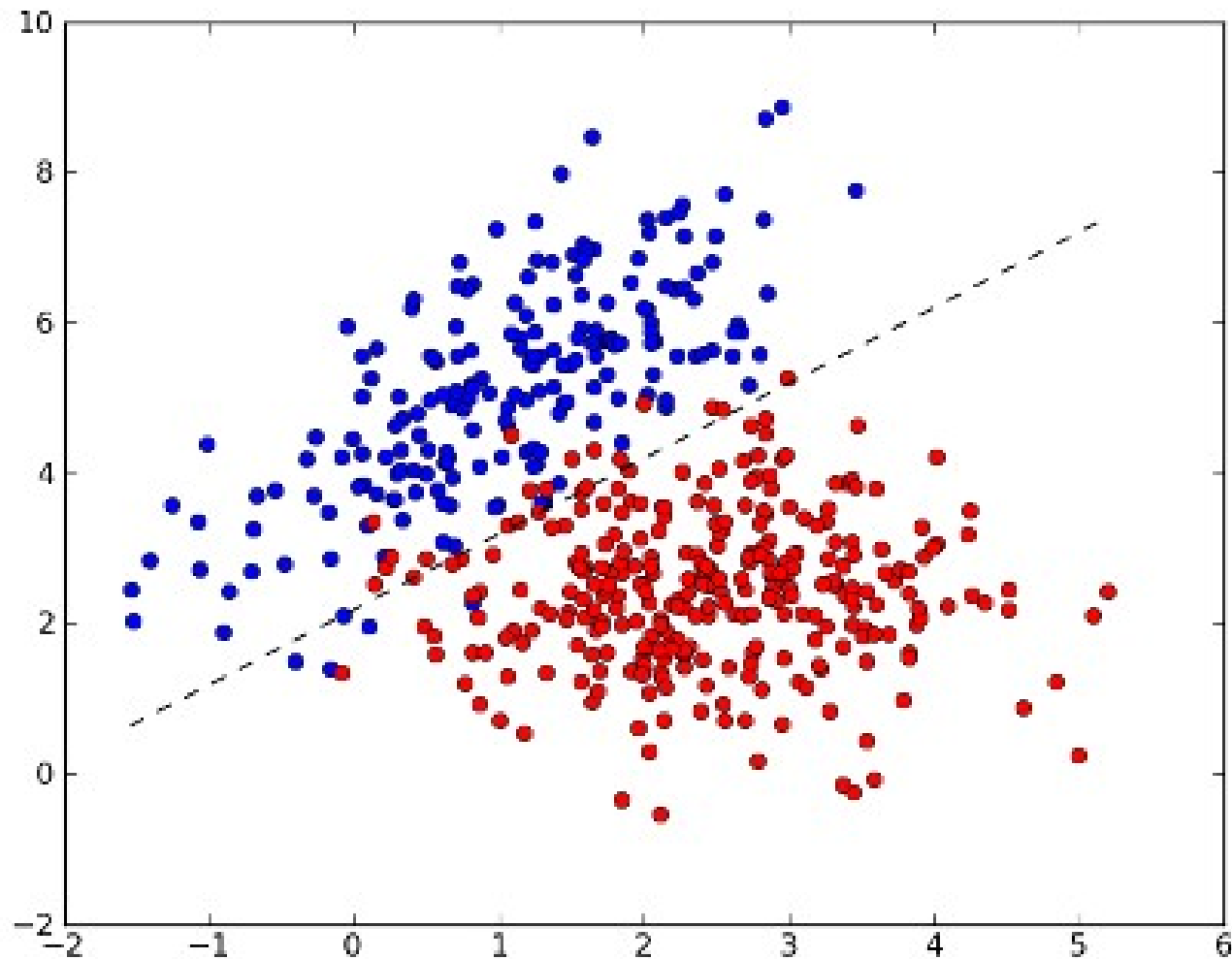


Машинное обучение

Лекция 6. Линейные алгоритмы классификации



Содержание лекции

- Общая формула линейного классификатора
- Метод стохастического градиента
- Частные случаи
- Обоснование метода СГ
- Выступ объекта для лин. классификатора
- ROC и AUC

Классификация линейной функцией

Обучающая выборка: $X^\ell = (x_i, y_i)_{i=1}^\ell$, $x_i \in \mathbb{R}^n$, $y_i \in \{-1, +1\}$

- 1 Модель классификации — *линейная*:

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

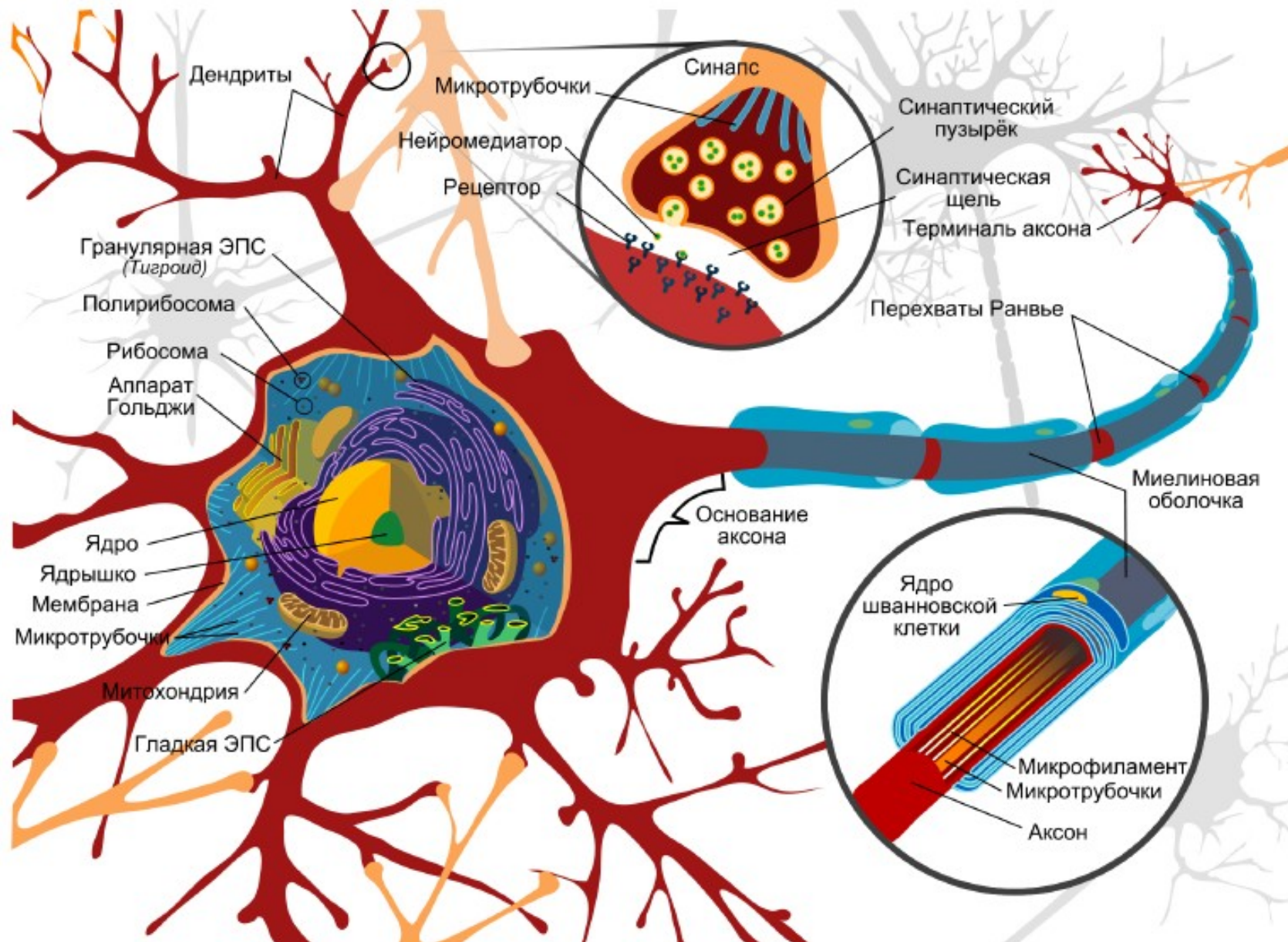
- 2 Функция потерь — бинарная или её аппроксимация:

$$\mathcal{L}(a, y) = [\langle x_i, w \rangle y_i < 0] \leq \mathcal{L}(\langle x_i, w \rangle y_i)$$

- 3 Метод обучения — *минимизация эмпирического риска*:

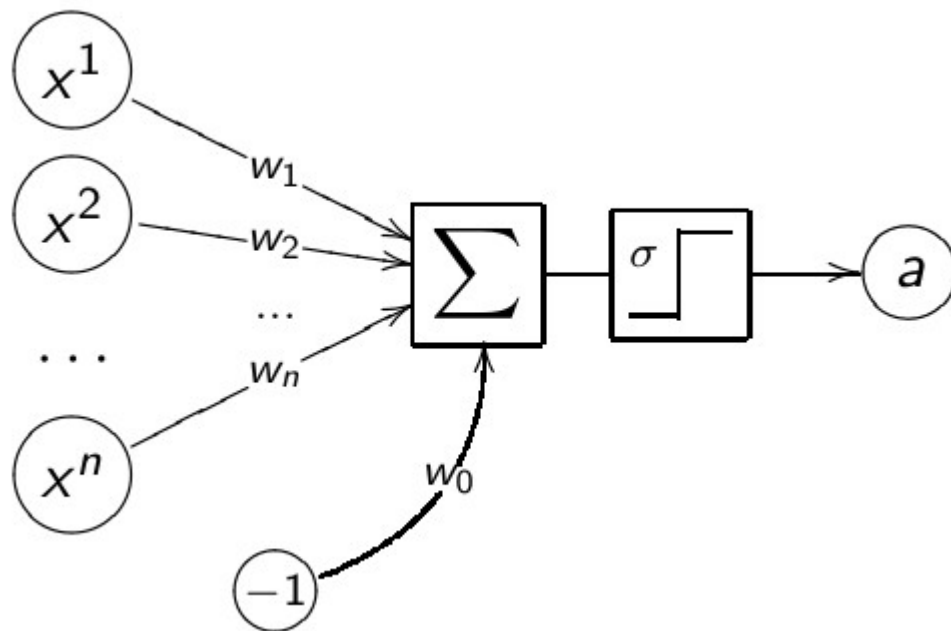
$$Q(w) = \sum_{i=1}^{\ell} [a(x_i, w) y_i < 0] \leq \sum_{i=1}^{\ell} \mathcal{L}(\langle x_i, w \rangle y_i) \rightarrow \min_w$$

Линейный классификатор – математическая модель нейрона



Линейная модель нейрона МакКаллока-Питтса (1943)

$$a(x, w) = \sigma(\langle w, x \rangle) = \sigma\left(\sum_{j=1}^n w_j f_j(x) - w_0\right)$$



Градиентный метод численной минимизации

Минимизация эмпирического риска:

$$Q(w) = \sum_{i=1}^{\ell} \mathcal{L}(g(w, x_i), y_i) = \sum_{i=1}^{\ell} \mathcal{L}_i(w) \rightarrow \min_w.$$

Численная минимизация методом *градиентного спуска*:

$w^{(0)}$:= начальное приближение;

$$w^{(t+1)} := w^{(t)} - h \cdot \nabla Q(w^{(t)}), \quad \nabla Q(w) = \left(\frac{\partial Q(w)}{\partial w_j} \right)_{j=0}^n,$$

где h — *градиентный шаг*, называемый также *темпом обучения*.

$$w^{(t+1)} := w^{(t)} - h \sum_{i=1}^{\ell} \nabla \mathcal{L}_i(w^{(t)}).$$

Идея ускорения сходимости:

брать (x_i, y_i) по одному и сразу обновлять вектор весов.

Метод стохастического градиента

Вход: выборка X^ℓ , темп обучения h , темп забывания λ

Выход: вектор весов w

- 1: инициализировать веса $w_j, j = 0, \dots, n$;
- 2: инициализировать оценку функционала: $\bar{Q} := \frac{1}{\ell} \sum_{i=1}^{\ell} \mathcal{L}_i(w)$;
- 3: **повторять**
- 4: выбрать объект x_i из X^ℓ случайным образом;
- 5: вычислить потерю: $\varepsilon_i := \mathcal{L}_i(w)$;
- 6: сделать градиентный шаг: $w := w - h \nabla \mathcal{L}_i(w)$;
- 7: оценить функционал: $\bar{Q} := (1 - \lambda) \bar{Q} + \lambda \varepsilon_i$;
- 8: **пока** значение \bar{Q} и/или веса w не сойдутся;

Пересчет функционала

Проблема: после каждого шага m по одному объекту x_i , не хотелось бы оценивать Q по всей выборке x_1, \dots, x_ℓ .

Решение: использовать рекуррентную формулу.

Среднее арифметическое $\bar{Q}_m = \frac{1}{m} \sum_{i=1}^m \varepsilon_i$:

$$\bar{Q}_m = (1 - \frac{1}{m})\bar{Q}_{m-1} + \frac{1}{m}\varepsilon_m.$$

Экспоненциальное скользящее среднее

$$\bar{Q}_m = \lambda\varepsilon_m + \lambda(1 - \lambda)\varepsilon_{m-1} + \lambda(1 - \lambda)^2\varepsilon_{m-2} + \lambda(1 - \lambda)^3\varepsilon_{m-3} + \dots$$

$$\bar{Q}_m := (1 - \lambda)\bar{Q}_{m-1} + \lambda\varepsilon_m.$$

Чем больше λ , тем быстрее забывается предыстория ряда.

Параметр λ называется *темпом забывания*.

Персептрон Розенблатта (1957)

- При синхронном возбуждении двух связанных нервных клеток синаптическая связь между ними усиливается.
- Математически (все x_j - бинарные):
 - $a(x_i, w) = y_i \implies w$ менять не нужно;
 - $a(x_i, w) = 0, y_i = 1 \implies w_j := w_j + h \cdot x_j$
 - $a(x_i, w) = 1, y_i = 0 \implies w_j := w_j - h \cdot x_j$
- Объединяем:

$$w := w - h(a(x_i, w) - y_i)x_i$$

Дельта-правило ADALINE

Задача регрессии: $x_i \in \mathbb{R}^{n+1}$, $y_i \in \mathbb{R}$

Адаптивный линейный элемент ADALINE [Видроу, Хофф 1960]:

$$a(x, w) = \langle w, x \rangle, \quad \mathcal{L}_i(w) = (\langle w, x_i \rangle - y_i)^2.$$

Градиентный шаг SG — **дельта-правило** (delta-rule):

$$w := w - h \underbrace{(\langle w, x_i \rangle - y_i)}_{\Delta_i} x_i,$$

Δ_i — ошибка алгоритма $a(x, w)$ на объекте x_i .

Формально совпадает с правилом персептрона Розенблатта!

Правило Хэбба

Задача классификации: $x_i \in \mathbb{R}^{n+1}$, $y_i \in \{-1, +1\}$,

$$a(x, w) = \text{sign}\langle w, x \rangle, \quad \mathcal{L}_i(w) = (-\langle w, x_i \rangle y_i)_+.$$

Градиентный шаг SG — **правило Хэбба** [1949]:

$$\text{если } \langle w, x_i \rangle y_i < 0 \text{ то } w := w + h x_i y_i,$$

То же самое для случая $y_i \in \{0, 1\}$,

$$a(x, w) = [\langle w, x \rangle > 0], \quad \mathcal{L}_i(w) = (a(x_i, w) - y_i) \langle w, x_i \rangle,$$

Градиентный шаг SG — **персептрон Розенблатта** [1957]:

$$w := w - h(a(x_i, w) - y_i) x_i.$$

Правило Хэбба

Теорема (Новиков, 1962)

Пусть выборка X^ℓ линейно разделима:

$\exists \tilde{w}, \exists \delta > 0: \langle \tilde{w}, x_i \rangle y_i > \delta$ для всех $i = 1, \dots, \ell$.

Тогда Алгоритм SG с правилом Хэбба находит вектор весов w ,

- разделяющий обучающую выборку без ошибок;
- при любом начальном положении $w^{(0)}$;
- при любом темпе обучения $h > 0$;
- независимо от порядка предъявления объектов x_i ;
- за конечное число исправлений вектора w ;
- если $w^{(0)} = 0$, то число исправлений $t_{\max} \leq \frac{1}{\delta^2} \max \|x_i\|^2$.

Доказательство

Рассмотрим $\cos(\widehat{\tilde{w}, w^t}) = \frac{\langle \tilde{w}, w^t \rangle}{\|w^t\|}$ после t -го исправления w^t , при $\|\tilde{w}\| = 1$.

При t -м исправлении $\langle x_i, w^{t-1} \rangle y_i < 0$. В силу линейной разделимости

$$\langle \tilde{w}, w^t \rangle = \langle \tilde{w}, w^{t-1} \rangle + h \langle \tilde{w}, x_i \rangle y_i > \langle \tilde{w}, w^{t-1} \rangle + h\delta > \langle \tilde{w}, w^0 \rangle + th\delta.$$

В силу ограниченности выборки, $\|x_i\| < D$:

$$\|w^t\|^2 = \|w^{t-1}\|^2 + h^2 \|x_i\|^2 + 2h \langle w^{t-1}, x_i \rangle y_i < \|w^{t-1}\|^2 + h^2 D^2 < \|w^0\|^2 + th^2 D^2.$$

Подставим эти соотношения в выражение для косинуса:

$$\cos(\widehat{\tilde{w}, w^t}) > \frac{\langle \tilde{w}, w^0 \rangle + th\delta}{\sqrt{\|w^0\|^2 + th^2 D^2}} \rightarrow \infty \text{ при } t \rightarrow \infty.$$

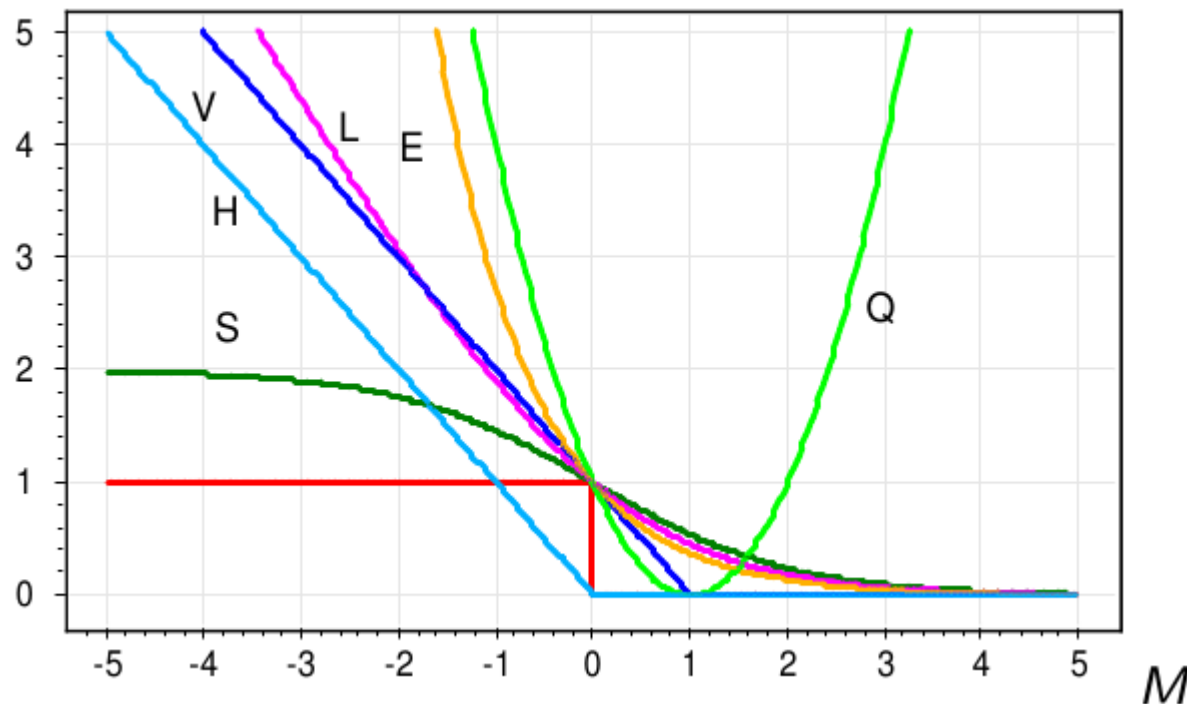
$\cos \leq 1$, значит при некотором t не найдётся ни одного $x_i \in X^\ell$ такого, что $\langle w^t, x_i \rangle y_i < 0$, то есть выборка окажется поделенной безошибочно.

Если $w^0 = 0$, то из условия $\cos = \frac{\sqrt{t}\delta}{D} \leq 1$ находим $t_{\max} = \left(\frac{D}{\delta}\right)^2$.

Понятие выступления для линейного классификатора

- Линейный классификатор:
$$a(x, w) = \text{sign}(x, w)$$
- $(x, w) = 0$ — разделяющая гиперплоскость,
- $M_i(w) = (w, x_i) y_i$ — отступ объекта x_i

Часто используемые непрерывные функции потерь



$$\begin{aligned} V(M) &= (1 - M)_+ \\ H(M) &= (-M)_+ \\ L(M) &= \log_2(1 + e^{-M}) \\ Q(M) &= (1 - M)^2 \\ S(M) &= 2(1 + e^M)^{-1} \\ E(M) &= e^{-M} \end{aligned}$$

$[M < 0]$

- кусочно-линейная (SVM);
- кусочно-линейная (Hebb's rule);
- логарифмическая (LR);
- квадратичная (FLD);
- сигмоидная (ANN);
- экспоненциальная (AdaBoost);
- пороговая функция потерь.

Начальное значение w

- 1 $w_j := 0$ для всех $j = 0, \dots, n$;
- 2 небольшие случайные значения:
 $w_j := \text{random} \left(-\frac{1}{2n}, \frac{1}{2n} \right)$;
- 3 $w_j := \frac{\langle y, f_j \rangle}{\langle f_j, f_j \rangle}$, $f_j = (f_j(x_i))_{i=1}^{\ell}$ — вектор значений признака.

Упражнение: доказать, что оценка w оптимальна, если

- 1) функция потерь квадратична и
- 2) признаки некоррелированы, $\langle f_j, f_k \rangle = 0$, $j \neq k$.

- 4 $w_j := \ln \frac{\sum_i [y_i=+1] f_j(x_i)}{\sum_i [y_i=-1] f_j(x_i)}$ — для классификации, $Y = \{-1, +1\}$
- 5 обучение по небольшой случайной подвыборке объектов;
- 6 мултистарт: многократные запуски из разных случайных начальных приближений и выбор лучшего решения.

Порядок предъявления x_i

Возможны варианты:

- 1 перетасовка объектов (shuffling):
попеременно брать объекты из разных классов;
- 2 чаще брать те объекты, на которых была допущена бóльшая ошибка
(чем меньше M_i , тем больше вероятность взять объект)
(чем меньше $|M_i|$, тем больше вероятность взять объект);
- 3 вообще не брать «хорошие» объекты, у которых $M_i > \mu_+$
(при этом немного ускоряется сходимость);
- 4 вообще не брать объекты-«выбросы», у которых $M_i < \mu_-$
(при этом может улучшиться качество классификации);

Параметры μ_+ , μ_- придётся подбирать.

Выбор шага h

- ① сходимость гарантируется (для выпуклых функций) при

$$h_t \rightarrow 0, \quad \sum_{t=1}^{\infty} h_t = \infty, \quad \sum_{t=1}^{\infty} h_t^2 < \infty,$$

в частности можно положить $h_t = 1/t$;

- ② метод скорейшего градиентного спуска:

$$\mathcal{L}_i(w - h \nabla \mathcal{L}_i(w)) \rightarrow \min_h,$$

позволяет найти адаптивный шаг h^* ;

Упражнение: доказать, что при квадратичной функции потерь $h^* = \|x_i\|^{-2}$.

- ③ пробные случайные шаги
— для «выбивания» из локальных минимумов;

Достоинства и недостатки

Достоинства:

- 1 легко реализуется;
- 2 легко обобщается на любые $g(x, w)$, $\mathcal{L}(a, y)$;
- 3 возможно динамическое (потокковое) обучение;
- 4 на сверхбольших выборках можно получить неплохое решение, даже не обработав все (x_i, y_i) ;
- 5 всё чаще применяется для Big Data

Недостатки:

- 1 возможна расходимость или медленная сходимость;
- 2 застревание в локальных минимумах;
- 3 подбор комплекса эвристик является искусством;
- 4 проблема переобучения;

Проблема переобучения

Возможные причины переобучения:

- 1 слишком мало объектов; слишком много признаков;
- 2 линейная зависимость (мультиколлинеарность) признаков:
пусть построен классификатор: $a(x, w) = \text{sign}\langle w, x \rangle$;
мультиколлинеарность: $\exists u \in \mathbb{R}^{n+1}: \forall x \quad \langle u, x \rangle \equiv 0$;
тогда $\forall \gamma \in \mathbb{R} \quad a(x, w) = \text{sign}\langle w + \gamma u, x \rangle$

Симптоматика:

- 1 слишком большие веса $|w_j|$ разных знаков;
- 2 неустойчивость $a(x, w)$;
- 3 $Q(X^\ell) \ll Q(X^k)$;

Терапия:

- 1 регуляризация (сокращение весов, weight decay);
- 2 ранний останов (early stopping);

Регуляризация

Штраф за увеличение нормы вектора весов:

$$\tilde{\mathcal{L}}_i(w) = \mathcal{L}_i(w) + \frac{\tau}{2} \|w\|^2 = \mathcal{L}_i(w) + \frac{\tau}{2} \sum_{j=1}^n w_j^2 \rightarrow \min_w.$$

Градиент:

$$\nabla \tilde{\mathcal{L}}_i(w) = \nabla \mathcal{L}_i(w) + \tau w.$$

Модификация градиентного шага:

$$w := w(1 - h\tau) - h\nabla \mathcal{L}_i(w).$$

Разные штрафы за ошибки

Задача классификации на два класса, $y_i \in \{-1, +1\}$.

Модель классификации: $a(x; w, w_0) = \text{sign}(g(x, w) - w_0)$.

Чем меньше w_0 , тем больше x_i : $a(x_i) = +1$.

Пусть λ_y — штраф за ошибку на объекте класса y .

Функция потерь теперь зависит от штрафов:

$$\mathcal{L}(a, y) = \lambda_{y_i} [a(x_i; w, w_0) \neq y_i] = \lambda_{y_i} [(g(x_i, w) - w_0)y_i < 0].$$

Проблема

На практике штрафы $\{\lambda_y\}$ могут пересматриваться

- Нужен удобный способ выбора w_0 в зависимости от $\{\lambda_y\}$, не требующий построения w заново.
- Нужна характеристика качества модели $g(x, w)$, не зависящая от штрафов $\{\lambda_y\}$ и численности классов.

ROC-кривая

ROC — «receiver operating characteristic».

- Каждая точка кривой соответствует некоторому $a(x; w, w_0)$.
- по оси X : доля ошибочных положительных классификаций (FPR — false positive rate):

$$\text{FPR}(a, X^\ell) = \frac{\sum_{i=1}^{\ell} [y_i = -1][a(x_i; w, w_0) = +1]}{\sum_{i=1}^{\ell} [y_i = -1]};$$

$1 - \text{FPR}(a)$ называется специфичностью алгоритма a .

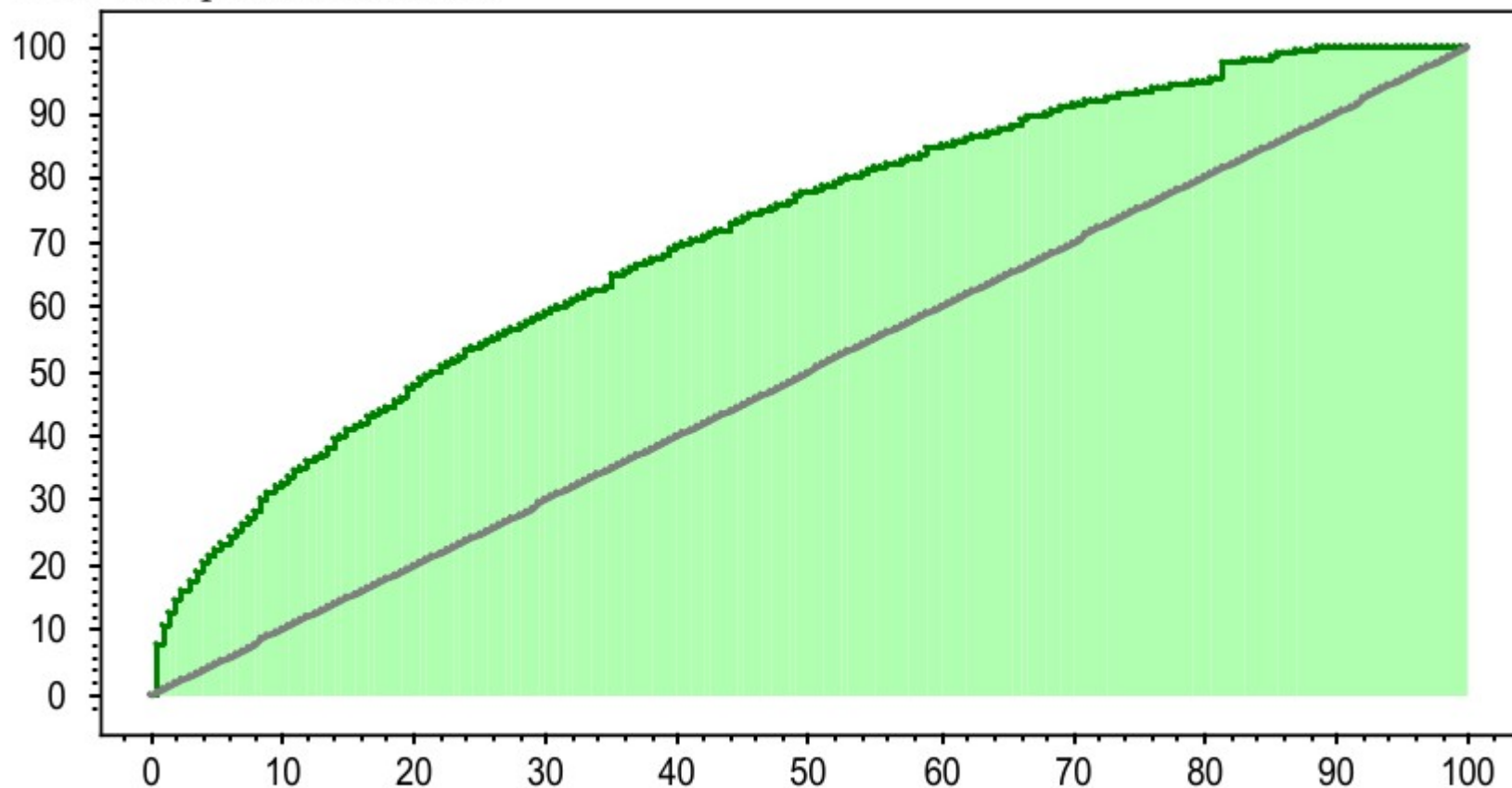
- по оси Y : доля правильных положительных классификаций (TPR — true positive rate):

$$\text{TPR}(a, X^\ell) = \frac{\sum_{i=1}^{\ell} [y_i = +1][a(x_i; w, w_0) = +1]}{\sum_{i=1}^{\ell} [y_i = +1]};$$

$\text{TPR}(a)$ называется также чувствительностью алгоритма a .

Пример

TPR, true positive rate, %



FPR, false positive rate, %

■ AUC, площадь под ROC-кривой — наихудшая ROC-кривая

Алгоритм построения ROC-кривой

Вход: выборка X^ℓ ; дискриминантная функция $g(x, w)$;

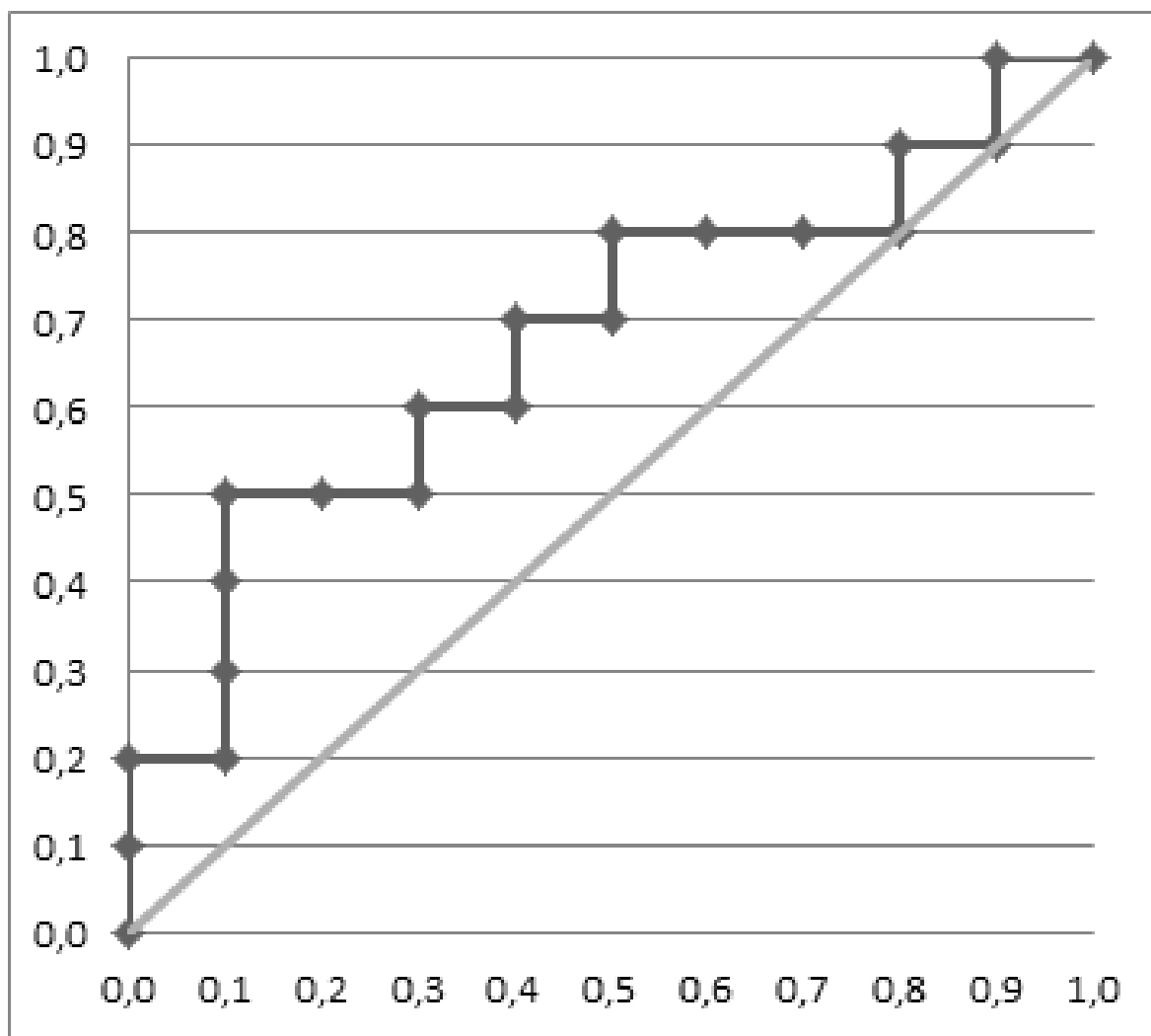
Выход: $\{(FPR_i, TPR_i)\}_{i=0}^\ell$, AUC — площадь под ROC-кривой.

- 1: $\ell_y := \sum_{i=1}^\ell [y_i = y]$, для всех $y \in Y$;
- 2: упорядочить выборку X^ℓ по убыванию значений $g(x_i, w)$;
- 3: поставить первую точку в начало координат:
 $(FPR_0, TPR_0) := (0, 0)$; $AUC := 0$;
- 4: **для** $i := 1, \dots, \ell$
- 5: **если** $y_i = -1$ **то** сместиться на один шаг вправо:
- 6: $FPR_i := FPR_{i-1} + \frac{1}{\ell_-}$; $TPR_i := TPR_{i-1}$;
 $AUC := AUC + \frac{1}{\ell_-} TPR_i$;
- 7: **иначе** сместиться на один шаг вверх:
- 8: $FPR_i := FPR_{i-1}$; $TPR_i := TPR_{i-1} + \frac{1}{\ell_+}$;

Пример

TPR

#	C	Score
1	P	0,9
2	P	0,8
3	N	0,7
4	P	0,6
5	P	0,55
6	P	0,54
7	N	0,53
8	N	0,52
9	P	0,51
10	N	0,505
11	P	0,4
12	N	0,39
13	P	0,38
14	N	0,37
15	N	0,36
16	N	0,35
17	P	0,34
18	N	0,33
19	P	0,3
20	N	0,1

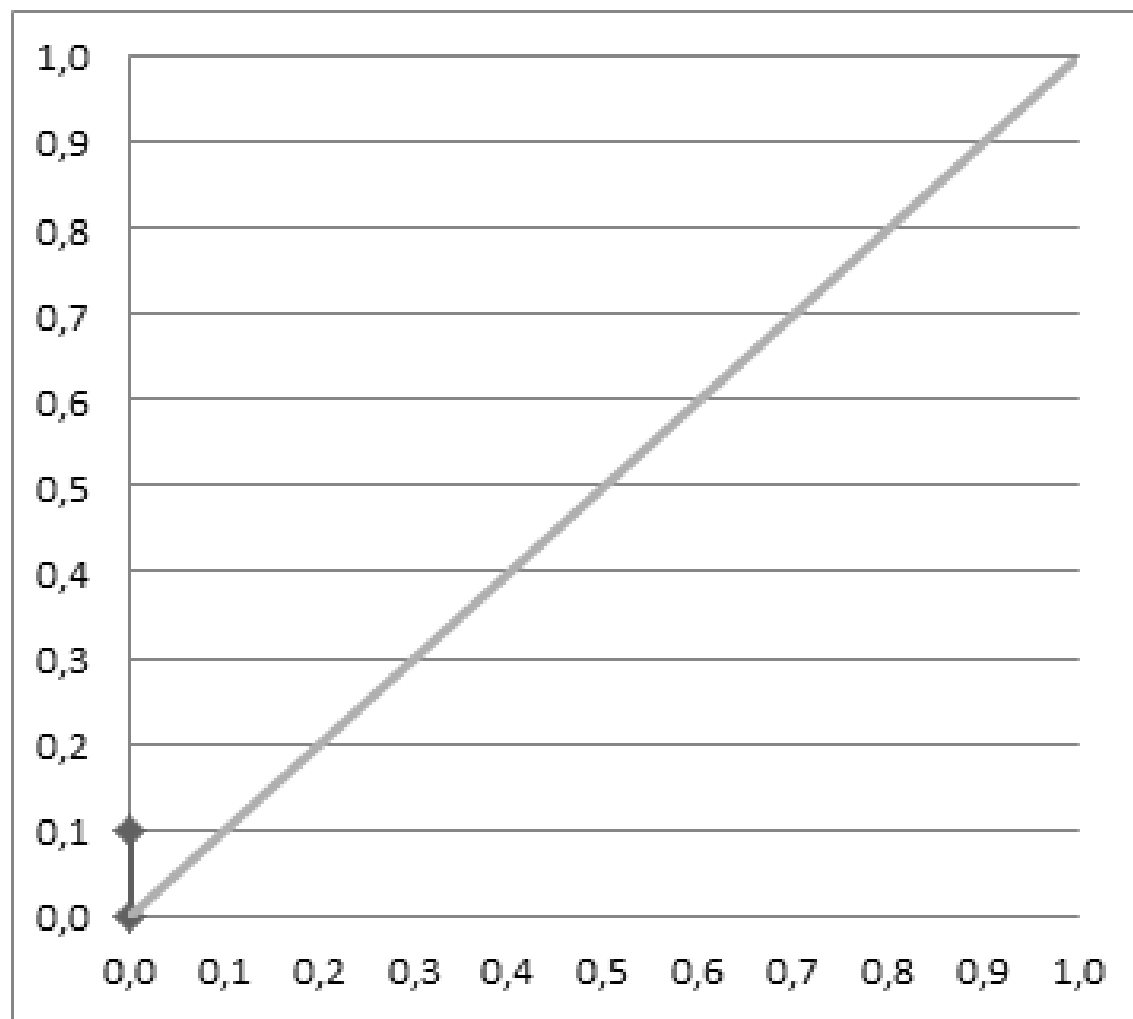


FPR

Пример

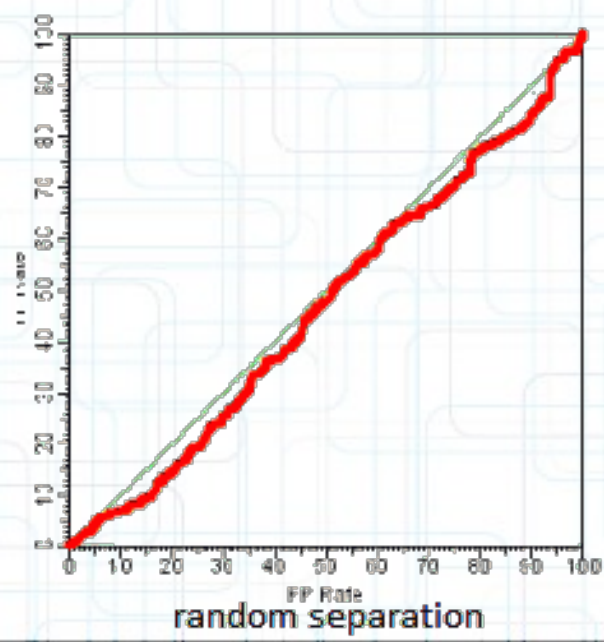
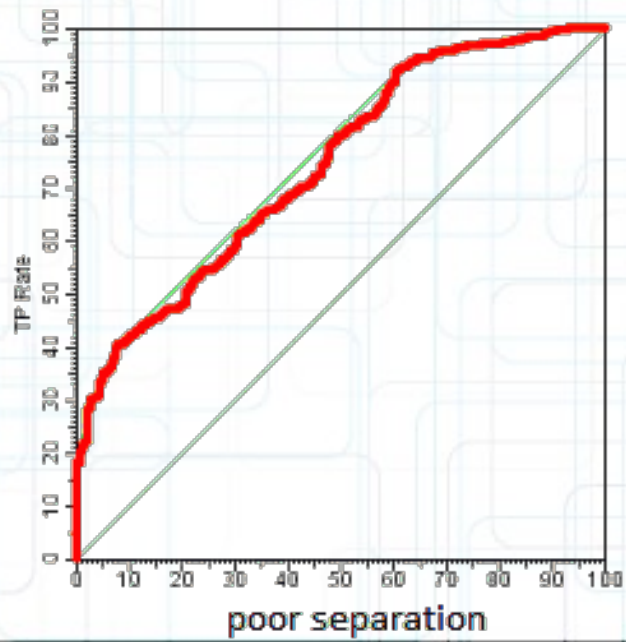
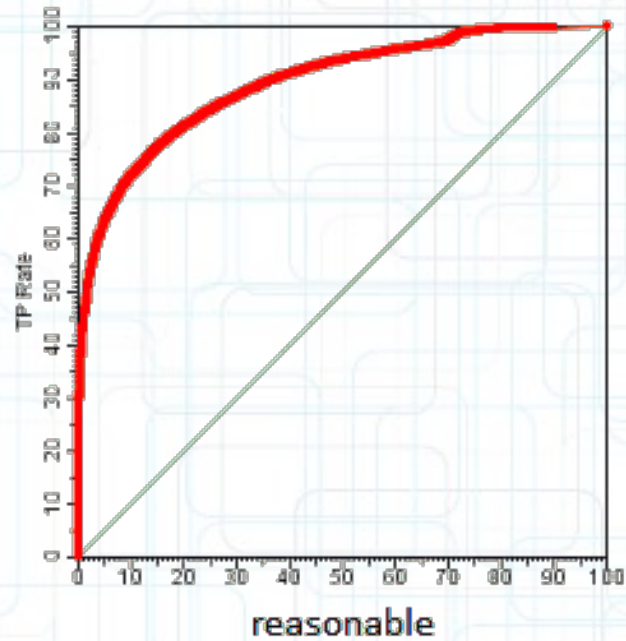
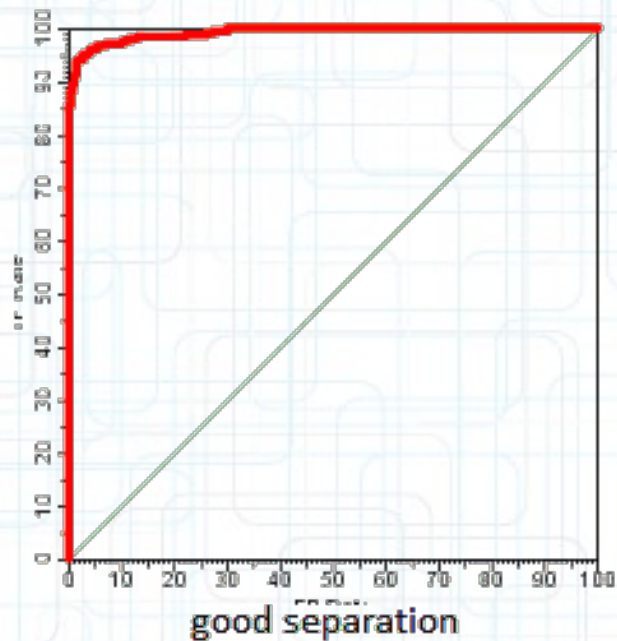
TPR

#	C	Score
1	P	0,9
2	P	0,8
3	N	0,7
4	P	0,6
5	P	0,55
6	P	0,54
7	N	0,53
8	N	0,52
9	P	0,51
10	N	0,505
11	P	0,4
12	N	0,39
13	P	0,38
14	N	0,37
15	N	0,36
16	N	0,35
17	P	0,34
18	N	0,33
19	P	0,3
20	N	0,1



FPR

Еще примеры



Градиентная максимизация AUC

Модель: $a(x_i, w, w_0) = \text{sign}(g(x_i, w) - w_0)$.

AUC — это доля правильно упорядоченных пар (x_i, x_j) :

$$\begin{aligned} \text{AUC}(w) &= \frac{1}{\ell_-} \sum_{i=1}^{\ell} [y_i = -1] \text{TPR}_i = \\ &= \frac{1}{\ell_- \ell_+} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} [y_i < y_j] [g(x_i, w) < g(x_j, w)] \rightarrow \max_w \end{aligned}$$

Явная максимизация аппроксимированного AUC:

$$\text{AUC}(w) \leq Q(w) = \sum_{i,j: y_i < y_j} \underbrace{\mathcal{L}(g(x_j, w) - g(x_i, w))}_{M_{ij}(w)} \rightarrow \min_w,$$

где $\mathcal{L}(M)$ — гладкая убывающая функция отступа,
 $M_{ij}(w)$ — новое понятие отступа для пар объектов.

Алгоритм стохастического градиента для AUC

Возьмём для простоты линейный классификатор:

$$g(x, w) = \langle x, w \rangle, \quad M_{ij}(w) = \langle x_j - x_i, w \rangle.$$

Вход: выборка X^ℓ , темп обучения h , темп забывания λ

Выход: вектор весов w

-
- 1: инициализировать веса $w_j, j = 0, \dots, n$;
 - 2: инициализировать оценку: $\bar{Q} := \frac{1}{\ell_+ \ell_-} \sum_{i,j: y_i < y_j} \mathcal{L}(M_{ij}(w))$;
 - 3: **повторять**
 - 4: выбрать **пару объектов** $(i, j): y_i < y_j$, случайным образом;
 - 5: вычислить потерю: $\varepsilon_{ij} := \mathcal{L}(M_{ij}(w))$;
 - 6: сделать градиентный шаг: $w := w - h \mathcal{L}'(M_{ij}(w))(x_j - x_i)$;
 - 7: оценить функционал: $\bar{Q} := (1 - \lambda) \bar{Q} + \lambda \varepsilon_{ij}$;
 - 8: **пока** значение \bar{Q} и/или веса w не сойдутся;