найдётся алгоритм $a \in A$ такой, что $n(a, X) = 0$. Рассмотрим $\mathrm{Orb}(m - l_1)$. Все алгоритмы этой орбиты допускают ровно $m - l_1$ ошибок на множестве X^m и l_1 ошибок на множестве X^{L-m}. Пусть A_1 — те из них, которые не допускает ни одной ошибки на множестве X^{l_1}. Множество A_1, очевидно, непусто. Алгоритмы из A_1 не ошибаются на $L - m - l_1$ объектах множества X^{L-m}. А поскольку $l_1 + l_2 = l \leqslant L - m$, то $L - m - l_1 \geqslant l_2$. Таким образом, во множестве A_1 существует алгоритм a', для которого $n\left(a', X^{l_2}\right) = 0$. По определению множества A_1 имеем $n\left(a', X\right) = 0$.

Рассмотрим теперь случай, когда $l_1 > \lfloor r/2 \rfloor$. Алгоритмы из $\mathrm{Orb}(m - \lfloor r/2 \rfloor)$ допускают ровно $\lfloor r/2 \rfloor$ ошибок на X^{L-m}, а правильные ответы дают на $L - m - \lfloor r/2 \rfloor$ объектах этого множества. Таким образом, множество $A_2 = \left\{a \in \mathrm{Orb}(m - \lfloor r/2 \rfloor) \colon n\left(a, X^{l_2}\right) = 0\right\}$ непусто, поскольку $l_2 = l - l_1 < l - \lfloor r/2 \rfloor \leqslant L - m - \lfloor r/2 \rfloor$. Алгоритмы из указанной орбиты допускают $m - \lfloor r/2 \rfloor$ ошибок на множестве X^m. Во множестве A_2 найдётся алгоритм a'', для которого $n\left(a'', X^{l_1}\right) = l_1 - \lfloor r/2 \rfloor$.

Итак, $n\left(a'', X^{l_2}\right) = 0$ и $n\left(a'', X^{l_1}\right) = l_1 - \lfloor r/2 \rfloor$. Но $n\left(a^h, X^{l_1}\right) \geqslant l_1 - (m - h) >$

$l_1 - m + (m - \lfloor r/2 \rfloor) = l_1 - \lfloor r/2 \rfloor = n\left(a'', X^{l_1}\right)$, что противоречит тому, что $a^h \in A(X)$.

На этом доказательство критерия завершется. ■

Следствие 3.4. *В ходе доказательства установлено, что при* $|X \cap X^m| > \lfloor r/2 \rfloor$ *алгоритмы из орбит, отличных от* b*-ой, не могут попасть во множество* $A(X)$.

Лемма 3.5. $\forall a^b \in \mathrm{Orb}(m - \lfloor r/2 \rfloor)$

если $X : |X \cap X^m| \leqslant \lfloor r/2 \rfloor$, *то*

$$a^b \in A(X) \Leftrightarrow n\left(a^b, X\right) = 0;$$

если $X : |X \cap X^m| > \lfloor r/2 \rfloor$, *то*

$$a^b \in A(X) \Leftrightarrow \begin{cases} n\left(a^b, X \cap X^{L-m}\right) = 0; \\ n\left(a^b, X \cap X^m\right) = |X \cap X^m| - \lfloor r/2 \rfloor. \end{cases}$$

Доказательство. Начнём со случая, когда $l_1 \leqslant \lfloor r/2 \rfloor$. В этом случае рассуждения полностью повторяют предыдущие (первая часть доказательства леммы 3.3, и мы приходим к выводу, что $a^b \in A(X) \Leftrightarrow n\left(a^b, X\right) = 0$.

Для $l_1 > \lfloor r/2 \rfloor$ ситуация меняется. Предположим, что $a^b \in \mathrm{Orb}(m - \lfloor r/2 \rfloor)$ и $a^b \in A(X)$, $|X \cap X^m| > \lfloor r/2 \rfloor$. Нашей ближайшей целью будет показать, что в этом случае $n\left(a^b, X^{l_2}\right) = 0$, а $n\left(a^b, X^{l_1}\right) = l_1 - \lfloor r/2 \rfloor$.

Предположим, что $n\left(a^b, X^{l_2}\right) \neq 0$. Поскольку $l_2 = l - l_1 < l - \lfloor r/2 \rfloor \leqslant L - m - \lfloor r/2 \rfloor$, в $\mathrm{Orb}(m - \lfloor r/2 \rfloor)$ найдётся алгоритм a_0^b, который на множестве X^m идентичен алгоритму a^b, в то время как $n\left(a_0^b, X^{l_2}\right) = 0$. Это противоречит тому, что $a^b \in A(X)$.

Равенство $n\left(a^b, X^{l_1}\right) = l_1 - \lfloor r/2 \rfloor$ следует из второй части доказательства леммы 3.3.

На этом доказательство критерия заканчивается, поскольку для разбиений $X : |X \cap X^m| > \lfloor r/2 \rfloor$ не существует алгоритмов из семейства A, допускающих на X меньше $l_1 - \lfloor r/2 \rfloor$ ошибок. ■

Лемма 3.6. *Орбитами действия группы* $S_m \times S_{L-m}$ *на множестве разбиений являются множества* $\{X : |X \cap X^m| = i\}$, *где* $i = \max(0, m-k), \ldots, \min(l, m)$.

Доказательство. Лемма очевидным образом следует из определений орбиты и действия группы перестановок на множестве разбиений. ■

Теорема 3.1 (Точная оценка для слоя шара). *Пусть $n(a_0, \mathbb{X}) = m$. Рассмотрим пересечение шара $B_r(a_0)$ с полным m-м слоем алгоритмов A_m. Тогда при обучении рандомизированным методом минимизации эмпирического риска и $l \leqslant L - m$ вероятность переобучения может быть записана в виде:*

$$Q_\varepsilon(\mu, \mathbb{X}) = \begin{cases} H_L^{\ell,m}\big(s_d(\varepsilon) + r/2\big), & s_d(\varepsilon) \geqslant 0; \\ 0, & s_d(\varepsilon) < 0, \end{cases}$$

где $s_d(\varepsilon) = \frac{\ell}{L}(m - \varepsilon k)$.

Доказательство. Лемма 3.1 дает возможность воспользоваться формулой (2.2):

$$Q_\varepsilon(\mu, \mathbb{X}) = \sum_{\omega \in \Omega(A)} \frac{|\omega|}{C_L^l} \sum_{\tau \in \Omega(\mathbb{X})} |X \in \tau : a_\omega \in A(X)| \frac{[\delta(a_\omega, X_\tau) \geqslant \varepsilon]}{|A(X_\tau)|}.$$

С учетом лемм 3.2 и 3.6

$$Q_\varepsilon(\mu, \mathbb{X}) = \sum_{h=m-\lfloor r/2 \rfloor}^{m} \frac{C_m^h C_{L-m}^{m-h}}{C_L^l} \sum_{i=0}^{\min(l,m)} \underbrace{|X : |X \cap X^m| = i, a^h \in A(X)|}_{S(i,h)} \frac{\big[\delta\left(a^h, X_i\right) \geqslant \varepsilon\big]}{|A(X_i)|},$$

где a^h — алгоритмы, представленные на рисунке 2, а X_i — произвольное разбиение, в которое входят ровно i объектов из X^m. Всюду далее будем пользоваться следующим обозначением: $\rho = \lfloor r/2 \rfloor$.

Разобьем сумму по i на два слагаемых:

$$Q_\varepsilon(\mu, \mathbb{X}) = \sum_{h=m-\rho}^{m} \frac{C_m^h C_{L-m}^{m-h}}{C_L^l} \sum_{i=0}^{\rho} S(i,h) \frac{\big[\delta\left(a^h, X_i\right) \geqslant \varepsilon\big]}{|A(X_i)|} + \\ + \sum_{h=m-\rho}^{m} \frac{C_m^h C_{L-m}^{m-h}}{C_L^l} \sum_{i=\rho+1}^{\min(l,m)} S(i,h) \frac{\big[\delta\left(a^h, X_i\right) \geqslant \varepsilon\big]}{|A(X_i)|}.$$

Из леммы 3.3 следует, что первое слагаемое соответствует случаю выбора алгоритмов, не допускающих ошибок на обучающей выборке. Из леммы 3.5 и из следствия 3.4 следует, что во втором слагаемом сумму по орбитам алгоритмов можно опустить, поскольку при данных разбиениях выбираться будут только алгоритмы из орбиты $m - \rho$, допускающие $i - \rho$ ошибок на X_i. Учитывая эти факты, имеем:

$$Q_\varepsilon(\mu, \mathbb{X}) = \big[s_d(\varepsilon) \geqslant 0\big] \sum_{h=m-\rho}^{m} \frac{C_m^h C_{L-m}^{m-h}}{C_L^l} \sum_{i=0}^{\rho} \frac{S(i,h)}{|A(X_i)|} + \\ + \frac{C_m^\rho C_{L-m}^\rho}{C_L^l} \sum_{i=\rho+1}^{\min(l,m)} \frac{S(i,h)}{|A(X_i)|} \big[i \leqslant s_d(\varepsilon) + r/2\big]. \qquad (3.1)$$

Значения $|A(X_i)|$ получаются легко из лемм 3.3 и 3.5:

$$|A(X_i)| = \begin{cases} \sum\limits_{j=m-\rho}^{m} C_{m-i}^{j} C_{k-m+i}^{m-j}, & i \leqslant \rho; \\ C_i^{\rho} C_{k-m+i}^{\rho}, & i > \rho. \end{cases}$$

Значения $S(i,h)$ также легко получаются из лемм 3.3 и 3.5:

$$S(i,h) = \begin{cases} C_{m-h}^{i} C_{L-2m+h}^{l-i}, & i \leqslant \rho; \\ C_{m-\rho}^{m-i} C_{L-m-\rho}^{l-i}, & i > \rho. \end{cases}$$

Подставим полученные значения в (3.1):

$$Q_\varepsilon(\mu, \mathbb{X}) = [s_d(\varepsilon) \geqslant 0] \sum_{h=m-\rho}^{m} \frac{C_m^h C_{L-m}^{m-h}}{C_L^l} \sum_{i=0}^{\rho} \frac{C_{m-h}^{i} C_{L-2m+h}^{l-i}}{\sum\limits_{j=m-\rho}^{m} C_{m-i}^{j} C_{k-m+i}^{m-j}} +$$

$$+ \frac{C_m^\rho C_{L-m}^\rho}{C_L^l} \sum_{i=\rho+1}^{\min(l,m)} [i \leqslant s_d(\varepsilon) + r/2] \frac{C_{m-\rho}^{m-i} C_{L-m-\rho}^{l-i}}{C_i^\rho C_{k-m+i}^\rho}.$$

Сократим вид второго слагаемого с помощью тождества $C_{n-k}^{m-k}/C_n^m = C_m^k/C_n^k$.

Заметим, что

$$\frac{C_{m-\rho}^{m-i}}{C_i^\rho} = \frac{C_{i-(i-m+\rho)}^{\rho-(i-m+\rho)}}{C_i^\rho} = \frac{C_\rho^{i-m+\rho}}{C_i^{i-m+\rho}} = \frac{C_\rho^{m-i}}{C_i^{m-\rho}}.$$

Далее,

$$\frac{C_{L-m-\rho}^{l-i}}{C_{L-m-l+i}^{\rho}} = \frac{C_{L-m-l+i-(i-l+\rho)}^{\rho-(i-l+\rho)}}{C_{L-m-l+i}^{\rho}} = \frac{C_\rho^{l-i}}{C_{L-m-l+i}^{i-l-\rho}}.$$

Теперь воспользуемся аналогичным тождеством $C_n^m C_m^k = C_n^k C_{n-k}^{m-k}$ и получим:

$$C_m^\rho C_\rho^{m-i} = C_m^i C_i^{m-\rho},$$
$$C_{L-m}^\rho C_\rho^{i-l+\rho} = C_{L-m}^{l-i} C_{L-m-l+i}^{\rho-l+i}.$$

Подстановка этих результатов во второе слагаемое дает:

$$\sum_{i=\rho+1}^{\min(l,m)} h_L^{l,m}(i) \, [i - \rho \leq s_d(\varepsilon)] = \sum_{i=\rho+1}^{\lfloor r/2 + s_d(\varepsilon) \rfloor} h_L^{l,m}(i). \tag{3.2}$$

Теперь займемся первым слагаемым. С учетом

$$C_m^h C_{m-h}^i = C_m^{m-h} C_{m-h}^i = C_m^i C_{m-i}^h,$$

$$C_{L-m}^{m-h} C_{L-2m+h}^{l-i} = C_{L-m}^{L-2m+h} C_{L-2m+h}^{l-i} = C_{L-m}^{l-i} C_{k-m+i}^{m-h}.$$

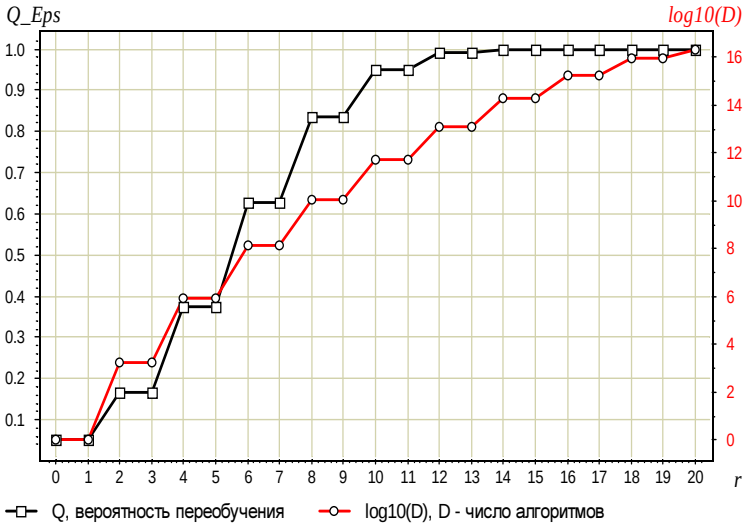


Рис. 3: Зависимость вероятности переобучения Q_ε и $\log_{10}|A|$ от радиуса шара r, при $l = k = 100,\ m = 10,\ \varepsilon = 0.05$.

первое слагаемое переписывается в виде:

$$\big[0 \le s_d(\varepsilon)\big] \sum_{h=m-\rho}^{m} \sum_{i=0}^{\rho} h_L^{l,m}(i) \frac{C_{m-i}^{h} C_{k-m+i}^{m-h}}{\sum\limits_{j=m-\rho}^{m} C_{m-i}^{j} C_{k-m+i}^{m-j}} =$$

$$= \big[0 \le s_d(\varepsilon)\big] \sum_{h=m-\rho}^{m} \sum_{i=0}^{\rho} h_L^{l,m}(i) \frac{h_k^{m,m-i}(h)}{\sum\limits_{j=m-\rho}^{m} h_k^{m,m-i}(j)} =$$

$$= \big[0 \le s_d(\varepsilon)\big] \sum_{i=0}^{\rho} h_L^{l,m}(i) \frac{\sum\limits_{h=m-\rho}^{m} h_k^{m,m-i}(h)}{\sum\limits_{j=m-\rho}^{m} h_k^{m,m-i}(j)} =$$

$$= \big[0 \le s_d(\varepsilon)\big] \sum_{i=0}^{\rho} h_L^{l,m}(i). \qquad (3.3)$$

Подстановка (3.2) и (3.3) в (3.1) завершает доказательство. ■

На рис. 3 представлена зависимость точной оценки вероятности переобучения для слоя шара Q_ε, а также числа алгоритмов в семействе, от радиуса шара r. Видно, что за счёт значительной «плотности» данного семейства вероятность переобучения может оставаться на приемлемо низком уровне при мощности семейства порядка тысяч и относительно небольшой длине выборки $l = k = 100$. Заметим, что VC-оценки в этой ситуации вырождены и существенно превышают единицу.

Теорема 3.2. *Пусть $n(a_0, \mathbb{X}) = m$. Рассмотрим пересечение шара $B_r(a_0)$ с полным*

m-м слоем алгоритмов A_m. Тогда при обучении рандомизированным методом минимизации эмпирического риска и $l > L - m$ вероятность переобучения может быть записана в виде:

$$Q_\varepsilon(\mu, \mathbb{X}) = \begin{cases} H_L^{\ell,m}\big(s_d(\varepsilon) + r/2\big), & s_d(\varepsilon) \geqslant m - k; \\ 0, & s_d(\varepsilon) < m - k, \end{cases}$$

где $s_d(\varepsilon) = \frac{\ell}{L}(m - \varepsilon k)$.

Доказательство этой теоремы полностью повторяет доказательство теоремы 3.1. Отличие случая $k < m$ от предыдущего заключается в новых условиях попадания алгоритмов во множество $A(X)$, описываемых следующими леммами.

Лемма 3.7. $\forall X, \forall a \in A \setminus \mathrm{Orb}(m - \lfloor r/2 \rfloor)$,

$$a \in A(X) \Leftrightarrow n(a, X) = m - k.$$

Лемма 3.8. $\forall a^b \in \mathrm{Orb}(m - \lfloor r/2 \rfloor)$

если $X : |X \cap X^m| \leq \lfloor r/2 \rfloor + m - k$, *то*

$$a^b \in A(X) \Leftrightarrow n\left(a^b, X\right) = m - k;$$

если $X : |X \cap X^m| > \lfloor r/2 \rfloor + m - k$, *то*

$$a^b \in A(X) \Leftrightarrow \begin{cases} n\left(a^b, X \cap X^{L-m}\right) = 0; \\ n\left(a^b, X \cap X^m\right) = |X \cap X^m| - \lfloor r/2 \rfloor. \end{cases}$$

Замечание. Обе полученные оценки не зависят от центра шара a_0, а только от номера слоя m, в котором лежит центр шара.

Следствие 3.9. *Вероятность переобучения семейства, состоящего из одного алгоритма* $a : n(a, \mathbb{X}) = m$, *представляется следующим образом:*

$$Q_\varepsilon(\mu, \mathbb{X}) = H_L^{\ell,m}\big(s_d(\varepsilon)\big)$$

Доказательство. Утверждение следует из теорем 3.1 и 3.2. ■

Следствие 3.10. *Вероятность переобучения семейства A_m, состоящего из всех алгоритмов m-го слоя, равна:*

$$Q_\varepsilon(\mu, \mathbb{X}) = \begin{cases} 1, & \text{при } \varepsilon k \leqslant m \leqslant L - \varepsilon\ell; \\ 0, & \text{иначе.} \end{cases}$$

Доказательство. Утверждение следует из теорем 3.1 и 3.2. ■

4 Эффект сходства и расслоения

В данном разделе исследуется совместное влияние сходства и расслоения на вероятность переобучения.

4.1 Точная оценка вероятности переобучения для хэммингова шара

Пусть семейство алгоритмов A представляет собой хэммингов шар $B_{r_0}(a_0)$, где $n(a_0, \mathbb{X}) = m$. Пусть, без ограничения общности, алгоритм a_0 допускает ошибки на первых m объектах генеральной выборки $\mathbb{X}$.

Получим точную оценку вероятности переобучения для шара. В дальнейшем множество, состоящее из первых m объектов, будем обозначать X^m. Множество, состоящее из последних $L - m$ объектов, будем обозначать X^{L-m}.

Лемма 4.1. *Группа $S_m \times S_{L-m}$, где S_m и S_{L-m} — симметрические группы перестановок, действующих на множествах X^m и X^{L-m} соответственно, является подгруппой группы симметрии семейства алгоритмов A.*

Доказательство. При доказательстве леммы 3.1 было установлено, что действие элементов группы $S_m \times S_{L-m}$ не меняет расстояние до центра шара a_0: для любых $a \in A$ и $\pi \in S_m \times S_{L-m}$ справедливо

$$d(a, a_0) = d(\pi(a), a_0).$$

Тогда, если $a \in B_{r_0}(a_0)$, то и $\pi(a) \in B_{r_0}(a_0)$. Поскольку действие элементов симметрической группы на алгоритмы инъективно, приходим к выводу, что $B_{r_0}(a_0) = = \pi(B_{r_0}(a_0))$. ■

Лемма 4.2. *Орбитами действия группы $S_m \times S_{L-m}$ на множестве алгоритмов A являются пересечения слоев $m - r_0, \ldots, m + r_0$ со сферами радиусов $0, 1, \ldots, r_0$ и центрами в алгоритме a_0.*

Доказательство. Пусть алгоритм $a \in A_p$: $d(a, a_0) = r_1$. Мы установили, что в этом случае $d(\pi(a), a_0) = d(a, a_0) = r_1$. Также мы знаем, что действие перестановки на алгоритм не меняет числа его ошибок на $\mathbb{X}$. Таким образом, $\pi(a)$ также принадлежит пересечению p-го слоя со сферой радиуса r_1.

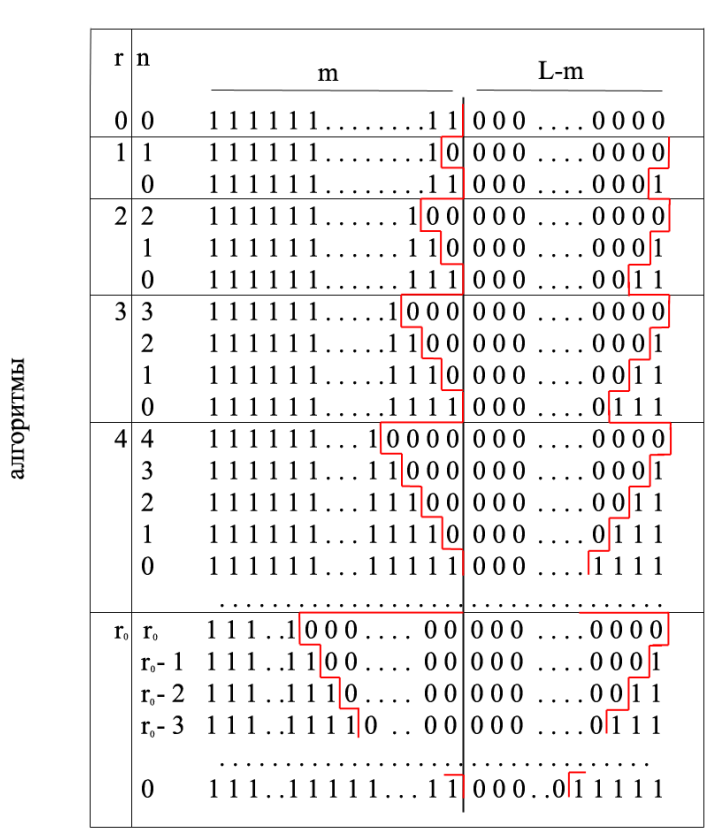


Рис. 4: Алгоритмы из орбит семейства A.

Осталось показать, что для любых $a_1, a_2 \in A_p$ таких, что $d(a_1, a_0) = d(a_2, a_0) = r_1$, найдётся перестановка $\pi \in S_m \times S_{L-m}$, при которой $\pi(a_1) = a_2$. Это уже было доказано в лемме 3.2. ■

На рис. 4 представлено по одному алгоритму из каждой орбиты семейства A. Пронумеруем орбиты двумя целочисленными индексами следующим образом: алгоритмы из орбиты $\mathrm{Orb}(r, n)$, $r = 0, \ldots, r_0$, $n = 0, \ldots, r$, принадлежат пересечению сферы радиуса r с центром в a_0 со слоем $r + m - 2n$. Обратим внимание на то, что алгоритм $a_{(r,n)}$ из орбиты $\mathrm{Orb}(r, n)$ имеет $m - n$ единиц и n нулей на множестве X^m и $r - n$ единиц, $L - m - r + n$ нулей на множестве X^{L-m}. Всего в орбите $\mathrm{Orb}(r, n)$ содержится $C_m^n C_{L-m}^{r-n}$ алгоритмов.

Лемма 4.3. $\forall X, \forall a \in A \setminus \mathrm{Orb}(r_0, r_0)$

$$a \in A(X) \Leftrightarrow n(a, X) = 0.$$

Доказательство. Достаточность очевидна. Докажем необходимость.

Введём обозначения: $X^{\ell_1} = |X^m \cap X|,\ X^{\ell_2} = |X^{L-m} \cap X|,\ \ell_1 = |X^{\ell_1}|,\ \ell_2 = |X^{\ell_2}|,$ $\ell = |X|,\ \ell = \ell_1 + \ell_2,\ X = X^{\ell_1} \cup X^{\ell_2}$.

Пусть $a \in A(X)$ и a принадлежит орбите $\mathrm{Orb}(r, n)$, отличной от $\mathrm{Orb}(r_0, r_0)$. Докажем, что алгоритм a не допускает ошибок на обучающей выборке X.

Начнем с рассмотрения случая $\ell_1 \leqslant r_0$. Алгоритмы орбиты $\mathrm{Orb}(\ell_1, \ell_1)$ имеют ровно ℓ_1 нулей на множестве X^m и не допускают ни одной ошибки на множестве X^{L-m}. Очевидно, существует $a^{\ell_1} \in \mathrm{Orb}(\ell_1, \ell_1) : n\left(a^{\ell_1}, X\right) = 0$.

В случае $\ell_1 > r_0$ в орбите $\mathrm{Orb}(r_0, r_0)$ существует алгоритм $a^{r_0} : n\left(a^{r_0}, X^{L-m}\right) = 0$ и $n\left(a^{r_0}, X^m\right) = \ell_1 - r_0$. Поскольку $n\left(a, X^m\right) \geqslant \ell_1 - n \geqslant \ell_1 - r_0 = n\left(a^{r_0}, X^m\right)$, то мы приходим к противоречию с тем, что $a \in A(X)$. ■

Следствие 4.4. *В ходе доказательства также установлено, что при* $|X \cap X^m| > r_0$ *алгоритмы из орбит, отличных от* $\mathrm{Orb}(r_0, r_0)$*, не могут попасть во множество* $A(X)$*.*

Лемма 4.5. $\forall a \in \mathrm{Orb}(r_0, r_0)$

если $X : |X \cap X^m| \leqslant r_0$*, то*

$$a \in A(X) \Leftrightarrow n(a, X) = 0;$$

если $X : |X \cap X^m| > r_0$*, то*

$$a \in A(X) \Leftrightarrow n\left(a, X \cap X^m\right) = |X \cap X^m| - r_0.$$

Доказательство. Случай $|X \cap X^m| \leqslant r_0$ повторяет первую часть доказательства леммы 4.3. Рассмотрим случай $|X \cap X^m| > r_0$.

Поскольку в этом случае ни один алгоритм из множества A не допускает меньше $|X \cap X^m| - r_0$ ошибок на обучающей выборке X, а алгоритмы из $\mathrm{Orb}(r_0, r_0)$ не ошибаются на алгоритмах множества X^{L-m}, то достаточность очевидна. Необходимость вытекает из того факта, что $\exists a \in \mathrm{Orb}(r_0, r_0) : n(a, X) = n(a, X^m) = |X \cap X^m| - r_0$. ■

Теорема 4.1 (Точная оценка для шара)**.** *Пусть* $n(a_0, \mathbb{X}) = m$*. Рассмотрим шар алгоритмов* $B_{r_0}(a_0)$*. Тогда при обучении рандомизированным методом минимизации эмпирического риска и* $r \leqslant \min(m, L - m)$ *вероятность переобучения может быть записана в следующем виде:*

$$Q_\varepsilon(\mu, \mathbb{X}) = \sum_{i=\max(0, m-k)}^{r_0} h_L^{\ell,m}(i) \frac{\sum\limits_{r_1=0}^{r_0} \sum\limits_{n_1=0}^{r_1} S(n_1, r_1, i)\left[m + r_1 - 2n_1 \geqslant \varepsilon k\right]}{\sum\limits_{r_2=0}^{r_0} \sum\limits_{n_2=0}^{r_2} S(n_2, r_2, i)} + \sum_{i=r_0+1}^{\lfloor s'_d(\varepsilon) \rfloor} h_L^{\ell,m}(i),$$

где $h_L^{\ell,m}(i) = \frac{C_m^i C_{L-m}^{\ell-i}}{C_L^\ell}$, $S(n,r,i) = C_{m-i}^{n-i} C_{k-m+i}^{r-n}$, $s'_d(\varepsilon) = \frac{\ell}{L}(m - \varepsilon k) + \frac{r_0 k}{L}$.

Доказательство. Воспользуемся формулой (2.2) с учетом лемм 4.1, 4.2 и 3.6:

$$Q_\varepsilon(\mu, \mathbb{X}) = \sum_{r=0}^{r_0} \sum_{n=0}^{r} \frac{C_m^n C_{L-m}^{r-n}}{C_L^l} \sum_{i=\max(0,m-k)}^{\min(l,m)} \underbrace{|X : |X \cap X^m| = i, a_{(r,n)} \in A(X)|}_{S(i,r,n)} \frac{\left[\delta\left(a_{(r,n)}, X_i\right) \geqslant \varepsilon\right]}{|A(X_i)|},$$

где $a(r,n)$ — алгоритмы, представленные на рисунке 4, а X_i — произвольное разбиение, в которое входят ровно i объектов из X^m.

Разобьем суммирование по орбитам разбиений (по i) на два слагаемых:

$$Q_\varepsilon(\mu, \mathbb{X}) = \sum_{r=0}^{r_0} \sum_{n=0}^{r} \frac{C_m^n C_{L-m}^{r-n}}{C_L^l} \sum_{i=\max(0,m-k)}^{r_0} S(i,r,n) \frac{\left[\delta\left(a_{(r,n)}, X_i\right) \geqslant \varepsilon\right]}{|A(X_i)|} +$$
$$+ \sum_{r=0}^{r_0} \sum_{n=0}^{r} \frac{C_m^n C_{L-m}^{r-n}}{C_L^l} \sum_{i=r_0+1}^{\min(l,m)} S(i,r,n) \frac{\left[\delta\left(a_{(r,n)}, X_i\right) \geqslant \varepsilon\right]}{|A(X_i)|}.$$

Из лемм 4.3 и 4.5, а также из следствия 4.4 следует, что первое слагаемое соответствует случаю выбора алгоритма, не допускающего ошибок на обучающей выборке X. Следствие 4.4 позволяет опустить суммирование по орбитам во втором слагаемом, поскольку при $i > r_0$ во множество $A(X_i)$ попадают только алгоритмы из $\mathrm{Orb}(r_0, r_0)$, допускающие в соответствии с леммой 4.5 ровно $i - r_0$ ошибок на X. С учетом этого:

$$Q_\varepsilon(\mu, \mathbb{X}) = \sum_{r=0}^{r_0} \sum_{n=0}^{r} \frac{C_m^n C_{L-m}^{r-n}}{C_L^l} \sum_{i=\max(0,m-k)}^{r_0} S(i,r,n) \frac{[r + m - 2n \geqslant \varepsilon k]}{|A(X_i)|} +$$
$$+ \frac{C_m^{r_0} C_{L-m}^0}{C_L^l} \sum_{i=r_0+1}^{\min(l,m)} S(i,r,n) \frac{\left[i \leqslant s_d(\varepsilon) + \frac{r_0 k}{L}\right]}{|A(X_i)|}. \tag{4.1}$$

Вычислим значение $S(i,r,n)$. В случае $i \leqslant r_0$ это число способов выбрать i из n объектов множества X^m и $l - i$ из $L - m - r + n$ объектов множества X^{L-m}, на которых не ошибается алгоритм $a_{(r,n)}$. В случае $i > r_0$ — число способов выбрать $i - r_0$ объектов из множества X^m, на которых алгоритм $a_{(r,n)}$ ошибается, и $l - i$ произвольных объектов из X^{L-m}. Итого:

$$S(i,r,n) = \begin{cases} C_n^i C_{L-m-r+n}^{l-i}, & i \leqslant r_0; \\ C_{m-r_0}^{i-r_0} C_{L-m}^{l-i}, & i > r_0. \end{cases}$$

Найдём значения $|A(X_i)|$. В случае $i > r_0$ в $A(X_i)$ попадают те алгоритмы $\mathrm{Orb}(r_0, r_0)$, которые ошибаются на множестве X^m $i - r_0$ раз. При $i \leqslant r_0$ из каждой орбиты во множество $A(X_i)$ попадают алгоритмы, не ошибающиеся ни на одном

объекте множеств X^m и X^{L-m}. Получаем:

$$|A(X_i)| = \begin{cases} \sum\limits_{r_1=0}^{r_0} \sum\limits_{n_1=0}^{r_1} C_{m-i}^{n_1-i} C_{k-m+i}^{r_1-n_1}, & i \leqslant r_0; \\ C_i^{r_0}, & i > r_0. \end{cases}$$

Подстановка полученных результатов в (4.1) дает:

$$Q_\varepsilon(\mu, \mathbb{X}) = \sum_{r=0}^{r_0} \sum_{n=0}^{r} \frac{C_m^n C_{L-m}^{r-n}}{C_L^l} \sum_{i=\max(0,m-k)}^{r_0} C_n^i C_{L-m-r+n}^{l-i} \frac{[r+m-2n \geqslant \varepsilon k]}{\sum\limits_{r_1=0}^{r_0} \sum\limits_{n_1=0}^{r_1} C_{m-i}^{n_1-i} C_{k-m+i}^{r_1-n_1}} + \\ + \frac{C_m^{r_0}}{C_L^l} \sum_{i=r_0+1}^{\min(l,m)} C_{m-r_0}^{i-r_0} C_{L-m}^{l-i} \frac{\left[i \leqslant s_d(\varepsilon) + \frac{r_0 k}{L}\right]}{C_i^{r_0}}. \tag{4.2}$$

Во втором слагаемом:

$$\frac{C_{m-r_0}^{i-r_0}}{C_i^{r_0}} = \frac{C_{m-r_0}^{m-i}}{C_i^{r_0}} = \frac{C_{i-(i-m+r_0)}^{r_0-(i-m+r_0)}}{C_i^{r_0}} = \frac{C_{r_0}^{i-m+r_0}}{C_i^{i-m+r_0}} = \frac{C_{r_0}^{m-i}}{C_i^{m-r_0}}$$

Далее, $C_m^{r_0} C_{r_0}^{m-i} = C_m^i C_i^{r_0-m+i} = C_m^i C_i^{m-r_0}$. Подстановка полученных формул во второе слагаемое (4.2) дает:

$$\frac{C_m^{r_0}}{C_L^l} \sum_{i=r_0+1}^{\min(l,m)} C_{m-r_0}^{i-r_0} C_{L-m}^{l-i} \frac{\left[i \leqslant s_d(\varepsilon) + \frac{r_0 k}{L}\right]}{C_i^{r_0}} = \\ = \sum_{i=r_0+1}^{\min(l,m)} h_L^{l,m}(i) \left[i \leqslant s_d(\varepsilon) + \frac{r_0 k}{L}\right] = \sum_{i=r_0+1}^{\lfloor s'_d(\varepsilon) \rfloor} h_L^{l,m}(i). \tag{4.3}$$

В первом слагаемом $C_m^n C_n^i = C_m^i C_{m-i}^{n-i}$, а $C_{L-m}^{r-n} C_{L-m-r+n}^{l-i} = C_{L-m}^{L-m-r+n} C_{L-m-r+n}^{l-i} = C_{L-m}^{l-i} C_{k-m+i}^{r-n}$. Заменив порядок суммирования, получим:

$$\sum_{r=0}^{r_0} \sum_{n=0}^{r} \frac{C_m^n C_{L-m}^{r-n}}{C_L^l} \sum_{i=\max(0,m-k)}^{r_0} C_n^i C_{L-m-r+n}^{l-i} \frac{[r+m-2n \geqslant \varepsilon k]}{\sum\limits_{r_1=0}^{r_0} \sum\limits_{n_1=0}^{r_1} C_{m-i}^{n_1-i} C_{k-m+i}^{r_1-n_1}} = \\ = \sum_{i=\max(0,m-k)}^{r_0} h_L^{l,m}(i) \frac{\sum\limits_{r=0}^{r_0} \sum\limits_{n=0}^{r} C_{m-i}^{n-i} C_{k-m+i}^{r-n} [r+m-2n \geqslant \varepsilon k]}{\sum\limits_{r_1=0}^{r_0} \sum\limits_{n_1=0}^{r_1} C_{m-i}^{n_1-i} C_{k-m+i}^{r_1-n_1}} \tag{4.4}$$

Подстановка (4.3) и (4.4) в (4.2) завершает доказательство. ■

Замечание. Полученная оценка снова не зависят от центра шара a_0, и зависит лишь от номера слоя m, в котором лежит центр шара.

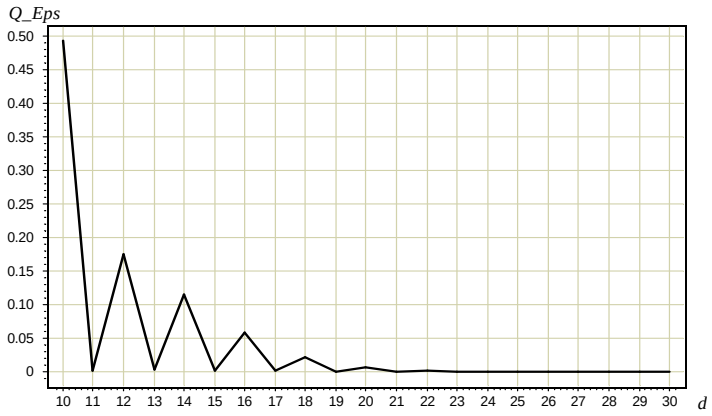


Рис. 5: Вклады слоёв шара в вероятность переобучения при $l = k = 100,\ m = 20,$ $r_0 = 10,\ \varepsilon = 0.05.$

Следствие 4.6. *Пусть* $n(a_1, \mathbb{X}) = n(a_2, \mathbb{X}) = m$ *и* $k \leqslant l$. *Тогда для* $\varepsilon \in \left[0; \frac{m-r}{k}\right]$ *вероятность переобучения семейства* $B_r(a_1)$ *не превосходит вероятность переобучения семейства* $B_r(a_2) \cap A_m$.

Доказательство. Доказательство этого утверждения напрямую следует из теорем 3.1, 3.2 и 4.1:

$$Q_\varepsilon(\mu, \mathbb{X}) \underset{\text{для } B_r(a_1)}{=} \sum_{i=\max(0,m-k)}^{r_0} h_L^{\ell,m}(i) \frac{\sum\limits_{r_1=0}^{r} \sum\limits_{n_1=0}^{r_1} S(n_1, r_1, i) \left[m + r_1 - 2n_1 \geqslant \varepsilon k\right]}{\sum\limits_{r_2=0}^{r} \sum\limits_{n_2=0}^{r_2} S(n_2, r_2, i)} + \sum_{i=r+1}^{\lfloor s'_d(\varepsilon) \rfloor} h_L^{\ell,m}(i) =$$

$$= \sum_{i=\max(0,m-k)}^{r} h_L^{\ell,m}(i) + \sum_{i=r+1}^{\lfloor s'_d(\varepsilon) \rfloor} h_L^{\ell,m}(i) =$$

$$= H_L^{l,m}\left(s_d(\varepsilon) + \tfrac{rk}{L}\right) \leqslant H_L^{l,m}\left(s_d(\varepsilon) + r/2\right) = \underset{\text{для } B_r(a_2) \cap A_m}{Q_\varepsilon(\mu, \mathbb{X})}.$$

■

Замечание. Обратим внимание на то, что мы получили гипергеометрическое распределение из двух слагаемых именно благодаря тому, что $\varepsilon \in \left[0; \frac{m-r}{k}\right]$. При этих ε значение $\lfloor s'_d(\varepsilon) \rfloor \geqslant r$, что и дает возможность «слить» две суммы в одну. В противном случае $\lfloor s'_d(\varepsilon) \rfloor$ строго меньше r, и значение гипергеометрической функции распределения следует брать не в точке $s'_d(\varepsilon)$, а в точке r. Также интересно отметить, что при $\varepsilon \in \left[0; \frac{m-r}{k}\right]$ первое слагаемое в оценке для шара в точности равняется $H_L^{l,m}(r)$.

На рис. 5 представлены точные значения вкладов слоев шара в его вероятность переобучения. Видно, что несколько нижних слоев шара дают большую часть вероят-

ности переобучения. Возникает вопрос: нельзя ли приближать оценку шара оценкой t его нижних слоев.

4.2 Приближение оценки шара t его нижними слоями

Теорема 4.2 (Точная оценка для нижних слоев шара). *Пусть $n(a_0, \mathbb{X}) = m$. Рассмотрим t нижних слоев шара алгоритмов $B_{r_0}(a_0)$. Тогда при обучении рандомизированным методом минимизации эмпирического риска и $r \leqslant \min(m, L-m)$ вероятность переобучения может быть записана в виде:*

$$Q_\varepsilon(\mu, \mathbb{X}) =$$

$$= \sum_{i=\max(0,m-k)}^{r_0} h_L^{\ell,m}(i) \frac{\sum\limits_{r_1=0}^{r_0}\sum\limits_{n_1=0}^{r_1} S'(n_1, r_1, i)\left[m + r_1 - 2n_1 \geqslant \varepsilon k\right]}{\sum\limits_{r_2=0}^{r_0}\sum\limits_{n_2=0}^{r_2} S'(n_2, r_2, i)} + \left[t \geqslant 1\right] \sum_{i=r_0+1}^{\lfloor s'_d(\varepsilon)\rfloor} h_L^{\ell,m}(i),$$

где $h_L^{\ell,m}(i) = \frac{C_m^i C_{L-m}^{\ell-i}}{C_L^\ell}$, $S'(n, r, i) = C_{m-i}^{n-i} C_{k-m+i}^{r-n}\left[m + r - 2n \leqslant m - r_0 + t - 1\right]$, $s'_d(\varepsilon) = \frac{\ell}{L}(m - \varepsilon k) + \frac{r_0 k}{L}$.

Эта теорема доказывается аналогично теореме 4.1. Леммы 4.1, 4.3 и 4.5 остаются справедливы для этого семейства алгоритмов. В лемме 4.2 множество слоев, в пересечении с которыми сферы дают орбиты алгоритмов, меняется на $m-r_0 \ldots m-r_0+t-1$.

Основное отличие в ходе доказательства — при использовании формулы (2.2) в начале доказательства множество суммируемых орбит алгоритмов сокращается добавлением проверочного множителя $[m + r - 2n \leqslant m - r_0 + t - 1]$ после знаков суммирования по индексам r и n.

На рис. 6 представлена зависимость точной оценки вероятности переобучения для t нижних слоев шара от параметра t. Снова видим, что существенные скачки происходят лишь на первых нескольких слоях.

На рис. 7 представлены результаты приближения оценки шара t его нижними слоями. Черным цветом изображена оценка шара. Красным, зеленым и синим — оценки 1, 2 и 3 его нижних слоев соответственно. Падение к нулю оценок первых слоев шара объясняется уменьшением нижнего слоя шара с ростом его радиуса. В определенный момент количество ошибок m, допускаемое алгоритмами нижнего слоя шара, становится меньше εk. В этом случае переобучение невозможно.

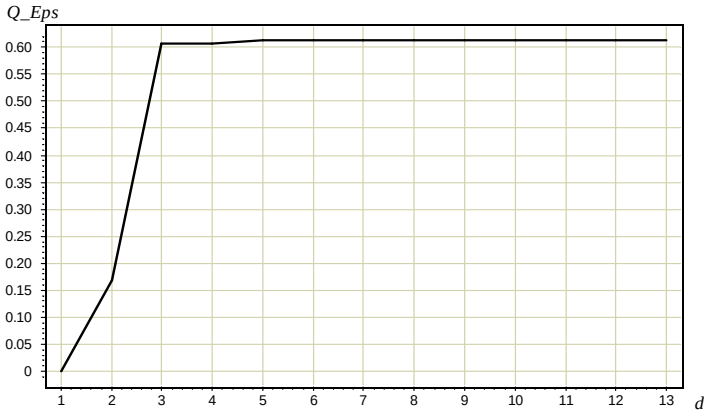


Рис. 6: Зависимость Q_ε от числа t нижних слоев шара, при $l = k = 100,\ m = 10,$ $r_0 = 6,\ \varepsilon = 0.05.$

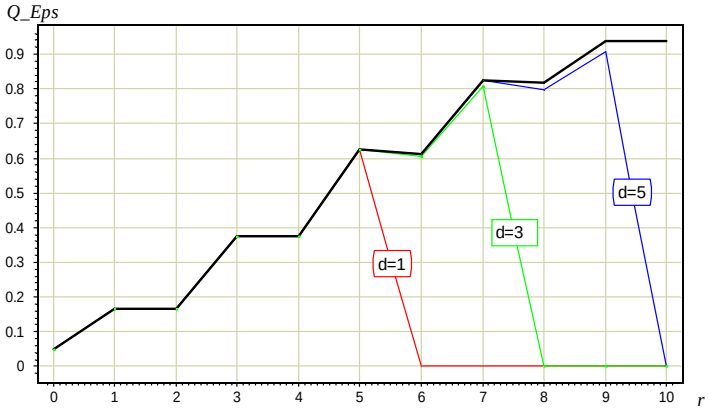


Рис. 7: Зависимость Q_ε от радиуса шара r для полного шара (верхняя кривая) и d его нижних слоёв, при $l = k = 100,\ m = 10,\ r_0 = 6,\ \varepsilon = 0.05.$

5 Задача распознавания вторичной структуры белка

В данной работе влияние эффектов расслоения и сходства алгоритмов на вероятность переобучения исследуется на примере прикладной задачи из области биоинформатики — задачи распознавания вторичной структуры белка по его первичной структуре. В роли алгоритмов выступают *закономерности*. Каждая закономерность — это элементарный классификатор, относящий подпоследовательности первичной структуры белка к одному из видов вторичной структуры. В качестве исходной выборки рассматривается банк данных PDB (Protein Data Bank), содержащий около 130 тысяч записей о белках.

5.1 Структура белка, постановка задачи

Белок — это *макромолекула*, то есть последовательность химически связанных молекул меньшего размера (*аминокислот*). Принято различать четыре уровня структуры белков [13].

- *Первичная структура* — последовательность аминокислот, составляющих данный белок. Образуется аминокислотами двадцати видов, и, соответственно, кодируется словом в двадцатибуквенном алфавите.

- *Вторичная структура* — последовательность локальных конформаций первичной структуры. Участки цепочки аминокислот, образующих первичную структуру, под действием ряда сил принимают один из нескольких видов (в данной работе принимается, что из трех видов) периодической структуры. Таким образом, вторичная структура кодируется словом в трехбуквенном алфавите, той же длины, что и первичная структура.

- *Третичная структура* — трехмерная конструкция «свернутой» вторичной структуры. Третичная структура может задаваться координатами и типами всех атомов, составляющих молекулу белка. В данной работе третичная структура не рассматривается.

- *Четвертичная структура* — совокупность трехмерных структур. В данной работе не рассматривается.

В настоящее время существует несколько способов установления различных структурных уровней белка. Первичная структура белка устанавливается выделе-

нием ДНК клетки и расшифровкой его генома. Вторичную структуру белка, содержащую всю необходимую информацию о его функциях и свойствах, устанавливают по третичной структуре. Ее, в свою очередь, получают непосредственно из белка, выделенного из живой клетки. Установление третичной структуры белка производится методами *ядерного магнитного резонанса (ЯМР)* и рентгено-структурного кристаллографического анализа. Эти методы чрезвычайно трудоемки, применимы не ко всем белкам и имеют погрешности. Поэтому актуальной задачей является *прогнозирование вторичной структуры белка по его первичной структуре.*

Задача прогнозирования вторичной структуры белка заключается в переводе последовательности 20-буквенного алфавита (первичной структуры) в последовательность 3-буквенного алфавита (вторичную структуру) той же длины.

Решение данной задачи представляется возможным благодаря следующей принятой в биологии гипотезе.

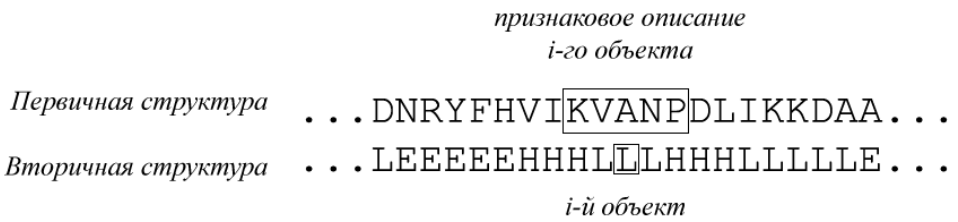
Гипотеза 5.1. *Первичная структура белка однозначно определяет структуру белка на всех последующих уровнях.*

5.2 Виды закономерностей и их переобученность

Для решения задачи распознавания вторичной структуры белка были выбраны логические алгоритмы классификации. В этом разделе рассматриваются метод поиска локальных закономерностей, которые затем будут использоваться для построения классификатора.

Гипотеза 5.2. *Состояние i-й позиции вторичной структуры белка определяется локальной окрестностью i-й позиции первичной структуры белка.*

Из этих соображений было принято решение объектами считать позиции вторичной структуры, их классами — буквы вторичной структуры $Y = \{H, E, L\}$, а признаковыми описаниями — подпоследовательности букв первичной структуры, лежащие в окрестности соответствующей позиции:



Закономерности будем искать среди предикатов, принимающих решение о покрытии объекта на основе его признакового описания.

Предикаты делятся на 6 типов $Y_\Phi = \left\{H, E, L, \bar{H}, \bar{E}, \bar{L}\right\}$: по два тиап предикатов, голосующих *за* класс и *против* одного из трёх классов. Индикатор ошибки предиката φ типа $c \in Y_\Phi$:

$$L\left(\varphi, x_i\right) = \left[\varphi(x_i) = 1 \text{ и } y_i = y\right],$$

если тип закономерностей c голосует против класса y и

$$L\left(\varphi, x_i\right) = \left[\varphi(x_i) = 1 \text{ и } y_i \neq y\right],$$

если тип c голосует за класс y.

Пусть $n(\varphi, X) = \sum_{x \in X} L\left(\varphi, x\right)$ и $p(\varphi, X) = \sum_{x \in X} \left(\varphi(x) - L\left(\varphi, x\right)\right)$. Информативность предиката φ, как функцию величин p и n, можно определять по-разному. В обзоре [9] приводится около 20 различных критериев информативности, используемых в методах поиска логических закономерностей. В данной работе используется критерий информативности, разработанный в теории бустинга для случая линейной композиции логических закономерностей [11]:

$$I(\varphi, X) = \sqrt{p(\varphi, X)} - \sqrt{n(\varphi, X)}. \tag{5.1}$$

Поскольку построение классификатора предполагает отбор наиболее информативных закономерностей, встает вопрос — не ведёт ли оптимизация закономерностей к переобучению.

Определим *переобученность* предиката как разность частоты его ошибок на контрольной и обучающей выборке:

$$\delta\left(\varphi, X, \bar{X}\right) = \frac{1}{k} n\left(\varphi, \bar{X}\right) - \frac{1}{\ell} n(\varphi, X),$$

где $\ell = |X|, k = |\bar{X}|$.

Методом обучения закономерностей типа $c \in Y_\Phi$ будем называть отображение μ_c, которое по обучающей выборке X строит набор закономерностей типа c:

$$\mu_c X = R_c \equiv \left\{\varphi_c^t | t = 1, \ldots, T_c\right\}.$$

В нашем случае метод обучения выделяет закономерности типа c, попавшие в определенным образом отбираемое множество наиболее информативных по критерию (5.1) закономерностей.

Нас будет интересовать, насколько велико значение переобученности закономерностей, выбираемых методом максимизации критерия информативности.

Все описанные эксперименты проводились на базе белков Protein Data Bank, доступной в интернете[1]. В результате слияния данных с совпадающими первичными структурами в базе находилось 39015 различных структур белков. Каждой из них соответствовало во многих случаях несколько различных вторичных структур (вплоть до 300). Это объясняется тем, что в базу попадают результаты экспериментов, проводимых разными лабораториями по различным методикам.

На этапе первичной обработки была проведена кластеризация базы белков. В один кластер сливались белки, первичные структуры которых отличались не более чем на 20% по нормированной метрике Левенштейна. Затем из каждого кластера определенным образом отбиралось по одному белку. Все эксперименты проводились на полученной таким образом базе, состоящей из 14928 пар первичных и вторичных структур, насчитывающей около 3.5 миллионов объектов. База делилась пополам на обучающую и контрольную выборки.

5.2.1 Закономерности-маски

Сначала были изучены предикаты, предложенные в [6]. Предикат φ представляется в виде маски длины d с r фиксированными позициями, в которых стоят буквы алфавита первичной структуры. Предикат φ выделяет объект x в том случае, если маска предиката совпадает с признаковым описанием объекта x. В приведенном выше примере объект выделяется предикатом $\varphi_1 = \{K--NP\}$ и не выделяется предикатом $\varphi_1 = \{KN--P\}$.

Достоинством данного вида закономерностей является простота их формы. Получив информативные закономерности, можно будет не только строить классификатор, но и пытаться понять их биохимический смысл.

Рассматривались предикаты с параметрами $d \in \{5, 7\}$ и $r \in \{3, 4\}$. В большинстве случаев более 95% предикатов каждого типа (всего их $20^r C_d^r$) выделяло хотя бы один объект обучающей выборки. На рисунке 8 приведено распределение предикатов с параметрами $d = 5, r = 3$ по числу покрытых объектов выборки. Всего таких предикатов 80000. Из них лишь около 20 покрывает больше 1000 объектов.

Для всех остальных значений параметров d и r картина не меняется. Во всех

[1]RCSB Protein Data Bank - www.pdb.org

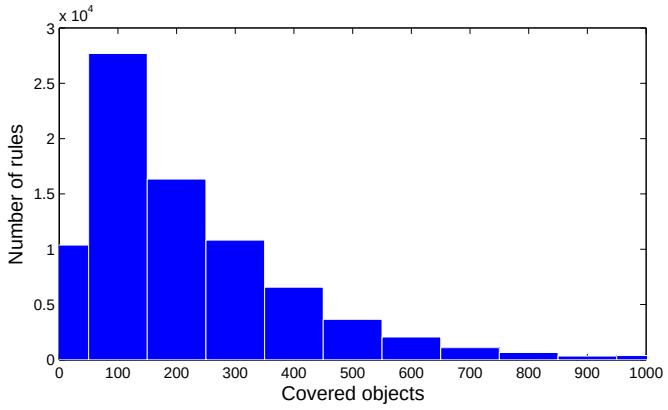


Рис. 8: Распределение предикатов-масок $(r = 3, d = 5)$ по числу выделяемых объектов.

случаях наибольшее число объектов выделяли предикаты, включавшие несколько букв «H» первичной структуры — такие, как $\{--HHH\}$, $\{-HHHH--\}$. Они покрывали около 5000 объектов, что составляет около 0.1% обучающей выборки. Они же одновременно являлись наиболее информативными.

Для выявления эффекта переобучения проводились следующие эксперименты. На обучающей выборке выделялось множество предикатов всех типов $\Phi(X)$. Найденные предикаты сортировались в порядке уменьшения информативности. Затем в каждом из 6-ти типов закономерностей множества $\Phi(X)$ отбиралось по K наиболее информативных. Таким образом, получалось 6 вариационных рядов информативности. Для каждого типа закономерностей вычислялась зависимость среднего значения и 95%-го доверительного интервала переобученности $n\big(\varphi_c^{(i)}, \bar{X}\big) - n\big(\varphi_c^{(i)}, X\big)$ закономерностей от порядкового номера i в вариационном ряду. Заметим, что поскольку в нашем случае $\ell = k$, мы домножали значение переобученности на длину обучающей выборки.

В случае переобученности закономерностей ожидалось значимое повышение начальных участков графиков. Однако в экспериментах этого не наблюдалось. Доверительные интервалы с запасом покрывают ноль, хоть и не симметричны относительно нуля, а средние значения лишь немного поднялись над нулем. В качестве примера на рисунке 9 приведены результаты для закономерностей-масок с параметрами $d = 7, r = 4$, голосующих против класса «L». Сплошной линией изображено среднее значение переобученностей, пунктирной — 95%-ые доверительные интервалы. Все кривые сглажены методом скользящего среднего с окном 100.

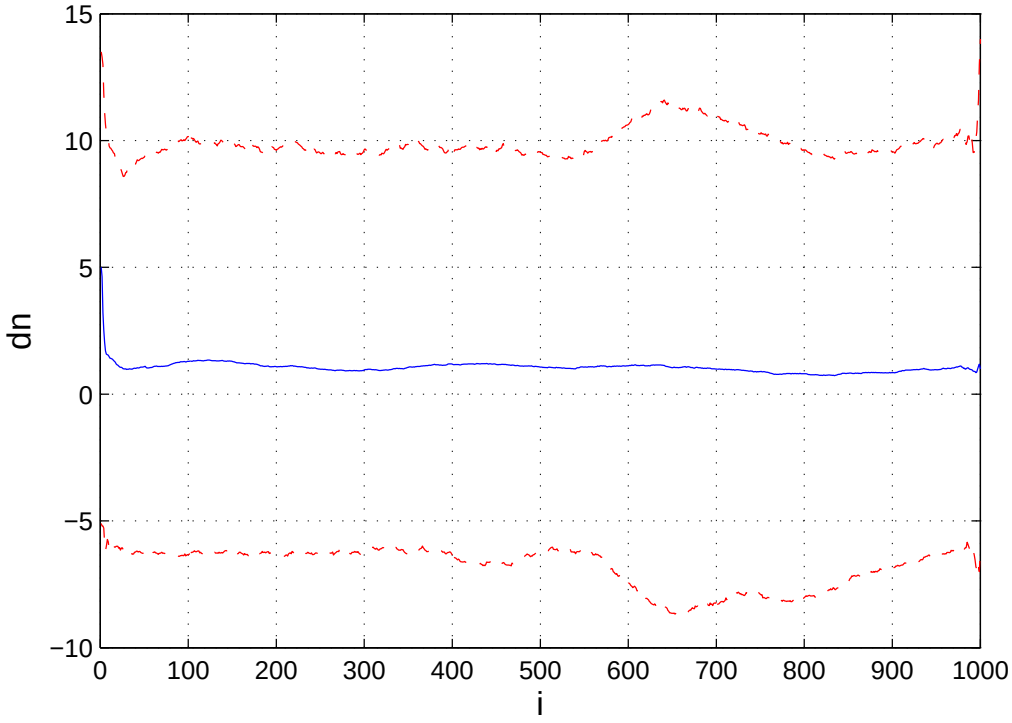


Рис. 9: Значения переобученностей (dn) наиболее информативных закономерностей типа «не L».

На рисунке 10 представлено распределение предикатов по информативности. Графики выровнены по горизонтальной оси, таким образом можно судить о количестве информативных предикатов каждого типа. Эти графики объясняют, почему переобучение практически отсутствует. Отбор закономерностей производится по критерию информативности, относительно которого семейство всех предикатов-масок является сильно расслоенным. Слои, содержащие наиболее информативные закономерности, как правило, не многочисленны; их число на порядки меньше длины выборки. В то же время, низкая переобученность закономерностей ещё не гарантирует построение надежного классификатора, поскольку высокоинформативных закономерностей, допускающих малую долю ошибок, явно не хватает для покрытия всей выборки.

На рисунке 11 представлены зависимости доли покрытий генеральной выборки от порога информативности I_0 отдельно для каждого типа закономерностей. На рисунке 12 представлено распределение 10% наиболее информативных предикатов по значениям n и p на обучающей выборке. Более темный цвет точек соответствует бóльшему значению информативности предикатов.

На графиках видно, что для покрытия 95% всей выборки необходимо взять порядка трети всех предикатов. Учитывая большое число «плохих» закономерностей, выделяющих мало объектов, возможность построения хорошего классификатора на основе закономерностей-масок с параметрами $d = 5, r = 3$ стоит под вопросом. Для остальных рассмотренных значений параметров d и r ситуация аналогичная.

Также предварительные эксперименты показали, что классификаторы, построенные на основе закономерностей-масок, переобучены.

Возникает вопрос — нельзя ли придумать простое правило объединения описанных предикатов в новые, удовлетворяющее следующим условиям:

- получающиеся закономерности должны по-прежнему иметь простой и понятный вид;
- новые закономерности должны выделять достаточно большое число объектов;
- при объединении информативность закономерностей должна увеличиваться в большинстве случаев.

5.2.2 Закономерности-подмножества

Рассмотрим закономерности другого вида. Закономерность φ задается r буквами алфавита первичной структуры и длиной окрестности d. Закономерность φ покрывает объект x, если в признаковом описании объекта x длины d встречается каждая из r букв предиката с учетом повторений. На примере приведенного ранее участка белка объект x_i покрывается предикатом $\varphi = (\{K, N\}, 5)$, но не покрывается предикатом $\varphi = (\{N, N\}, 5)$, где 5 — длина окрестности.

Из определения следует, что предикат-подмножество может быть получен путём слияния всех предикатов-масок, имеющих одинаковый набор букв и ту же длину окрестности d. Например, предикаты-маски $\{AB - C\}$, $\{C - BA\}$ и прочие, всего $4! = 24$ штук, сливаются в предикат-подмножество $\varphi = (\{A, B, C\}, 4)$.

Эксперименты показали, что слияние предикатов-масок в предикаты-подмножества в большинстве случаев повышают информативность закономерностей. Были рассмотрены предикаты с $d \in \{5, 7\}$ и $r \in \{3, 4\}$. На обучающей выборке снова выделялись практически все предикаты (их всего C^r_{r+19} — общее количество наборов при выборе r элементов из 20 с возвращением и без учета порядка.).

На рисунке 13 показано распределение числа предикатов-подмножеств с пара-

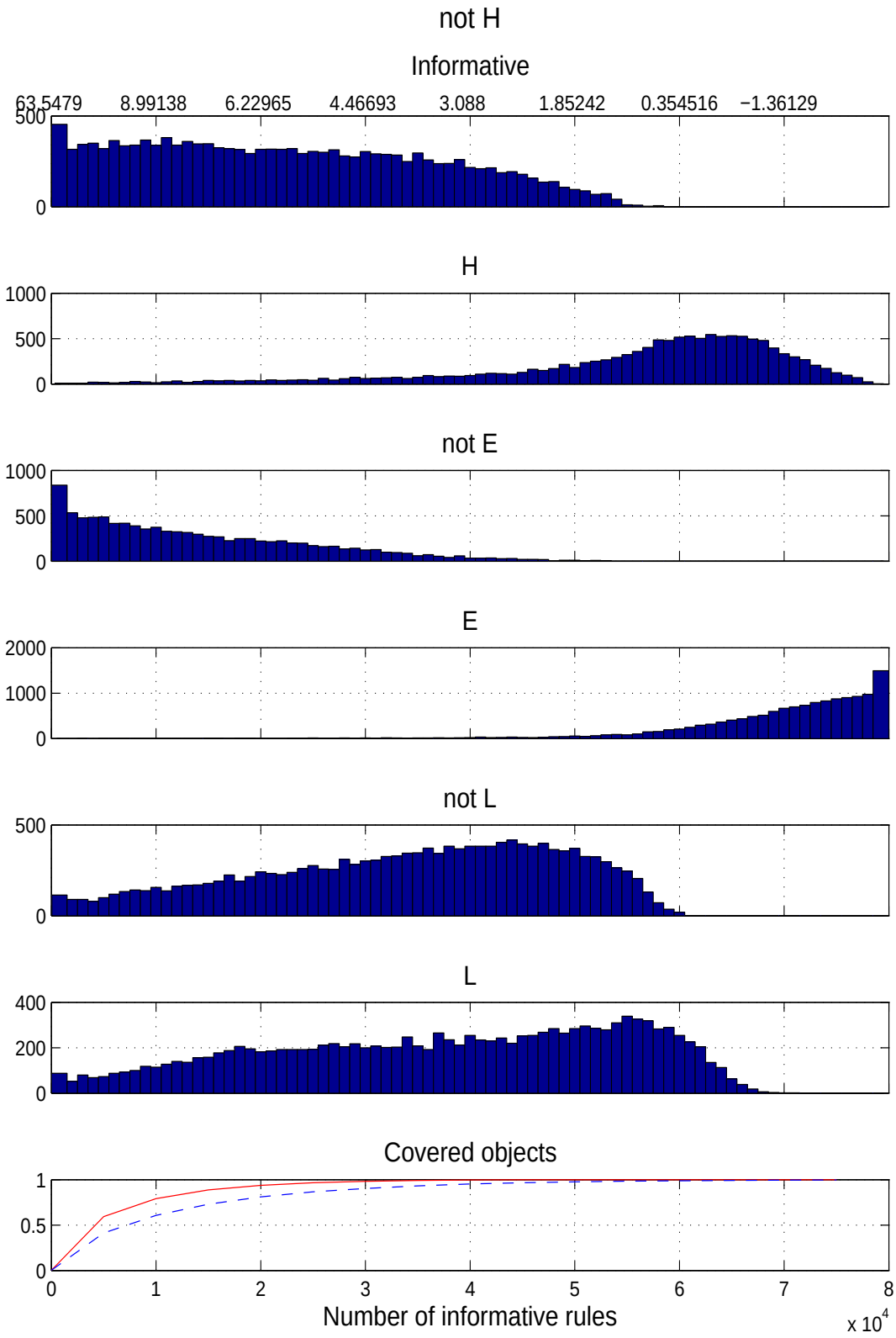


Рис. 10: *вверху* — распределение предикатов-масок ($d = 5, r = 3$) по информативности; *внизу* — зависимость доли покрытия генеральной выборки от числа наиболее информативных предикатов, пунктир — доля правильно выделенных объектов.

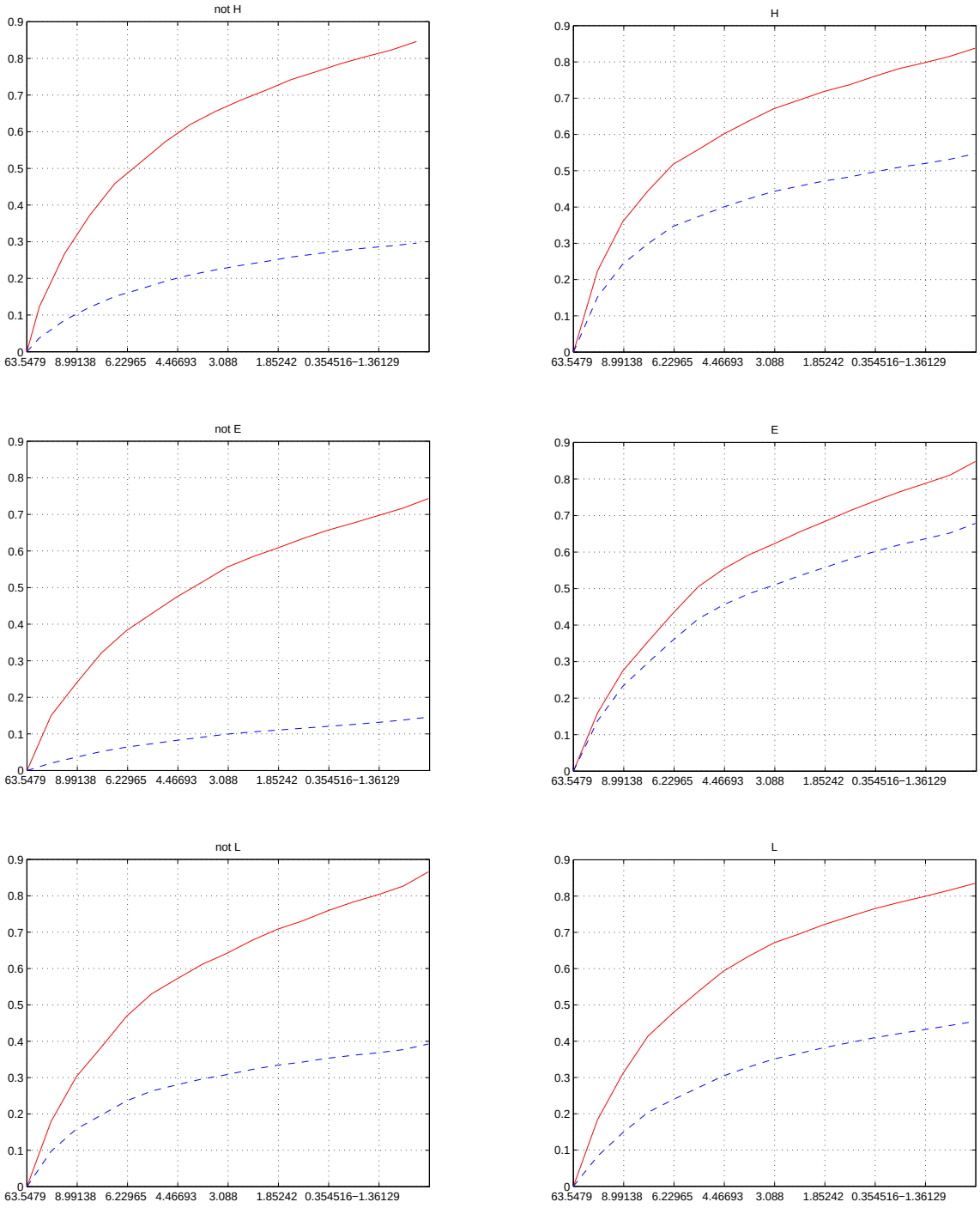


Рис. 11: Зависимости доли выделенных объектов и ошибочно выделенных объектов (пунктир) от порога информативности I_0 для предикатов-масок $(d = 5, r = 3)$.

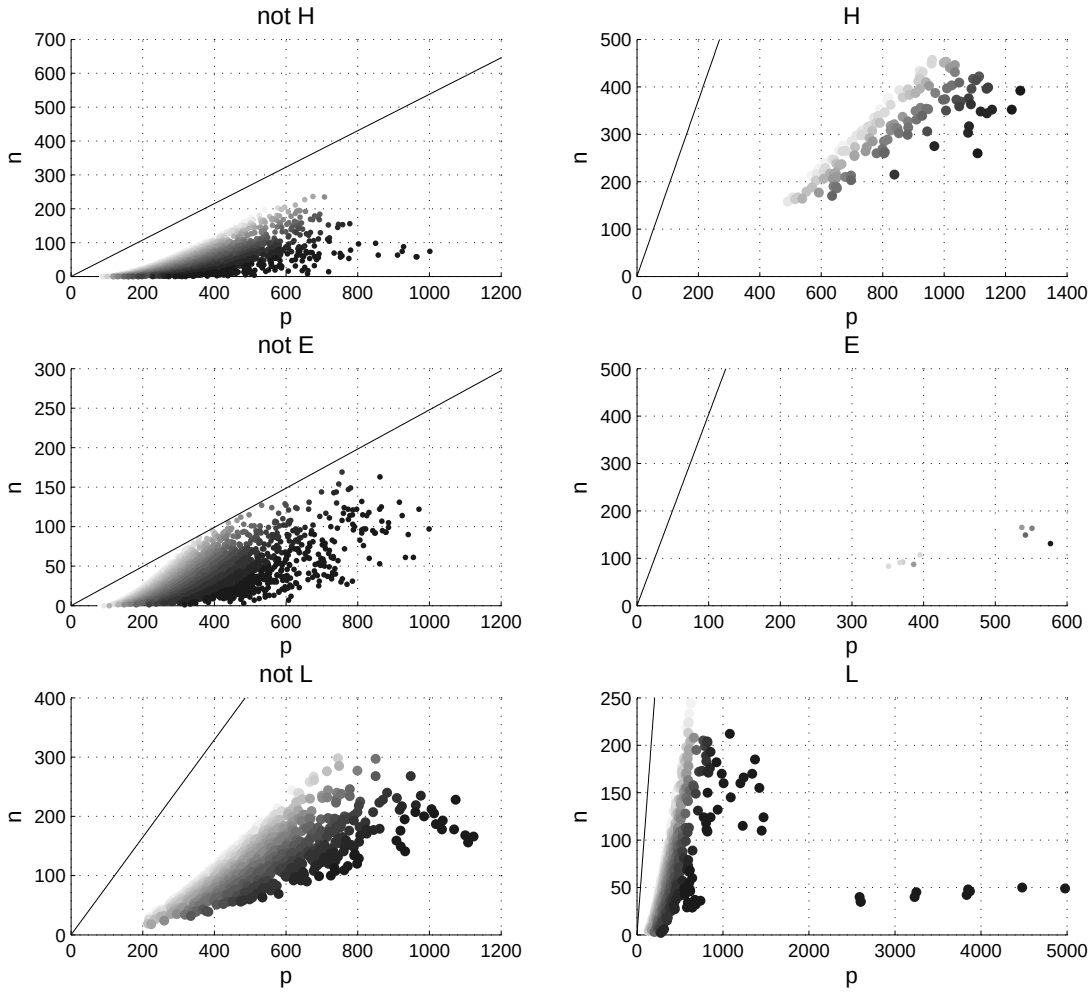


Рис. 12: Распределение первых 10% наиболее информативных предикатов-масок ($d = 5, r = 3$) по значениям $p(\varphi, X)$ и $n(\varphi, X)$.

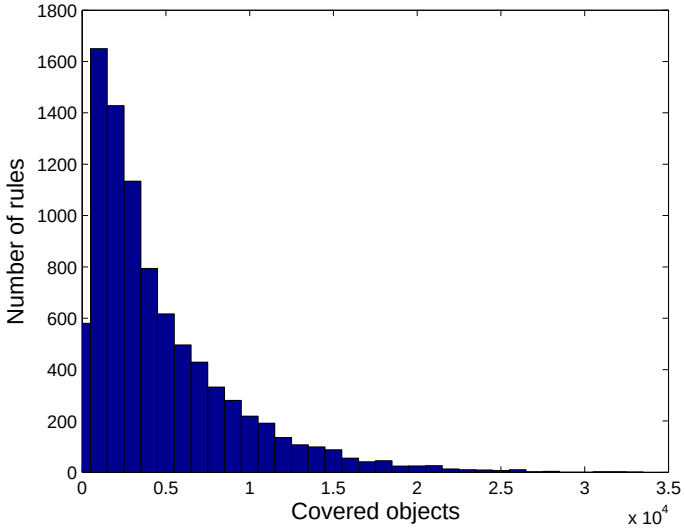


Рис. 13: Распределение предикатов-подмножеств $(r = 4, d = 7)$ по числу выделяемых объектов.

метрами $r = 4$ и $d = 7$ по числу покрываемых объектов генеральной выборки. Видно, что число покрываемых предикатами объектов существенно выросло.

Для выявления переобучения новых закономерностей был повторен эксперимент, описанный в прошлом разделе. Результаты опять не показали видимого подъема переобученности наиболее информативных закономерностей. На рисунке 14 представлена зависимость значения $n\big(\varphi_c^{(i)}, \bar{X}\big) - n\big(\varphi_c^{(i)}, X\big)$ для типа предикатов-подмножеств $(r = 4, d = 7)$, голосующих против «L». Пунктиром изображен 95%-й доверительный интервал, сплошной линией — среднее значение. Все кривые были сглажены методом скользящего среднего с окном 100.

На рисунках 15, 16 и 17 представлены графики для предикатов-подмножеств $(r = 4, d = 7)$, аналогичные приведенным в предыдущем разделе. Вместе с числом ошибок предикатов возросло и число покрываемых ими объектов: теперь для покрытия 95% генеральной выборки надо брать около четверти всех закономерностей. В предыдущем случае мы были вынуждены для обеспечения покрытия выборки искать информативные закономерности среди множества предикатов, большая часть которых были статистически ненадежными.

С помощью предикатов-подмножеств были сделаны первые попытки построения алгоритма простого голосования. Порог информативности отбора закономерностей настраивался из соображений покрытия выборки: находилось наименьшее значение информативности I_0, при котором предикаты с $I(\varphi, X) > I_0$ покрывали не менее

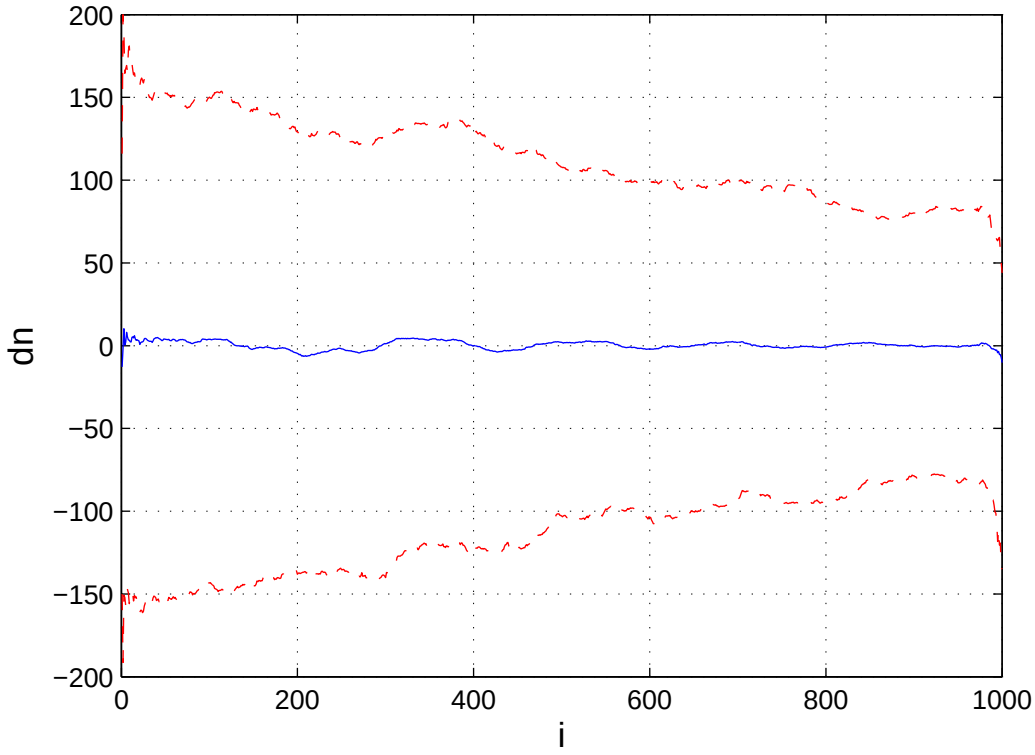


Рис. 14: Значения переобученностей (dn) наиболее информативных закономерностей типа «не L».

95% обучающей выборки. Затем для объектов обучающей выборки подсчитывались суммы голосов каждого из 6 типов отобранных закономерностей. Настройка весовых коэффициентов полученных значений проводилась двумя независимыми образами.

Сначала грубо сравнивались доли голосовавших правил, при этом доли закономерностей, голосовавших против класса, брались с отрицательным знаком. Этот метод дал крайне плохие результаты: среднее значение ошибки на обучении $\sim 43\%$, что почти совпадало со значением ошибки и на контрольной выборке. С учетом малого числа хороших закономерностей, голосовавших за и против класса «E», были получены средние значения ошибок описанного алгоритма классификации на множестве объектов выборки, не принадлежавших классу «E». Для обучающей и контрольной выборки ошибка составила порядка 30%.

Также были предприняты попытки настройки весовых коэффициентов с помощью метода опорных векторов. Задача классификации на 3 класса с количеством голосов предикатов 6-ти типов в качестве признаков решалась с помощью многоклассового модуля программной реализации SVM^{light} [10], свободно доступной в

сети. Метод опорных векторов дал те же результаты: около 42% ошибок на всем обучении и контроле и порядка 30% ошибок на объектах, не принадлежавших классу «E». В дальнейшем планируется проведение рядя экспериментов по оптимизации параметров метода опорных векторов.

5.3 Основные выводы

- Какие из рассмотренных закономерностей лучше подходят для построения классификатора решать рано. Этот вопрос требует поиска новых видов закономерностей, более детального изучения возможных критериев отбора информативных закономерностей и дальнейших экспериментов по построению классификатора.

- Отсутствие переобучения у закономерностей, отобранных по критерию информативности 5.1, объясняется малым количеством информативных алгоритмов и сильным расслоением семейства предикатов по значению информативности.

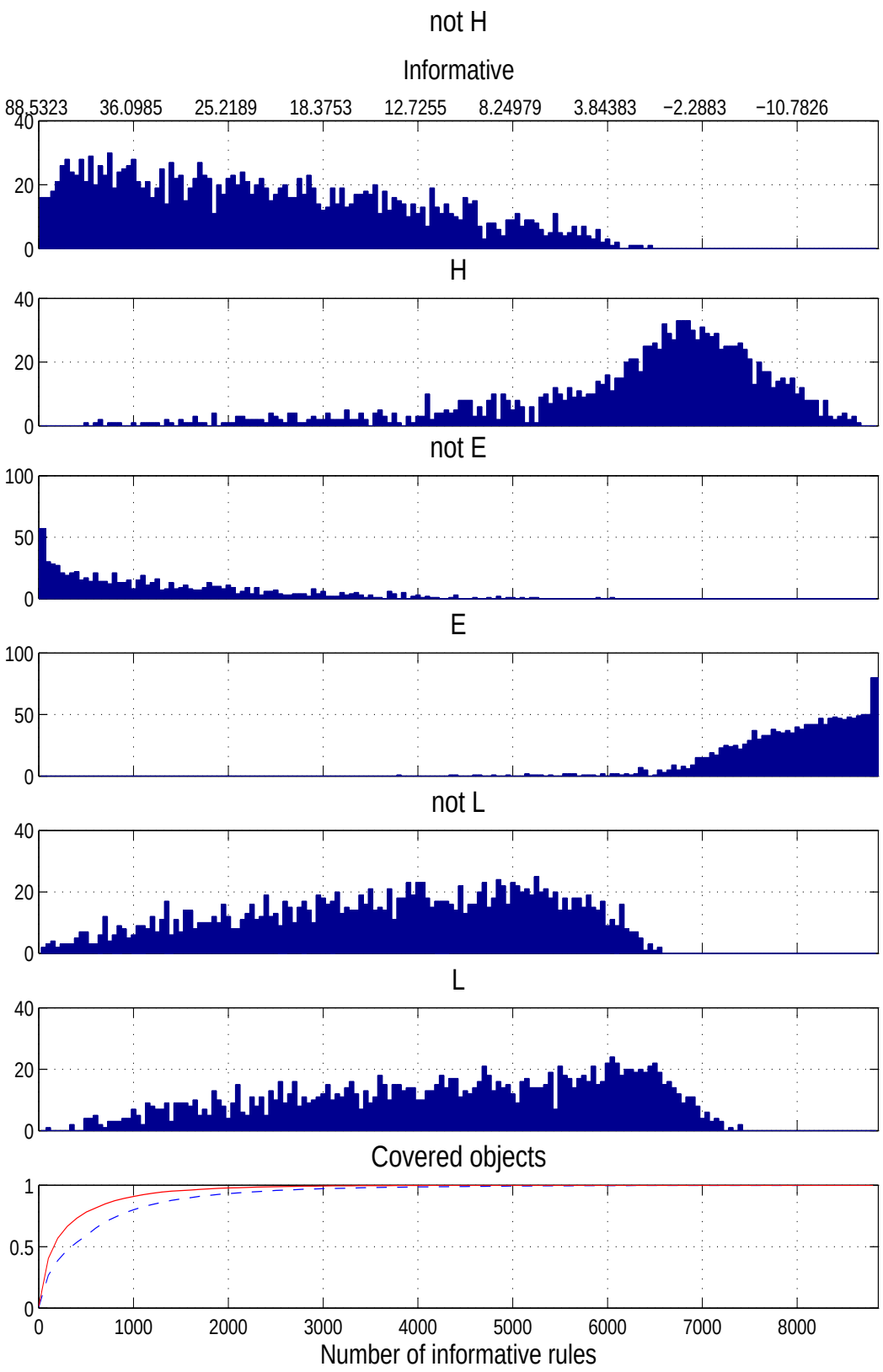


Рис. 15: *Вверху* — распределение предикатов-подмножеств ($d = 7, r = 4$) по информативности. *Внизу* — зависимость доли покрытия генеральной выборки от числа наиболее информативных предикатов, пунктир — доля правильно выделенных объектов.

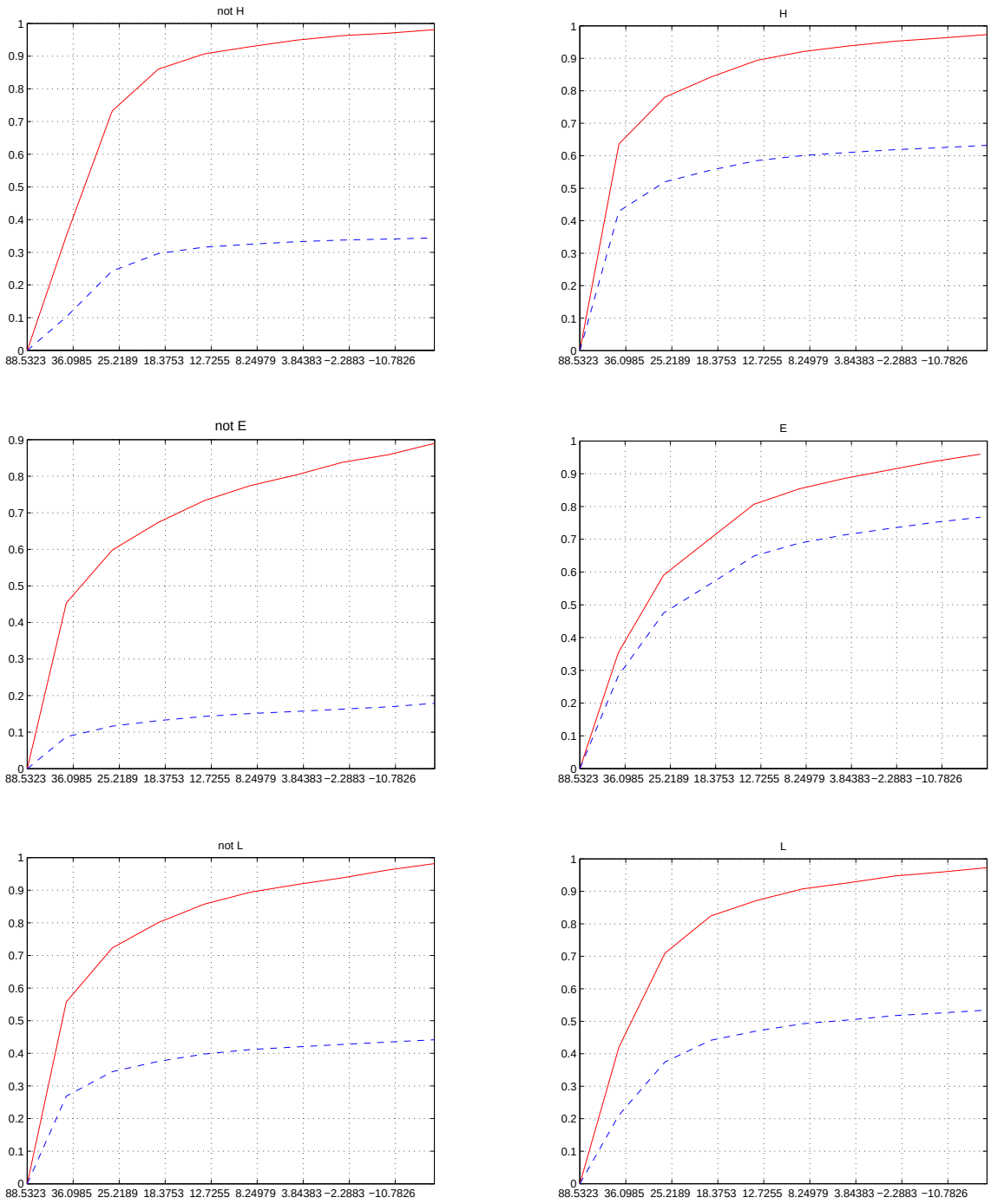


Рис. 16: Зависимости доли выделенных объектов и ошибочно выделенных объектов (пунктир) от порога информативности I_0 для предикатов-подмножеств ($d = 7, r = 4$).

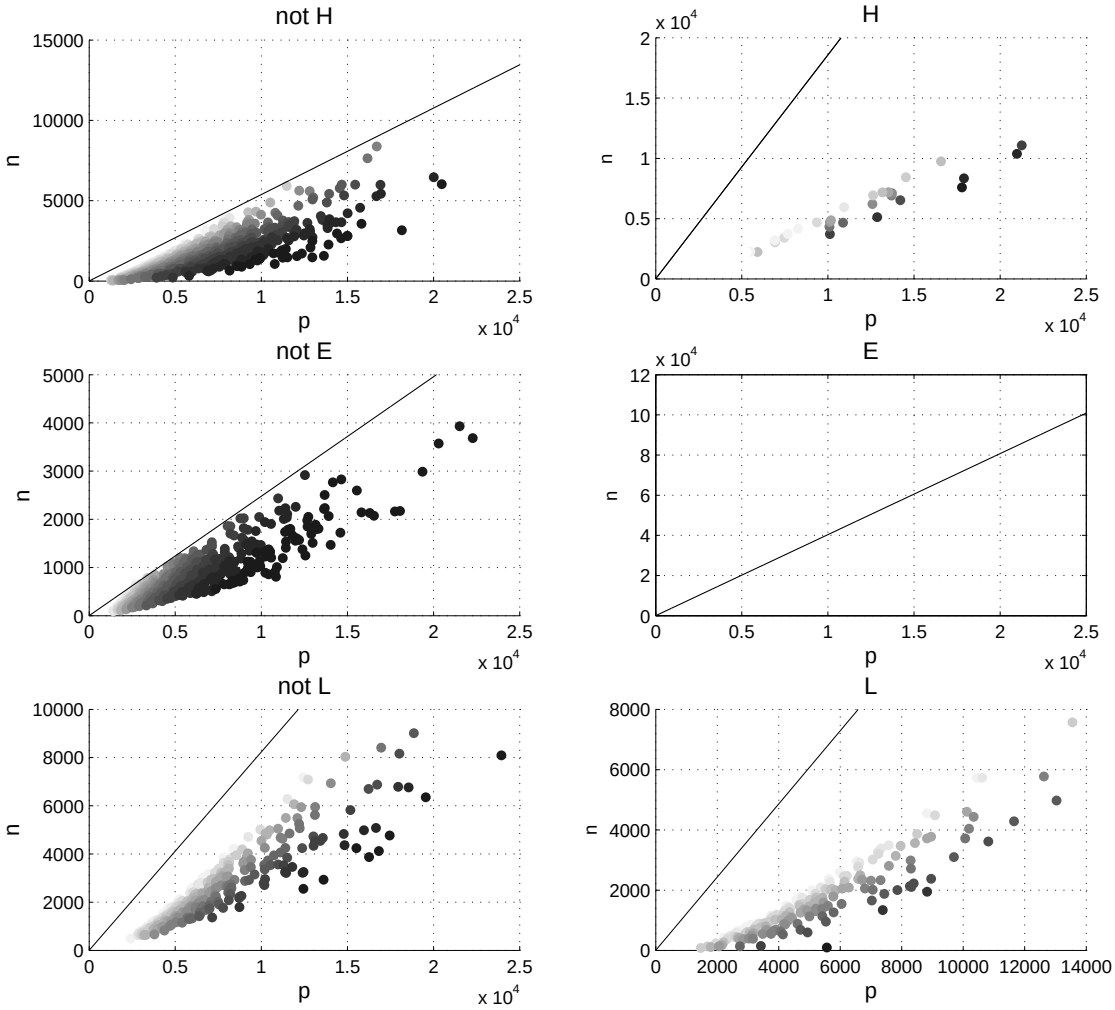


Рис. 17: Распределение первых 20% наиболее информативных предикатов-подмножеств ($d = 7, r = 4$) по значениям $p(\varphi, X)$ и $n(\varphi, X)$.

6 Заключение

В качестве основных результатов, полученных в данной работе, можно отметить следующие:

- получено обобщение теоретико-группового подхода, предложенного в [7], упрощающее получение точных оценок вероятности переобучения для некоторых семейств алгоритмов;
- получены точные оценки вероятности переобучения для трех модельных семейств — шара алгоритмов, t нижних слоев шара и пересечения шара со слоем алгоритмов;
- показано, что учёт сходства алгоритмов существенно улучшает точность оценок вероятности переобучения;
- показано, что вероятность переобучения расслоенных семейств возможно аппроксимировать несколькими нижними слоями;
- в задаче распознавания вторичной структуры белка показано, что рассмотренные закономерности практически не переобучаются.

В дальнейшем планируется продолжение расширения класса модельных семейств, для которых возможно получение точных оценок вероятности переобучения. Планируется подробно изучить эффект, описанный в разделе 3.1, связанный с достаточностью рассмотрения небольшого числа алгоритмов семейства для приближения его вероятности переобучения. Также будет продолжено исследование возможности приближения вероятности переобучения семейства несколькими его нижними слоями.

Список литературы

[1] *Вапник В. Н., Червоненкис А. Я.* О равномерной сходимости частот появления событий к их вероятностям // *Теория вероятностей и её применения.* — 1971. — Т. 16, № 2. — С. 264–280.

[2] *Вапник В. Н., Червоненкис А. Я.* Теория распознавания образов. — М.: Наука, 1974.

[3] *Воронцов, К. В.* Комбинаторная теория надёжности обучения по прецедентам: Дис. док. физ.-мат. наук: 05-13-17 — Вычислительный центр РАН — 2010. http://www.machinelearning.ru/wiki/images/b/b6/Voron10doct.pdf

[4] *Воронцов К. В.* Точные оценки вероятности переобучения // Доклады РАН, 2009. — Т. 429, № 1. — С. 15–18.

[5] *Воронцов К. В.* Комбинаторный подход к оценке качества обучаемых алгоритмов // Математические вопросы кибернетики. Под редакцией О. Б. Лупанов. — М.: Физматлит, 2004. — Т. 13 — С. 5–136.

[6] *Рудаков К. В., Торшин И. Ю.* Вопросы разрешимости задачи распознавания вторичной структуры белка. — Информатика и ее применения, 2010. — Т. 4, Вып. 2.

[7] *Фрей А. И.* Точные оценки вероятности переобучения для симметричных семейств алгоритмов // Математические методы распознавания образов: 14-ая Всеросс. конф.: Докл. М.: МАКС Пресс, 2009. — С. 66–69.

[8] *Bax E. T.* Similar classifiers and VC error bounds: Tech. Rep. CalTech CS TR97 14: — 1997.

[9] *Furnkranz J., Flach P. A.* Roc 'n' rule learning-towards a better understanding of covering algorithms // Machine Learning. — 2005. — Vol. 58, no. 1. — Pp. 39–77.

[10] *Joachims T.* Making large-Scale SVM Learning Practical. Advances in Kernel Methods — Support Vector Learning. — 1999, Pp. 169–184.

[11] *Schapire R. E., Singer Y.* Improved boosting using confidence-rated predictions // Machine Learning. — 1999. — Vol. 37, no. 3. — Pp. 297–336.

[12] *Skala M.* Hypergeometric tail inequalities: ending the insanity, 2009. http://ansuz.sooke.bc.ca/professional/hypergeometric.pdf

[13] *Torshin I. Y.* Bioinformatics in the Post-Genomic Era: The Role of Biophysics, Nova Biomedical Books, NY, 2006, ISBN: 1-60021-048

[14] *Vapnik V.* Statistical Learning Theory. — Wiley, New York — 1998.