

Линейные методы классификации и регрессии: метод стохастического градиента

Воронцов Константин Вячеславович

vokov@forecsys.ru

<http://www.MachineLearning.ru/wiki?title=User:Vokov>

Этот курс доступен на странице вики-ресурса

<http://www.MachineLearning.ru/wiki>

«Машинное обучение (курс лекций, К.В.Воронцов)»

Видеолекции: <http://shad.yandex.ru/lectures>

3 марта 2016

- 1 Метод стохастического градиента**
 - Минимизация эмпирического риска
 - Линейный классификатор
 - Метод стохастического градиента
- 2 Эвристики для метода стохастического градиента**
 - Инициализация весов и порядок объектов
 - Выбор величины градиентного шага
 - Проблема переобучения, метод сокращения весов
- 3 Вероятностные функции потерь**
 - Вероятностная модель классификации
 - Логистическая регрессия
 - Калибровка Платта

Обучение регрессии — это оптимизация

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

- 1 Модель регрессии — *линейная*:

$$a(x, w) = \langle x, w \rangle = \sum_{j=1}^n f_j(x) w_j, \quad w \in \mathbb{R}^n$$

- 2 Функция потерь — *квадратичная*:

$$\mathcal{L}(a, y) = (a - y)^2$$

- 3 Метод обучения — *метод наименьших квадратов*:

$$Q(w) = \sum_{i=1}^{\ell} (a(x_i, w) - y_i)^2 \rightarrow \min_w$$

- 4 Проверка по тестовой выборке $X^k = (\tilde{x}_i, \tilde{y}_i)_{i=1}^k$:

$$\bar{Q}(w) = \frac{1}{k} \sum_{i=1}^k (a(\tilde{x}_i, w) - \tilde{y}_i)^2$$

Обучение классификации — тоже оптимизация

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*:

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

- 4 Проверка по тестовой выборке $X^k = (\tilde{x}_i, \tilde{y}_i)_{i=1}^k$:

$$\bar{Q}(w) = \frac{1}{k} \sum_{i=1}^k [\langle \tilde{x}_i, w \rangle \tilde{y}_i < 0]$$

Понятие отступа для разделяющих классификаторов

Задача классификации с двумя классами: $y_i \in \{-1, +1\}$

Разделяющий классификатор: $a(x, w) = \text{sign } g(x, w)$,
 $g(x, w)$ — разделяющая (дискриминантная) функция,
 w — вектор параметров,
 $g(x, w) = 0$ — уравнение разделяющей поверхности

Определение

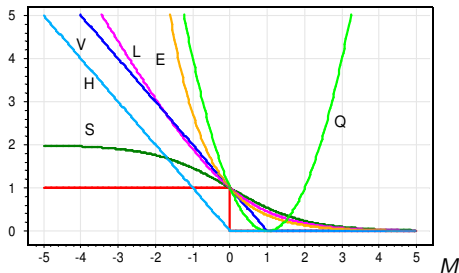
$M_i(w) = g(x_i, w)y_i$ — отступ (margin) объекта x_i ;

$M_i(w) < 0 \iff$ алгоритм $a(x, w)$ ошибается на x_i ;

Линейный классификатор: $a(x, w) = \text{sign} \langle x, w \rangle$:
 $\langle x, w \rangle = 0$ — уравнение разделяющей гиперплоскости,
 $M_i(w) = \langle w, x_i \rangle y_i$ — отступ объекта x_i .

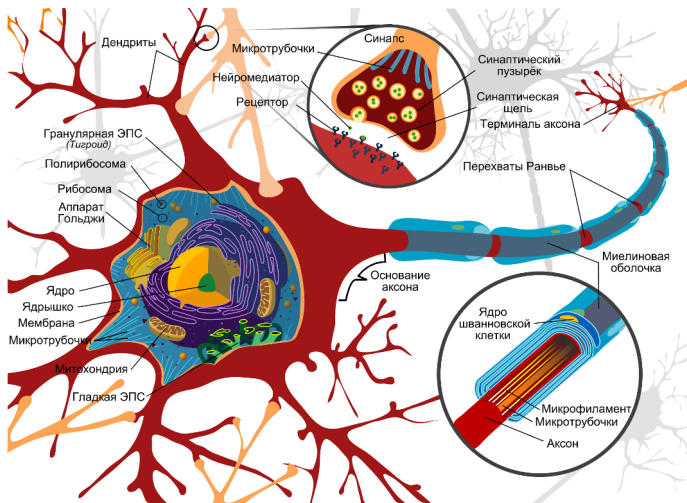
Непрерывные аппроксимации пороговой функции потерь

Часто используемые непрерывные функции потерь $\mathcal{L}(M)$:



- | | |
|-----------------------------|-----------------------------------|
| $V(M) = (1 - M)_+$ | — кусочно-линейная (SVM); |
| $H(M) = (-M)_+$ | — кусочно-линейная (Hebb's rule); |
| $L(M) = \log_2(1 + e^{-M})$ | — логарифмическая (LR); |
| $Q(M) = (1 - M)^2$ | — квадратичная (FLD); |
| $S(M) = 2(1 + e^M)^{-1}$ | — сигмоидная (ANN); |
| $E(M) = e^{-M}$ | — экспоненциальная (AdaBoost); |
| $[M < 0]$ | — пороговая функция потерь. |

Линейный классификатор — математическая модель нейрона



Линейный классификатор — математическая модель нейрона

Линейная модель нейрона МакКаллока-Питтса [1943]:

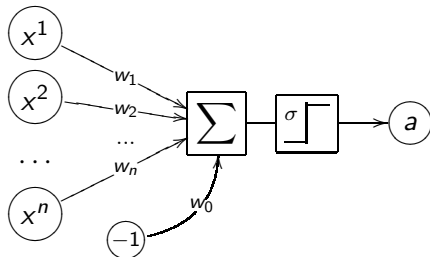
$$a(x, w) = \sigma(\langle w, x \rangle) = \sigma\left(\sum_{j=1}^n w_j f_j(x) - w_0\right),$$

$\sigma(z)$ — функция активации (например, sign),

w_j — весовые коэффициенты синаптических связей,

w_0 — порог активации,

$w, x \in \mathbb{R}^{n+1}$, если ввести константный признак $f_0(x) \equiv -1$



Персептрон Розенблатта [1957]

Задача классификации с двумя классами, $y_i \in \{0, 1\}$
и бинарными признаками $f_j(x) \in \{0, 1\}$,

$$a(x, w) = [\langle w, x \rangle > 0].$$

Эвристика исправления весов в случае ошибки:

- $a(x_i, w) = y_i \implies w$ менять не нужно;
- $a(x_i, w) = 0, y_i = 1 \implies w_j := w_j + h \cdot f_j(x_i)$;
- $a(x_i, w) = 1, y_i = 0 \implies w_j := w_j - h \cdot f_j(x_i)$;

Объединим это в одну рекуррентную формулу коррекции весов:

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

Градиентный метод численной минимизации

Минимизация эмпирического риска (регрессия, классификация):

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

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.

Алгоритм SG (Stochastic Gradient)

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;

Robbins, H., Monro S. A stochastic approximation method // Annals of Mathematical Statistics, 1951, 22 (3), p. 400–407.

Откуда взялась такая оценка функционала?

Проблема: вычисление оценки Q по всей выборке x_1, \dots, x_ℓ намного дольше градиентного шага по одному объекту x_i .

Решение: использовать приближённую рекуррентную формулу.

Среднее арифметическое:

$$\bar{Q}_m = \frac{1}{m}\varepsilon_m + \frac{1}{m}\varepsilon_{m-1} + \frac{1}{m}\varepsilon_{m-2} + \dots$$

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

Экспоненциальное скользящее среднее:

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

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

Параметр λ — *темп забывания* предыстории ряда.

Алгоритм SAG (Stochastic Average Gradient)

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

Schmidt M., Le Roux N., Bach F. Minimizing finite sums with the stochastic average gradient // arXiv.org, 2013.

Частный случай №1: 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!

Частный случай №2: правило Хэбба и персептрон Розенблатта

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]:

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

Обоснование Алгоритма SG с правилом Хэбба

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

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 теоремы Новикова

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 $t_{\max} = \left(\frac{D}{\delta}\right)^2$.

Варианты инициализации весов

- 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 обучение по небольшой случайной подвыборке объектов;
- 5 мультистарт: многократные запуски из разных случайных начальных приближений и выбор лучшего решения.

Варианты порядка предъявления объектов

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.

Варианты выбора градиентного шага

- 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 $h^* = \|x_j\|^{-2}$.

- 3 пробные случайные шаги для «выбивания» итерационного процесса из локальных минимумов;
- 4 метод Левенберга-Марквардта (второго порядка)

Диagonalный метод Левенберга-Марквардта

Метод Ньютона-Рафсона, $\mathcal{L}_i(w) \equiv \mathcal{L}(\langle w, x_i \rangle y_i)$:

$$w := w - h(\mathcal{L}_i''(w))^{-1} \nabla \mathcal{L}_i(w),$$

где $\mathcal{L}_i''(w) = \left(\frac{\partial^2 \mathcal{L}_i(w)}{\partial w_j \partial w_{j'}} \right)$ — гессиан, $n \times n$ -матрица

Эвристика. Считаем, что гессиан диагонален:

$$w_j := w_j - h \left(\frac{\partial^2 \mathcal{L}_i(w)}{\partial w_j^2} + \mu \right)^{-1} \frac{\partial \mathcal{L}_i(w)}{\partial w_j},$$

h — темп обучения, можно полагать $h = 1$

μ — параметр, предотвращающий обнуление знаменателя.

Отношение h/μ есть темп обучения на ровных участках функционала $\mathcal{L}_i(w)$, где вторая производная обнуляется.

SG: 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 подходит для задач с большими данными

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:

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

Проявления переобучения:

- слишком большие веса $|w_j|$ разных знаков;
- неустойчивость дискриминантной функции $\langle w, x \rangle$;
- $Q(X^\ell) \ll Q(X^k)$;

Способы уменьшения переобучения:

- регуляризация (сокращение весов, weight decay);

Регуляризация (сокращение весов)

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:

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

Методы подбора коэффициента регуляризации τ :

- 1 скользящий контроль;
- 2 стохастическая адаптация;
- 3 двухуровневый байесовский вывод.

Принцип максимума правдоподобия

Пусть $X \times Y$ — в.п. с плотностью $p(x, y|w) = P(y|x, w)p(x)$.
Пусть X^ℓ — простая (i.i.d.) выборка: $(x_i, y_i)_{i=1}^\ell \sim p(x, y|w)$

Оценка максимального правдоподобия для w :

$$\prod_{i=1}^{\ell} P(y_i|x_i, w) \overbrace{p(x_i)} \rightarrow \max_w$$

Функционал логарифма правдоподобия (log-likelihood):

$$L(w) = \sum_{i=1}^{\ell} \log P(y_i|x_i, w) \rightarrow \max_w.$$

В случае двух классов, $y_i \in Y = \{0, 1\}$, удобно записывать модель условной вероятности $\pi(x, w) = P(y=1|x, w)$:

$$L(w) = \sum_{i=1}^{\ell} y_i \log \pi(x_i, w) + (1 - y_i) \log(1 - \pi(x_i, w)) \rightarrow \max_w,$$

Связь правдоподобия и аппроксимации эмпирического риска

Пусть $X \times Y$ — в.п. с плотностью $p(x, y|w) = P(y|x, w)p(x)$.
Пусть X^ℓ — простая (i.i.d.) выборка: $(x_i, y_i)_{i=1}^\ell \sim p(x, y|w)$

- *Максимизация правдоподобия:*

$$L(w) = \sum_{i=1}^{\ell} \log P(y_i|x_i, w) \rightarrow \max_w.$$

- *Минимизация аппроксимированного эмпирического риска:*

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

Эти два принципа эквивалентны, если положить

$$-\log P(y_i|x_i, w) = \mathcal{L}(y_i g(x_i, w)).$$

$$\boxed{\text{модель } P(y|x, w)} \Leftrightarrow \boxed{\text{модель } g(x, w) \text{ и } \mathcal{L}(M)}.$$

Вероятностные смысл регуляризации

$P(y|x, w)$ — вероятностная модель данных;

$p(w; \gamma)$ — априорное распределение параметров модели;

γ — вектор гиперпараметров;

Теперь не только появление выборки X^ℓ ,
но и появление модели w также полагается стохастическим.

Совместное правдоподобие данных и модели:

$$p(X^\ell, w) = p(X^\ell | w) p(w; \gamma).$$

Принцип максимума совместного правдоподобия:

$$L(w) = \ln p(X^\ell, w) = \sum_{i=1}^{\ell} \log P(y_i | x_i, w) + \underbrace{\log p(w; \gamma)}_{\text{регуляризатор}} \rightarrow \max_{w, \gamma}$$

Пример: квадратичный (гауссовский) регуляризатор

Пусть $w \in \mathbb{R}^n$ имеет n -мерное гауссовское распределение:

$$p(w; \sigma) = \frac{1}{(2\pi\sigma)^{n/2}} \exp\left(-\frac{\|w\|^2}{2\sigma}\right), \quad \|w\|^2 = \sum_{j=1}^n w_j^2,$$

т. е. все веса независимы, имеют нулевое матожидание и равные дисперсии σ ; σ — гиперпараметр.

Логарифмируя, получаем квадратичный регуляризатор:

$$-\ln p(w; \sigma) = \frac{1}{2\sigma} \|w\|^2 + \text{const}(w).$$

Вероятностный смысл коэффициента регуляризации: $\tau = \sigma^{-1}$.

Двухклассовая логистическая регрессия

Линейная модель классификации для двух классов $Y = \{-1, 1\}$:

$$a(x) = \text{sign}\langle w, x \rangle, \quad x, w \in \mathbb{R}^n.$$

Отступ $M = \langle w, x \rangle y$.

Логарифмическая функция потерь:

$$\mathcal{L}(M) = \log(1 + e^{-M}).$$

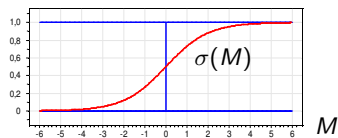
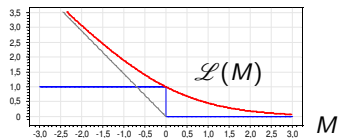
Модель условной вероятности:

$$P(y|x, w) = \sigma(M) = \frac{1}{1 + e^{-M}},$$

где $\sigma(M)$ — сигмоидная функция,
важно: $\sigma(M) + \sigma(-M) = 1$.

Задача обучения логистической регрессии:

$$L(w) = \sum_{i=1}^{\ell} \log(1 + \exp(-\langle w, x_i \rangle y_i)) \rightarrow \min_w$$



Двухклассовая логистическая регрессия с регуляризацией

Линейная модель классификации для двух классов $Y = \{-1, 1\}$:

$$a(x) = \text{sign}\langle w, x \rangle, \quad x, w \in \mathbb{R}^n.$$

Отступ $M = \langle w, x \rangle y$.

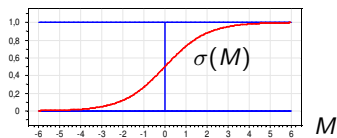
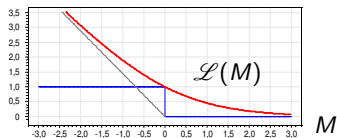
Логарифмическая функция потерь:

$$\mathcal{L}(M) = \log(1 + e^{-M}).$$

Модель условной вероятности:

$$P(y|x, w) = \sigma(M) = \frac{1}{1 + e^{-M}},$$

где $\sigma(M)$ — сигмоидная функция,
 важно: $\sigma(M) + \sigma(-M) = 1$.



Задача обучения **регуляризованной** логистической регрессии:

$$L(w) = \sum_{i=1}^{\ell} \log(1 + \exp(-\langle w, x_i \rangle y_i)) + \frac{\tau}{2} \|w\|^2 \rightarrow \min_w$$

Многоклассовая логистическая регрессия

Линейный классификатор при произвольном числе классов $|Y|$:

$$a(x) = \arg \max_{y \in Y} \langle w_y, x \rangle, \quad x, w_y \in \mathbb{R}^n.$$

Вероятность того, что объект x относится к классу y :

$$P(y|x, w) = \frac{\exp\langle w_y, x \rangle}{\sum_{z \in Y} \exp\langle w_z, x \rangle} = \text{SoftMax}_{y \in Y} \langle w_y, x \rangle,$$

функция SoftMax: $\mathbb{R}^Y \rightarrow \mathbb{R}^Y$ переводит произвольный вектор в нормированный вектор дискретного распределения.

Задача максимизации регуляризованного правдоподобия:

$$Q(w) = - \sum_{i=1}^{\ell} \log P(y_i|x_i, w) + \frac{\tau}{2} \sum_{y \in Y} \|w_y\|^2 \rightarrow \min_w.$$

Калибровка Платта

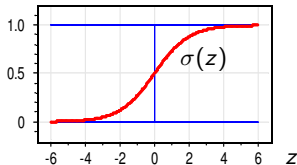
Пусть для простоты классов два, $Y = \{-1, +1\}$.

Задача. Для классификатора вида $a(x) = \text{sign } g(x, w)$ построить функцию оценки условной вероятности $P(y|x)$.

Модель условной вероятности:

$$\pi(x; a, b) = P(y=1|x) = \sigma(ag(x, w) + b)$$

где $\sigma(z) = \frac{1}{1+e^{-z}}$ — сигмоидная функция



Калибровка коэффициентов a, b по *контрольной* выборке методом максимума правдоподобия:

$$\sum_{y_i=-1} \log(1 - \pi(x_i; a, b)) + \sum_{y_i=+1} \log \pi(x_i; a, b) \rightarrow \max_{a, b}$$

Резюме в конце лекции

- Метод стохастического градиента (SG, SAG) подходит для любых моделей и функций потерь
- Хорошо подходит для обучения по большим данным
- *Аппроксимация пороговой функции потерь $\mathcal{L}(M)$* позволяет использовать градиентную оптимизацию
- Функции $\mathcal{L}(M)$, штрафующие за приближение к границе классов, увеличивают зазор между классами, благодаря этому повышается надёжность классификации
- *Регуляризация* решает проблему мультиколлинеарности и также снижает переобучение
- *Логистическая регрессия* — метод классификации, оценивающий условные вероятности классов $P(y|x)$