Вычислимые комбинаторные оценки обобщающей способности

Евгений Соколов
ВМК МГУ

21 сентября 2013 г.

$\mathbb{X}$ — объекты; $\mathbb{Y}$ — ответы;
$y^*\colon \mathbb{X} \to \mathbb{Y}$ — неизвестная зависимость.

Дано: $x_i = (x_i^1, \ldots, x_i^n)$ — *обучающие объекты* с известными ответами $y_i = y^*(x),\ \ i = 1, \ldots, \ell$:

$$\begin{pmatrix} x_1^1 & \ldots & x_1^n \\ \ldots & \ldots & \ldots \\ x_\ell^1 & \ldots & x_\ell^n \end{pmatrix} \xrightarrow{y^*} \begin{pmatrix} y_1 \\ \ldots \\ y_\ell \end{pmatrix}$$

Найти: алгоритм $a\colon \mathbb{X} \to \mathbb{Y}$, способный давать правильные ответы на новых объектах $\tilde{x}_i = (\tilde{x}_i^1, \ldots, \tilde{x}_i^n),\ \ i = 1, \ldots, k$:

$$\begin{pmatrix} \tilde{x}_1^1 & \ldots & \tilde{x}_1^n \\ \ldots & \ldots & \ldots \\ \tilde{x}_k^1 & \ldots & \tilde{x}_k^n \end{pmatrix} \xrightarrow{a?} \begin{pmatrix} a(\tilde{x}_1) \\ \ldots \\ a(\tilde{x}_k) \end{pmatrix}$$

Модель алгоритмов — параметрическое семейство отображений

$$A = \{g(x, \theta) \mid \theta \in \Theta\},$$

где $g\colon \mathbb{X} \times \Theta \to \mathbb{Y}$ — фиксированная функция,
Θ — множество допустимых значений параметра θ.

В задачах обучения по прецедентам выделяются два этапа:

1. *Метод обучения* $\mu\colon (\mathbb{X} \times \mathbb{Y})^\ell \to A$ по обучающей выборке $X = (x_i, y_i)_{i=1}^\ell$ выбирает из A алгоритм $a = \mu(X)$.
2. Найденный алгоритм a применяется для вычисления прогнозов $\tilde{y}_i = a(\tilde{x}_i)$ на новой выборке $\bar{X} = \{\tilde{x}_1, \ldots, \tilde{x}_k\}$.

Принцип минимизации эмпирического риска

Эмпирический риск — частота ошибок алгоритма a на X:

$$Q(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} \big[a(x_i) \neq y_i\big].$$

Минимизация эмпирического риска — пример метода обучения:

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

Проблема обобщающей способности:

- будет ли алгоритм $a = \mu(X)$ приближать y^* на всём $\mathbb{X}$?
- найдём ли мы «закон природы» или *переобучимся*, т. е. подгоним функцию $g(x, \theta)$ под заданные точки (x_i, y_i)?
- будет ли $Q(a, \bar{X})$ мало́ на новых данных — *контрольной выборке* $\bar{X} = (\tilde{x}_i, \tilde{y}_i)_{i=1}^{k}$, $\tilde{y}_i = y^*(\tilde{x}_i)$?

Задача обучения по прецедентам

$\mathbb{X} = \{x_1, \ldots, x_L\}$ — конечное *генеральное множество* объектов;
$\mathbb{A} = \{a_1, \ldots, a_D\}$ — конечное множество *алгоритмов* (гипотез);
$I(a, x) = [\text{алгоритм } a \text{ ошибается на объекте } x]$;

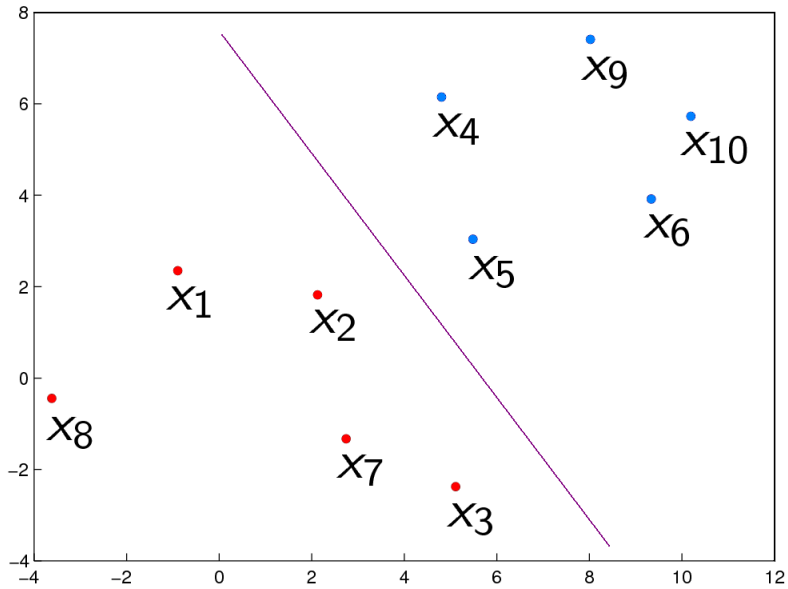
$L \times D$-*матрица ошибок* с попарно различными столбцами:

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

$m(a, X) = \sum\limits_{x \in X} I(a, x)$ — *число ошибок* $a \in \mathbb{A}$ на выборке $X \subset \mathbb{X}$;

$\nu(a, X) = \frac{1}{|X|} m(a, X)$ — *частота ошибок* a на выборке X;

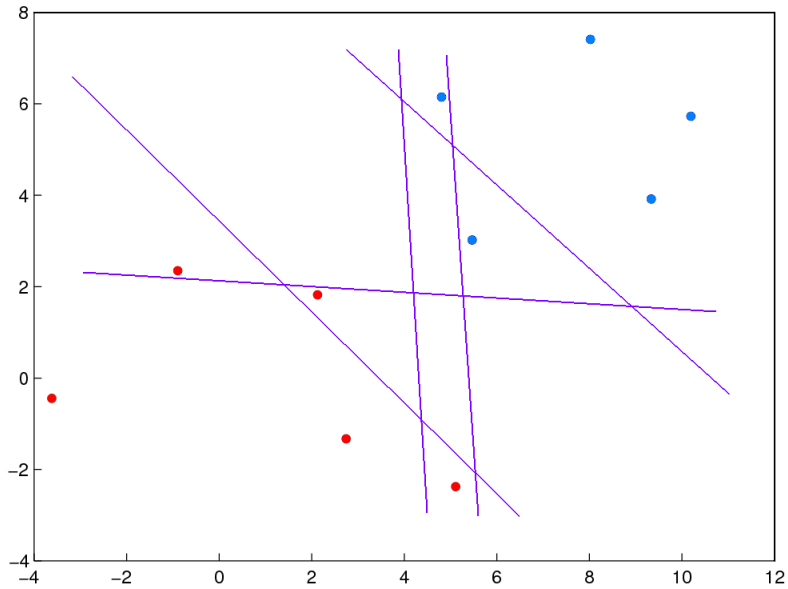
Пример. Матрица ошибок линейных классификаторов



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

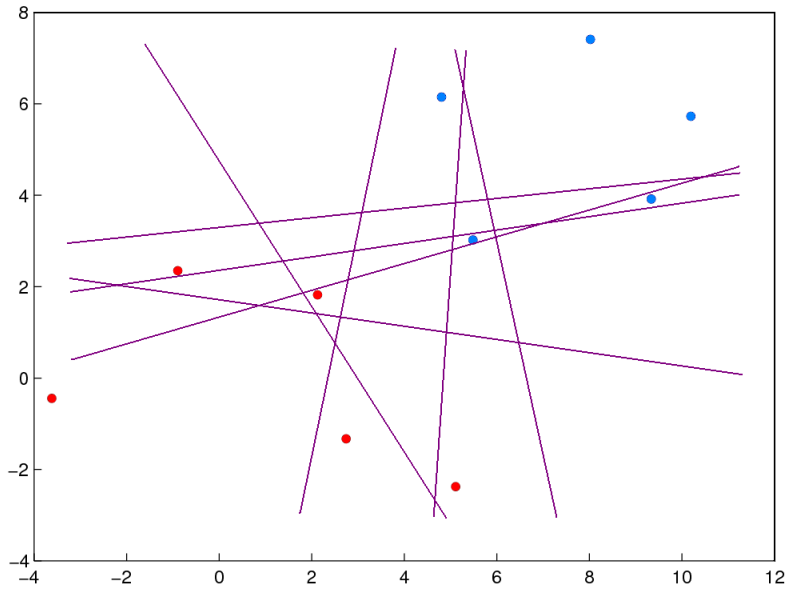
Пример. Матрица ошибок линейных классификаторов



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 обучения по прецедентам

Опр. *Метод обучения* $\mu\colon 2^{\mathbb{X}} \to \mathbb{A}$ произвольной выборке $X \subset \mathbb{X}$ ставит в соответствие некоторый алгоритм $a \in \mathbb{A}$.

Опр. Метод *минимизации эмпирического риска*:

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

Проблема надёжности обучения по прецедентам: насколько большим может оказаться $m(\mu X, \bar{X})$?

Единственная вероятностная аксиома:
пусть все разбиения $X \sqcup \bar{X} = \mathbb{X}$ равновероятны,
X — наблюдаемая обучающая выборка, $|X| = \ell$,
$\bar{X}$ — скрытая контрольная выборка, $|\bar{X}| = k$.

$\mathsf{P} \equiv \mathsf{E} \equiv \frac{1}{C_L^\ell} \sum\limits_{X \subset \mathbb{X}}$ — доля разбиений выборки.

$\delta(\mu, X, \bar{X}) = \nu(\mu X, \bar{X}) - \nu(\mu X, X)$ — *переобученность* μ на X.

Функционалы обобщающей способности:

- Полный скользящий контроль (Complete Cross-Validation):
$$\mathrm{CCV}(\mu, \mathbb{X}) = \mathsf{E}\, \nu\big(\mu X, \bar{X}\big).$$
- Ожидаемая переобученность (Expected OverFitting):
$$\mathrm{EOF}(\mu, \mathbb{X}) = \mathsf{E}\, \delta(\mu, X, \bar{X}).$$
- Вероятность большой частоты ошибок на контроле:
$$R_\varepsilon(\mu, \mathbb{X}) = \mathsf{P}\big[\nu(\mu X, \bar{X}) \geqslant \varepsilon\big].$$
- Вероятность переобучения:
$$Q_\varepsilon(\mu, \mathbb{X}) = \mathsf{P}\big[\delta(\mu, X, \bar{X}) \geqslant \varepsilon\big].$$

Определим бинарные отношения на множестве алгоритмов $\mathbb{A}$:

частичный порядок $a \leqslant b$: $I(a,x) \leqslant I(b,x)$ для всех $x \in \mathbb{X}$;

предшествование $a \prec b$: $a \leqslant b$ и $\|b-a\| = 1$.

Определение

Граф расслоения–связности $\langle \mathbb{A}, E \rangle$:

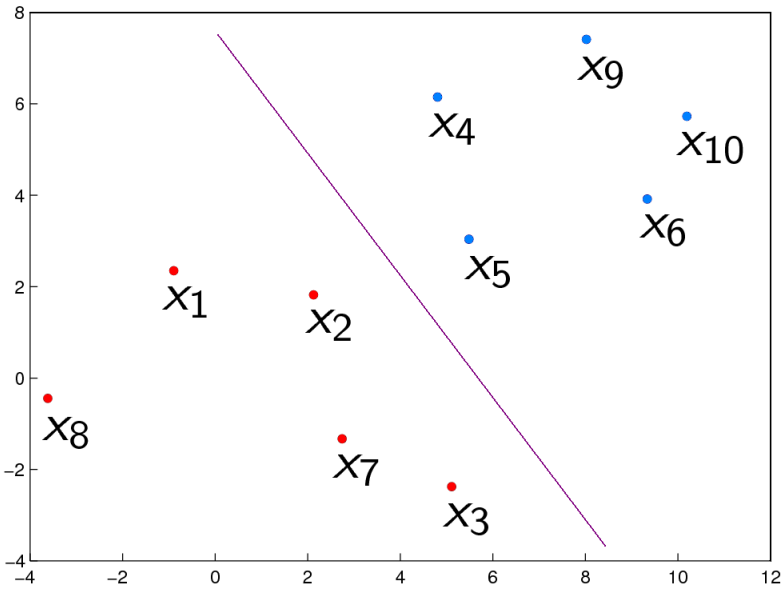
$\mathbb{A}$ — *множество попарно различных векторов ошибок;*

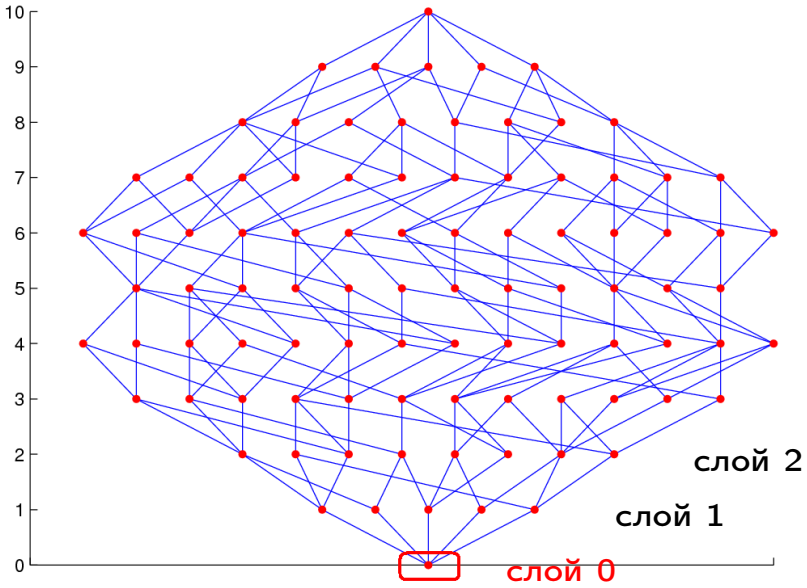
$E = \{(a,b) \colon a \prec b\}$.

Свойства графа расслоения–связности:

- это подграф диаграммы Хáссе отношения порядка $\leqslant$ на $\mathbb{A}$;
- каждому ребру (a,b) соответствует объект $x_{ab} \in \mathbb{X}$, такой, что $I(a,x_{ab}) = 0, \quad I(b,x_{ab}) = 1$;
- граф является многодольным со слоями $A_m = \{a \in \mathbb{A} \colon m(a,\mathbb{X}) = m\}, \quad m = 0,\ldots,L$;

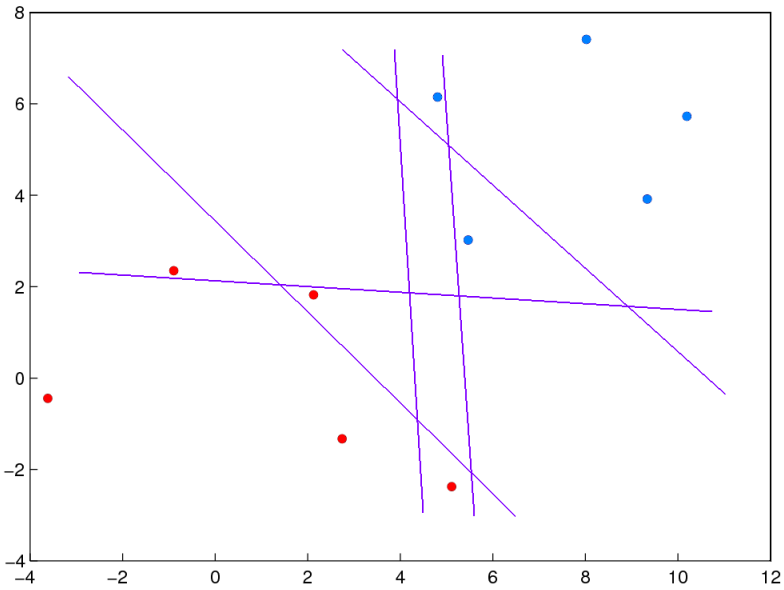
Пример. Семейство линейных алгоритмов классификации

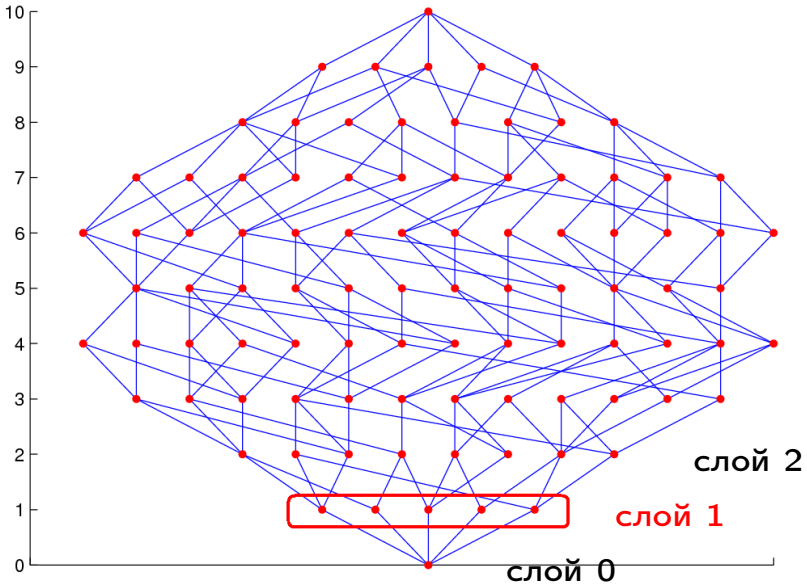




	слой 0
x_1	0
x_2	0
x_3	0
x_4	0
x_5	0
x_6	0
x_7	0
x_8	0
x_9	0
x_{10}	0

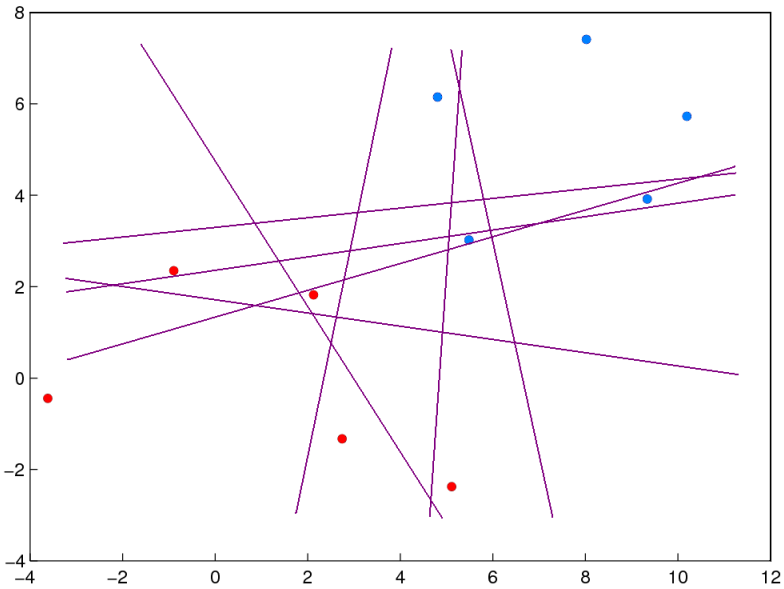
Пример. Семейство линейных алгоритмов классификации

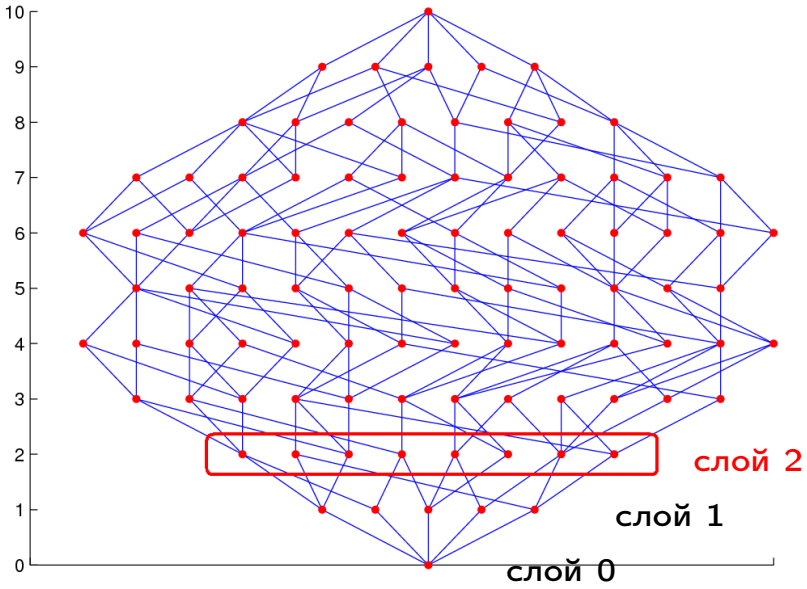




	слой 0	слой 1				
x_1	0	1	0	0	0	0
x_2	0	0	1	0	0	0
x_3	0	0	0	1	0	0
x_4	0	0	0	0	1	0
x_5	0	0	0	0	0	1
x_6	0	0	0	0	0	0
x_7	0	0	0	0	0	0
x_8	0	0	0	0	0	0
x_9	0	0	0	0	0	0
x_{10}	0	0	0	0	0	0

Пример. Семейство линейных алгоритмов классификации





	слой 0	слой 1					слой 2								
x_1	0	1	0	0	0	0	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	...

Определение

Верхняя связность $u(a)$ алгоритма a — это число всех рёбер, исходящих из вершины a:

$$u(a) = |X_a|, \quad X_a = \left\{ x_{ab} \in \mathbb{X} \;\middle|\; a \prec b \right\};$$

X_a называется порождающим множеством алгоритма a.

Определение

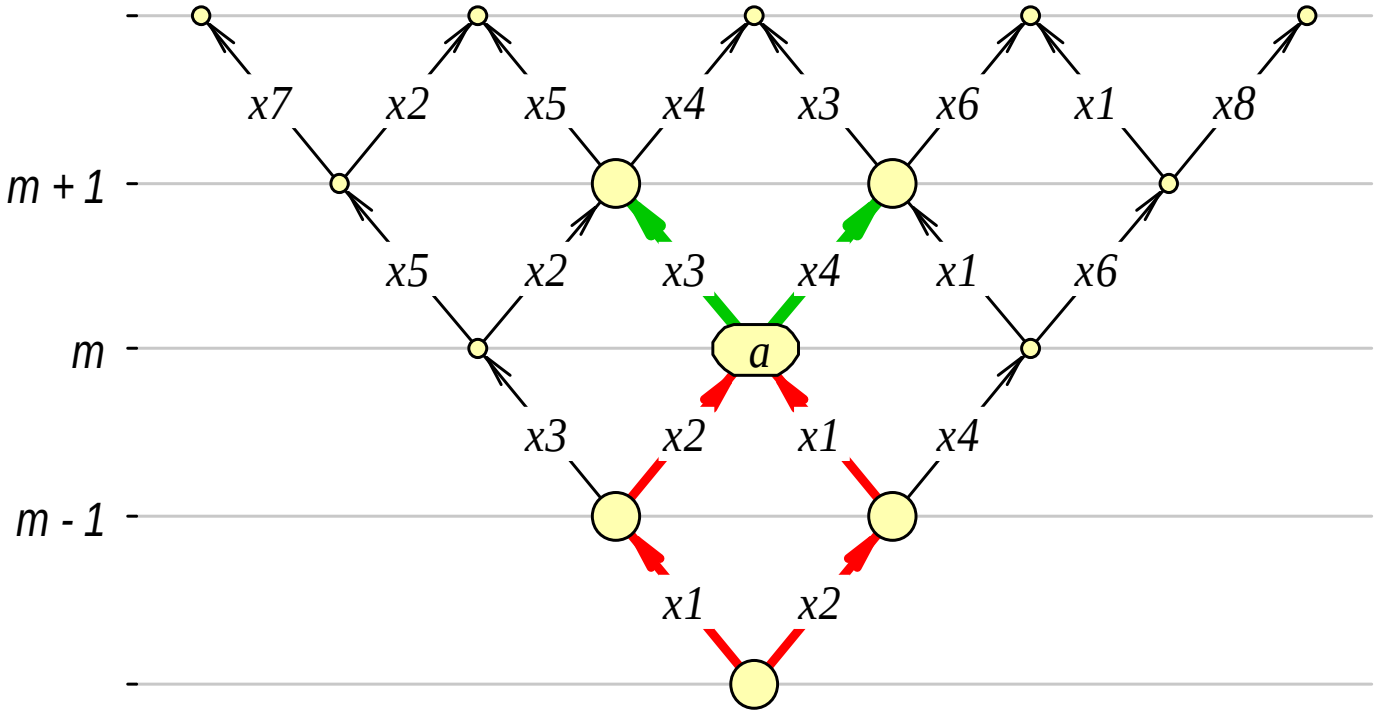
Неполноценность $q(a)$ алгоритма a — это число различных объектов, соответствующих всем рёбрам на путях, ведущих в a:

$$q(a) = |X'_a|, \quad X'_a = \left\{ x \in \mathbb{X} \;\middle|\; \exists b \in \mathbb{A} \colon b < a,\ I(b,x) < I(a,x) \right\};$$

X'_a называется запрещающим множеством алгоритма a.

Пример: двумерная сеть алгоритмов

Верхняя связность алгоритма a: $X_a = \{x3, x4\}$, $u(a) = |X_a| = 2$;
Неполноценность алгоритма a: $X'_a = \{x1, x2\}$, $q(a) = |X'_a| = 2$;



Основная лемма: если $\mu X = a$, то $X_a \subseteq X$ и $X'_a \subseteq \bar{X}$.

Теорема (Воронцов, Решетняк, Ивахненко, 2010)

Для любого метода минимизации эмпирического риска μ, *любых* $\mathbb{X}$, $\mathbb{A}$ *и* $\varepsilon \in (0,1)$

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

где $u = |X_a|$ — *верхняя связность алгоритма* a,
$q = |X'_a|$ — *неполноценность алгоритма* a,
$m = m(a, \mathbb{X})$ — *число ошибок алгоритма* a,

$$\mathcal{H}_L^{\ell,\, m}(z) = \sum_{s=0}^{\lfloor z \rfloor} \frac{C_m^s C_{L-m}^{\ell-s}}{C_L^\ell}, \quad z = 0, \ldots, \ell$$

— *функция гипергеометрического распределения.*

- Минимизация эмпирического риска — «модельный» метод обучения, не применяемый на практике (т.к. функционал не является ни гладким, ни выпуклым).
- В то же время для МЭР удалось получить слабо завышенные оценки вероятности переобучения.

Вопрос: можно ли моделировать реальные методы обучения с помощью МЭР?

Изучим вопрос о возможности моделирования логистической регрессии.

Семейство линейных классификаторов: $a_w(x) = \operatorname{sign}\langle w, x\rangle$.

МЭР: $\mu X = \operatorname*{argmin}_w \sum_{i=1}^{\ell} [\langle w, x_i\rangle y_i < 0]$.

Логистическая регрессия: $\mu_{\mathsf{LR}} X = \operatorname*{argmin}_w \sum_{i=1}^{\ell} \log\left(1 + \exp(-\langle w, x_i\rangle y_i)\right)$.

Комбинаторные оценки вероятности переобучения имеют вид

$$B_\varepsilon = \sum_{a \in \mathbb{A}} b(a),$$

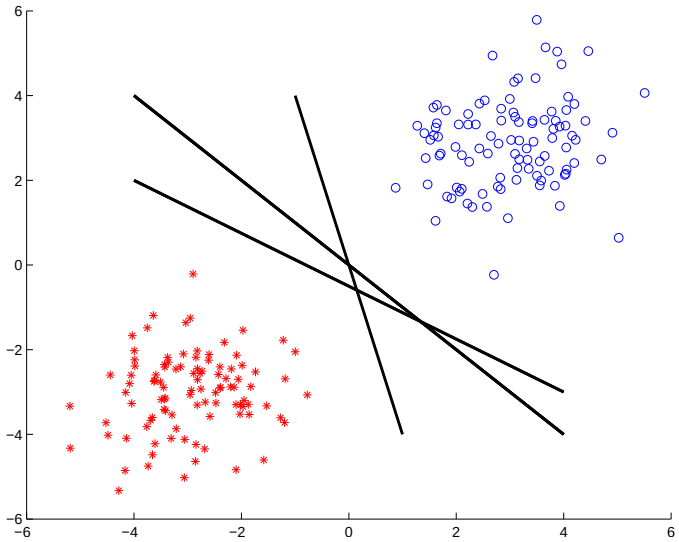
где a — вершины графа расслоения-связности.

Существенный вклад вносят лишь алгоритмы из нижних слоев!

Для вычисления вероятности переобучения МЭР необходимо решить две задачи:

1. Обход графа расслоения-связности семейства линейных классификаторов.
2. Приближенное вычисление Q_ε по небольшой доле алгоритмов.

Семейство линейных классификаторов



$$a(x) = \operatorname{sign}\langle w, x\rangle$$

Все три классификатора лежат в одном *классе эквивалентности*, т.к. имеют одинаковые векторы ошибок.

Вопросы:

1. Как устроены классы эквивалентности линейных классификаторов?
2. Как для данного алгоритма найти соседние с ним?

Преобразование двойственности

- Как объекты, так и веса w линейного классификатора являются векторами из $\mathbb{R}^d$.
- Объекты интерпретируются как точки, веса — как нормали гиперплоскостей.
- Как обменять их ролями?

Определим *преобразование двойственности* $\mathcal{D}$:

- произвольная точка $x \in \mathbb{R}^d$ переводится в гиперплоскость:

$$\mathcal{D}(x) = \{w \in \mathbb{R}^d \mid \langle w, x \rangle = 0\}.$$

- произвольная гиперплоскость $h = \{x \in \mathbb{R}^d \mid \langle w, x \rangle = 0\}$ переводится в точку:

$$\mathcal{D}(h) = w.$$

Преобразование двойственности

С помощью преобразования двойственности переведем выборку объектов $\mathbb{X} = \{x_1, \ldots, x_L\}$ в выборку гиперплоскостей $\mathbb{H} = \{h_1, \ldots, h_L\}$.

Гиперплоскость $h_i \in \mathbb{H}$ делит все пространство на два полупространства:

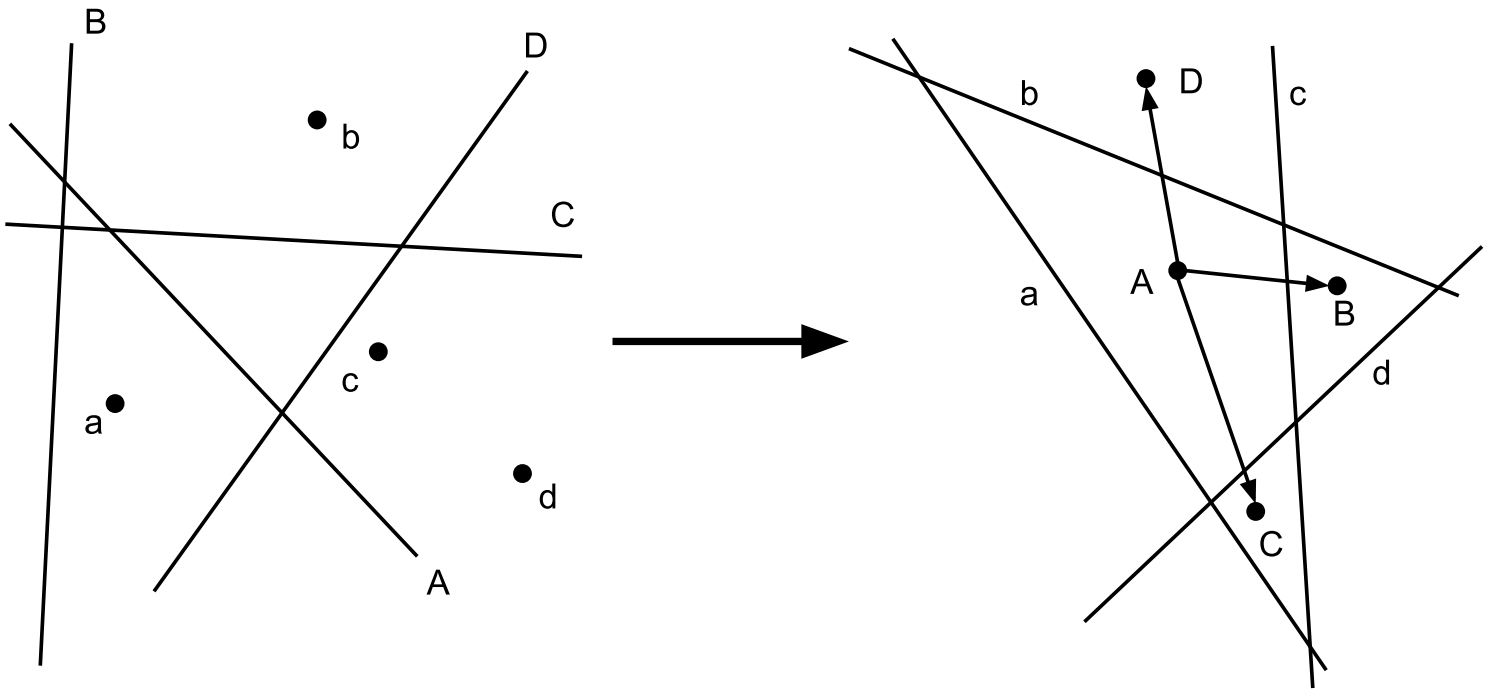
$$h_i^+ = \{w \in \mathbb{R}^d \mid \operatorname{sign}\langle w, x_i\rangle = y_i\},$$
$$h_i^- = \{w \in \mathbb{R}^d \mid \operatorname{sign}\langle w, x_i\rangle = -y_i\}$$

В одном полупространстве лежат линейные классификаторы, правильно классифицирующие x_i, в другом — неправильно классифицирующие.

Пусть a — линейный классификатор, $(s_1, \ldots, s_L)$, $s_i \in \{-, +\}$ — его вектор ошибок.
Тогда множество всех линейных классификаторов с такими же векторами ошибок можно найти как

$$C(a) = \bigcap_{i=1}^{L} h_i^{s_i}.$$

Преобразование двойственности



Конфигурация гиперплоскостей

Гиперплоскости из $\mathbb{H}$ разбивают $\mathbb{R}^d$ на выпуклые многогранники, называемые *ячейками конфигурации (cells of arrangement)*.

Само же разбиение пространства на эти области называется *конфигурацией гиперплоскостей (arrangement of hyperplanes)*.

Лемма

Пусть $\mathbb{X}$ — выборка в общем положении, имеющая константный признак, $\mathbb{H}$ — двойственный ей набор гиперплоскостей, $\mathbb{A}$ — семейство линейных классификаторов для этой выборки.
Тогда:

- *Существует взаимно однозначное соответствие между ячейками конфигурации гиперплоскостей $\mathbb{H}$ и алгоритмами из $\mathbb{A}_h$.*
- *Если a_1 и a_2 — соседние алгоритмы, то соответствующие им ячейки C_{a_1} и C_{a_2} имеют общую грань размерности $(d-1)$.*

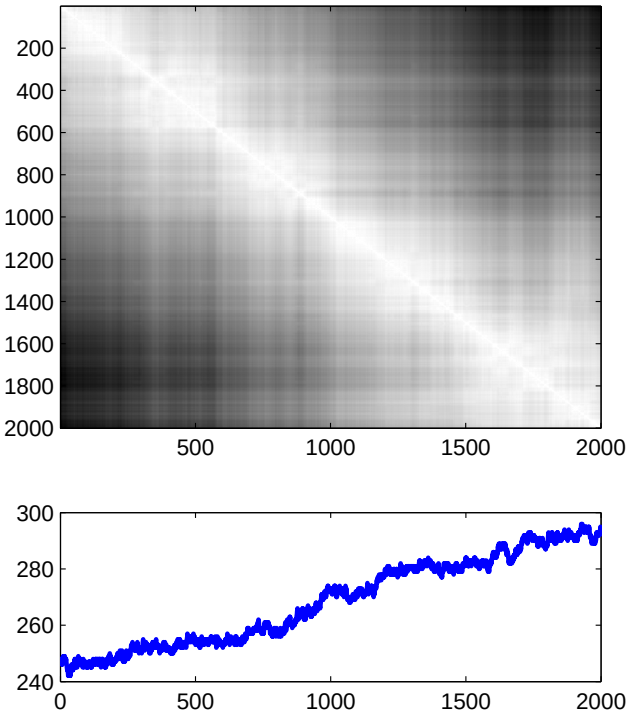
Поиск вдоль луча

- пусть задан линейный классификатор $a_{w_0}(x)$, которому соответствует точка w_0 в некоторой ячейке конфигурации;
- соседние с ним алгоритмы в графе расслоения-связности соответствуют соседним ячейкам;
- построим в двойственном пространстве луч $\{w_0 + tu \mid t \geqslant 0\}$ в направлении u;
- пересечение луча с гиперплоскостью h_i определяется уравнением $\langle w_0 + tu, x_i \rangle = 0$;
- решение: $t_i = -\frac{\langle w_0, x_i \rangle}{\langle u, x_i \rangle}$;
- пусть $t_{(1)}$ и $t_{(2)}$ — первое и второе наименьшие положительные значения из $\{t_i\}$;
- соседом алгоритма w_0 вдоль направления u будет алгоритм $w' = w_0 + \frac{1}{2}(t_{(1)} + t_{(2)})$.

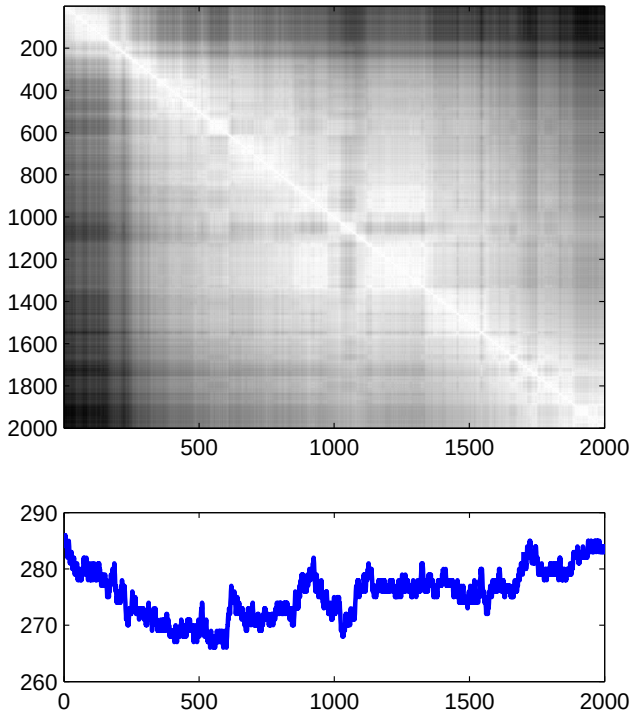
Приближенное вычисление Q_ε

Предлагается с помощью случайных блужданий получать выборку алгоритмов, и затем вычислять оценку только по ней.

Простое случайное блуждание:



Случайное блуждание с притяжением:



Вход: Стартовая вершина v_0, размер выборки n, число блужданий N_{RW}, вероятность перехода вверх $p_{\mathbf{up}}$.

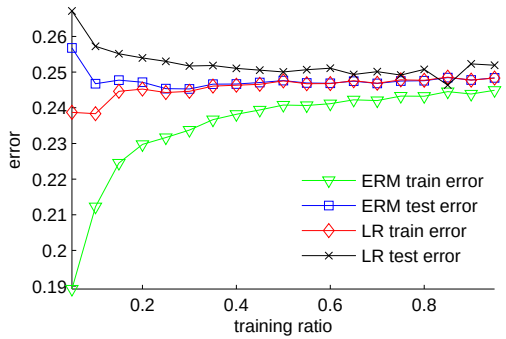
Выход: Выборка $v_1, v_2, \ldots, v_n$.

1: $u_i = v_0$, $i = 1, \ldots, N_{\mathsf{RW}}$; // *текущие вершины в каждом из блужданий*
2: $k = 0$; // *текущие размер выборки*
3: **цикл**
4: **для** $i = 1, \ldots, N_{\mathsf{RW}}$
5: Найти соседа u_i' алгоритма u_i вдоль случайного направления;
6: **если** u_i' расположен выше u_i в графе **то**
7: **с вероятностью** $p_{\mathbf{up}}$
8: $u_i = u_i'$;
9: **иначе**
10: $u_i = u_i$;
11: $k = k + 1$; $v_k = u_i$;
12: **если** $k = n$ **то**
13: **выход**

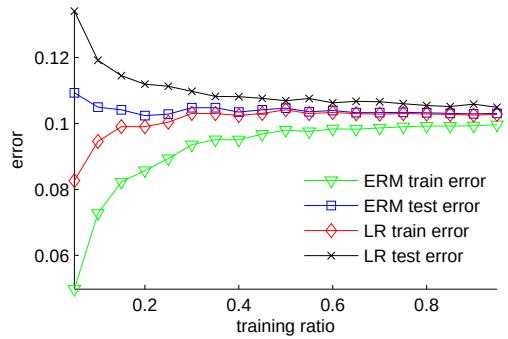
Постановка эксперимента

- генеральная выборка $\mathbb{X}$ разбивается на обучающую $\mathbb{X}^L$ и контрольную $\mathbb{X}^K$;
- логистическая регрессия обучается на $\mathbb{X}^L$, ее качество оценивается по контрольной выборке $\mathbb{X}^K$;
- по обучающей выборке $\mathbb{X}^L$ сэмплируется выборка алгоритмов $a_1, \ldots, a_n$;
- обучающая выборка многократно разбивается на подвыборки X^ℓ и X^k, по которым оценивается ошибка на обучении и контроле для ПМЭР (оценивание по Монте-Карло);
- оценка для ошибки на контроле X^k ПМЭР выступает в качестве приближения для ошибки на контроле $\mathbb{X}^K$ логистической регрессии.

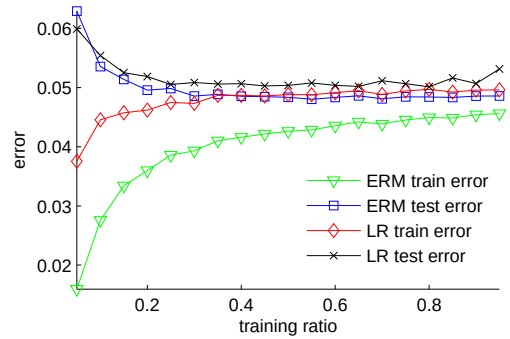
Эксперимент 1, точность оценок



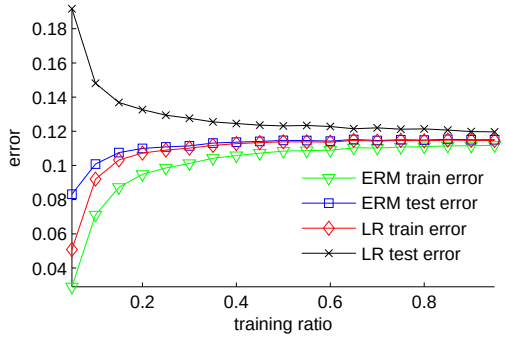
UCI:wine



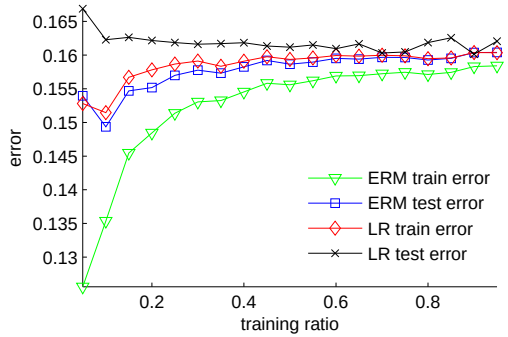
UCI:waveform



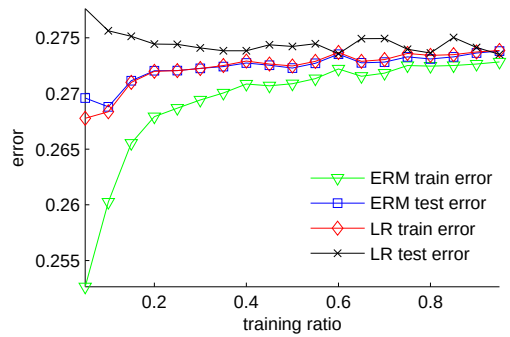
UCI:pageblocks



UCI:Optdigits



UCI:pendigits



UCI:Letter

- Сравнение кривых обучения логистической регрессии и ПМЭР.
- Частоты ошибок МЭР оцениваются только по обучающей выборке.
- Вывод: оценки для МЭР подходят и для логистической регрессии.
- Вывод: оценки хорошо вычисляются случайными блужданиями.

Эксперимент 2, сравнение оценок

Все оценки вычислены по выборке, полученной случайным блужданием.

- MC — оценка Монте-Карло
- VC — оценка Вапника-Червоненкиса
- SC — комбинаторная оценка расслоения-связности

Task	TrainErr	TestErr	Overfit	MC	VC	SC	PAC DD
UCI:glass	0.046	0.075	0.029	0.078	0.211	0.140	0.738
UCI:Liver dis.	0.299	0.314	0.015	0.060	0.261	0.209	1.067
UCI:Ionosphere	0.049	0.125	0.077	0.052	0.150	0.112	1.153
UCI:Wdbc	0.001	0.056	0.055	0.032	0.071	0.043	0.705
UCI:Australian	0.122	0.136	0.013	0.030	0.137	0.110	0.678
UCI:pima	0.220	0.227	0.007	0.028	0.159	0.127	0.749

Выводы:

- полученная блужданием выборка хорошо подходит для вычисления оценок;
- комбинаторные оценки значительно точнее PAC-Bayes оценок.

Линейные композиции линейных классификаторов

Будем рассматривать композиции вида

$$a(x) = \operatorname{sign} \sum_{i=1}^{p} \operatorname{th}\langle w_i, x \rangle$$

- Базовый алгоритм — линейный классификатор низкой размерности, построенный логистической регрессией.
- Критерий отбора признаков $F = \{f_1, \ldots, f_d\}$ — обращенная комбинаторная оценка:

$$\nu(\mu X, \bar{X}) \leqslant \nu(\mu X, X) + \varepsilon(\mathbb{A}, \mu, \ell, k, \eta) \to \min_{G \subseteq F};$$

  (берем $\eta = 1/2$).
- Для вычисления комбинаторной оценки на каждом наборе признаков используется случайные блуждания.
- Композиция — простое голосование, настраивается методом ComBoost.

Построение композиции методом ComBoost

Вход: Выборка X; параметры T, ℓ_0, ℓ_1;
Выход: Базовые линейные классификаторы $b_1, \ldots, b_T$

1: Инициализировать веса и отсутпы:
$w_i = 1$, $M_i = 0$ для всех $i = 1, \ldots, \ell$;
2: **для** $t = 1, \ldots, T$, пока не выполнен критерий останова
3: Обучить базовый алгоритм:
$b_t := \operatorname{argmin}_b Q(b, X, W)$;
4: Обновить значения отступов:
$M_i = M_i + y_i b_t(x_i)$ для всех $i = 1, \ldots, L$;
5: упорядочить выборку X по возрастанию отступов M_i;
6: Отобрать объекты для обучения следующего базового алгоритма:
$w_i = [\ell_0 \leqslant i \leqslant \ell_1]$;

Эксперимент с композициями

	statlog	waveform	wine	faults
CombCV	**85,35**	87,56	**71,63**	**73,62**
CombQeps	**85,08**	**87,66**	71,08	**71,65**
CV	84,04	**88,13**	**71,52**	70,86
Emp	80,77	87,34	71,49	71,09
PacBayes DD	82,13	87,17	64,68	67,67

Показан процент правильных ответов на контрольной выборке (с усреднением по 10 разбиениям). **Жирным** выделены два лучших результата на каждой задаче.

Критерии отбора признаков:

- CombCV и CombQeps — оценка Монте-Карло и комбинаторная оценка, посчитанные по сэмплированной выборке алгоритмов.
- CV — оценка кросс-валидации для логистической регрессии.
- Emp — ошибка на обучении.
- PacBayes DD — одна из последних оценок, полученных в рамках теории статистического обучения.

Спасибо за внимание!

Евгений Соколов

sokolov.evg@gmail.com