Линейные методы классификации

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)»

20 февраля 2010

Содержание

Задача построения разделяющей поверхности

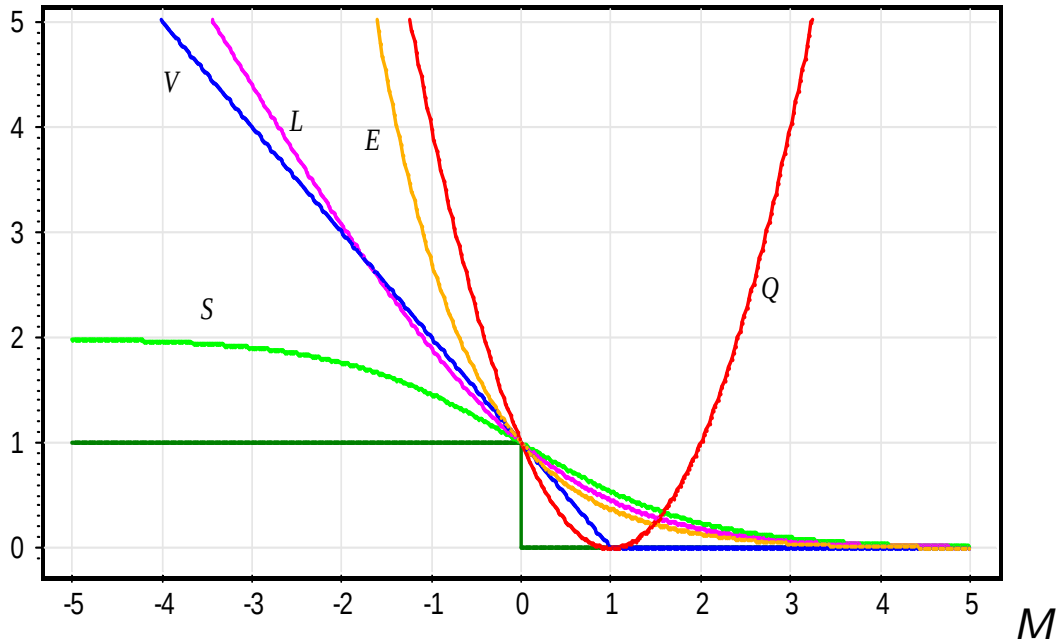
- Задача классификации с двумя классами, $Y = \{-1, +1\}$: по обучающей выборке $X^\ell = (x_i, y_i)_{i=1}^\ell$ построить алгоритм классификации $a(x, w) = \operatorname{sign} f(x, w)$, где
  $f(x, w)$ — *дискриминантная функция*,
  w — вектор параметров.
- $f(x, w) = 0$ — разделяющая поверхность;
  $M_i(w) = y_i f(x_i, w)$ — *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 $\mathscr{L}(M)$:



$Q(M) = (1 - M)^2$ — квадратичная (ЛДФ);
$V(M) = (1 - M)_+$ — кусочно-линейная (SVM);
$S(M) = 2(1 + e^M)^{-1}$ — сигмоидная (нейросети);
$L(M) = \log_2(1 + e^{-M})$ — логарифмическая (LR);
$E(M) = e^{-M}$ — экспоненциальная (AdaBoost).

Связь с принципом максимума правдоподобия

Пусть $X \times Y$ — в.п. с плотностью $p(x, y|w)$.
Пусть X^ℓ — 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

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

$$\boxed{\text{модель } p} \rightleftarrows \boxed{\text{модель } f \text{ и функция потерь } \mathscr{L}}.$$

Линейный классификатор

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

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

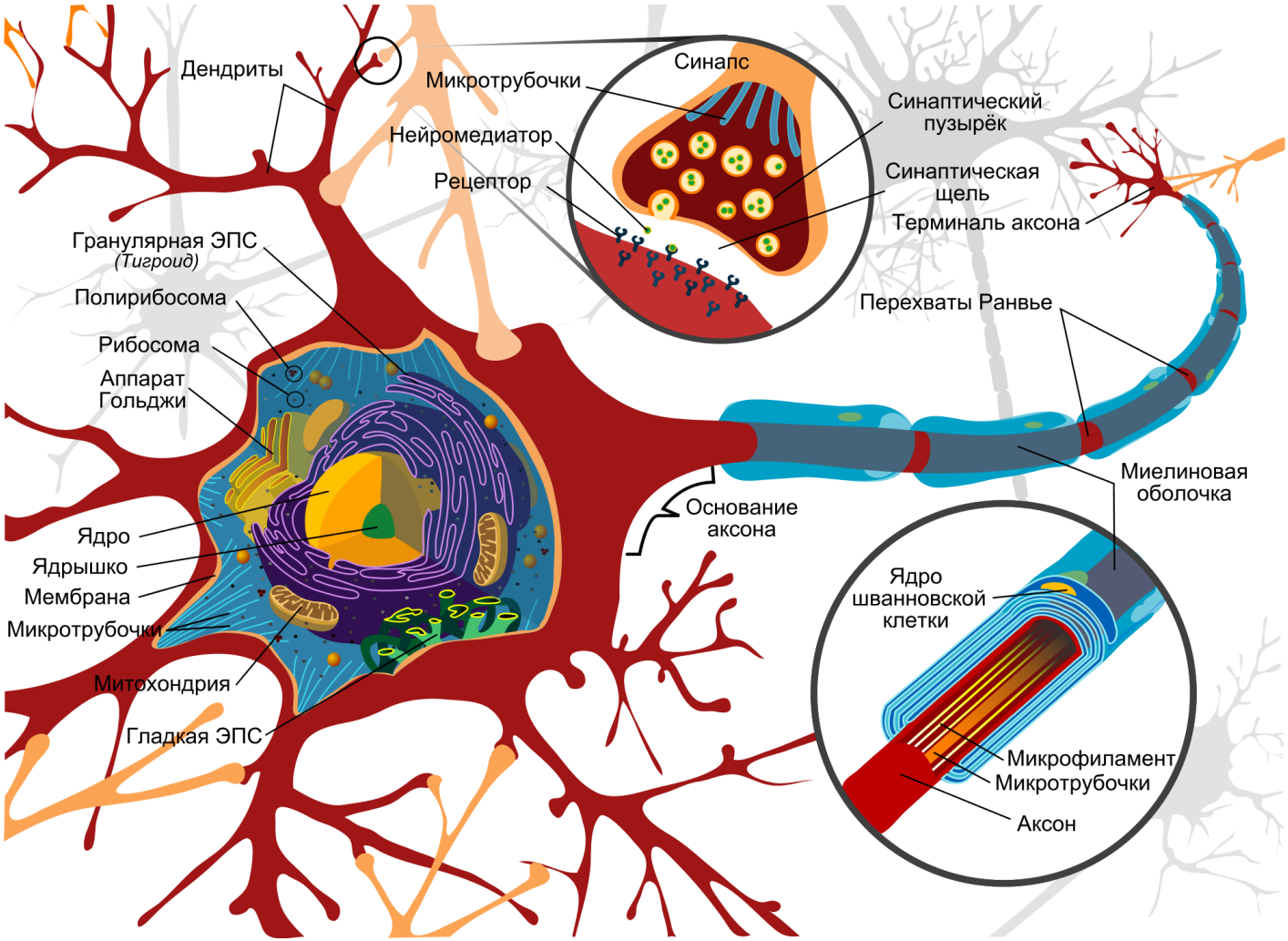
где $w_0, w_1, \ldots, w_n \in \mathbb{R}$ — коэффициенты (веса признаков);
Введём константный признак $f_0 \equiv -1$.
Векторная запись:

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

Отступы объектов x_i:

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

Похож ли нейрон на линейный классификатор?

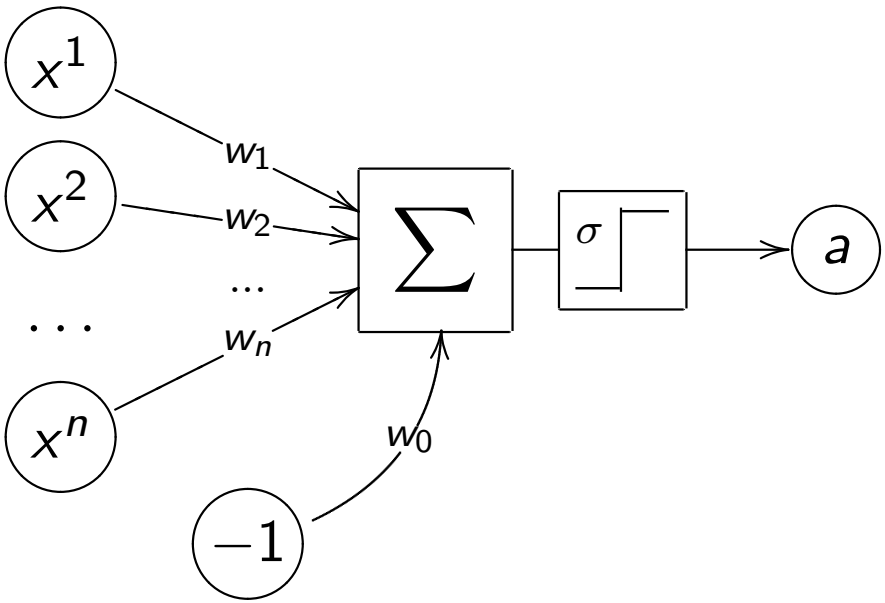


Математическая модель нейрона

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: $X = \mathbb{R}^n$, $Y \subseteq \mathbb{R}$,

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

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

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

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. наивный линейный байесовский классификатор
   по небольшой случайной подвыборке объектов;
4. $w_j := \frac{\langle y, f_j \rangle}{\langle f_j, f_j \rangle}$,
   где $f_j = \left(f_j(x_i)\right)_{i=1}^{\ell}$ — вектор значений j-го признака.

Упражнение: в последнем случае доказать, что если функция потерь квадратична и признаки некоррелированы, то оценка w является оптимальной.

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);
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^*;
3. пробные случайные шаги
— для «выбивания» из локальных минимумов;

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

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;
3. наличие «шумовых» неинформативных признаков;

Симптоматика:

1. резкое увеличение $\|w\|$;
2. неустойчивость классификации;
3. $Q(X^\ell) \ll Q(X^k)$;

Терапия:

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

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;

Обобщение: байесовская регуляризация

$p(x, y|w)$ — 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 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 $|w_j| = u_j + v_j$;

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

если $u_j = v_j = 0$, то вес $w_j = 0$ и признак не учитывается.

Регуляризация в линейных классификаторах

- В случае мультиколлинеарности
  — решение $Q(w) \to \min_w$ 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 и другие варианты...
- Выбор параметра регуляризации:
  — 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 $Q(w) \to \min$.

Логистическая регрессия: базовые предположения

- $X = \mathbb{R}^n$, $Y = \pm 1$, выборка $X^\ell = (x_i, y_i)_{i=1}^\ell$ i.i.d. из

$$p(x, y) = P_y p_y(x) = \mathsf{P}(y|x)p(x)$$

- Функции правдоподобия $p_y(x)$ экспонентные:

$$p_y(x) = \exp\big(c_y(\delta)\langle\theta_y, x\rangle + b_y(\delta, \theta_y) + d(x, \delta)\big),$$

где $\theta_y \in \mathbb{R}^n$ — параметр *сдвига*;
δ — параметр *разброса*;
b_y, c_y, d — произвольные числовые функции;
причём параметры $d(\cdot)$ и δ не зависят от y.

Класс экспонентных распределений очень широк:
равномерное, нормальное, гипергеометрическое, пуассоновское, биномиальное, Г-распределение, и др.

Пример: гауссовская плотность — экспонентная

Многомерное нормальное распределение, $\mu \in \mathbb{R}^n$, $\Sigma \in \mathbb{R}^{n\times n}$, является экспонентным:

параметр сдвига $\theta = \Sigma^{-1}\mu$;

параметр разброса $\delta = \Sigma$.

$$\mathcal{N}(x;\mu,\Sigma) = (2\pi)^{-\frac{n}{2}}|\Sigma|^{-\frac{1}{2}}\exp\Big(-\tfrac{1}{2}(x-\mu)^{\mathsf{T}}\Sigma^{-1}(x-\mu)\Big) =$$

$$= \exp\Big(\underbrace{\mu^{\mathsf{T}}\Sigma^{-1}x}_{\langle\theta,x\rangle} - \underbrace{\tfrac{1}{2}\mu^{\mathsf{T}}\Sigma^{-1}\Sigma\Sigma^{-1}\mu}_{b(\delta,\theta)} -$$

$$- \underbrace{\tfrac{1}{2}x^{\mathsf{T}}\Sigma^{-1}x - \tfrac{n}{2}\ln(2\pi) - \tfrac{1}{2}\ln|\Sigma|}_{d(x,\delta)}\Big).$$

Основная теорема

Оптимальный байесовский классификатор для двух классов:

$$a(x) = \text{sign}\big(\lambda_+ \text{P}(+1|x) - \lambda_- \text{P}(-1|x)\big) = \text{sign}\left(\frac{p_+(x)}{p_-(x)} - \frac{\lambda_- P_-}{\lambda_+ P_+}\right).$$

Теорема

Если p_y экспонентны, параметры $d(\cdot)$ и δ не зависят от y,
и среди признаков $f_1(x), \ldots, f_n(x)$ есть константа,
то байесовский классификатор линеен:

$$a(x) = \text{sign}\big(\langle w, x\rangle - w_0\big), \qquad w_0 = \ln(\lambda_-/\lambda_+);$$

апостериорные вероятности классов:

$$\text{P}(y|x) = \sigma\big(\langle w, x\rangle y\big),$$

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

Обоснование логарифмической функции потерь

Максимизация логарифма правдоподобия выборки:

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

Подставим: $p(x, y) = \mathrm{P}(y|x) \cdot p(x) = \sigma\big(\langle w, x\rangle y\big) \cdot \mathrm{const}(w)$

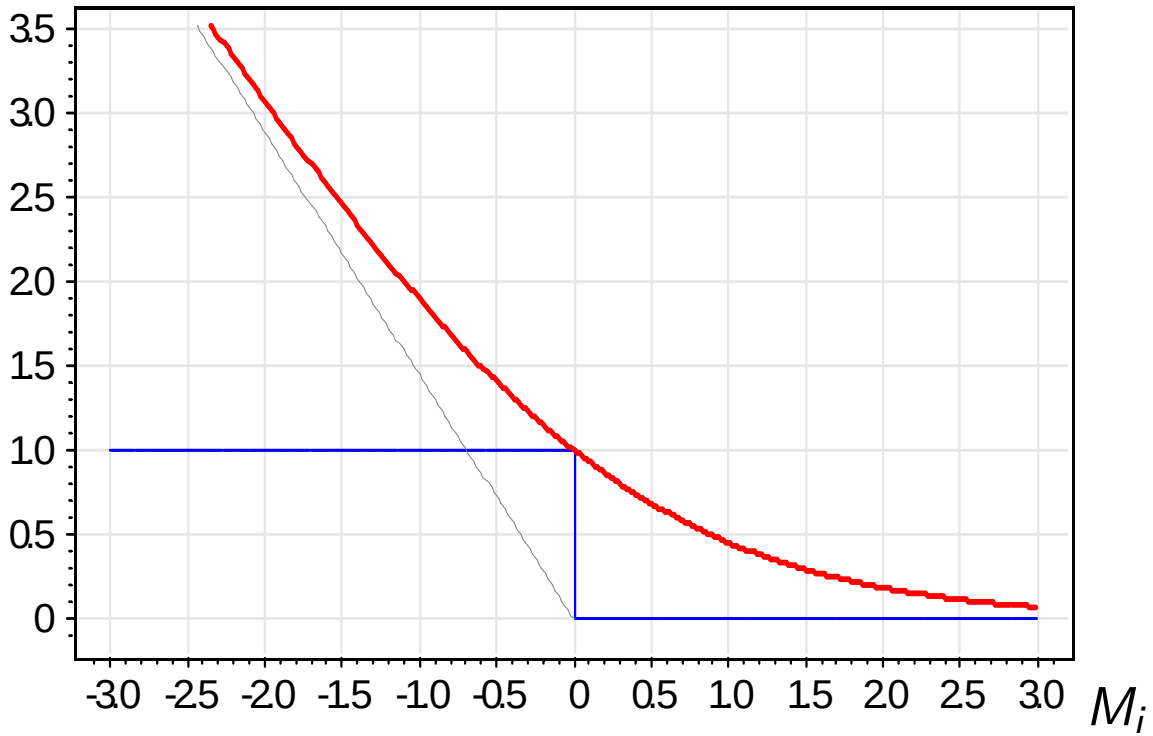
$$L(w, X^\ell) = \sum_{i=1}^{\ell} \log \sigma\big(\langle w, x_i\rangle y_i\big) + \mathrm{const}(w) \to \max_w .$$

Максимизация $L(w)$ эквивалентна минимизации $\widetilde{Q}(w)$:

$$\widetilde{Q}(w, X^\ell) = \sum_{i=1}^{\ell} \log\big(1 + \exp(-\underbrace{\langle w, x_i\rangle y_i}_{M_i(w)})\big) \to \min_w .$$

Логарифмическая функция потерь

Логарифмическая функция потерь $\mathscr{L}(M_i) = \log_2\left(1 + e^{-M_i}\right)$ и её наклонная асимптота.



Градиентный метод

Производная сигмоидной функции: $\sigma'(z) = \sigma(z)\sigma(-z)$.
Вектор градиента функционала $\widetilde{Q}(w)$:

$$\nabla\widetilde{Q}(w) = -\sum_{i=1}^{\ell} y_i x_i \sigma\big(-M_i(w)\big).$$

Градиентный шаг в методе стохастического градиента:

$$w^{(t+1)} := w^{(t)} + \eta\, y_i x_i\, \sigma\big(-M_i(w^{(t)})\big),$$

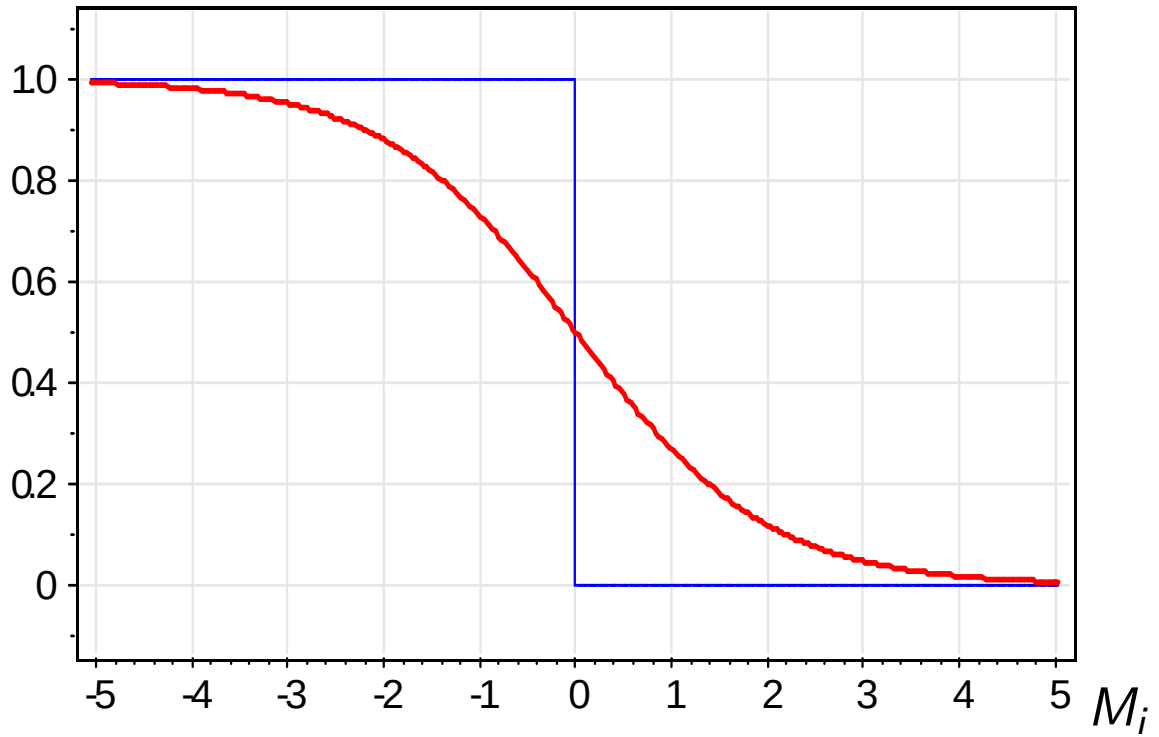
где (x_i, y_i) — предъявляемый прецедент, η — темп обучения.

Оказывается, это «сглаженный вариант» правила Хэбба:

$$w^{(t+1)} := w^{(t)} + \eta\, y_i x_i \,\big[M_i(w^{(t)}) < 0\big].$$

Сглаженное правило Хэбба

Правило Хэбба: пороговое $[M_i < 0]$ и сглаженное $\sigma(-M_i)$:



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

Бинаризация признаков и скоринговая карта

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

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

$$R(x) = \sum_{y \in Y} D_{xy} \, \mathrm{P}(y|x) = \sum_{y \in Y} D_{xy} \sigma\big(\langle w, x \rangle y\big),$$

где D_{xy} — величина потери для (x, y).

Недостатки:

трудно проверить, выполняются ли базовые предположения;
т. е. оценка вероятности носит эвристический характер;

Методика VaR (Value at Risk):

1000 раз: для каждого x разыгрывается исход y; $V = \sum_{i=1}^{\ell} D_{xy}$;
строится эмпирическое распределение величины V;
определяется α-квантиль распределения.

Принцип максимума ширины разделяющей полосы

Линейный классификатор:

$$a(x) = \operatorname{sign}\big(\langle w, x\rangle - w_0\big), \quad w, x \in \mathbb{R}^n,\ w_0 \in \mathbb{R}.$$

Пусть выборка $X^\ell = (x_i, y_i)_{i=1}^\ell$ линейно разделима:

$$\exists w, w_0: \quad M_i(w, w_0) = y_i\big(\langle w, x_i\rangle - w_0\big) > 0, \quad i = 1, \ldots, \ell$$

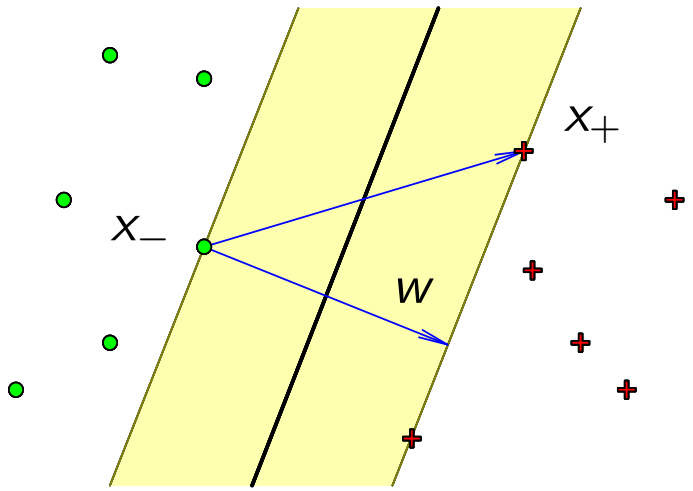
Нормировка: $\min\limits_{i=1,\ldots,\ell} M_i(w, w_0) = 1.$

Разделяющая полоса:

$$\big\{x:\ -1 \leqslant \langle w, x\rangle - w_0 \leqslant 1\big\}.$$

Ширина полосы:

$$\frac{\langle x_+ - x_-, w\rangle}{\|w\|} = \frac{2}{\|w\|} \to \max.$$



Обоснование кусочно-линейной функции потерь

Линейно разделимая выборка

$$\begin{cases} \|w\|^2 \to \min\limits_{w,w_0}; \\ M_i(w,w_0) \geqslant 1, \quad i = 1,\ldots,\ell. \end{cases}$$

Переход к линейно неразделимой выборке (эвристика)

$$\begin{cases} \frac{1}{2}\|w\|^2 + C\sum\limits_{i=1}^{\ell}\xi_i \to \min\limits_{w,w_0,\xi}; \\ M_i(w,w_0) \geqslant 1 - \xi_i, \quad i = 1,\ldots,\ell; \\ \xi_i \geqslant 0, \quad i = 1,\ldots,\ell. \end{cases}$$

Эквивалентная задача безусловной минимизации:

$$Q(w,w_0) = \sum_{i=1}^{\ell}\big(1 - M_i(w,w_0)\big)_+ + \frac{1}{2C}\|w\|^2 \to \min_{w,w_0}.$$

Задача поиска седловой точки функции Лагранжа

Функция Лагранжа: $\mathscr{L}(w, w_0, \xi; \lambda, \eta) =$

$$= \frac{1}{2}\|w\|^2 - \sum_{i=1}^{\ell} \lambda_i \big(M_i(w, w_0) - 1\big) - \sum_{i=1}^{\ell} \xi_i \big(\lambda_i + \eta_i - C\big),$$

λ_i — переменные, двойственные к ограничениям $M_i \geqslant 1 - \xi_i$;
η_i — переменные, двойственные к ограничениям $\xi_i \geqslant 0$.

$$\begin{cases} \mathscr{L}(w, w_0, \xi; \lambda, \eta) \;\to\; \min\limits_{w, w_0, \xi} \; \max\limits_{\lambda, \eta}; \\ \xi_i \geqslant 0, \quad \lambda_i \geqslant 0, \quad \eta_i \geqslant 0, \quad i = 1, \ldots, \ell; \\ \lambda_i = 0 \;\text{ либо }\; M_i(w, w_0) = 1 - \xi_i, \quad i = 1, \ldots, \ell; \\ \eta_i = 0 \;\text{ либо }\; \xi_i = 0, \quad i = 1, \ldots, \ell; \end{cases}$$

Необходимые условия седловой точки функции Лагранжа

Функция Лагранжа: $\mathscr{L}(w, w_0, \xi; \lambda, \eta) =$

$$= \frac{1}{2}\|w\|^2 - \sum_{i=1}^{\ell} \lambda_i \big(M_i(w, w_0) - 1\big) - \sum_{i=1}^{\ell} \xi_i \big(\lambda_i + \eta_i - C\big),$$

Необходимые условия седловой точки функции Лагранжа:

$$\frac{\partial \mathscr{L}}{\partial w} = w - \sum_{i=1}^{\ell} \lambda_i y_i x_i = 0 \quad \Longrightarrow \quad w = \sum_{i=1}^{\ell} \lambda_i y_i x_i;$$

$$\frac{\partial \mathscr{L}}{\partial w_0} = -\sum_{i=1}^{\ell} \lambda_i y_i = 0 \quad \Longrightarrow \quad \sum_{i=1}^{\ell} \lambda_i y_i = 0;$$

$$\frac{\partial \mathscr{L}}{\partial \xi_i} = -\lambda_i - \eta_i + C = 0 \quad \Longrightarrow \quad \eta_i + \lambda_i = C, \quad i = 1, \ldots, \ell.$$

Понятие опорного вектора

Типизация объектов:

1. $\lambda_i = 0;\ \ \eta_i = C;\ \ \xi_i = 0;\ \ M_i \geqslant 1.$
   — периферийные (неинформативные) объекты.

2. $0 < \lambda_i < C;\ \ 0 < \eta_i < C;\ \ \xi_i = 0;\ \ M_i = 1.$
   — опорные граничные объекты.

3. $\lambda_i = C;\ \ \eta_i = 0;\ \ \xi_i > 0;\ \ M_i < 1.$
   — опорные-нарушители.

Определение

Объект x_i называется *опорным*, если $\lambda_i \neq 0$.

Двойственная задача

$$\begin{cases} -\mathscr{L}(\lambda) = -\sum\limits_{i=1}^{\ell} \lambda_i + \frac{1}{2} \sum\limits_{i=1}^{\ell} \sum\limits_{j=1}^{\ell} \lambda_i \lambda_j y_i y_j \langle x_i, x_j \rangle \;\to\; \min\limits_{\lambda}; \\ 0 \leqslant \lambda_i \leqslant C, \quad i = 1, \ldots, \ell; \\ \sum\limits_{i=1}^{\ell} \lambda_i y_i = 0. \end{cases}$$

Решение прямой задачи выражается через решение двойственной:

$$\begin{cases} w = \sum\limits_{i=1}^{\ell} \lambda_i y_i x_i; \\ w_0 = \langle w, x_i \rangle - y_i, \qquad \text{для любого } i\text{: } \lambda_i > 0,\ M_i = 1. \end{cases}$$

Линейный классификатор:

$$a(x) = \operatorname{sign}\Big(\sum_{i=1}^{\ell} \lambda_i y_i \langle x_i, x \rangle - w_0 \Big).$$

Нелинейное обобщение SVM

Переход к спрямляющему пространству более высокой размерности: $\psi\colon X \to H$.

Определение

Функция $K\colon X \times X \to \mathbb{R}$ — *ядро*, если $K(x, x') = \langle \psi(x), \psi(x') \rangle$ при некотором $\psi\colon X \to H$, где H — гильбертово пространство.

Теорема

Функция $K(x, x')$ является ядром тогда и только тогда, когда
- она симметрична: $K(x, x') = K(x', x)$;
- и неотрицательно определена:
  $\int_X \int_X K(x, x') g(x) g(x') dx dx' \geqslant 0$ для любой $g\colon X \to \mathbb{R}$.

Конструктивные методы синтеза ядер

1. $K(x,x') = \langle x, x' \rangle$ — ядро;
2. константа $K(x,x') = 1$ — ядро;
3. произведение ядер $K(x,x') = K_1(x,x')K_2(x,x')$ — ядро;
4. $\forall \psi : X \to \mathbb{R}$ произведение $K(x,x') = \psi(x)\psi(x')$ — ядро;
5. $K(x,x') = \alpha_1 K_1(x,x') + \alpha_2 K_2(x,x')$ при $\alpha_1, \alpha_2 > 0$ — ядро;
6. $\forall \varphi : X \to X$ если K_0 ядро, то $K(x,x') = K_0(\varphi(x), \varphi(x'))$ — ядро;
7. если $s\colon X \times X \to \mathbb{R}$ — симметричная интегрируемая функция, то $K(x,x') = \int_X s(x,z)s(x',z)\,dz$ — ядро;
8. если K_0 — ядро и функция $f : \mathbb{R} \to \mathbb{R}$ представима в виде сходящегося степенного ряда с неотрицательными коэффициентами, то $K(x,x') = f\big(K_0(x,x')\big)$ — ядро;

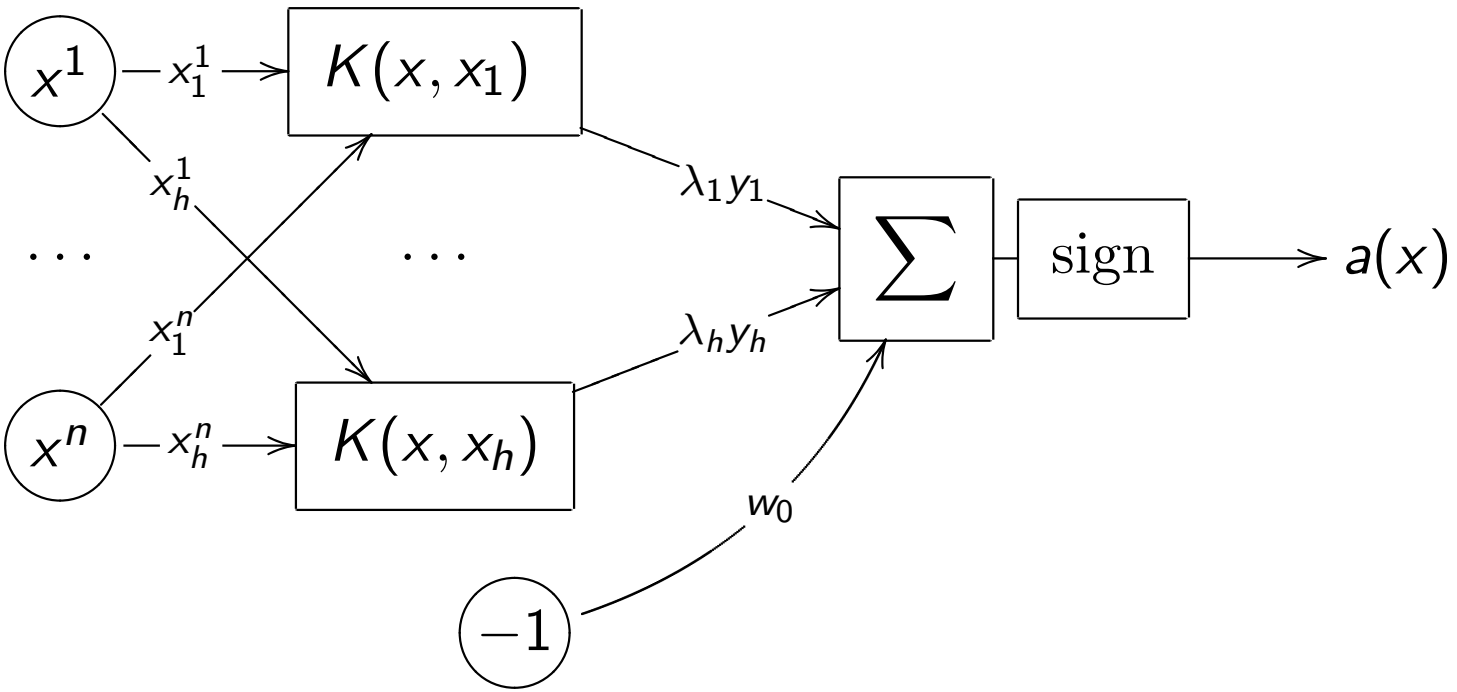
Примеры ядер

1. $K(x,x') = \langle x,x'\rangle^2$
   — квадратичное ядро;
2. $K(x,x') = \langle x,x'\rangle^d$
   — полиномиальное ядро с мономами степени d;
3. $K(x,x') = \big(\langle x,x'\rangle + 1\big)^d$
   — полиномиальное ядро с мономами степени $\leqslant d$;
4. $K(x,x') = \operatorname{th}\big(k_0 + k_1\langle x,x'\rangle\big)$
   — нейросеть с сигмоидными функциями активации;
5. $K(x,x') = \exp\big(-\beta\|x-x'\|^2\big)$
   — сеть радиальных базисных функций;

SVM как двухслойная нейронная сеть

Перенумеруем объекты так, чтобы $x_1, \ldots, x_h$ были опорными.

$$a(x) = \text{sign}\left(\sum_{i=1}^{h} \lambda_i y_i K(x, x_i) - w_0\right).$$



Преимущества и недостатки SVM

Преимущества SVM перед SG:

- Задача выпуклого квадратичного программирования имеет единственное решение.
- Число нейронов скрытого слоя определяется автоматически — это число опорных векторов.

Недостатки SVM:

- Неустойчивость к шуму.
- Нет общих подходов к оптимизации $K(x, x')$ под задачу.
- Приходится подбирать константу C.

Машина релевантных векторов RVM

Положим, как и в SVM, при некоторых $\lambda_i \geqslant 0$

$$w = \sum_{i=1}^{\ell} \lambda_i y_i x_i,$$

причём опорным векторам x_i соответствуют $\lambda_i \neq 0$.

Проблема: Какие из коэффициентов λ_i лучше обнулить?

Идея: байесовский регуляризатор зависит не от w, а от λ_i.
Пусть λ_i независимые, гауссовские, с дисперсиями α_i:

$$p(\lambda) = \frac{1}{(2\pi)^{\ell/2}\sqrt{\alpha_1 \cdots \alpha_\ell}} \exp\left(-\sum_{i=1}^{\ell} \frac{\lambda_i^2}{2\alpha_i}\right);$$

$$\widetilde{Q}(w) = \sum_{i=1}^{\ell} \mathscr{L}\big(M_i(w(\lambda))\big) + \frac{1}{2}\sum_{i=1}^{\ell}\left(\ln \alpha_i + \frac{\lambda_i^2}{\alpha_i}\right) \to \min_{\lambda,\alpha}.$$

Преимущества RVM перед SVM

- Опорных векторов, как правило, меньше (более «разреженное» решение).
- Шумовые выбросы уже не входят в число опорных.
- Не надо искать параметр регуляризации (вместо этого α_i оптимизируются в процессе обучения).
- Аналогично SVM, можно использовать ядра.

Балансировка ошибок I и II рода. Постановка задачи

Задача классификации на два класса, $Y = \{-1, +1\}$;
λ_y — штраф за ошибку на объекте класса y;
модель алгоритмов $a(x, w, w_0) = \text{sign}\big(f(x, w) - w_0\big)$,

В логистической регрессии:

- $f(x, w) = \langle x, w \rangle$ — не зависит от $\{\lambda_y\}$;
- $w_0 = \ln \frac{\lambda_{-1}}{\lambda_{+1}}$ — зависит только от $\{\lambda_y\}$.

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

Постановка задачи

- Нужен удобный способ выбора w_0 в зависимости от $\{\lambda_y\}$, не требующий построения w заново.
- Нужна характеристика качества классификатора, инвариантная относительно выбора $\{\lambda_y\}$.

Определение ROC-кривой

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):

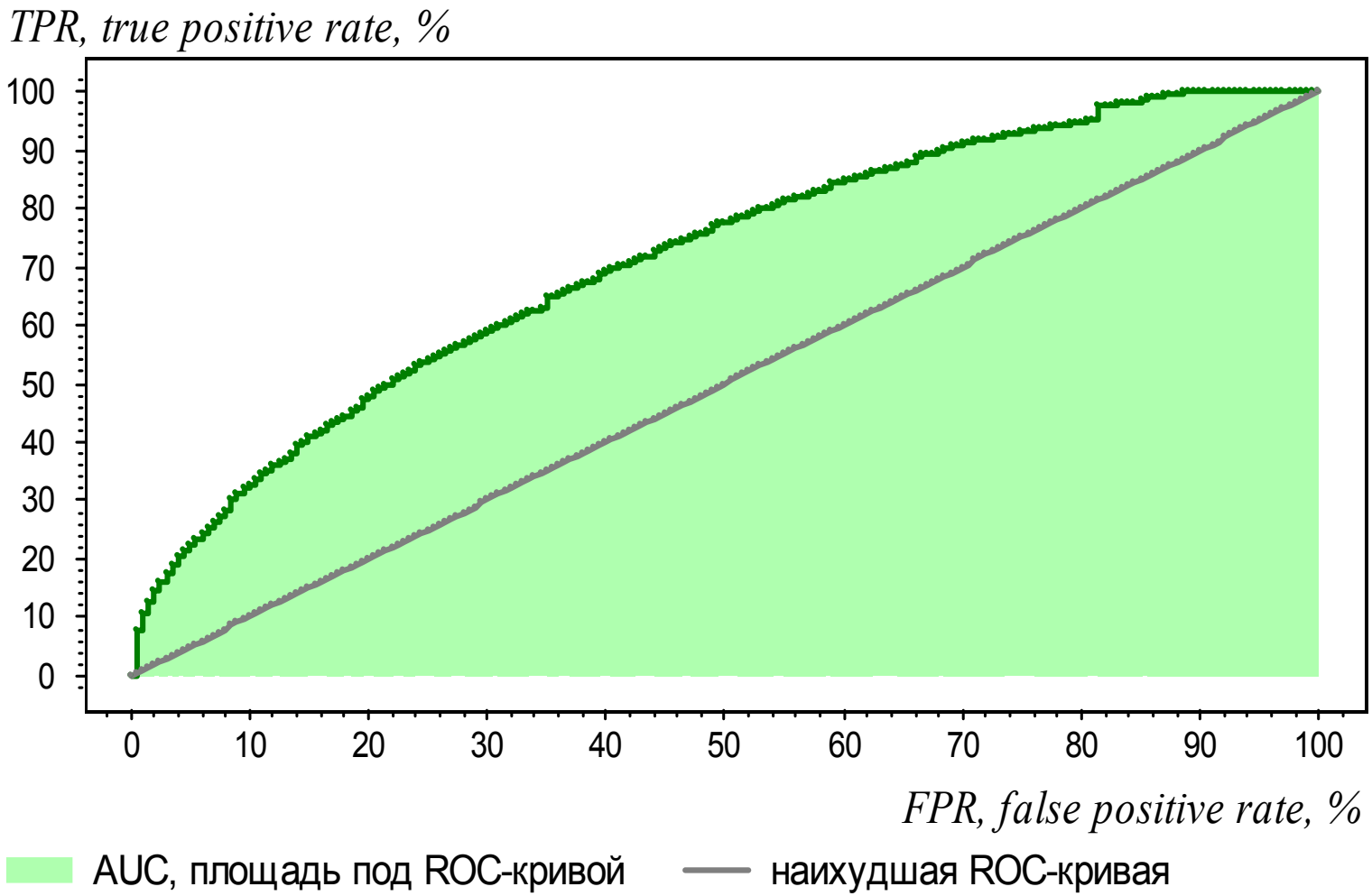
$$\text{FPR}(a, X^\ell) = \frac{\sum_{i=1}^{\ell} [y_i = -1][a(x_i; w, w_0) = +1]}{\sum_{i=1}^{\ell} [y_i = -1]};$$

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

1: $\ell_y := \sum_{i=1}^{\ell} [y_i = y]$, для всех $y \in Y$;
2: упорядочить выборку X^ℓ по убыванию значений $f(x_i, w)$;
3: поставить первую точку в начало координат:
$(\mathrm{FPR}_0, \mathrm{TPR}_0) := (0, 0)$; $\mathrm{AUC} := 0$;
4: **для** $i := 1, \ldots, \ell$
5: **если** $y_i = -1$ **то** сместиться на один шаг вправо:
6: $\mathrm{FPR}_i := \mathrm{FPR}_{i-1} + \frac{1}{\ell_-}$; $\mathrm{TPR}_i := \mathrm{TPR}_{i-1}$;
$\mathrm{AUC} := \mathrm{AUC} + \frac{1}{\ell_-}\mathrm{TPR}_i$;
7: **иначе** сместиться на один шаг вверх:
8: $\mathrm{FPR}_i := \mathrm{FPR}_{i-1}$; $\mathrm{TPR}_i := \mathrm{TPR}_{i-1} + \frac{1}{\ell_+}$;

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.
- На практике ROC-кривую строят по контрольной выборке.
- Существуют методы обучения, минимизирующие AUC.