Линейные методы классификации: метод стохастического градиента

К. В. Воронцов
vokov@forecsys.ru

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

март 2014

Содержание

1. **Градиентные методы обучения**
   - Минимизация эмпирического риска
   - Линейный классификатор
   - Метод стохастического градиента
2. **Порождающие и разделяющие модели**
   - Принцип максимума правдоподобия
   - Регуляризация правдоподобия
   - Примеры
3. **Балансировка ошибок и ROC-кривая**
   - Определение ROC-кривой
   - Эффективное построение ROC-кривой
   - Градиентная максимизация AUC

Задача построения разделяющей поверхности

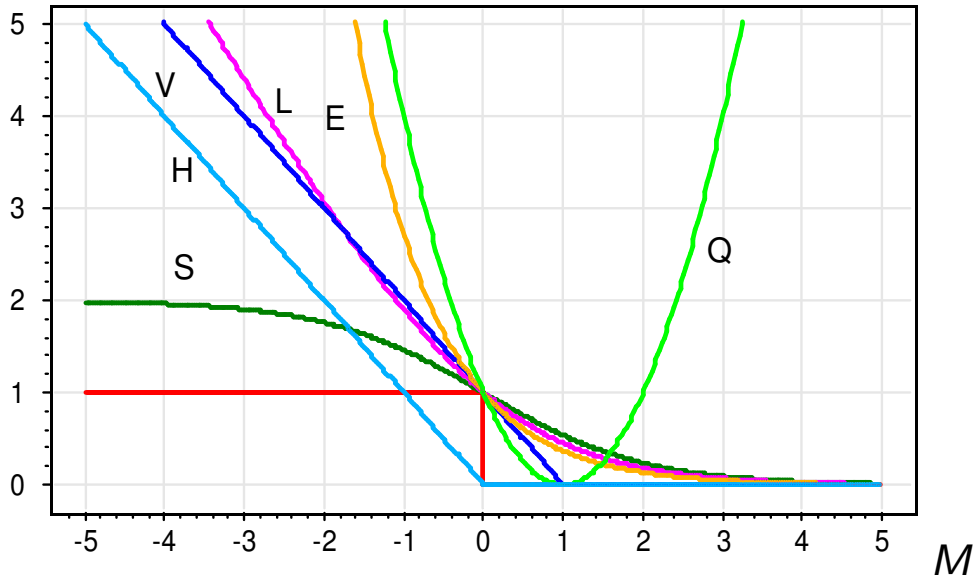
- Задача классификации с двумя классами, $Y = \{-1, +1\}$:
  по обучающей выборке $X^\ell = (x_i, y_i)_{i=1}^\ell$ построить
  алгоритм классификации $a(x, w) = \operatorname{sign} f(x, w)$, где
  $f(x, w)$ — *разделяющая (дискриминантная) функция*,
  w — вектор параметров.
- $f(x, w) = 0$ — разделяющая поверхность;
  $M_i(w) = y_i f(x_i, w)$ — *отступ* (margin) объекта x_i;
  $M_i(w) < 0 \iff$ алгоритм $a(x, w)$ ошибается на x_i.
- *Минимизация эмпирического риска*:

$$Q(w) = \sum_{i=1}^{\ell} \big[M_i(w) < 0\big] \quad \leqslant \quad \widetilde{Q}(w) = \sum_{i=1}^{\ell} \mathscr{L}\big(M_i(w)\big) \to \min_w;$$

функция потерь $\mathscr{L}(M)$ невозрастающая, неотрицательная.

Непрерывные аппроксимации пороговой функции потерь

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



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

Линейный классификатор

$f_j : X \to \mathbb{R}$, $j = 1, \ldots, n$ — числовые признаки;
Линейный алгоритм классификации:

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

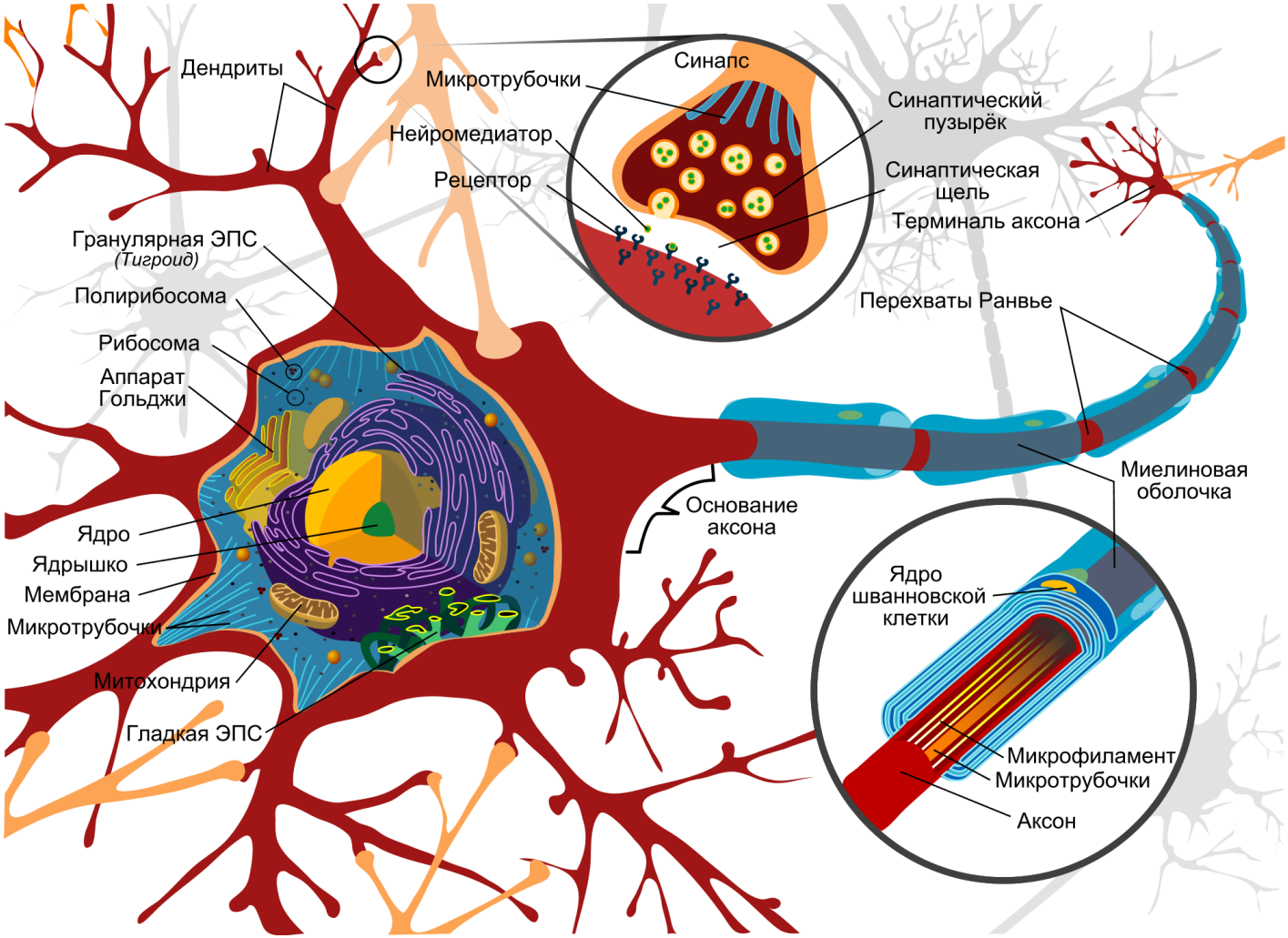
где $w_0, w_1, \ldots, w_n \in \mathbb{R}$ — коэффициенты (веса признаков);
Введём константный признак $f_0 \equiv -1$.
Векторная запись:

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

Отступы объектов x_i:

$$M_i(w) = \langle w, x_i \rangle y_i.$$

Похож ли нейрон на линейный классификатор?

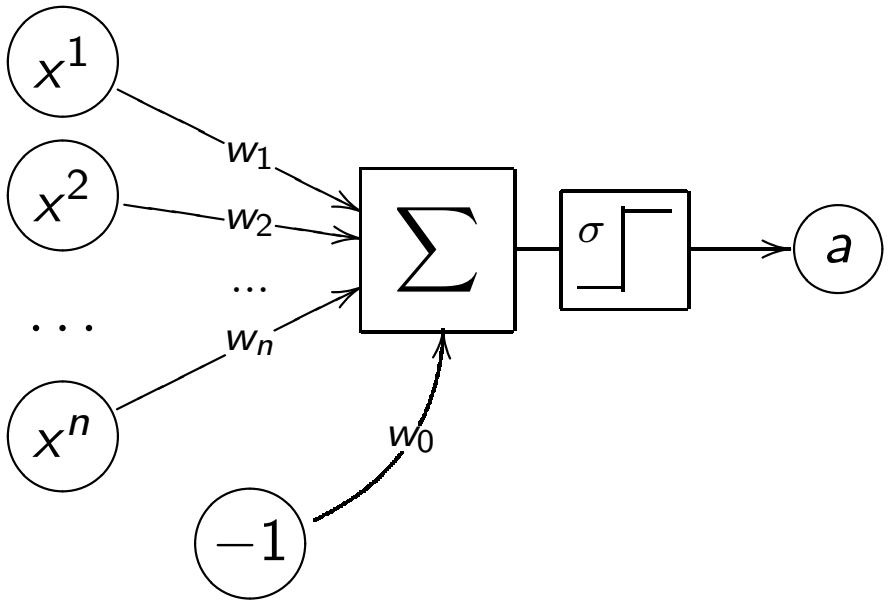


Математическая модель нейрона

Линейная модель нейрона МакКаллока-Питтса [1943]:

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

где $\sigma(s)$ — функция активации (в частности, sign).



Градиентный метод численной минимизации

Минимизация аппроксимированного эмпирического риска:

$$Q(w; X^\ell) = \sum_{i=1}^{\ell} \mathscr{L}\big(\langle w, x_i\rangle y_i\big) \to \min_w .$$

Численная минимизация методом *градиентного спуска*:

$$w^{(0)} := \text{начальное приближение;}$$

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

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

$$w^{(t+1)} := w^{(t)} - \eta \sum_{i=1}^{\ell} \mathscr{L}'\big(\langle w^{(t)}, x_i\rangle y_i\big)\, x_i\, y_i.$$

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 η; параметр λ;

Выход:

веса $w_0, w_1, \ldots, w_n$;

1: инициализировать веса w_j, $j = 0, \ldots, n$;
2: инициализировать текущую оценку функционала:
$Q := \sum_{i=1}^{\ell} \mathscr{L}\big(\langle w, x_i\rangle y_i\big)$;
3: **повторять**
4: выбрать объект x_i из X^ℓ (например, случайно);
5: вычислить потерю: $\varepsilon_i := \mathscr{L}\big(\langle w, x_i\rangle y_i\big)$;
6: градиентный шаг: $w := w - \eta \mathscr{L}'\big(\langle w, x_i\rangle y_i\big) x_i y_i$;
7: оценить значение функционала: $Q := (1-\lambda)Q + \lambda\varepsilon_i$;
8: **пока** значение Q и/или веса w не стабилизируются;

Частный случай №1: дельта-правило ADALINE

Задача регрессии: $X = \mathbb{R}^{n+1}$, $Y \subseteq \mathbb{R}$,

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

Адаптивный линейный элемент ADALINE
[Видроу и Хофф, 1960]:

$$a(x, w) = \langle w, x \rangle$$

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

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

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

Частный случай №2: правило Хэбба

Задача классификации: $X = \mathbb{R}^{n+1}$, $Y = \{-1, +1\}$,

$$\mathscr{L}(a, y) = (-\langle w, x\rangle y)_+.$$

Линейный классификатор:

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

Градиентный шаг — правило Хэбба [1949]:

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

Если $X = \{0, 1\}^n$, $Y = \{0, +1\}$, то правило Хэбба переходит в правило перцептрона Розенблатта [1957]:

$$w := w - \eta\big(a(x_i, w) - y_i\big)x_i.$$

Обоснование Алгоритма SG с правилом Хэбба

Задача классификации: $X = \mathbb{R}^{n+1}$, $Y = \{-1, 1\}$.

Теорема (Новиков, 1962)

Пусть выборка X^ℓ линейно разделима:

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

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 $\eta > 0$;
- независимо от порядка предъявления объектов x_i;
- за конечное число исправлений вектора w;
- если $w^{(0)} = 0$, то число исправлений $t_{\max} \leqslant \frac{1}{\delta^2} \max \|x_i\|$.

SG: Инициализация весов

Возможны варианты:

1. $w_j := 0$ для всех $j = 0, \ldots, n$;
2. небольшие случайные значения:
   $w_j := \text{random}\left(-\frac{1}{2n}, \frac{1}{2n}\right)$;
3. $w_j := \frac{\langle y, f_j \rangle}{\langle f_j, f_j \rangle}$, $f_j = \left(f_j(x_i)\right)_{i=1}^{\ell}$ — 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. многократные запуски из разных случайных начальных приближений и выбор лучшего решения.

SG: Порядок 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.

SG: Выбор величины градиентного шага

Возможны варианты:

1. сходимость гарантируется (для выпуклых функций) при
$$\eta_t \to 0, \quad \sum_{t=1}^{\infty} \eta_t = \infty, \quad \sum_{t=1}^{\infty} \eta_t^2 < \infty,$$
в частности можно положить $\eta_t = 1/t$;
2. *метод скорейшего градиентного спуска*:
$$Q\big(w - \eta \nabla Q(w)\big) \to \min_{\eta},$$
позволяет найти *адаптивный шаг* η^*;

   Упражнение: доказать, что при квадратичной функции потерь $\eta^* = \|x_i\|^{-2}$.
3. пробные случайные шаги
   — для «выбивания» из локальных минимумов;

SG: Достоинства и недостатки

Достоинства:

1. легко реализуется;
2. легко обобщается на любые f, $\mathscr{L}$;
3. возможно динамическое (потоковое) обучение;
4. на сверхбольших выборках не обязательно брать все x_i;

Недостатки:

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

SG: 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: $\exists u \in \mathbb{R}^{n+1}$: $\langle u, x\rangle \equiv 0$;
   тогда $\forall \gamma \in \mathbb{R} \quad a(x, w) = \text{sign}\langle w + \gamma u, x\rangle$

Симптоматика:

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

Терапия:

1. сокращение весов (weight decay);
2. ранний останов (early stopping);

SG: Сокращение весов

Штраф за увеличение нормы вектора весов:

$$Q_\tau(w; X^\ell) = Q(w; X^\ell) + \frac{\tau}{2}\|w\|^2 \to \min_w.$$

Градиент:

$$\nabla Q_\tau(w) = \nabla Q(w) + \tau w.$$

Модификация градиентного шага:

$$w := w(1 - \eta\tau) - \eta\nabla Q(w).$$

Подбор параметра регуляризации τ:

1. скользящий контроль;
2. стохастическая адаптация;
3. байесовский вывод второго уровня;

Принцип максимума правдоподобия

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

- *Максимизация правдоподобия:*

$$L(w; X^\ell) = \ln \prod_{i=1}^{\ell} p(x_i, y_i|w) = \sum_{i=1}^{\ell} \ln p(x_i, y_i|w) \to \max_w.$$

- *Минимизация аппроксимированного эмпирического риска:*

$$\widetilde{Q}(w; X^\ell) = \sum_{i=1}^{\ell} \mathscr{L}\big(y_i f(x_i, w)\big) \to \min_w;$$

- Эти два принципа эквивалентны, если положить

$$-\ln p(x_i, y_i|w) = \mathscr{L}\big(y_i f(x_i, w)\big).$$

порождающая модель p $\rightleftarrows$ разделяющая модель f и $\mathscr{L}$

Обобщение: вероятностная (байесовская) регуляризация

$p(x, y|w)$ — вероятностная модель порождения 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 случайным.

Совместное правдоподобие данных и модели:

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

Принцип максимума совместного правдоподобия:

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

Пример 1: 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 параметра регуляризации: $\tau = \frac{1}{\sigma}$.

Пример 2: лапласовский регуляризатор

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

$$p(w; C) = \frac{1}{(2C)^n} \exp\left(-\frac{\|w\|_1}{C}\right), \quad \|w\|_1 = \sum_{j=1}^{n} |w_j|,$$

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

Логарифмируя, получаем регуляризатор по L_1-норме:

$$-\ln p(w; C) = \frac{1}{C} \sum_{j=1}^{n} |w_j| + \text{const}(w).$$

Почему этот регуляризатор приводит к отбору признаков?

Пример 2: лапласовский регуляризатор

$$Q(w, X^\ell) = \sum_{i=1}^{\ell} \ln p(x_i, y_i|w) + \frac{1}{C}\sum_{j=1}^{n} |w_j| \to \min_{w,C}.$$

Почему этот регуляризатор приводит к отбору признаков:

Замена переменных: $u_j = \frac{1}{2}\big(|w_j| + w_j\big)$, $v_j = \frac{1}{2}\big(|w_j| - w_j\big)$.
Тогда $w_j = u_j - v_j$ и $|w_j| = u_j + v_j$;

$$\begin{cases} Q(u,v) = \sum_{i=1}^{\ell} \mathscr{L}\big(M_i(u-v, w_0)\big) + \frac{1}{C}\sum_{j=1}^{n}(u_j + v_j) \to \min_{u,v} \\ u_j \geqslant 0, \quad v_j \geqslant 0, \quad j = 1, \ldots, n; \end{cases}$$

чем больше C, тем больше ограничений-неравенств активны,
но если $u_j = v_j = 0$, то вес $w_j = 0$ и признак не учитывается.

Регуляризация в линейных классификаторах

- В случае мультиколлинеарности
  — решение $Q(w) \to \min\limits_{w}$ неединственно или неустойчиво;
  — классификатор $a(x; w)$ неустойчив;
  — переобучение: $Q(X^\ell) \ll Q(X^k)$.
- Регуляризация — это выбор наиболее устойчивого решения
  — Гаусс — без отбора признаков;
  — Лаплас — с отбором признаков;
  — возможны комбинации (ElasticNet) и другие варианты...
- Выбор параметра регуляризации τ:
  — с помощью скользящего контроля;
  — с помощью оценок обобщающей способности;
  — стохастическая адаптация;
  — байесовский вывод второго уровня.

Зоопарк методов

- Вид разделяющей поверхности $f(x, w)$:
  — линейная $f(x, w) = \langle x, w \rangle$;
  — нелинейная;
- Вид непрерывной аппроксимации функции потерь $\mathscr{L}(M)$:
  — логарифмическая $\mathscr{L}(M) = \log(1 + e^{-M})$ ... LR;
  — кусочно-линейная $\mathscr{L}(M) = (1 - M)_+$ ... SVM;
  — экспоненциальная $\mathscr{L}(M) = e^{-M}$ ... AdaBoost;
- Вид регуляризатора $-\log p(w; \gamma)$:
  — равномерный ... персептроны, LR;
  — гауссовский с равными дисперсиями ... SVM, RLR;
  — гауссовский с неравными дисперсиями ... RVM;
  — лапласовский ... приводит к отбору признаков;
- Вид численного метода оптимизации $Q(w) \to \min$.

Балансировка ошибок I и II рода

Задача классификации на два класса, $Y = \{-1, +1\}$;
Модель классификации: $a(x, w, w_0) = \text{sign}\big(f(x, w) - w_0\big)$.

$a(x_i, w) = -1$, $y_i = +1$ — ложно-отрицательная классификация («пропуск цели», ошибка I рода)

$a(x_i, w) = +1$, $y_i = -1$ — ложно-положительная классификация («ложная тревога», ошибка II рода)

На практике цена ошибок I и II рода может быть неизвестна или многократно пересматриваться.

Постановка задачи

- Выбирать w_0 без обучения w заново.
- Ввести характеристику качества классификатора, инвариантную относительно выбора цены потерь.

Определение ROC-кривой

ROC — «receiver operating characteristic».

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

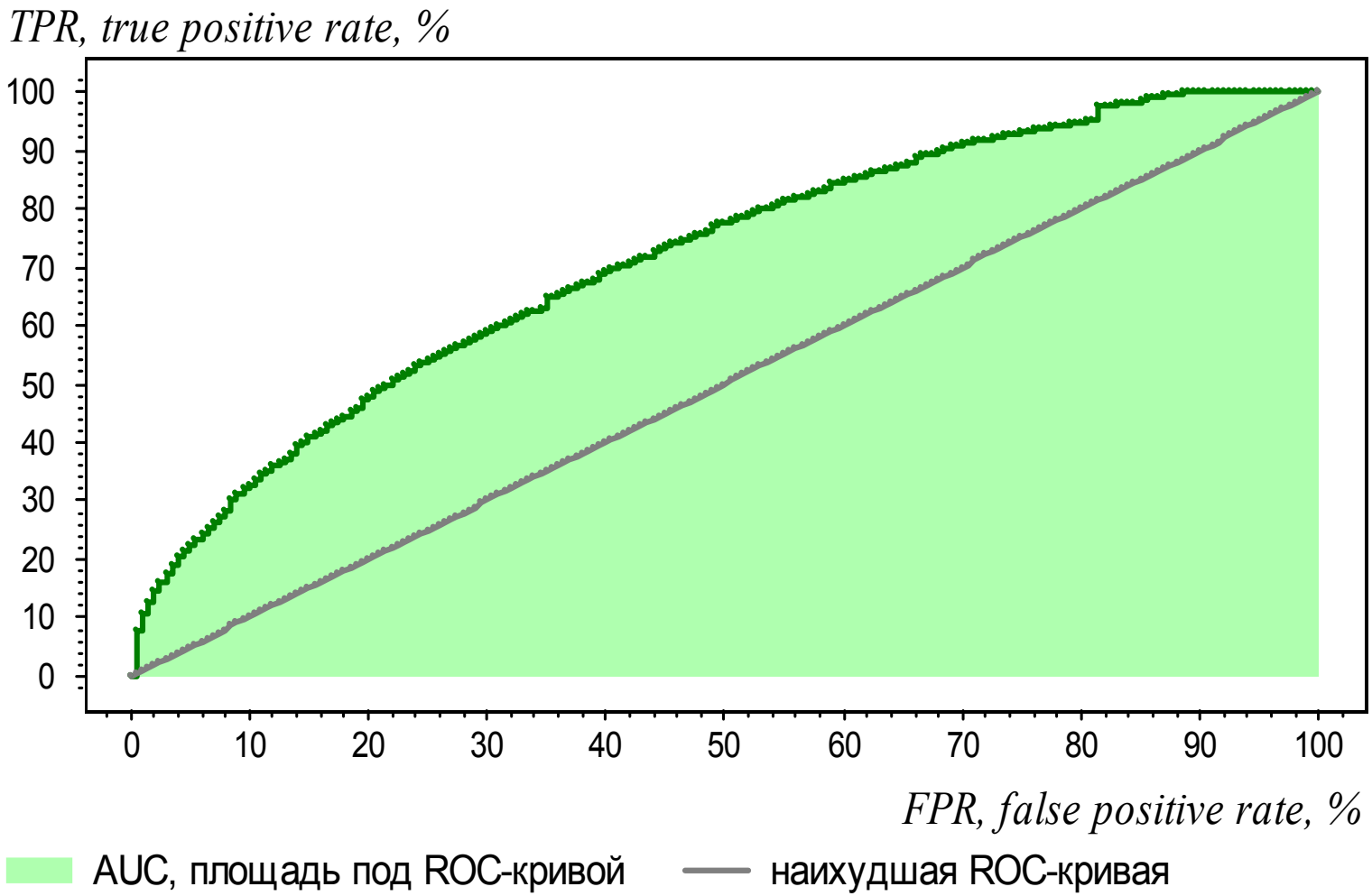
$$\text{FPR}(a, X^\ell) = \frac{\sum_{i=1}^{\ell} [y_i = -1][a(x_i; w, w_0) = +1]}{\sum_{i=1}^{\ell} [y_i = -1]};$$

  $1 - \text{FPR}(a)$ называется *специфичностью* алгоритма a.
- по оси Y: доля *верно-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 ROC-кривой



Алгоритм эффективного построения ROC-кривой

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

Выход: $\{(\mathrm{FPR}_i, \mathrm{TPR}_i)\}_{i=0}^{\ell}$, AUC — площадь под ROC-кривой.

$\ell_y := \sum_{i=1}^{\ell} [y_i = y]$, для всех $y \in Y$;

упорядочить выборку X^ℓ по убыванию значений $f(x_i, w)$;

поставить первую точку в начало координат:
$(\mathrm{FPR}_0, \mathrm{TPR}_0) := (0, 0)$; $\mathrm{AUC} := 0$;

для $i := 1, \ldots, \ell$

  если $y_i = -1$ **то** сместиться на один шаг вправо:

    $\mathrm{FPR}_i := \mathrm{FPR}_{i-1} + \frac{1}{\ell_-}$; $\mathrm{TPR}_i := \mathrm{TPR}_{i-1}$;

    $\mathrm{AUC} := \mathrm{AUC} + \frac{1}{\ell_-}\mathrm{TPR}_i$;

  иначе сместиться на один шаг вверх:

    $\mathrm{FPR}_i := \mathrm{FPR}_{i-1}$; $\mathrm{TPR}_i := \mathrm{TPR}_{i-1} + \frac{1}{\ell_+}$;

Градиентная максимизация AUC

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

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

$$\text{AUC} = \frac{1}{\ell_-}\sum_{i=1}^{\ell}\big[y_i = -1\big]\text{TPR}_i = $$

$$= \frac{1}{\ell_-\ell_+}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\big[y_i < y_j\big]\big[f(x_i, w) < f(x_j, w)\big] \to \max_w.$$

Явная максимизация аппроксимированного AUC:

$$Q(w) = \sum_{i,j:\ y_i<y_j} \mathscr{L}\big(\underbrace{f(x_j, w) - f(x_i, w)}_{M_{ij}(w)}\big) \to \min_w,$$

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

Резюме в конце лекции

- Методы обучения линейных классификаторов отличаются
  — видом функции потерь;
  — видом регуляризатора;
  — численным методом оптимизации.
- *Аппроксимация пороговой функции потерь* гладкой убывающей функцией отступа $\mathscr{L}(M)$ повышает качество классификации (за счёт увеличения зазора)
  и облегчает оптимизацию.
- *Регуляризация* решает проблему мультиколлинеарности
  и также снижает переобучение.
- *Максимизация AUC* не зависит от соотношения штрафов за ошибки I и II рода.