

# КОНЕЧНО-СХОДЯЩИЕСЯ АЛГОРИТМЫ В МАТЕМАТИЧЕСКОЙ ДИАГНОСТИКЕ\*

В. А. Бондарко  
vbondarko@gmail.com

2 марта 2022 г.

**1°.** Конечно-сходящиеся алгоритмы решения систем неравенств были первоначально предложены В. А. Якубовичем [1] для решения задачи о настройке весов перцептрона Розенблатта [2]. Постановка этой задачи следующая: в  $\mathbb{R}^n$  имеются два множества точек (два *класса*)

$$P = \{p_j\}_{j=1}^{m_1} \text{ и } Q = \{q_j\}_{j=1}^{m_2}.$$

Требуется строго разделить их гиперплоскостью, т. е. отыскать такой вектор  $\omega \in \mathbb{R}^n$  и скаляр  $\beta$ , что

$$\begin{aligned} \langle \omega, p_j \rangle + \beta &> 0 \quad \forall j \in 1 : m_1, \\ \langle \omega, q_j \rangle + \beta &< 0 \quad \forall j \in 1 : m_2. \end{aligned} \tag{1}$$

*Весами* перцептрона называют набор из компонент  $\omega$  и величины  $\beta$ .

В терминологии теории распознавания образов обучение перцептрона с учителем, т.е. по известным представителям классов  $P$  и  $Q$ , заключается в решении системы неравенств (1). После обучения перцептрон классифицирует предъявляемые ему точки  $x_i \in \mathbb{R}^n$  в соответствии со знаком величины  $\langle \omega, x_i \rangle + \beta$ .

**2°.** Нетрудно видеть, что задача настройки весов почти в точности совпадает с задачей диагностики, как она поставлена в недавно прочитанном на этом семинаре докладе [3]. Единственное существенное различие состоит в том, что для настройки требуется априорная разрешимость системы неравенств (1), а алгоритм диагностики должен ответить на вопрос о разрешимости этой системы. К счастью, предложенный В. А. Якубовичем алгоритм настройки,

---

\*Семинар по оптимизации, машинному обучению и искусственному интеллекту «O&ML»  
<http://www.apmath.spbu.ru/oml/>

который в его честь получил название ЯВА, вполне способен дать ответ на этот вопрос.

Чтобы пояснить логику алгоритма, переформулируем задачу в духе двойственности: определим набор векторов

$$\varphi_i = - \begin{bmatrix} p_i \\ 1 \end{bmatrix} \text{ при } i = 1, \dots, m_1, \quad \varphi_{m_1+i} = \begin{bmatrix} q_i \\ 1 \end{bmatrix} \text{ при } i = 1, \dots, m_2,$$

и зациклим этот набор в бесконечную последовательность  $\{\varphi_k\}_{k=1}^{\infty}$ . Искомые величины  $\omega$  и  $\beta$  объединим в один вектор  $\tau = \begin{bmatrix} \omega \\ \beta \end{bmatrix}$  и обозначим  $\ell_k(\tau) = \langle \tau, \varphi_k \rangle$ , тогда система неравенств (1) будет эквивалентна неравенствам

$$\ell_k(\tau) < 0, \quad k = 1, 2, \dots \quad (2)$$

Отыскав решение этой системы, мы можем построить вектор  $\omega$  по первым  $n$  компонентам решения, а последнюю компоненту выбрать в качестве  $\beta$ .

Фиксируем произвольную стартовую точку  $\tau_1$  и при  $k = 1, 2, \dots$  положим

$$\tau_{k+1} = \begin{cases} \tau_k - \frac{\ell_k(\tau_k) + 1}{|\varphi_k|^2} \varphi_k, & \text{если } \ell_k(\tau_k) \geq 0, \\ \tau_k, & \text{если } \ell_k(\tau_k) < 0. \end{cases} \quad (3)$$

Необходимо подчеркнуть, что увеличивая  $k$  (номер шага алгоритма), мы автоматически переходим к следующему значению  $\varphi_k$ : в первый момент проверяется значение  $\ell_1(\tau_1) = \langle \tau_1, \varphi_1 \rangle$ , затем с полученным значением  $\tau_2$  (которое может, в частности, и не отличаться от  $\tau_1$ ) мы проверяем знак  $\ell_2(\tau_2) = \langle \tau_2, \varphi_2 \rangle$ , и т. д.

Чтобы сформулировать правило останова, сделаем следующее предположение: если у системы неравенств (2) существует решение  $\tau_*$ , то

$$\ell_k(\tau_*) < -1, \quad k = 1, 2, \dots, \quad (4)$$

$$|\tau_*| \leq R, \quad (5)$$

причем значение  $R$  должно быть известно. Неравенство (4) общности не уменьшает, поскольку значение  $\ell_k(\tau)$  линейно зависит от аргумента. Условие (5) не слишком ограничительное, поскольку оценка  $|\tau_*|$  может быть сколь угодно грубой.

Итак, правило останова:

- 1) Поскольку последовательность  $\{\varphi_k\}_{k=1}^{\infty}$  получена зацикливанием конечного набора  $\{\varphi_i\}_{i=1}^{m_1+m_2}$ , выполнение  $m_1 + m_2$  раз подряд неравенств  $\ell_k(\tau_k) < 0$  — это сигнал к остановке алгоритма с итоговым вектором  $\tau_k$ , который задает разделяющую гиперплоскость.

2) Если количество коррекций вектора  $\tau_k$  превысило величину

$$M_{\max} := (|\tau_1| + R)^2 \max_j |\varphi_j|^2, \quad (6)$$

то алгоритм останавливается с итоговым выводом, что разделение классов невозможно.

Это один из вариантов алгоритма ЯВА, который в более общем виде зависит еще от нескольких параметров. Здесь для уменьшения громоздкости этим параметрам приданы конкретные значения. В результате этот вариант алгоритма ЯВА становится похож на алгоритм Розенблатта:

$$\tau_{k+1} = \begin{cases} \tau_k - C \frac{\ell_k(\tau_k)}{|\varphi_k|^2} \varphi_k, & \text{если } \ell_k(\tau_k) \geq 0, \\ \tau_k, & \text{если } \ell_k(\tau_k) < 0, \end{cases} \quad k = 1, 2, \dots,$$

где  $C$  — произвольная положительная константа. Стоит еще отметить, что алгоритм ЯВА в общем виде предназначен для решения счетной системы неравенств (2) при  $\ell_k(\tau) = \alpha_k - \gamma_k \langle \tau, \varphi_k \rangle$ , причем векторы  $\varphi_k$ , вообще говоря, не обязательно должны выбираться из конечного множества.

**ТЕОРЕМА.** *Если классы  $P$  и  $Q$  разделимы, то алгоритм (3) порождает последовательность векторов  $\tau_k$ , которая за конечное число шагов  $N$  после  $M$  коррекций сходится к решению неравенств (2). Справедливы оценки*

$$M \leq M_{\max}, \quad N \leq (m_1 + m_2) M. \quad (7)$$

*Если первое неравенство нарушено, то классы  $P$  и  $Q$  неразделимы.*

**Доказательство.** Определим функцию Ляпунова  $V(\tau) = |\tau - \tau_*|^2$  и убедимся, что при каждой коррекции вектора  $\tau$  она убывает на отделенную от нуля величину. Тем самым теорема будет доказана.

Итак, пусть в момент  $k$  неравенство (2) для  $\tau = \tau_k$  нарушено:  $\ell_k(\tau_k) \geq 0$ . Тогда

$$d_k := V(\tau_k) - V(\tau_{k+1}) = |\tau_k - \tau_*|^2 - |\tau_{k+1} - \tau_*|^2 = \langle \tau_k - \tau_{k+1}, \tau_k + \tau_{k+1} - 2\tau_* \rangle.$$

Учитывая, что  $\tau_k + \tau_{k+1} = 2\tau_k - (\tau_k - \tau_{k+1})$ , получаем

$$\begin{aligned} d_k &= -|\tau_k - \tau_{k+1}|^2 + 2 \langle \tau_k - \tau_{k+1}, \tau_k \rangle - 2 \langle \tau_k - \tau_{k+1}, \tau_* \rangle = \\ &= -\frac{[\ell_k(\tau_k) + 1]^2}{|\varphi_k|^2} + 2 \frac{[\ell_k(\tau_k) + 1]}{|\varphi_k|^2} (\langle \tau_k, \varphi_k \rangle + 1 - 1) - 2 \frac{[\ell_k(\tau_k) + 1]}{|\varphi_k|^2} \langle \tau_*, \varphi_k \rangle = \\ &= \frac{[\ell_k(\tau_k) + 1]^2}{|\varphi_k|^2} - 2 \frac{[\ell_k(\tau_k) + 1]}{|\varphi_k|^2} [\ell_k(\tau_*) + 1]. \end{aligned}$$

Коррекции  $\tau_k$  производятся при условии  $\ell_k(\tau_k) \geq 0$ , поэтому  $[\ell_k(\tau_k) + 1] > 1$ , а  $\ell_k(\tau_*) < -1$  по сделанным предположениям. Следовательно, обозначив  $\beta := \max |\varphi_k|^2$ , получаем  $d_k > 1/\beta$ . Значения  $V(\tau)$  неотрицательны, поэтому

$$V(\tau_1) \geq V(\tau_1) - V(\tau_K) = \sum_{k=1}^K \delta_k \geq M/\beta \Rightarrow M \leq V(\tau_1)\beta,$$

т. е. первая из оценок (7) справедлива. Вторая оценка очевидна, поскольку коррекции не могут происходить реже, чем один раз за  $m_1 + m_2$  шагов.  $\square$

**З а м е ч а н и е.** Из доказательства теоремы очевидно, что приведенная там оценка количества коррекций чрезмерно грубая. Сделано это специально с целью максимально упростить описание алгоритма и формулировку теоремы. Если желательно уточнить оценку числа коррекций, то можно присвоить некоторой величине  $s_1$  нулевое стартовое значение, а при  $k = 1, 2, \dots$  дополнить алгоритм (3) следующим образом:

$$s_{k+1} = \begin{cases} s_k + \frac{[\ell_k(\tau_k) + 1]^2}{|\varphi_k|^2}, & \text{если } \ell_k(\tau_k) \geq 0, \\ s_k, & \text{если } \ell_k(\tau_k) < 0. \end{cases} \quad (8)$$

Правило останова при этом изменятся:

- 1) Если неравенства  $\ell_k(\tau_k) < 0$  выполнены  $m_1 + m_2$  раз подряд, то алгоритм останавливается с итоговым вектором  $\tau_k$ , который задает разделяющую гиперплоскость.
- 2) Если при некотором  $k$  значение  $s_k$  превысило  $(|\tau_1| + R)^2$ , то алгоритм останавливается с итоговым выводом, что разделение классов невозможно.

Проделанные в ходе доказательства теоремы рассуждения имеют явный геометрический смысл. Рассмотрим для простоты графического изображения случай двумерных векторов  $\tau$  и  $\varphi$ . Если на  $k$ -м шаге вектор  $\tau_k$  находится в полупространстве  $\{\tau : \ell_k(\tau) > 0\}$ , т.е. левее и выше синей прямой на рис. 1, то происходит коррекция, и новое значение  $\tau_{k+1}$  попадает на границу полупространства  $\{\tau : \ell_k(\tau) < -1\}$ . При этом он приближается ко всем точкам этого полупространства, в том числе и к  $\tau_*$  — неизвестному решению всех неравенств (2), причем уменьшение этого расстояния отделено от нуля. Ясно, что таких коррекций может быть только конечное число.

Следующие два рисунка отображают работу моделирующей программы. Элементами классов  $P$  и  $Q$  служат точки на плоскости, они задаются нажатием левой/правой кнопки мыши и изображаются в виде красных/синих звездочек в левом поле каждого рисунка. Правое поле показывает коррекции векторов  $\tau_k$  в сопряженном пространстве (не слишком наглядно, поскольку

вместо пространственного изображения представляется плоская картинка). В случае успешной сходимости в левом поле изображается разделяющая прямая (рис. 2), а при невозможности разделения, как на рис. 3, алгоритм просто останавливается.

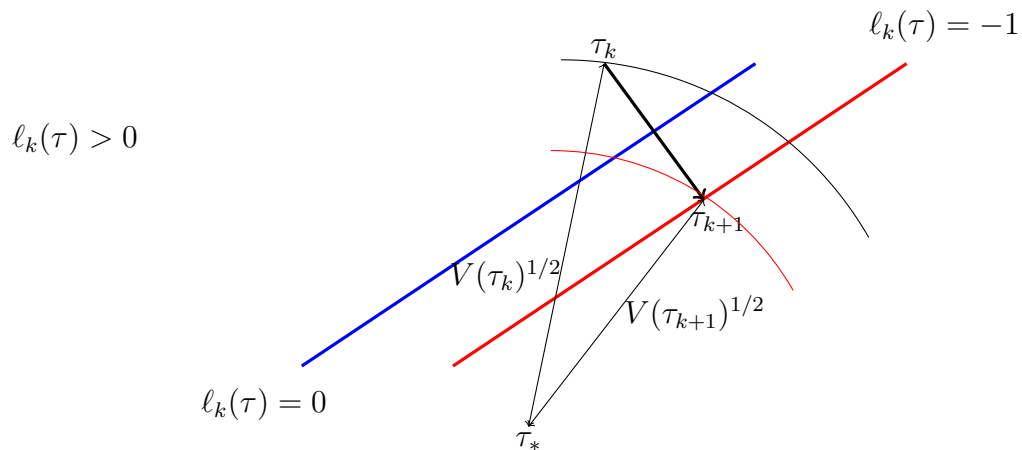


Рис. 1. Убывание расстояния до решения в результате коррекции

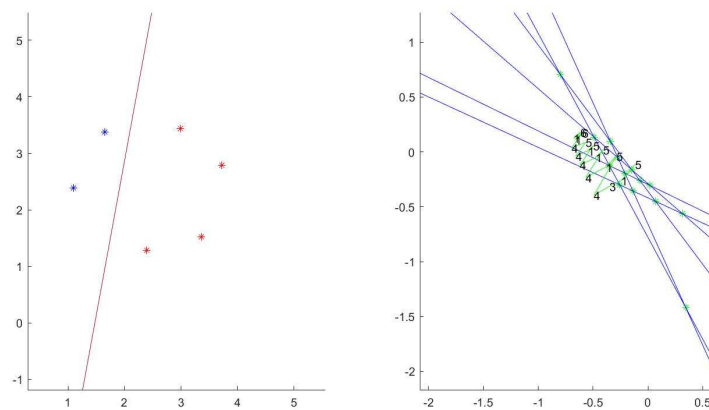


Рис. 2. Успешное разделение классов  $P$  и  $Q$

**3°.** Вернемся к геометрическому смыслу алгоритма ЯВА. Сразу приходит в голову, что полупространства тут не по-существу. Вместо них неравенства  $\ell_k(\tau) \leq 0$  могут выделять любые выпуклые множества. Задающие алгоритм формулы изменятся, но логика сохранится: если на очередном шаге вектор  $\tau_k$

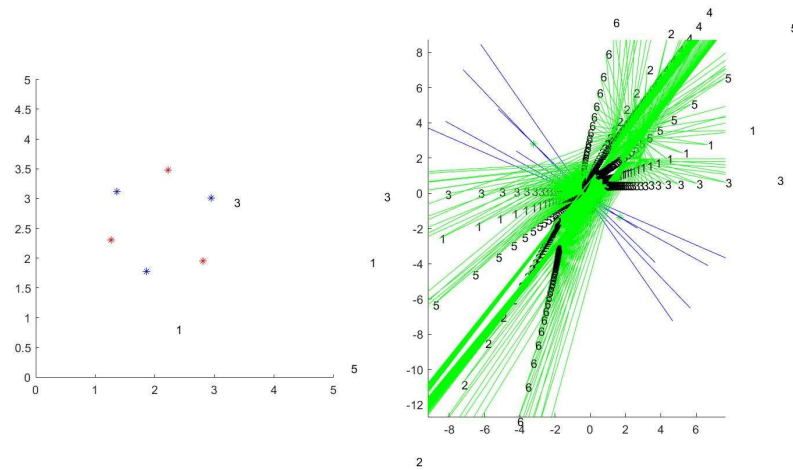


Рис. 3. Разделение классов  $P$  и  $Q$  невозможно

попал внутрь множества, то его следует оставить на месте. Если же он оказался вовне, то его следует спроектировать на границу множества  $\ell_k(\tau) \leq 0$ , или придвинуть к нему, или даже перенести вовнутрь. Главное, чтобы при этом существенно убывало расстояние от  $\tau_k$  до решения всей системы неравенств.

Первым и очень продуктивным шагом в этом направлении стало рассмотрение неравенств вида

$$|y_k - \langle \tau, \varphi_k - 1 \rangle| \leq C, \quad k = 0, 1, \dots \quad (9)$$

Эти неравенства выделяют в пространстве векторов  $\tau$  "полоску", т.е. промежуток между двумя гиперплоскостями. По этой причине первый алгоритм для решения этой системы получил название "Полоска-1", а затем появились "Полоска-2", "Векторная полоска", "Квадратичная полоска" и пр. Проще всего описать "Полоску-1": поскольку неравенства (9) задают пересечение полупространств  $y_k - \langle \tau, \varphi_k - 1 \rangle \geq -C$  и  $y_k - \langle \tau, \varphi_k - 1 \rangle \leq C$ , то достаточно просто дважды применить алгоритм ЯВА.

Неравенства (9) возникают при решении задачи адаптивного управления. Постановка этой задачи поясняется блок-схемой на рис. 4.

На объект управления с неизвестными параметрами поступает неизмеряемое возмущающее воздействие  $v_k$  и управляющее воздействие  $u_k$ . Измеряемый выход  $y_k$  и управляющее воздействие  $u_k$  доступны алгоритму адаптации  $A$ , который непосредственно в замкнутом контуре выделяет необходимую информацию  $\tau_k$  об объекте и на ее основе настраивает регулятор  $C(\tau_k)$ . Регулятор, в свою очередь, формирует управляющее воздействие. Система адаптивна в заданном классе объектов и возмущений, если после конечного переходного процесса она обеспечивает достижение некоторой цели управления. В идеальном случае при этом еще и оптимизируется некий функционал качества управления.

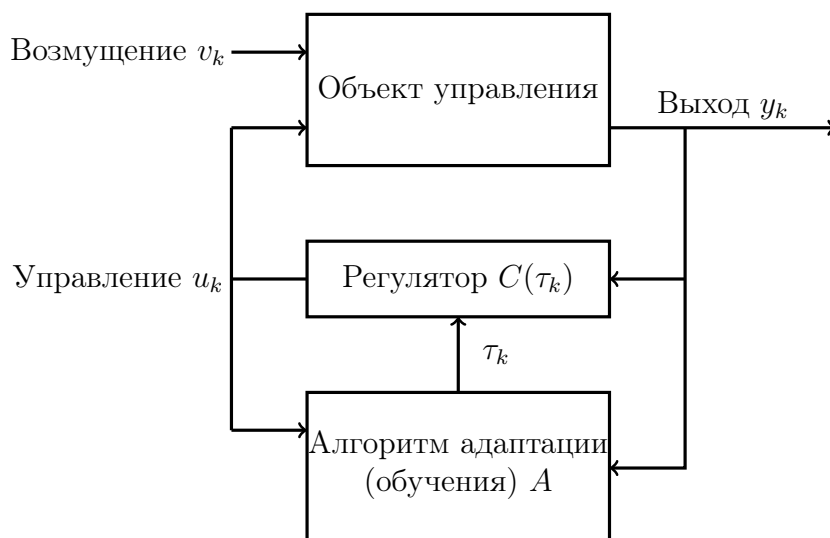


Рис. 4. Блок-схема адаптивного полосового фильтра–усилителя

Рассмотрим простейший пример. Уравнение объекта управления —

$$y_{k+1} = ay_k + bu_k + v_{k+1}, \quad k = 0, 1, \dots \quad (10)$$

Параметры  $a$  и  $b$  неизвестны, возмущение  $v_k$  неизвестно, но ограничено по абсолютной величине:

$$|v_k| \leq 1 \quad \forall k. \quad (11)$$

Требуется минимизировать  $\sup_{\{v_k\}} \overline{\lim}_k |y_k|$ , т.е. вогнать выход в максимально узкие границы. Оптимальное управление определяется из уравнения  $ay_k + bu_k = 0$ , но оно нереализуемо, поскольку  $a$  и  $b$  неизвестны.

Адаптивный регулятор: уравнение (10) перепишем в виде  $y_{k+1} - \langle \tau_*, \varphi_k \rangle = v_{k+1}$ , где  $\tau_* = \begin{bmatrix} a \\ b \end{bmatrix}$ ,  $\varphi_k = \begin{bmatrix} y_k \\ u_k \end{bmatrix}$ , тогда неравенства (11) при  $\tau = \tau_*$  эквивалентны тому, что

$$|y_{k+1} - \langle \tau, \varphi_k \rangle| < C \quad (12)$$

при  $\tau = \tau_*$ ,  $C = 1$ . Имея любое решение этих неравенств, мы можем определять управление из уравнения  $\langle \tau, \varphi_k \rangle = 0$ , обеспечивая при этом  $|y_k| \leq C$ . Решение системы будем искать алгоритмом "Полоска-1", т.е., по-существу, алгоритмом ЯВА. Для сходимости алгоритма приходится требовать неравенства  $C > 1$ , но значение  $C$  можно выбрать сколь угодно близко к единице.

Результаты моделирования представлены на рисунках 5 и 6. На рис. 5 зеленая ломаная линия изображает эволюцию вектора  $\tau_k$ , а красные и синие линии — границы полосок. Красная звездочка изображает вектор  $\tau_*$ , т.е. соответствует истинным значениям параметров. Видно, что предельное значение  $\tau_k$  не совпадает с  $\tau_*$ , но это и не требуется.

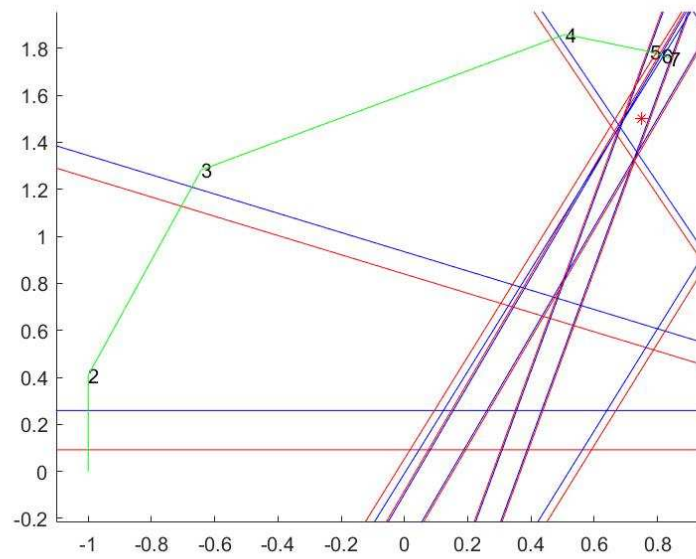
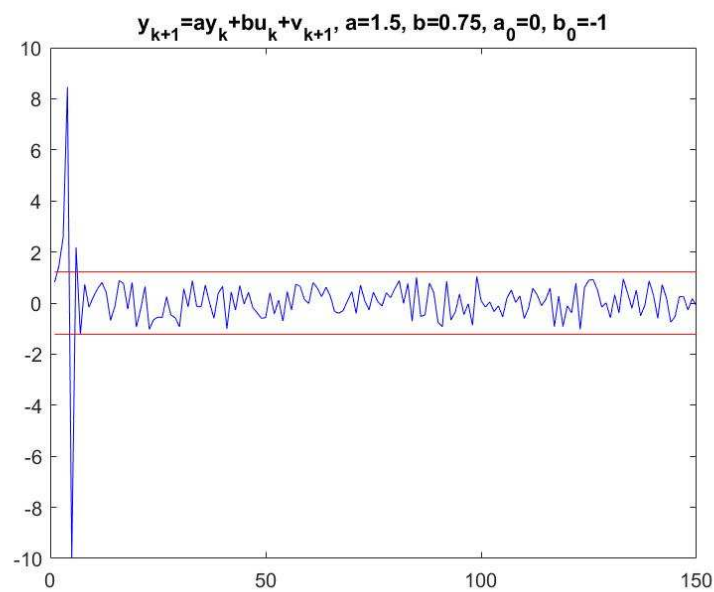
Рис. 5. Эволюция векторов  $\tau_k$ 

Рис. 6. Наблюдаемый выход адаптивной системы

На рис. 6 изображен наблюдаемый выход системы и границы, в которых он должен находиться после окончания переходного периода. Этот период заканчивается уже на седьмом шаге.



## ЛИТЕРАТУРА

1. Якубович В. А. *Некоторые общие теоретические принципы построения обучаемых опознающих систем. I* // Вычислительная техника и вопросы программирования. Л.: Изд-во ЛГУ, 1965. С. 3–71.
2. Rosenblatt F. (1962) Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington DC.
3. Малоземов В. Н., Плоткин А. В. *Строгое линейное отделение двух конечных множеств и линейное программирование* // Семинар «О&МЛ». Избранные доклады. 24 февраля 2022 г.  
(<http://www.apmath.spbu.ru/oml/rep22.shtml#0224>)