

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)»

МФТИ • 8 сентября 2023

- 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
 - Пример. Задача кредитного скоринга

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*:

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

- 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*:

$$a(x, w) = \text{sign}\langle x, w \rangle = \text{sign} \sum_{j=1}^n w_j f_j(x)$$

- 2 Функция потерь — бинарная или **её аппроксимация**:

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

- 3 Метод обучения — *минимизация эмпирического риска*:

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

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

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

Разделяющие классификаторы (margin-based classifier)

Разделяющий классификатор: $a(x, w) = \text{sign } g(x, w)$

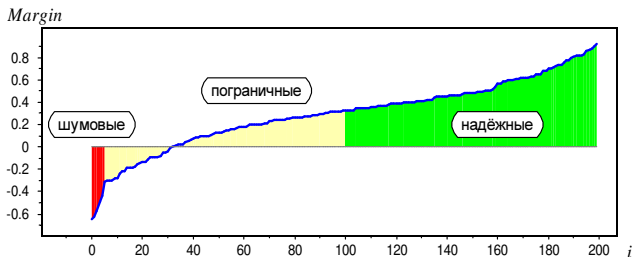
$g(x, w)$ — разделяющая (дискриминантная) функция

$g(x, w) = 0$ — уравнение разделяющей поверхности

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

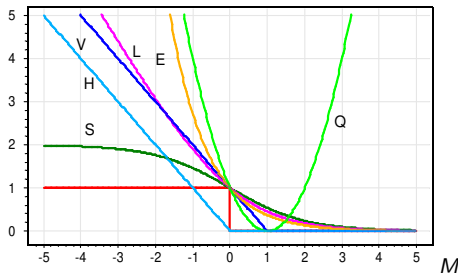
$M_i(w) < 0 \iff$ алгоритм $a(x, w)$ ошибается на x_i

Ранжирование объектов по возрастанию отступов $M_i(w)$:



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

Линейная модель нейрона МакКаллока-Питтса [1943]:

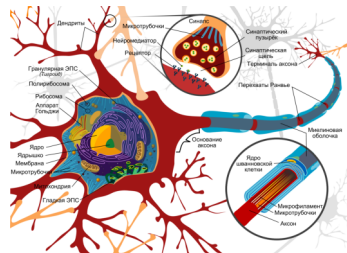
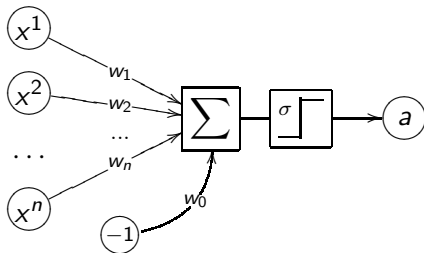
$$a(x, w) = \sigma(\langle w, x \rangle) = \sigma\left(\sum_{j=1}^n w_j f_j(x) - w_0\right),$$

$\sigma(z)$ — 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 $f_0(x) \equiv -1$



Градиентный метод численной минимизации

Минимизация эмпирического риска:

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

Метод *градиентного спуска*:

$w^{(0)}$:= начальное приближение;

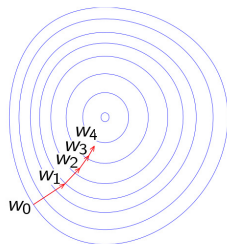
$$w^{(t+1)} := w^{(t)} - h \cdot \nabla Q(w^{(t)}), \quad \nabla Q(w) = \left(\frac{\partial Q(w)}{\partial w_j} \right)_{j=0}^n,$$

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: $\bar{Q} := \lambda \varepsilon_i + (1 - \lambda) \bar{Q}$;

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 рекуррентная 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:

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

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

Экспоненциальное скользящее среднее:

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

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

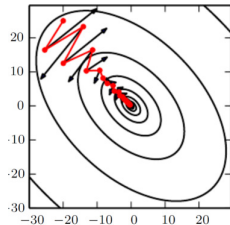
Параметр λ (порядка $\frac{1}{m}$) — темп забывания предыстории ряда.

Метод накопления инерции (momentum)

Momentum — экспоненциальное скользящее среднее градиента по последним $\approx \frac{1}{1-\gamma}$ итерациям [Б.Т.Поляк, 1964]:

$$v := \gamma v + (1-\gamma) \mathcal{L}'_i(w)$$

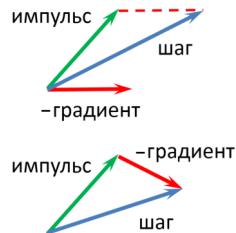
$$w := w - hv$$



NAG (Nesterov's accelerated gradient) — стохастический градиент с инерцией [Ю.Е.Нестеров, 1983]:

$$v := \gamma v + (1-\gamma) \mathcal{L}'_i(w - hv)$$

$$w := w - hv$$



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.

Эта 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 объекты, на которых ошибка больше:
чем меньше M_i , тем больше вероятность взять объект;
- 3 чаще брать объекты, на которых уверенность меньше:
чем меньше $|M_i|$, тем больше вероятность взять объект;
- 4 вообще не брать «хорошие» объекты, у которых $M_i > \mu_+$
(при этом немного ускоряется сходимость);
- 5 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^* ;

При квадратичной функции потерь $h^* = \|x_i\|^{-2}$.

- 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.

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: $\exists u \in \mathbb{R}^n: \forall x \in X \langle u, x \rangle = 0$
неединственность решения: $\forall \gamma \in \mathbb{R} \ a(x, w) = \text{sign}\langle w + \gamma u, x \rangle$

Проявления переобучения:

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

Простой способ уменьшить переобучение:

- регуляризация $\|w\| \rightarrow \min$ (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.

SG: Достоинства и недостатки

Достоинства:

- 1 легко реализуется;
- 2 легко обобщается на любые $g(x, w)$, $\mathcal{L}(a, y)$;
- 3 легко добавить регуляризацию
- 4 возможно динамическое (потокое) обучение;
- 5 на сверхбольших выборках можно получить неплохое решение, даже не обработав все (x_i, y_i) ;
- 6 подходит для задач с большими данными

Недостатки:

- 1 подбор комплекса эвристик является искусством (не забыть про переобучение, застревание, расходимость)

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 :

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

Логарифм правдоподобия (log-likelihood, log-loss):

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

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 (Maximum Likelihood, ML):*

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

- *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

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

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

Вероятностный 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 апостериорной вероятности
(Maximum a Posteriori Probability, MAP):

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

Примеры: априорные распределения Гаусса и Лапласа

Пусть веса w_j независимы, $E w_j = 0$, $D w_j = C$.

Распределение Гаусса и квадратичный (L_2) регуляризатор:

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

Распределение Лапласа и абсолютный (L_1) регуляризатор:

$$p(w; C) = \frac{1}{(2C)^n} \exp\left(-\frac{\|w\|}{C}\right), \quad \|w\| = \sum_{j=1}^n |w_j|,$$
$$-\ln p(w; C) = \frac{1}{C} \|w\| + \text{const}$$

C — гиперпараметр, $\tau = \frac{1}{C}$ — коэффициент регуляризации.

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 $M = \langle w, x \rangle y$.

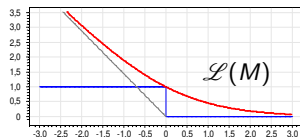
Логарифмическая функция потерь:

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

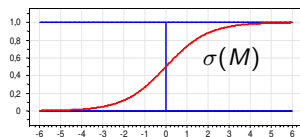
Модель условной вероятности:

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

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



M



M

Максимизация правдоподобия (logistic loss) с регуляризацией:

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

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.

Максимизация правдоподобия (log-loss) с регуляризацией:

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

Пример. Бинаризация признаков и скоринговая карта

Задача кредитного скоринга:

- x_i — заёмщики
- $y_i = -1$ (bad), $+1$ (good)

Бинаризация признаков $f_j(x)$:

$$b_{jk}(x) = [f_j(x) \text{ из } k\text{-го интервала}]$$

Линейная модель классификации:

$$a(x, w) = \text{sign} \sum_{j,k} w_{jk} b_{jk}(x).$$

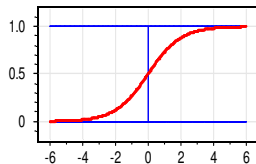
Вес признака w_{jk} равен его вкладу в общую сумму баллов (score).

признак j	интервал k	w_{jk}
Возраст	до 25	5
	25 - 40	10
	40 - 50	15
	50 и больше	10
Собственность	владелец	20
	совладелец	15
	съемщик	10
	другое	5
Работа	руководитель	15
	менеджер среднего звена	10
	служащий	5
	другое	0
Стаж	1/безработный	0
	1..3	5
	3..10	10
	10 и больше	15
Работа_мужа /жены	нет/домохозяйка	0
	руководитель	10
	менеджер среднего звена	5
	служащий	1

Оценивание рисков в скоринге

Логистическая регрессия не только определяет веса w , но и оценивает *апостериорные вероятности* классов

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



Оценка *риска* (математического ожидания) потерь объекта x :

$$R(x) = \sum_{y \in Y} D_{xy} P(y|x),$$

где D_{xy} — величина потери для объекта x с исходом y .

Оценка говорит о том, сколько мы потеряем в среднем.
Но сколько мы потеряем в худшем случае?

Методика VaR (Value at Risk)

Стохастическое моделирование: $N = 10^4$ раз

- для каждого x_i разыгрывается исход $y_i \sim P(y|x_i)$;
- вычисляется сумма потерь по портфелю $V = \sum_{i=1}^{\ell} D_{x_i y_i}$;

99%-квантиль эмпирического распределения потерь определяет величину резервируемого капитала



Резюме в конце лекции

- Метод стохастического градиента (SG) 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 $P(y|x)$