

# Комбинаторная теория надёжности обучения по прецедентам

Воронцов Константин Вячеславович

Диссертация на соискание ученой степени  
доктора физико-математических наук  
05.13.17 — теоретические основы информатики

Научный консультант — чл.-корр РАН К. В. Рудаков

ВЦ РАН, 22 апреля 2010

## Содержание

- 1 Проблема обобщающей способности**
  - Завышенность оценок обобщающей способности
  - Цели и методика исследования
  - Постановка задачи
- 2 Слабая вероятностная аксиоматика**
  - Основная аксиома
  - Закон больших чисел в слабой аксиоматике
  - VC-оценки в слабой аксиоматике
- 3 Измерение факторов завышенности VC-оценок**
  - Эксперимент на реальных данных
  - Эксперимент с цепочками алгоритмов
  - Эксперимент с парой алгоритмов
- 4 Точные оценки обобщающей способности**
  - Порождающие и запрещающие множества
  - Модельные семейства алгоритмов
  - Эффекты расслоения и связности
  - Оценки полного скользящего контроля

## Задача обучения по прецедентам

### Актуальность

Для повышения качества алгоритмов классификации и прогнозирования необходимо иметь достаточно точные оценки обобщающей способности.

### Проблема, требующая решения

Известные оценки сильно завышены.

### Цель работы

Разработка нового математического аппарата для получения точных (не завышенных, не асимптотических) оценок обобщающей способности.

## Теория Вапника-Червоненкиса (VC-теория)

Пусть  $A$  — конечное множество (семейство) алгоритмов;

$P(a)$  — вероятность ошибки алгоритма  $a \in A$ ;

$\nu(a, X)$  — частота ошибок алгоритма  $a$  на выборке  $X$  длины  $\ell$ ;

$\mu X$  — алгоритм, обученный методом  $\mu$  по выборке  $X$ ;

### Теорема (Вапник и Червоненкис, 1971)

Для любых  $P, \mu$ , конечного  $A$  и  $\varepsilon \in (0, 1)$

$$P_X \left[ P(\mu X) - \nu(\mu X, X) \geq \varepsilon \right] \leq$$

$$\stackrel{(1)}{\leq} P_X \left[ \max_{a \in A} (P(a) - \nu(a, X)) \geq \varepsilon \right] \leq$$

$$\stackrel{(2)}{\leq} |A| \cdot P_X \left[ P(a) - \nu(a, X) \geq \varepsilon \right] \stackrel{(3)}{\leq} |A| \cdot \frac{3}{2} e^{-\frac{\varepsilon \ell^2}{4}}.$$

## Проблема сильной завышенности VC-оценки

Требуемая длина обучающей выборки при заданных точности  $\varepsilon$ , надёжности  $\eta = 0.01$  и ёмкости  $h = VCdim(A)$

$h$	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 0.2$
0	35900	1440	360	91
2	259300	7619	1600	316
5	632633	18260	3770	741
10	1262928	36396	7521	1470
20	2531001	72918	15069	2936
50	6348132	182980	37821	7381
100	7373100	295440	73821	14811

### Вывод

На практике достаточно на порядки меньшего числа объектов.

## Причины завышенности VC-оценки

### Основная причина — чрезмерная общность:

- не учитываются свойства конкретного метода обучения  $\mu$ ;
- не учитываются свойства конкретной выборки  $X$ ;
- VC-оценка — это оценка «худшего случая».

### Верхние оценки по ходу доказательства:

- 1 принцип равномерной сходимости;
- 2 неравенство Буля (union bound);
- 3 неравенство Хёффдинга (Бернштейна, Чернова, и т.п.).

## Развитие VC-теории, 1968–2009

- Оценки равномерной сходимости [Вапник, Червоненкис, 1968]
- Theory of learnable (PAC-learning) [Valiant, 1982]
- Data-dependent bounds [Haussler, 1992]
- Concentration inequalities [Talagrand, 1995]
- Connected function classes [Sill, 1995]
- Similar classifiers VC bounds [Bax, 1997]
- Margin based bounds [Bartlett, 1998]
- Self-bounding learning algorithms [Freund, 1998]
- Rademacher complexity [Koltchinskii, 1998]
- Adaptive microchoice bounds [Langford, Blum, 2001]
- Algorithmic stability [Bousquet, Elisseeff, 2002]
- Algorithmic luckiness [Herbrich, Williamson, 2002]
- Shell bounds [Langford, 2002]
- PAC-Bayes bounds [McAllester, 1999; Langford, 2005]

## Проблема завышенности остаётся

- Ни одна из теорий не устраняет *всех* причин завышенности;
- Ни одна из теорий не даёт *точных* оценок вероятности переобучения;
- Большинство теорий основаны на принципе равномерной сходимости.

Подробный обзор теорий обобщающей способности представлен в Главе 2 диссертации.



## Цели и методика исследования

### Цель диссертационной работы

Создание нового математического аппарата для получения точных оценок вероятности переобучения.

### Основные этапы исследования

- 1 Введение слабой вероятностной аксиоматики (Глава 1).
- 2 Экспериментальное измерение факторов завышенности и понимание причин завышенности VC-оценок (Глава 3).
- 3 Разработка общих методов получения точных оценок и исследование модельных частных случаев (Глава 4).
- 4 Применение — создание новых методов обучения (Глава 5).

## Чем не устраивает классический вероятностный подход

Вероятность большого отклонения частоты  $\nu$  от вероятности  $P$ :

$$P_\varepsilon = P_X\{P - \nu(X) > \varepsilon\}.$$

### Недостатки

- Невозможно измерить вероятность  $P_\varepsilon$  как частоту, если вероятность  $P$  неизвестна.
- Невозможно получить точное выражение для  $P_\varepsilon$ ; оценки  $P_\varepsilon$  либо завышенные, либо асимптотические.

Причина недостатков — вероятность  $P$  инфинитарна.

Предлагается оценивать частоту  $\nu(\bar{X})$  на скрытой выборке  $\bar{X}$ :

$$Q_\varepsilon = P_{X, \bar{X}}\{\nu(\bar{X}) - \nu(X) > \varepsilon\}.$$

## Задача обучения по прецедентам

Объекты  $\mathbb{X} = \{x_1, \dots, x_L\}$ ; алгоритмы  $A = \{a_1, \dots, a_D\}$ ;

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

$\vec{a}(\mathbb{X}) = (I(a, x_i))_{i=1}^L$  — вектор ошибок алгоритма  $a \in A$ ;

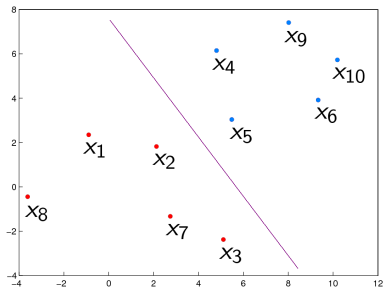
$n(a, X)$  — число ошибок алгоритма  $a \in A$  на выборке  $X \subset \mathbb{X}$ ;

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$\dots$	$a_D$	
$x_1$	1	1	0	0	0	1	$\dots$	1	X — наблюдаемая (обучающая) выборка длины $\ell$
$\dots$	0	0	0	0	1	1	$\dots$	1	
$x_\ell$	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 $k = L - \ell$
$\dots$	0	0	0	1	0	0	$\dots$	1	
$x_L$	0	1	1	1	1	1	$\dots$	0	

### Задача обучения по прецедентам:

Зная только  $X$ , выбрать алгоритм с малым  $n(a, \mathbb{X})$ .

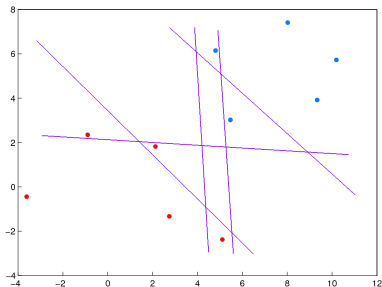
## Пример. Матрица ошибок линейных классификаторов



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

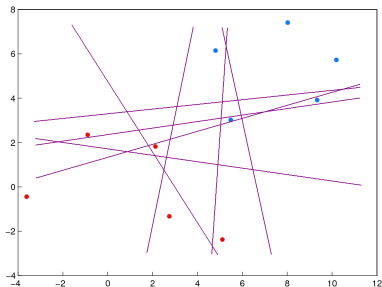
## Пример. Матрица ошибок линейных классификаторов



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

Допустим, что для решения задачи обучения по прецедентам выбран некоторый *метод обучения*  $\mu: \mathbb{X}^\ell \rightarrow A$ .

**Пример** — метод минимизации эмпирического риска:

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

**Опр.** *Переобученность* алгоритма  $a$  на разбиении  $X \sqcup \bar{X} = \mathbb{X}$ :

$$\delta(a, X, \bar{X}) = \frac{1}{k} n(a, \bar{X}) - \frac{1}{\ell} n(a, X).$$

**Опр.** *Переобучение* — это событие  $[\delta(\mu X, X, \bar{X}) \geq \varepsilon]$ .

**Задача** — оценить *вероятность переобучения*:

$$Q_\varepsilon(\mu, \mathbb{X}) = P_{X, \bar{X}} [\delta(\mu X, X, \bar{X}) \geq \varepsilon] \leq \eta(\varepsilon) \quad ?$$

## Слабая вероятностная аксиоматика: основная аксиома

Пусть  $\mathbb{X} = \{x_1, \dots, x_L\}$  — конечное множество объектов.

### Аксиома (единственное вероятностное допущение)

Все  $C_L^\ell$  разбиений  $X \sqcup \bar{X} = \mathbb{X}$  равновероятны, где

$X$  — наблюдаемая обучающая выборка длины  $\ell = |X|$ ;

$\bar{X}$  — скрытая контрольная выборка длины  $k = |\bar{X}| = L - \ell$ ;

Теория меры и предельный переход  $L \rightarrow \infty$  не используются.

**Вероятность** понимается только как доля разбиений выборки:

$$Q_\varepsilon(\mu, \mathbb{X}) = \mathbf{P}[\delta(\mu, X, \bar{X}) \geq \varepsilon] = \frac{1}{C_L^\ell} \sum_{\substack{X, \bar{X} \\ X \sqcup \bar{X} = \mathbb{X}}} [\delta(\mu, X, \bar{X}) \geq \varepsilon].$$

Это аналог стандартной гипотезы о независимости наблюдений.



## Аналог закона больших чисел в слабой аксиоматике

Пусть  $|A| = 1$ ,  $\mu X = a$  для всех  $X \subset \mathbb{X}$ .

Обозначим  $m = n(a, \mathbb{X})$ ,  $s = n(a, X)$ .

### Теорема (точная оценка)

Вероятность большого уклонения частот описывается функцией **гипергеометрического распределения** (ГГР):

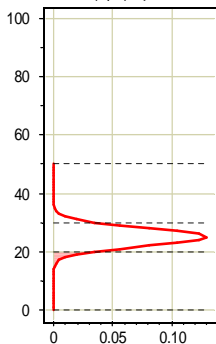
$$Q_\varepsilon(a, \mathbb{X}) = H_L^{\ell, m} \left( \frac{\ell}{L} (m - \varepsilon k) \right),$$

где  $H_L^{\ell, m}(z) = \sum_{s=0}^{\lfloor z \rfloor} \frac{C_m^s C_{L-m}^{\ell-s}}{C_L^\ell}$  — левый «хвост» ГГР.

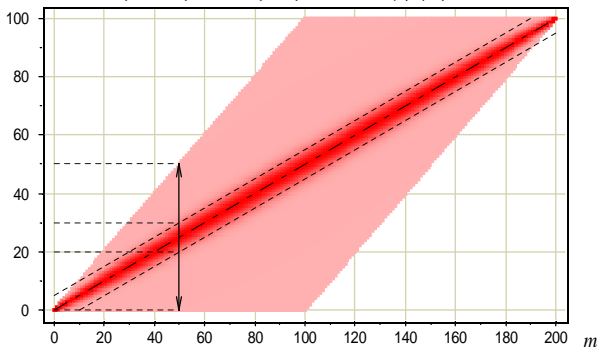
**Вывод:** основная аксиома обеспечивает возможность предсказания скрытого  $n(a, \bar{X})$  по наблюдаемому  $n(a, X)$ .

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

$s$   $h(s|m)$  при  $m=50$



$s$  Гипергеометрическое распределение  $h(s|m)$  при  $L=200, k=100$



Предсказание  $n(a, \bar{X})$  по  $n(a, X)$  возможно благодаря узости гипергеометрического пика (концентрации вероятности).

Закон больших чисел:  $\frac{1}{\ell} n(a, X), \frac{1}{k} n(a, \bar{X}) \rightarrow \frac{1}{L} n(a, \mathbb{X})$  при  $\ell, k \rightarrow \infty$ .

## В слабой аксиоматике также передоказаны:

- сходимости эмпирических распределений (критерий Смирнова, в том числе для дискретных распределений);
- доверительные интервалы для квантилей;
- критерий знаков, Уилкоксона–Манна–Уитни, и другие непараметрические критерии;
- оценки Вапника–Червоненкиса;

### Открытая проблема

Насколько значительную часть теории вероятностей, математической статистики, теории информации возможно переформулировать в рамках слабой аксиоматики?

## Обобщение оценки Вапника–Червоненкиса

### Определение

*Степень некорректности* метода обучения  $\mu$  на выборке  $\mathbb{X}$ :

$$\sigma(\mu, \mathbb{X}) = \max_{X \subset \mathbb{X}: |X|=\ell} \nu(\mu X, X).$$

### Теорема

Для любых  $\mathbb{X}$ ,  $A$ ,  $\mu$ ,  $\sigma(\mu, \mathbb{X}) \leq \sigma$ ,  $\varepsilon \in (0, 1)$

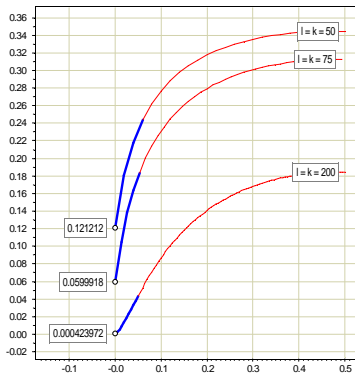
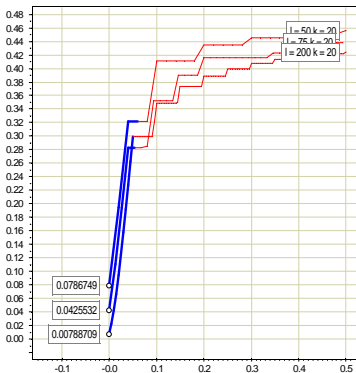
$$Q_\varepsilon(\mu, \mathbb{X}) \leq |A| \max_{m \in M(\varepsilon, \sigma)} H_L^{\ell, m}(s(\varepsilon, \sigma)),$$

где

$$M(\varepsilon, \sigma) = \{m: \varepsilon k \leq m \leq k + \sigma \ell\};$$

$$s(\varepsilon, \sigma) = \min\left\{\frac{\ell}{L}(m - \varepsilon k), \sigma \ell\right\}.$$

## Обобщение оценки Вапника–Червоненкиса (эксперимент)

Зависимость  $Q_\epsilon$  от степени некорректности  $\sigma$  при  $|A| = 1$ :при  $l = k = 50, 75, 200$ .при  $l = 50, 75, 200$ ;  $k = 20$ .

## Обобщение оценки Вапника–Червоненкиса (выводы)

### Выводы

- Даже малая некорректность приводит к резкому увеличению вероятности переобучения  $Q_\epsilon$ .
- Учёт корректности в некоторых случаях уменьшает VC-оценку в сотни раз.
- Однако схема доказательства остаётся та же, ни одна из причин завышенности VC-оценки не устраняется.

### Задачи следующего этапа

- Измерить факторы завышенности количественно в эксперименте на реальных данных.
- Выделить наиболее значимые факторы завышенности.

## Идея эмпирического измерения факторов завышенности

В слабой аксиоматике вероятность измеряется по случайному подмножеству разбиений  $X_n \sqcup \bar{X}_n = \mathbb{X}$ ,  $n = 1, \dots, N$ :

$$\hat{Q}_\varepsilon(\mu, \mathbb{X}) = \frac{1}{N} \sum_{n=1}^N [\delta(\mu X_n, X_n, \bar{X}_n) \geq \varepsilon]$$

Степень завышенности раскладывается в произведение факторов:

$$\hat{Q}_\varepsilon(\mu, \mathbb{X}) \cdot r_1 \cdot r_2 \cdot r_3 \cdot r_4 = |A| \cdot e^{-\varepsilon^2 \ell}$$

**Факторы завышенности VC-оценки:**

- $r_1 \geq 1$ : расслоение (принцип равномерной сходимости)
- $r_2 \geq 1$ : сходство (применение неравенства Буля)
- $r_3 \geq 1$ : оценка профиля разнообразия сверху константой
- $r_4 \geq 1$ : экспоненциальная аппроксимация ГГР

## Вычислительный эксперимент

- 7 задач классификации на два класса (из репозитория UCI)
- $20 \times 2$ -кратный скользящий контроль,  $\ell = k$
- Логический алгоритм классификации Forecsys LogicPro<sup>®</sup>  
[Воронцов, Кочедыков, Ивахненко]

Задача	$L$	$n$	средняя ошибка не тестовых данных				
			C4.5	C5.0	RIPPER	SLIPPER	LogicPro
crx	690	15	15.5	14.0	15.2	15.7	$14.3 \pm 0.2$
german	1000	20	27.0	28.3	28.7	27.2	$28.5 \pm 1.0$
hepatitis	155	19	18.8	20.1	23.2	17.4	$16.7 \pm 1.7$
horse-colic	300	25	16.0	15.3	16.3	15.0	$16.4 \pm 0.5$
hypothyroid	3163	25	0.4	0.4	0.9	0.7	$0.8 \pm 0.04$
liver	345	6	37.5	31.9	31.3	32.2	$29.2 \pm 1.6$
promoters	106	57	18.1	22.7	19.0	18.9	$12.0 \pm 2.0$

$L$  — объём полной выборки;  $n$  — число признаков.



## Результаты эксперимента

### Причины завышенности VC-оценок

(пороги  $\varepsilon_0, \varepsilon_1, \varepsilon_2$  соответствуют надёжности  $\hat{Q}_\varepsilon = 0.05, 0.1, 0.01$ ).

Задача	$y$	$r_1$	$r_2(\varepsilon_0)$	$r_3(\varepsilon_0)$	$r_4(\varepsilon_0)$	$\Delta[\varepsilon_1, \varepsilon_2]$	$\Delta(\varepsilon_0)$
crx	0	890	680	3.1	32.6	[10; 41]	24
	1	690	1700	1.6	11.6	[11; 180]	12
german	1	8 950	1500	1.7	10.9	[38; 530]	54
	2	37 000	9000	1.2	9.9	[1.0; 2.2]	1.9
hepatitis	0	23	280	13.4	9.5	[11; 148]	83
	1	55	680	2.4	22.5	[12; 27]	15
horse-colic	1	72	4500	2.1	7.2	[2; 9]	7
	2	140	3400	3.6	7.3	[3; 6]	6
hypothyroid	0	61 000	400	32.2	16.5	[3; 220]	21
	1	153 000	460	3.8	28.7	[2; 44]	30
promoters	0	94	340	5.9	9.8	[36; 230]	72
	1	150	790	3.4	6.9	[9; 22]	18

## Выводы

Результаты экспериментов на 7 реальных задачах из UCI:

$|A| \sim 10^6 \dots 10^{11}$  — число различных алгоритмов в  $A$ ;

$\Delta \sim 10^0 \dots 10^2$  — число эффективно используемых алгоритмов (эффективный локальный коэффициент разнообразия, ЭЛКР).

### Основные причины завышенности

- Не учитывается *расслоение* множества алгоритмов:

чем выше  $m = n(a, \mathbb{X})$ , тем меньше  $P[\mu X = a]$   
(завышенность в  $r_1 = 10^2 \dots 10^5$  раз);

- Не учитывается *связность* множества алгоритмов:

чем больше схожих алгоритмов, тем сильнее завышенность  
(завышенность в  $r_2 = 10^3 \dots 10^4$  раз).

На практике множество  $A$ , как правило, расслоено и связно.

## Связные семейства и цепочки алгоритмов

**Опр. 1.** Семейство  $A$  *связное*, если  $\forall a \in A \exists a' \in A: \rho(a, a') = 1$ .

$\rho(a, a')$  — хэммингово расстояние между векторами ошибок:

$$\rho(a, a') = \sum_{i=1}^L |I(a, x_i) - I(a', x_i)|, \quad \forall a, a' \in A.$$

**Опр. 2.** *Цепочка алгоритмов* — последовательность  $a_0, a_1, \dots, a_D$  такая, что  $\rho(a_{d-1}, a_d) = 1$  для всех  $d = 1, \dots, D$ .

**Опр. 3.** *Цепочка алгоритмов монотонная*, если  $I(a_{d-1}, x) \leq I(a_d, x)$  для всех  $x \in \mathbb{X}$  и  $d = 1, \dots, D$ .

**Пример:**

	$a_0$	$a_1$	$a_2$	$a_3$	$\dots$	$a_D$
$x_1$	0	1	1	1	1	1
$x_2$	0	0	1	1	1	1
$x_3$	0	0	0	1	1	1
$\dots$	0	0	0	0	1	1
$x_L$	0	0	0	0	0	1

## Эксперимент с монотонной цепочкой алгоритмов

Цель эксперимента: понять, как *связность* и *расслоение* влияют на вероятность переобучения.

Цепочка с расслоением:

	$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$
$x_1$	1	1	1	1	1	1	1
$x_2$	$0 \rightarrow 1$	1	1	1	1	1	1
$x_3$	0	$0 \rightarrow 1$	1	1	1	1	1
$x_4$	0	0	$0 \rightarrow 1$	1	1	1	1
$x_5$	0	0	0	$0 \rightarrow 1$	1	1	1
$x_6$	0	0	0	0	$0 \rightarrow 1$	1	1

Цепочка без расслоения:

	$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$
$x_1$	1	$1 \rightarrow 0$	0	0	0	0	0
$x_2$	$0 \rightarrow 1$	1	$1 \rightarrow 0$	0	0	0	0
$x_3$	0	0	$0 \rightarrow 1$	1	$1 \rightarrow 0$	0	0
$x_4$	0	0	0	0	$0 \rightarrow 1$	1	1
$x_5$	0	0	0	0	0	0	0
$x_6$	0	0	0	0	0	0	0

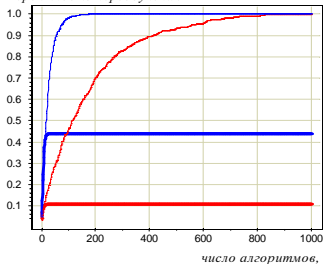
Для каждой цепочки генерируется *не-цепочка* путём случайной перестановки единиц в каждом столбце.

Итого имеем 4 модельных семейства.

## Цепочки и не-цепочки; с расслоением и без

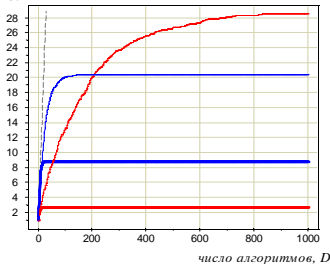
Зависимость  $Q_\varepsilon$  от  $D$  при  $\ell = k = 100$ ,  $\varepsilon = 0.05$ ,  $n(a_0, \mathbb{X}) = 10$ .

Вероятность переобучения



— Цепочка с расслоением      — Цепочка без расслоения  
— Не-цепочка с расслоением      — Не-цепочка без расслоения

ЭЛКР



## Выводы

- Связность приводит к замедлению роста  $Q_\varepsilon(D)$ .
- Расслоение понижает уровень горизонтальной асимптоты.

## Эксперимент с монотонной цепочкой алгоритмов

### Основные выводы

- Без расслоения или без связности переобучение ( $Q_\varepsilon = \frac{1}{2}$ ) наступает при  $|A|$  порядка десятков.
- На практике «хорошие» семейства обязаны быть расслоенными и обладать той или иной структурой сходства алгоритмов, например, связностью.

### Дальнейшая цель

- Изучение структур расслоения и сходства (связности) в семействах алгоритмов.

## «Игрушечный пример»: двухэлементное множество алгоритмов

Пусть алгоритмы  $a_1, a_2$  допускают  $m_1, m_2$  ошибок на  $X^L$ :

$$\begin{aligned}
 a_1 &= (\overbrace{11111111}^{m_1} 00000000000000000000); \\
 a_2 &= (000 \overbrace{111111111111}^{m_2} 0000000000000000).
 \end{aligned}$$

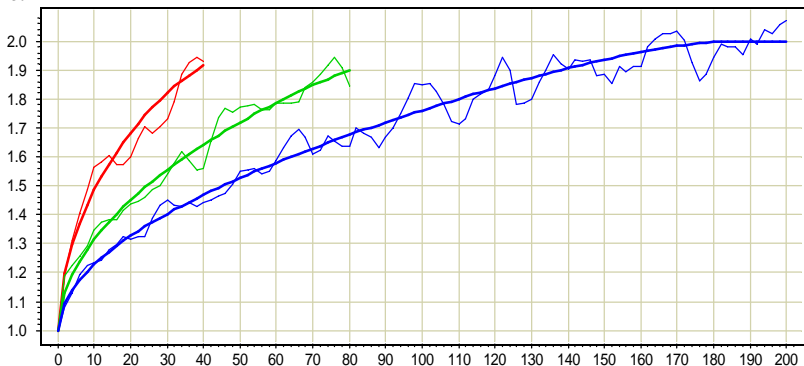
### Теорема (точная оценка вероятности переобучения)

$$\begin{aligned}
 Q_\varepsilon &= \sum_{s_0=0}^{m_0} \sum_{s_1=0}^{m_1} \sum_{s_2=0}^{m_2} \frac{C_{m_0}^{s_0} C_{m_1}^{s_1} C_{m_2}^{s_2} C_{L-m_0-m_1-m_2}^{\ell-s_0-s_1-s_2}}{C_L^\ell} \times \\
 &\quad \times \left( [s_1 < s_2] [s_0 + s_1 \leq \frac{\ell}{L}(m_0 + m_1 - \varepsilon k)] + \right. \\
 &\quad \left. + [s_1 \geq s_2] [s_0 + s_2 \leq \frac{\ell}{L}(m_0 + m_2 - \varepsilon k)] \right).
 \end{aligned}$$

## Эксперимент №1. Два алгоритма одинакового качества

$$\ell = k = 100; \quad \varepsilon = 0.05; \quad \underline{m_1 = m_2}; \quad m = 20, 40, 100$$

ЭЛКР



*хэммингово расстояние между алгоритмами*

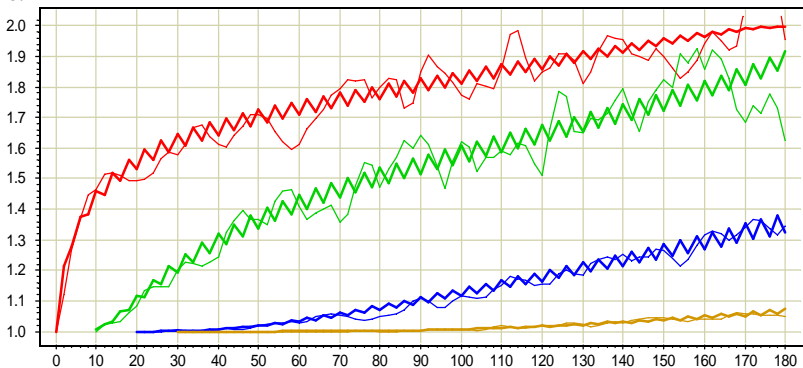
— m=20    — m=40    — m=100



## Эксперимент №2. Два алгоритма разного качества (расслоение)

$$\ell = k = 100; \quad \varepsilon = 0.05; \quad \underline{m_0} = 20; \quad \underline{d} \equiv m_2 - m_1 = 0, 10, 20, 30$$

ЭЛКР



*хэммингово расстояние между алгоритмами*

— d=0    — d=10    — d=20    — d=30

## Двуэлементное множество алгоритмов

### Выводы

- Оптимизация при неполной информации ведёт к переобучению, даже если вариантов выбора только два.
- Если варианты почти одинаковы, то переобучения почти нет (эффект сходства).
- Если один из вариантов явно хуже, то переобучения почти нет (эффект расслоения).

### Задача следующего этапа

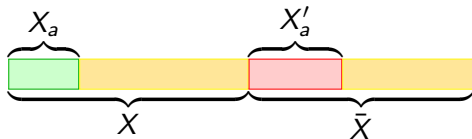
Развить математические методы получения точных оценок  $Q_\epsilon$  для множеств алгоритмов с расслоением и связностью.

## Гипотеза о порождающих и запрещающих объектах

### Гипотеза (1)

Для каждого  $a \in A$  можно указать пару непересекающихся подмножеств объектов  $X_a \subset \mathbb{X}$ ,  $X'_a \subset \mathbb{X}$  такую, что:

$$(\mu X = a) \Leftrightarrow (X_a \subseteq X) \text{ и } (X'_a \subseteq \bar{X}), \quad \forall X \subset \mathbb{X}.$$



Опр.  $X_a$  — множество объектов, **порождающих** алгоритм  $a$ .

Опр.  $X'_a$  — множество объектов, **запрещающих** алгоритм  $a$ .

Опр.  $\mathbb{X} \setminus (X_a \cup X'_a)$  — множество объектов, **нейтральных** для  $a$ .

## Обозначения и основная лемма

Введём для каждого  $a \in A$  следующие обозначения:

$L_a = L - |X_a| - |X'_a|$  — число нейтральных объектов;

$\ell_a = \ell - |X_a|$  — число нейтральных обучающих объектов;

### Лемма (о вероятности получения алгоритма)

*Если гипотеза (1) справедлива, то вероятность получить в результате обучения алгоритм  $a$  равна доле разбиений, при которых объекты из  $X_a$  и  $X'_a$  остаются на своих местах:*

$$P_a = P[\mu X = a] = \frac{C_{L_a}^{\ell_a}}{C_L^{\ell}}.$$

## Ещё обозначения и основная теорема

$$m_a = n(a, \mathbb{X}) - n(a, X_a) - n(a, X'_a)$$

— число ошибок алгоритма  $a$  на нейтральных объектах;

$$s_a(\varepsilon) = \frac{\ell}{L} (n(a, \mathbb{X}) - \varepsilon k) - n(a, X_a)$$

— наибольшее число ошибок переобученного алгоритма  $a$  на нейтральных обучающих объектах  $X \setminus X_a$ .

### Теорема (точная оценка вероятности переобучения)

*Если гипотеза (1) справедлива, то*

$$P[\delta_\mu(X) \geq \varepsilon] = \sum_{a \in A} P_a H_{L_a}^{\ell_a, m_a}(s_a(\varepsilon)).$$

## Обобщение основной гипотезы

### Гипотеза (2)

Для каждого  $a \in A$  можно указать такой **набор пар** непересекающихся подмножеств объектов  $X_{av}, X'_{av} \subset \mathbb{X}$ ,  $v \in V_a$  и такой коэффициент  $c_{av} \in \mathbb{R}$ , что

$$[\mu X=a] = \sum_{v \in V_a} c_{av} [X_{av} \subseteq X] [X'_{av} \subseteq \bar{X}].$$

Обозначения: для каждого  $a \in A$  и каждого  $v \in V_a$

$$L_{av} = L - |X_{av}| - |X'_{av}|;$$

$$l_{av} = l - |X_{av}|;$$

$$m_{av} = n(a, \mathbb{X}) - n(a, X_{av}) - n(a, X'_{av});$$

$$s_{av}(\varepsilon) = \frac{l}{L} (n(a, \mathbb{X}) - \varepsilon k) - n(a, X_{av}).$$

## Обобщение: лемма и основная теорема

### Лемма (о вероятностях получения алгоритмов)

Если гипотеза (2) справедлива, то вероятность получить в результате обучения алгоритм с вектором ошибок  $a$

$$P[\mu X = a] = \sum_{v \in V_a} c_{av} P_{av}; \quad P_{av} = \frac{C_{L_{av}}^{\ell_{av}}}{C_L^{\ell}}.$$

### Теорема (точная оценка вероятности переобучения)

Если гипотеза (2) справедлива, то

$$Q_{\varepsilon} = \sum_{a \in A} \sum_{v \in V_a} c_{av} P_{av} H_{L_{av}}^{\ell_{av}, m_{av}}(s_{av}(\varepsilon)).$$

## Сильное ли ограничение накладывает гипотеза (2)?

Оказывается, почти не накладывает. **Это общий случай!**

### Теорема

Пусть векторы ошибок алгоритмов  $a_1, \dots, a_D$  попарно различны и метод  $\mu$  минимизирует эмпирический риск.

Тогда справедлива гипотеза (2), причём  $c_{av} = 1$ .

Доказательство конструктивно, но «тавтологично» — строится система подмножеств  $(X_{av}, X'_{av})_{v \in V_a} \equiv (X, \bar{X})_{\mu X=a}$ , что приводит к вычислительно неэффективным оценкам.

**В общем случае система подмножеств не единственна.**

### Открытая проблема

Как искать системы подмножеств с наименьшими  $|X_{av}|$ ,  $|X'_{av}|$ ?



## Монотонная цепочка алгоритмов

Чем интересна *монотонная цепочка*:

- это простейший пример связного семейства с расслоением;
- это модель «хорошего» однопараметрического семейства.

	$a_0$	$a_1$	$a_2$	$a_3$	$\dots$	$a_D$
$x_1$	0	1	1	1	1	1
$x_2$	0	0	1	1	1	1
$x_3$	0	0	0	1	1	1
$x_4$	0	0	0	0	1	1
$\dots$	0	0	0	0	0	1
$\dots$	0	0	0	0	0	0
$\dots$	0	0	0	0	0	0
$\dots$	1	1	1	1	1	1
$x_L$	1	1	1	1	1	1

## Вероятность переобучения монотонной цепочки

Пусть  $\mu$  — пессимистичная минимизация эмпирического риска (выбор алгоритма по принципу «худший из лучших»):

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

### Теорема (точная оценка вероятности переобучения)

Пусть  $a_0, a_1, \dots, a_D$  — монотонная цепочка,  $n(a_0, \mathbb{X}) = m$ ,  $k \leq D \leq L - m$ . Тогда

$$P_d = P[\mu X = a_d] = \frac{C_{L-d-1}^{\ell-1}}{C_L^\ell};$$

$$Q_\varepsilon(\mu, \mathbb{X}) = \sum_{d=0}^k P_d H_{L-d-1}^{\ell-1, m} \left( \frac{\ell}{L} (m + d - \varepsilon k) \right).$$

## Идея доказательства

Перенумеруем объекты так, чтобы  $a_d$  ошибался на  $x_1, \dots, x_d$ .

	$a_0$	$a_1$	$a_2$	$a_3$	$\dots$	$a_D$
$x_1$	0	1	1	1	1	1
$x_2$	0	0	1	1	1	1
$x_3$	0	0	0	1	1	1
$x_4$	0	0	0	0	1	1
$\dots$	0	0	0	0	0	1
$\dots$	0	0	0	0	0	0
$\dots$	0	0	0	0	0	0
$\dots$	1	1	1	1	1	1
$x_L$	1	1	1	1	1	1

$(\mu X = a_d) \Leftrightarrow (x_{d+1} \in X) \text{ и } (x_1, \dots, x_d \in \bar{X}), \text{ при } d \leq k;$

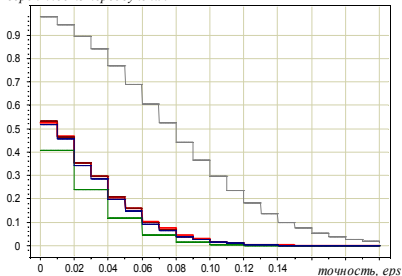
$(\mu X = a_d)$  невозможно, при  $d > k$ .

Таким образом, справедлива Гипотеза (1).

## Вычислительный эксперимент

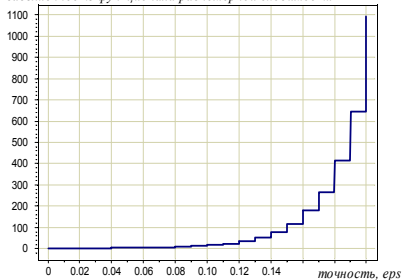
Зависимость  $Q_\epsilon$  от точности  $\epsilon$  при  $\ell = k = 100$ ,  $m = 20$ :

вероятность переобучения



— точная оценка — лесс.МЭР — ранд.МЭР  
— равномерная — опт.МЭР

завышенность функционала равномерной сходимости



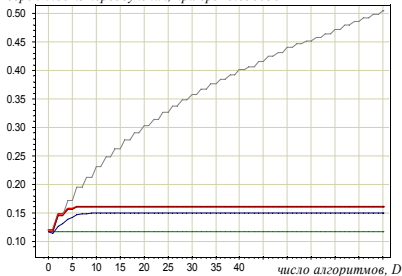
## Вывод

- Равномерная оценка сильно завышена при больших  $\epsilon$ .

## Вычислительный эксперимент

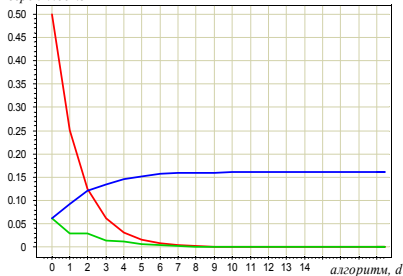
Зависимость  $Q_\epsilon$  от  $D$  при  $\ell = k = 100$ ,  $m = 20$ ,  $\epsilon = 0.05$ :

вероятность переобучения, при  $\epsilon = 0.0500501$



— точная оценка — песс.МЭР — ранд.МЭР  
— равномерная — опт.МЭР

вероятность



— вероятность алгоритма  $P(d)$  — вероятность переобучения  
— вклад  $d$ -го алгоритма в  $Q_{\epsilon ps}$

## Выводы

- Монотонная цепочка почти не переобучается.
- Существенны вклады только 5–6 нижних слоёв.

## Другие модельные семейства алгоритмов,

для которых в данной работе получены точные оценки:

- унимодальная цепочка алгоритмов;
- единичная окрестность лучшего алгоритма;
- слой булева куба;
- интервал булева куба;
- $d$  нижних слоёв интервала булева куба.

Оценки, полученные другими авторами:

- связные семейства [Д. Кочедыков, И. Решетняк].
- монотонные и унимодальные  $h$ -мерные сетки [П. Ботов];
- симметричные семейства алгоритмов [А. Фрей];
- связка монотонных цепочек [А. Фрей];
- хэммингов шар, слои хэммингова шара [И. Толстихин];

## Единичная окрестность лучшего алгоритма

**Опр.** *Единичная окрестность* — последовательность  $a_0, a_1, \dots, a_D$ :  
 $n(a_d, \mathbb{X}) = n(a_0, \mathbb{X}) + 1$  и  $\rho(a_0, a_d) = 1$  для всех  $d = 1, \dots, D$ .  
 Алгоритм  $a_0$  — центр окрестности.

	$a_0$	$a_1$	$a_2$	$a_3$	$\dots$	$a_D$
$x_1$	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
$\dots$	0	0	0	0	0	1
$\dots$	0	0	0	0	0	0
$\dots$	0	0	0	0	0	0
$\dots$	1	1	1	1	1	1
$x_L$	1	1	1	1	1	1

## Единичная окрестность лучшего алгоритма

Пусть  $\mu$  — пессимистичная минимизация эмпирического риска; причём в случае неоднозначности метод  $\mu$  выбирает алгоритм с меньшим номером.

### Теорема (точная оценка)

Пусть  $m = n(a_0, \mathbb{X})$ ;  $L \geq m + D$ . Тогда

$$P[\delta_\mu(X) \geq \varepsilon] = P_0 H_{L-D}^{\ell-D, m} \left( \frac{\ell}{L} (m - \varepsilon k) \right) + \sum_{d=1}^D P_d H_{L-d}^{\ell-d+1, m} \left( \frac{\ell}{L} (m + 1 - \varepsilon k) \right);$$

$$P_0 = \frac{C_{L-D}^k}{C_L^k}; \quad P_d = \frac{C_{L-d}^{k-1}}{C_L^k}, \quad d = 1, \dots, D;$$



## Слой булева куба

Пусть  $A = \{a: n(a, \mathbb{X}) = m\}$  —  $m$ -й слой булева куба,  $|A| = C_L^m$ .

### Теорема

Пусть  $\mu$  — метод минимизации эмпирического риска.

Тогда для любого  $\varepsilon \in [0, 1]$

$$Q_\varepsilon = [\varepsilon k \leq m \leq L - \varepsilon l].$$

### Выводы

- Алгоритмы нижних слоёв,  $m < \varepsilon k$ , не вносят вклад в переобучение.
- Нижний слой множества алгоритмов не должен быть «слишком богатым», т. е. полным слоем.

## Интервал булева куба

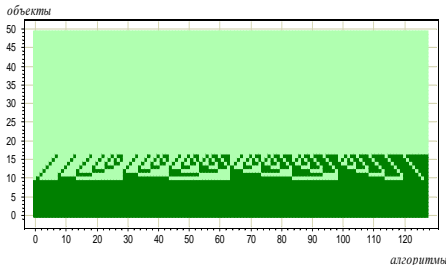
Пусть  $A$  — интервал ранга  $m$  в  $L$ -мерном булевом кубе.

$m_0$  «внутренних» объектов  $x_j$ :  $I(a, x) = 0, \forall a \in A$ ;

$m_1$  «шумовых» объектов  $x_j$ :  $I(a, x) = 1, \forall a \in A$ ;

$m$  «пограничных» объектов: реализуются все  $2^m$  векторов ошибок.

**Пример.** Матрица ошибок  
интервала булева куба  
при  $L = 50$ ,  
 $m = 7$ ,  
 $m_0 = 33$ ,  
 $m_1 = 10$ .



## Слои интервала булева куба

### Теорема

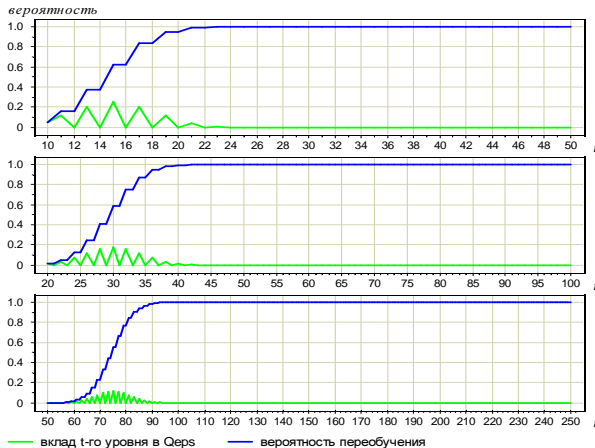
Пусть  $\mu$  — пессимистичная минимизация эмпирического риска,  
 $A$  — нижние  $t$  слоёв интервала ранга  $m$ ,  
 число шумовых объектов равно  $m_1$ .

Тогда для любого  $\varepsilon \in [0, 1]$

$$Q_\varepsilon = \sum_{s=0}^m \sum_{s_1=0}^{m_1} \frac{C_m^s C_{m_1}^{s_1} C_{L-m-m_1}^{\ell-s-s_1}}{C_L^\ell} [s_1 \leq \frac{\ell}{L}(m_1 + \min\{t, m-s\} - \varepsilon k)].$$

## Интервал булева куба: результаты экспериментов

Зависимость вероятности переобучения  $Q_\varepsilon$  от номера слоя  $t$  при  $\varepsilon = 0.05$ ,  $\ell = k = \{100, 200, 500\}$ ,  $m_1 = 0.1\ell$ ,  $m = 0.4\ell$ .



## Интервал булева куба: выводы

### Выводы

- Переобучение наступает довольно быстро — при доле пограничных объектов  $\frac{m}{L}$  порядка  $\varepsilon$ .
- Интервал — слишком «богатое» семейство алгоритмов.
- В реальных семействах алгоритмов пограничный слой, по всей видимости, устроен иначе.

## Постановка задачи рекуррентного вычисления $Q_\varepsilon$

$\mathfrak{I}(a) = \langle X_{av}, X'_{av}, c_{av} \rangle_{v \in V_a}$  — информация об алгоритме  $a \in A$ , необходимая для вычисления вероятности переобучения  $Q_\varepsilon$ .

Расслоение:  $n(a_0, \mathbb{X}) \leq n(a_1, \mathbb{X}) \leq \dots \leq n(a_D, \mathbb{X})$ .

Дополнительное предположение:  $n(a_0, \mathbb{X}) = 0$ .

Пусть  $\mu_d$  — пессимистичный метод обучения, выбирающий алгоритмы только из подмножества  $A_d = \{a_0, \dots, a_d\}$ .

**Задача (пересчёт  $Q_\varepsilon$  при добавлении алгоритма  $a_d$ )**

*Известна информация  $\mathfrak{I}(a_t)$  относительно метода  $\mu_{d-1}$  для всех алгоритмов  $a_t$ ,  $t \leq d-1$ .*

*Вычислить информацию  $\mathfrak{I}(a_t)$  относительно метода  $\mu_d$  для всех алгоритмов  $a_t$ ,  $t \leq d$ .*

## Теоремы о рекуррентном вычислении $Q_\varepsilon$

### Теорема (об информации $\mathfrak{I}(a_d)$ )

$$[\mu_d X = a_d] = [X'_d \subseteq \bar{X}], \quad X'_d = \{x_i \in \mathbb{X} : I(a_d, x_i) = 1\}.$$

### Теорема (о коррекции информации $\mathfrak{I}(a_t)$ , $t < d$ )

Для каждого  $v \in V_t$  такого, что  $X_{tv} \cap X'_d = \emptyset$

- 1) если  $X'_d \setminus X'_{tv} = \{x_i\}$  — одноэлементное множество, то присоединить  $x_i$  к  $X_{tv}$ ;
- 2) если  $|X'_d \setminus X'_{tv}| > 1$ , то добавить в  $V_t$  индекс  $w$ , положив  $c_{tw} = -c_{tv}$ ,  $X_{tw} = X_{tv}$ ,  $X'_{tw} = X'_{tv} \cup X'_d$ ;
- 3) если  $|X'_d \setminus X'_{tv}| = 0$ , то удалить из  $V_t$  индекс  $v$ .

## Упрощённое рекуррентное вычисление $Q_\varepsilon$

### Теорема (О верхних и нижних оценках)

Если иногда пропускать шаг 2) при  $c_{tv} = 1$ ,  
то вычисляемое значение  $Q_\varepsilon$  может только увеличиться.

Если иногда пропускать шаг 2) при  $c_{tv} = -1$ ,  
то вычисляемое значение  $Q_\varepsilon$  может только уменьшиться.

### Теорема (Об упрощённом рекуррентном вычислении $Q_\varepsilon$ )

Если всегда пропускать шаг 2), то шаг 3) не будет выполняться  
никогда, и будет получена верхняя оценка  $Q_\varepsilon$ .

Рекуррентное вычисление  $Q_\varepsilon$  может занять время  $O(L2^D)$ .  
Упрощённое рекуррентное вычисление  $Q_\varepsilon$  занимает  $O(LD^2)$ .  
Его можно сократить до  $O(LD)$  и даже до  $O(L)$ .



## Расслоение и связность

Расслоение множества алгоритмов  $A = \bigsqcup_{m=0}^L A_m$ , где

$A_m = \{a \in A: n(a, \mathbb{X}) = m\}$  —  $m$ -й слой множества алгоритмов.

Отношение частичного порядка на алгоритмах:

$$a \prec a' \Leftrightarrow (I(a, x) \leq I(a', x), \forall x \in \mathbb{X}).$$

Связность  $q(a)$  алгоритма  $a \in A$  — число сравнимых с ним алгоритмов в следующем слое:

$$q(a) = \#\left\{a': a \prec a' \text{ и } n(a, \mathbb{X}) + 1 = n(a', \mathbb{X})\right\}.$$

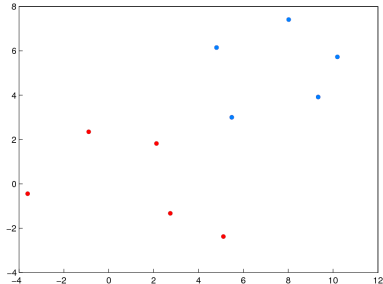
Граф связности множества алгоритмов  $A$ :

— вершины — алгоритмы,

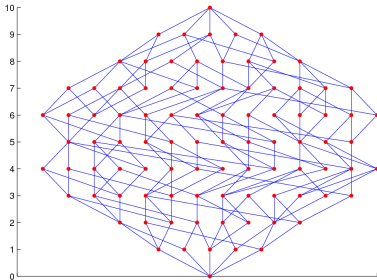
— рёбра  $(a, a')$  — пары алгоритмов  $a \prec a'$  в соседних слоях.

## Граф расслоения и связности (пример)

Линейно разделяемая выборка  
длины  $L = 10$



Граф связности семейства  
линейных классификаторов



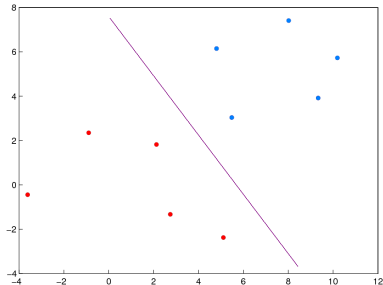
Вершины графа — это алгоритмы.

Рёбра — алгоритмы, различающиеся только на одном объекте.

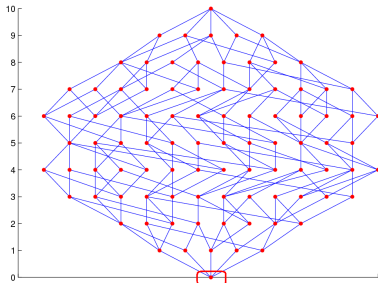
Горизонтальные слои соответствуют уровням числа ошибок  $m$ .

## Граф расслоения и связности (пример)

Линейно разделяемая выборка  
длины  $L = 10$



Слой  $m = 0$  из 1 алгоритма



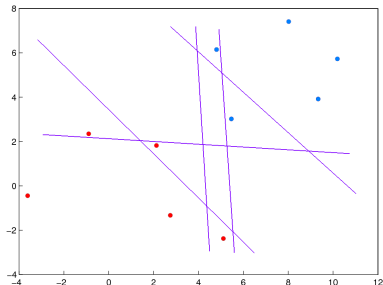
Вершины графа — это алгоритмы.

Рёбра — алгоритмы, различающиеся только на одном объекте.

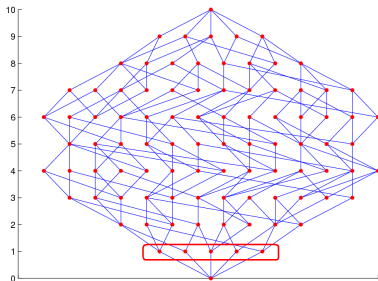
Горизонтальные слои соответствуют уровням числа ошибок  $m$ .

## Граф расслоения и связности (пример)

Линейно разделимая выборка  
длины  $L = 10$



Слой  $m = 1$  из 5 алгоритмов



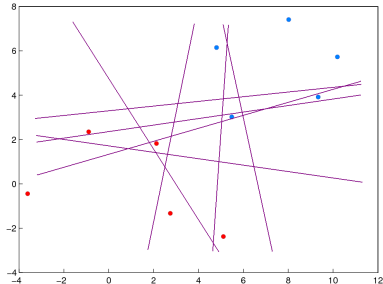
Вершины графа — это алгоритмы.

Рёбра — алгоритмы, различающиеся только на одном объекте.

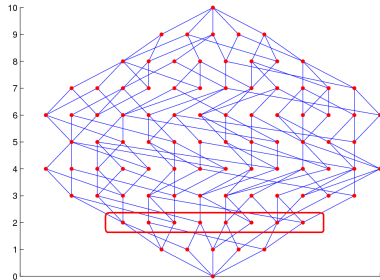
Горизонтальные слои соответствуют уровням числа ошибок  $m$ .

## Граф расслоения и связности (пример)

Линейно разделимая выборка  
длины  $L = 10$



Слой  $m = 2$  из 8 алгоритмов



Вершины графа — это алгоритмы.

Рёбра — алгоритмы, различающиеся только на одном объекте.

Горизонтальные слои соответствуют уровням числа ошибок  $m$ .

## Оценка $Q_\varepsilon$ через профиль расслоения–связности

**Опр.** Профиль расслоения–связности  $\Delta_{mq}$  — это число алгоритмов в  $m$ -м слое со связностью  $q$ .

### Теорема

Пусть векторы ошибок всех алгоритмов  $a \in A$  попарно различны, и в  $A$  есть корректный на  $\mathbb{X}$  алгоритм.

Тогда справедлива верхняя оценка вероятности переобучения

$$Q_\varepsilon \leq \sum_{m=\lceil \varepsilon k \rceil}^L \sum_{q=0}^{\ell} \Delta_{mq} \frac{C_{L-m-q}^{\ell-q}}{C_L^\ell}.$$

# Оценка $Q_\varepsilon$ через профиль расслоения и профиль связности

## Теорема

Пусть справедливы условия предыдущей теоремы и профиль расслоения–связности сепарабелен:

$$\Delta_{mq} \leq \Delta_m \lambda_q.$$

Тогда справедлива верхняя оценка вероятности переобучения

$$Q_\varepsilon \leq \underbrace{\sum_{m=\lceil \varepsilon k \rceil}^k \Delta_m \frac{C_{L-m}^\ell}{C_L^\ell}}_{VC\text{-оценка}} \underbrace{\sum_{q=0}^L \lambda_q \left( \frac{\ell}{L-m} \right)^q}_{\text{поправка на связность}}.$$

При известных  $\Delta_m$ ,  $\lambda_q$  вычисления  $Q_\varepsilon$  займут  $O(L)$ .

## Эксперименты и выводы

В экспериментах с линейными классификаторами:

- средняя связность = размерности пространства (с очень высокой точностью);
- гипотеза сепарабельности выполнялась (с достаточной точностью);

### Выводы

- **Учёт расслоения и связности существенно уточняет оценку (экспоненциально по размерности пространства).**
- Оценка зависит не от одной скалярной характеристики сложности, а от «профиля», в отличие от VC-оценок.
- Как использовать эту оценку на практике? (пока открытый вопрос)



## Функционал полного скользящего контроля

Выше рассматривалась только *вероятность переобучения*

$$Q_\varepsilon(\mu, \mathbb{X}) = \frac{1}{C_L^\ell} \sum_{(X, \bar{X})} [\nu(\mu X, \bar{X}) - \nu(\mu X, X) \geq \varepsilon].$$

Все результаты легко переносятся и на функционал

$$R_\varepsilon(\mu, \mathbb{X}) = \frac{1}{C_L^\ell} \sum_{(X, \bar{X})} [\nu(\mu X, \bar{X}) \geq \varepsilon].$$

Однако для функционала *полного скользящего контроля*

$$CCV(\mu, \mathbb{X}) = \frac{1}{C_L^\ell} \sum_{(X, \bar{X})} \nu(\mu X, \bar{X}).$$

техника получения оценок совсем другая.

**Недостаток CCV:** не учитывается дисперсия частоты ошибок.

## Метод ближайшего соседа

Пусть  $\rho(x, x')$  — функция расстояния на множестве  $\mathbb{X}$ .

$$a(x; X) = y(\arg \min_{x' \in X} \rho(x, x')).$$

### Определение (профиль компактности выборки $\mathbb{X}$ )

доля объектов, у которых  $m$ -й сосед  $x_{im}$  лежит в другом классе:

$$K(m, \mathbb{X}) = \frac{1}{L} \sum_{i=1}^L [y(x_i) \neq y(x_{im})]; \quad m = 1, \dots, L-1,$$

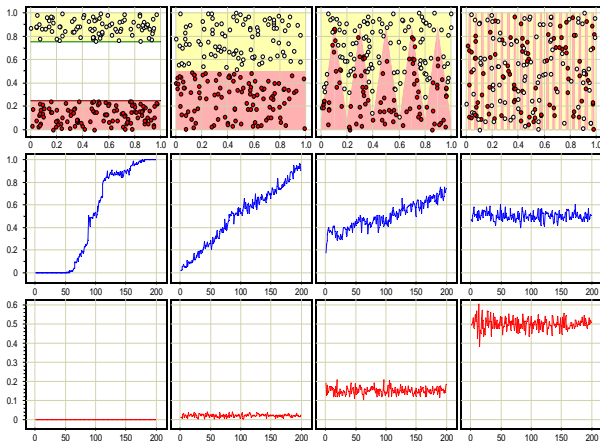
### Теорема (точная оценка для метода ближайшего соседа)

$$\text{CCV}(\mu, \mathbb{X}) = \sum_{m=1}^k K(m, \mathbb{X}) \frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}}.$$

## Профили компактности для серии модельных задач

средний ряд: профили компактности,

нижний ряд: зависимость  $CCV$  от длины контроля  $k = |\bar{X}|$ .



## Свойства профиля компактности и оценки ССV

### Выводы

- Полученная оценка ССV является *точной* (не завышенной, не асимптотической).
- ССV практически не зависит от длины контроля  $k$  (всегда ли? — открытый вопрос).
- Для минимизации ССV важен только начальный участок профиля, т. к.  $\frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}} \rightarrow 0$  экспоненциально по  $m$ .
- Минимизация ССV приводит к эффективному отбору эталонных объектов, без переобучения [М. Иванов, 2009].

**Замечание.** VC-теория вообще не даёт содержательных оценок для метода ближайшего соседа, т.к. ёмкость данного семейства алгоритмов бесконечна.

## Монотонные алгоритмы классификации: определения

Задача классификации:  $\mathbb{X}$  — ч. у. множество,  $Y = \{0, 1\}$ ,  
 $A$  — множество монотонных отображений  $a: \mathbb{X} \rightarrow Y$ .

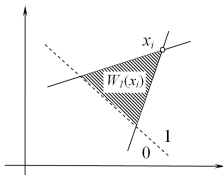
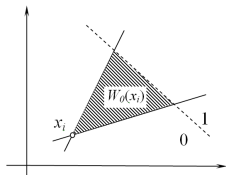
**Опр.** Степень немонотонности выборки  $\mathbb{X}$ :

$$\theta(\mathbb{X}) = \min_{a \in A} \nu(a, \mathbb{X}).$$

**Опр.** Верхний и нижний клин объекта  $x_i \in \mathbb{X}$ :

$$W_0(x_i) = \{x \in \mathbb{X} : x_i < x \text{ и } y(x) = 0\};$$

$$W_1(x_i) = \{x \in \mathbb{X} : x < x_i \text{ и } y(x) = 1\}.$$



## Профиль монотонности выборки

### Определение (Профиль монотонности выборки $\mathbb{X}$ )

доля объектов  $x_i \in \mathbb{X}$  с клином мощности  $m$ :

$$M(m, \mathbb{X}) = \frac{1}{L} \sum_{i=1}^L [ |W_{y(x_i)}(x_i)| = m ]; \quad m = 0, \dots, L-1.$$

### Теорема

Пусть  $\mu$  — метод минимизации эмпирического риска в классе всех монотонных функций,  $\theta$  — степень немонотонности выборки  $\mathbb{X}$ . Тогда

$$\text{CCV}(\mu, \mathbb{X}) \leq \sum_{m=0}^{\theta L + k - 1} M(m, \mathbb{X}) H_{L-1}^{\ell, m}(\theta L).$$

## Свойства профиля монотонности и оценки CCV

### Выводы

- Невырожденность:  $CCV(\mu, \mathbb{X}) \leq 1$ .
- Для минимизации CCV важен только начальный участок профиля, т. к.  $H_{L-1}^{\ell, m}(\theta L) \rightarrow 0$  по  $m$  при малых  $\theta$ .
- Для минимизации CCV отношение порядка на множестве объектов  $\mathbb{X}$  должно быть близко к линейному вблизи границы классов.
- Минимизация CCV приводит к повышению обобщающей способности алгоритмической композиции с монотонной корректирующей операцией [И. Гуз, 2008].

**Замечание.** VC-теория даёт сильно завышенные оценки для монотонных семейств алгоритмов (эффективная ёмкость определяется максимальной длиной антицепи).

## Результаты, выносимые на защиту

- 1 Слабая вероятностная аксиоматика.
- 2 VC-оценки, учитывающие степень некорректности метода обучения.
- 3 Методика измерения факторов завышенности VC-оценок.
- 4 Метод получения точных оценок вероятности переобучения путём выделения порождающих и запрещающих множеств.
- 5 Рекуррентный алгоритм вычисления вероятности переобучения.
- 6 Блочный метод вывода точных оценок вероятности переобучения.
- 7 Точные оценки вероятности переобучения для 7 модельных семейств алгоритмов.
- 8 Верхние оценки вероятности переобучения через профиль расслоения и связности.
- 9 Точные оценки CCV для метода ближайшего соседа через профиль компактности выборки.
- 10 Верхние оценки CCV для монотонных алгоритмов через профиль монотонности выборки.