Министерство образования и науки Российской Федерации

Калужский филиал
федерального государственного бюджетного образовательного
учреждения высшего образования
**«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»**
(КФ МГТУ им. Н.Э. Баумана)

И.И. Кручинин
(к.т.н. доцент)

Лекция: «Линейные классификаторы» по курсу «Введение в машинное обучение»

Калуга – 2018

Краткое содержание:

1. Линейные методы классификации
2. Аппроксимация эмпирического риска
3. Принцип максимума совместного правдоподобия данных и модели
4. Регуляризаторы
5. Независимые параметры с неравными дисперсиями
6. Линейная модель классификации
7. Модель МакКаллока – Питтса
8. Коннективизм и нейронные сети
9. Алгоритм персептрона
10. Метод стохастического градиента
11. Классические частные случаи
12. Эвристики для улучшения градиентных методов обучения
13. Логистическая регрессия
14. Метод стохастического градиента для логистической регрессии
15. Оценка апостериорных вероятностей
16. Вероятностный вывод и оценивание рисков
17. Метод опорных векторов
18. Оптимальная разделяющая гиперплоскость
19. Линейная дискриминантная функция
20. Линейно не разделимая выборка
21. Ядра и спрямляющие пространства
22. Машина опорных векторов как двухслойная нейронная сеть
23. Метод релевантных векторов
24. Оптимизация порога решающего правила

Линейные методы классификации

Линейные модели широко используются в машинном обучении благодаря их относительной простоте, в некоторых случаях хорошей интерпретируемости и наличию глубоко проработанных численных методов. Простейшим обоснованием линейного классификатора служит его аналогия с нервной клеткой — нейроном. Персептронные принципы обучения, первоначально заимствованные из нейрофизиологии, затем нашли математические обоснования и с точки зрения градиентных методов минимизации эмпирического риска, и с точки зрения байесовской теории классификации, и с точки зрения статистических оценок обобщающей способности.

Аппроксимация и регуляризация эмпирического риска

Рассмотрим задачу классификации с двумя классами, $Y = \{-1,+1\}$.

Пусть модель алгоритмов представляет собой параметрическое семейство отображений $a(x,w) = \operatorname{sign} f(x,w)$, где w— вектор параметров. Функция $f(x,w)$ называется дискриминантной функцией. Пока мы не будем предполагать, что она линейна по параметрам. Если $f(x,w) > 0$, то алгоритм aотносит объект xк классу +1, иначе к классу −1. Уравнение $f(x,w) = 0$ описывает разделяющую поверхность.

Как обычно, задача обучения классификатора $a(x,w)$ заключается в том, чтобы настроить вектор параметров w, имея обучающую выборку пар $X^\ell = (x_i,y_i)^\ell_{i=1}$.

Опр. 4.1. Величина $M_i(w) = y_i f(x_i,w)$ называется отступом (margin) объекта x_iотносительно алгоритма классификации $a(x,w) = \operatorname{sign} f(x,w)$.

Если $M_i(w) < 0$, то алгоритм $a(x,w)$ допускает ошибку на объекте x_i. Чем больше отступ $M_i(w)$, тем правильнее и надёжнее классификация объекта x_i.

Минимизация аппроксимированного эмпирического риска. Определим функцию потерь вида $\mathrm{L}\big(M_i(w)\big)$, где $\mathrm{L}(M)$ — монотонно невозрастающая функция отступа, мажорирующая пороговую функцию потерь: $[M<0] \le \mathrm{L}(M)$. Тогда минимизацию

суммарных потерь можно рассматривать как приближённый метод минимизации эмпирического риска — числа ошибок на обучающей выборке:

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

.

Некоторые применяемые на практике функции потерь показаны на рис. Квадратичная функция потерь не является монотонной, тем не менее, она соответствует линейному дискриминанту Фишера. Кусочно-линейная функция потерь соответствует методу опорных векторов (SVM), сигмоидная используется в нейронных сетях, логистическая — в логистической регрессии, экспоненциальная — в алгоритме бустингаAdaBoost, см. ??. Каждый из перечисленных методов имеет свою разумную мотивацию, однако все они приводят к разным функциям потерь. Все эти методы будут рассмотрены далее.

Исходя из эвристического принципа максимизации отступов, невозможно ответить на ряд вопросов: какие ещё функции L(*M*) допустимы, какие из них более предпочтительны и в каких ситуациях.

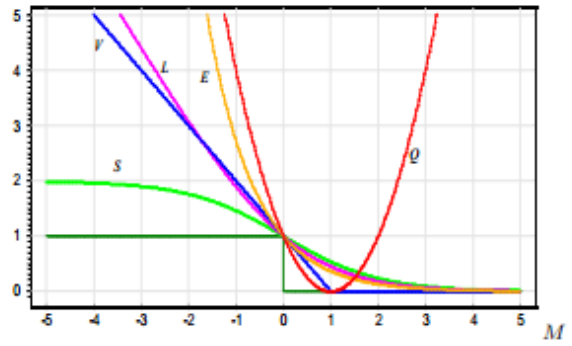


Рис. 7. Непрерывные аппроксимации пороговой функции потерь $[M < 0]$.

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

Вероятностная модель данных позволяет по-другому взглянуть на задачу. Допустим, множество *X*× *Y*является вероятностным пространством, и вместо модели разделяющей поверхности *f*(*x*,*w*) задана параметрическая модель совместной плотности распределения объектов и классов *p*(*x*,*y*|*w*).

Для настройки вектора параметров w по обучающей выборке X^ℓ применим *принцип максимума правдоподобия*. Найдём такое значение вектора параметров w, при котором наблюдаемая выборка X^ℓ максимально правдоподобна (совместная плотность распределения объектов выборки максимальна). Если выборка X^ℓ простая (состоит из независимых наблюдений, взятых из одного и того же распределения $p(x, y|w)$), то правдоподобие представляется в виде произведения:

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

(4.1)

Удобнее рассматривать логарифм правдоподобия:

$$L(w, X^\ell) = \ln p(X^\ell|w) = \sum_{i=1}^{\ell} \ln p(x_i, y_i|w) \to \max_w . \qquad (4.2)$$

Сопоставляя (4.2) с правой частью (4.1), легко заключить, что эти две задачи эквивалентны, если положить

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

Зная модель плотности $p(x,y|w)$, можно выписать вид разделяющей поверхности fи функцию L . С другой стороны, задавая вид разделяющей поверхности и функцию потерь L , казалось бы, из чисто эвристических соображений, мы тем самым неявно принимаем некоторую вероятностную модель данных.

Принцип максимума совместного правдоподобия данных и модели. Допустим, что наряду с параметрической моделью плотности распределения $p(x,y|w)$ имеется ещё априорное распределение в пространстве параметров модели $p(w)$. Никакая модель не может быть идеальной, поэтому вряд ли параметрическое семейство плотностей $p(x,y|w)$ содержит ту самую неизвестную плотность, которая породила выборку. Будем считать, что выборка может быть порождена каждой из плотностей $p(x,y|w)$ с вероятностью $p(w)$.

Чтобы ослабить априорные ограничения, вместо фиксированной функции $p(w)$ вводится параметрическое семейство априорных распределений $p(w;\gamma)$, где γ— неизвестная и не случайная величина, называемая гиперпараметром. Исследователю разрешается варьировать значение гиперпараметра. При этом свойства модели могут изменяться радикальным образом, и появляется возможность найти такое значение гиперпараметра, при котором модель наиболее адекватна.

Принцип максимума правдоподобия теперь будет записываться по-другому, поскольку не только появление выборки X^ℓ, но и появление модели wтакже является случайным. Их совместное появление описывается, согласно формуле условной вероятности, плотностью распределения $p(X^\ell,w;\gamma) = p(X^\ell|w)p(w;\gamma)$. Таким образом, приходим к принципу максимума совместного правдоподобия данных и модели:

$$L_\gamma(w, X^\ell) = \ln p(X^\ell, w; \gamma) = \sum_{i=1}^{\ell} \ln p(x_i, y_i|w) + \ln p(w; \gamma) \to \max_w. \quad (4.3)$$

Функционал L_γ распадается на два слагаемых: логарифм правдоподобия (4.2) и регуляризатор, не зависящий от данных. Второе слагаемое ограничивает вектор параметров модели, не позволяя ему быть каким угодно.

Нормальный регуляризатор. Пусть вектор $w \in R^n$ имеет нормальное распределение, все его компоненты независимы и имеют равные дисперсии σ. Будем считать σ гиперпараметром. Логарифмируя, получаем квадратичный регуляризатор:

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

где const(w) — слагаемое, не зависящее от w, которым можно пренебречь, поскольку оно не влияет на решение оптимизационной задачи (4.3). Минимизация регуляризованного эмпирического риска приводит в данном случае к выбору такого решения w, которое не слишком сильно отклоняется от нуля. В линейных классификаторах это позволяет избежать проблем мультиколлинеарности и переобучения.

Лапласовский регуляризатор. Пусть вектор $w \in R^n$ имеет априорное распределение Лапласа, все его компоненты независимы и имеют равные дисперсии. Тогда

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

Распределение Лапласа имеет более острый пик и более тяжёлые «хвосты», по сравнению с нормальным распределением. Его дисперсия равна $2C^2$.

Самое интересное и полезное для практики свойство этого регуляризатора заключается в том, что он приводит к отбору признаков. Это происходит из-за того, что априорная плотность не является гладкой в точке $w = 0$. Запишем оптимизационную задачу настройки вектора параметров w:

$$Q(w) = \sum_{i=1}^{\ell} \mathscr{L}_i(w) + \frac{1}{C}\sum_{j=1}^{n} |w_j| \to \min_w,$$

где $L_i(w) = L(y_i f(x_i, w))$ — некоторая ограниченная гладкая функция потерь. Сделаем замену переменных, чтобы функционал стал гладким. Каждой переменной w_j поставим в соответствие две новые неотрицательные переменные $u_j = \frac{1}{2}(|w_j| + w_j)$ и $v_j = \frac{1}{2}(|w_j| - w_j)$. Тогда $w_j = u_j - v_j$ и $|w_j| = u_j + v_j$. В новых переменных функционал

становится гладким, но добавляются ограничения-неравенства:

$$Q(u,v) = \sum_{i=1}^{\ell} \mathscr{L}_i(u-v) + \frac{1}{C}\sum_{j=1}^{n}(u_j + v_j) \to \min_{u,v}$$
$$u_j \geqslant 0, \quad v_j \geqslant 0, \quad j = 1,\ldots,n. \qquad ;$$

Для любого jхотя бы одно из ограничений $u_j>0$ и $v_j>0$ является активным, то есть обращается в равенство, иначе второе слагаемое в $Q(u,v)$ можно было бы уменьшить, не изменив первое. Если гиперпараметр C устремить к нулю, то активными в какой-то момент станут все $2n$ограничений. Постепенное уменьшение гиперпараметра C приводит к увеличению числа таких j, для которых оба ограничения активны, $u_j = v_j = 0$, откуда следует, что $w_j = 0$. В линейных моделях это означает, что значения j-го признака игнорируются, и его можно исключить из модели.

Таким образом, для тех моделей, в которых обнуление коэффициента w_jозначает исключение j-го признака, регуляризация Лапласа автоматически приводит к отбору признаков (featuresselection). Параметр Cпозволяет регулировать селективность метода. Чем меньше C, тем большее коэффициентов w_jокажутся нулевыми.

Независимые параметры с неравными дисперсиями. Возьмём гауссовскую модель априорного распределения $p(w)$, $w \in \mathbb{R}^n$с независимыми параметрами w_j, но теперь не будем предполагать, что дисперсии C_jпараметров w_jодинаковы:

$$p(w) = \frac{1}{(2\pi)^{n/2}\sqrt{C_1 \cdots C_n}} \exp\left(-\sum_{j=1}^{n} \frac{w_j^2}{2C_j}\right). \qquad (4.4)$$

Будем считать дисперсии C_jнеизвестными и определять их наравне с самими параметрами w_jисходя из принципа максимума совместного правдоподобия:

$$Q(w) = \sum_{i=1}^{\ell} \mathscr{L}_i(w) + \frac{1}{2}\sum_{j=1}^{n}\left(\ln C_j + \frac{w_j^2}{C_j}\right) \to \min_{w,C}.$$

Результатом такой модификации также является отбор признаков. Если $C_j \to 0$, то параметр w_j стремится к нулю, и на практике часто достигает машинного нуля. Если $C_j \to \infty$, то на параметр w_j фактически не накладывается никаких ограничений, и он может принимать любые значения.

Линейная модель классификации

Пусть X— пространство объектов; $Y = \{-1,1\}$ — множество допустимых ответов; объекты описываются nчисловыми признаками $f_j: X \to \mathbb{R}$, $j = 1,\ldots,n$. Вектор $x = (x^1,\ldots,x^n) \in \mathbb{R}^n$, где $x^j = f_j(x)$, называется признаковым описанием объекта x.

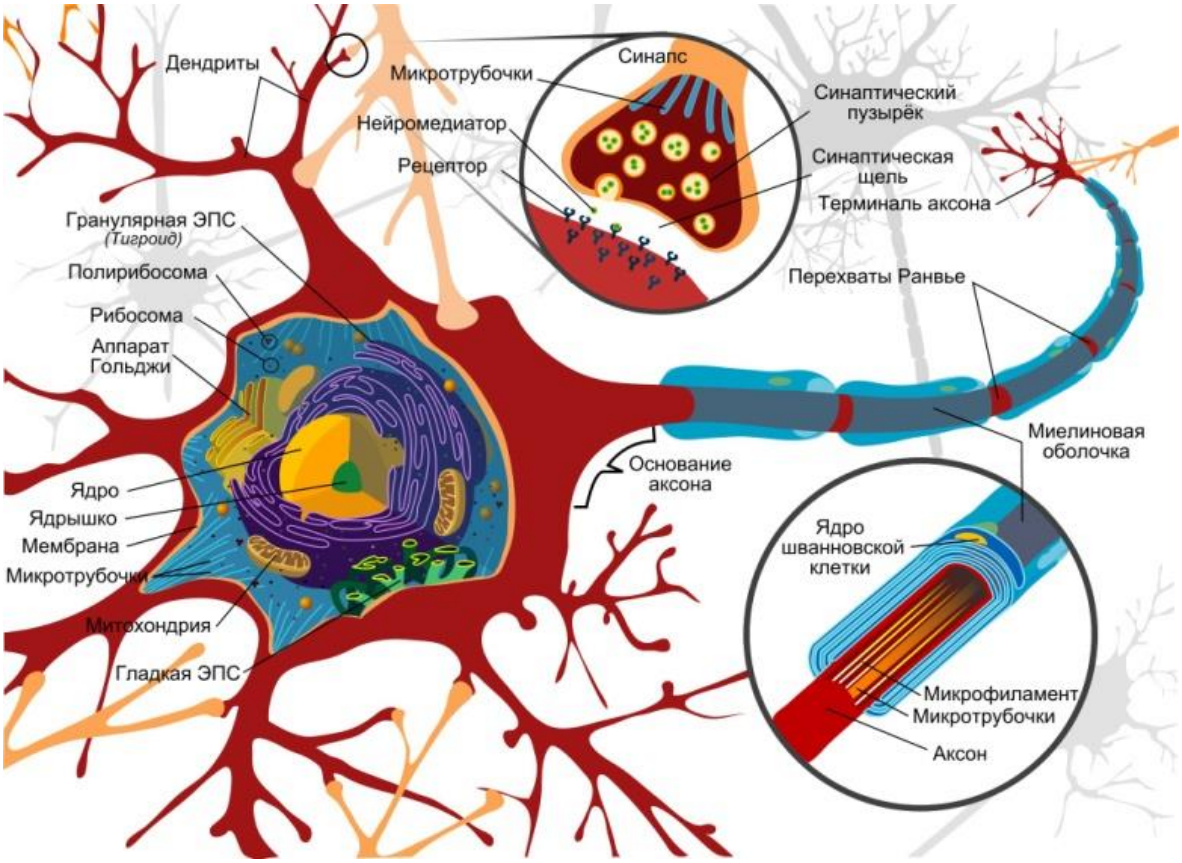


Рис. . Нервная клетка .

Если дискриминантная функция определяется как скалярное произведение вектора *x*и вектора параметров $w \in \mathbb{R}^n$, то получается линейный классификатор:

$$a(x,w) = \operatorname{sign}\big(\langle w,x\rangle - w_0\big) = \operatorname{sign}\left(\sum_{j=1}^{n} w_j f_j(x) - w_0\right). \qquad (4.5)$$

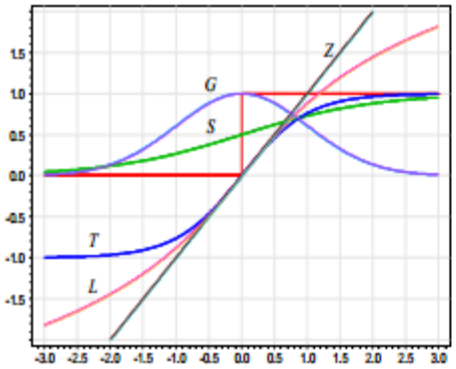
Уравнение h*w*,*x*i = 0 задаёт гиперплоскость, разделяющую классы в пространстве $\mathbb{R}^n$. Если вектор *x*находится по одну сторону гиперплоскости с её направляющим вектором *w*, то объект *x*относится к классу +1, иначе — к классу −1.

Параметр w_0 иногда опускают. Иногда полагают, что среди признаков есть константа, $f_j(x) \equiv -1$, и тогда роль свободного коэффициента w_0 играет параметр w_j.

Устройство нервной клетки и модель МакКаллока-Питтса. Линейный классификатор или персептрон является простейшей математической моделью нервной клетки — нейрона, Рис . Нейрон имеет множество разветвлённых отростков — дендритов, и одно длинное тонкое волокно — аксон, на конце которого находятся синапсы, примыкающие к дендритам других нервных клеток. Нервная клетка может находиться в двух состояниях: обычном и возбуждённом. Клетка возбуждается, когда в ней

накапливается достаточное количество положительных зарядов. В возбуждённом состоянии клетка генерирует электрический импульс величиной около 100 мВ и длительностью около 1 мс, который проходит по аксону до синапсов. Синапс при приходе импульса выделяет вещество, способствующее проникновению положительных зарядов внутрь соседней клетки, примыкающей к данному синапсу. Синапсы имеют разную способность концентрировать это вещество, причём некоторые даже препятствуют его выделению — они называются тормозящими. После возбуждения клетки наступает период релаксации — некоторое время она не способна генерировать новые импульсы.

Нервную клетку можно рассматривать как устройство, которое на каждом такте своей работы принимает заряды величиной $x^j = f_j(x)$ от n входов — синапсов, примыкающих к её дендритам. Поступающие заряды складываются с весами w_j. Если вес w_j положительный, то j-й синапс возбуждающий, если отрицательный,



$\theta(z) = [z \geqslant 0]$	пороговая функция Хевисайда;
$\sigma(z) = (1 + e^{-z})^{-1}$	сигмоидная функция (S);
$\mathrm{th}(z) = 2\sigma(2z) - 1$	гиперболический тангенс (T);
$\ln(z + \sqrt{z^2 + 1})$	логарифмическая функция (L);
$\exp(-z^2/2)$	гауссовская функция (G);
z	линейная функция (Z);

Рис. 9. Стандартные функции активации $\varphi(z)$.

то тормозящий. Если суммарный заряд превышает *порог активации* w_0, то нейрон возбуждается и выдаёт на выходе $+1$, иначе выдаётся -1.

Функцию $\varphi(z) = \mathrm{sign}(z)$, преобразующую значение суммарного импульса в выходное значение нейрона, называют *функцией активации*. В общем случае это не обязательно пороговая функция. Используют также «сглаженную» пороговую функцию — гиперболический тангенс $\varphi(z) = \mathrm{th}(z)$ и другие, см. Рис. 9.

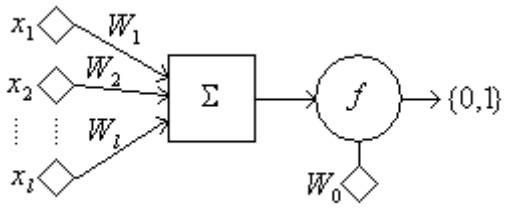
Таким образом, линейный классификатор (4.5) является математической моделью нейрона. Эту модель предложили в 1943 году МакКаллок и Питтс.

Коннективизм и нейронные сети. Нервная система состоит из огромного числа связанных друг с другом нейронов, распространяющих направленные волны импульсов. Скорость распространения импульсов составляет приблизительно 100 м/с. Человек способен решать сложные задачи распознавания и принятия решений за десятые доли секунды, откуда следует, что необходимые для этого нейровычисления выполняются не более чем за 10^2 последовательных тактов. Кора головного мозга человека содержит порядка 10^{11} нейронов, и каждый нейрон имеет синаптические связи с 10^3–10^4 других нейронов. Нейровычисления выполняются с большой степенью параллелизма, и для принятия одного решения может быть задействовано огромное число нейронов.

Есть гипотеза, что, соединив большое число элементарных классификаторов (скажем, линейных, но обязательно через нелинейные функции активации), возможно создать универсальную машину, способную обучаться решению любых задач, подобно тому, как это делает человеческий мозг. На основе этой идеи, высказанной ещё в 50-е годы и названной принципом коннективизма, строятся искусственные нейронные сети.

На самом деле механизмы функционирования нервных клеток гораздо сложнее описанных выше. В нейрофизиологии известны десятки различных типов нейронов, и многие из них функционируют иначе. Однако в теории искусственных нейронных сетей не ставится задача максимально точного воспроизведения функций биологических нейронов. Цель в том, чтобы подсмотреть некоторые принципы в живой природе и использовать их для построения обучаемых устройств.

Алгоритм персептрона



Математическая модель нейрона. В алгоритме персептрона в основу положен принцип действия нейрона. Обобщенная схема нейрона представлена на рисунке. Здесь $x_1, x_2, \ldots, x_l$ – компоненты вектора признаков $x = (x_1, x_2, \ldots, x_l)$; Σ – *сумматор*; $W_1, W_2, \ldots, W_l$ – синоптические веса; f – *функция* активации; W_0 – порог. Выходом сумматора является величина $\sum_{i=1}^{l} W_i x_i$, которая является входом (аргументом) функции активации. *Значение функции* активацииивычисляетсянаосновеопределениязнакасуммы $\sum_{i=1}^{l} W_i x_i + W_0$:

$$f(v) = \begin{cases} 0 \text{ при } v < 0 \\ 1 \text{ при } v > 0 \end{cases}.$$

Таким образом, *нейрон* представляет собой линейный *классификатор* с дискриминантной функцией $g(x) = \sum_{i=1}^{l} W_i x_i + W_0$.

Тогда задача построения линейного классификатора для заданного *множества* прецедентов сводится к задаче обучения нейрона, т.е. подбора

соответствующих весов $W_1, W_2, \ldots, W_l$ и порога W_0. Обучение состоит в коррекции синоптических весов и порога.

Алгоритм персептрона. *Алгоритм* персептрона представляет собой последовательную итерационную процедуру. Каждый шаг состоит в предъявлении нейрону очередного вектора-прецедента и коррекции весов W_i по результатам классификации. При этом прецеденты предъявляются циклически, т.е. после предъявления последнего снова предъявляется первый. Процесс обучения заканчивается, когда *нейрон* правильно классифицирует все прецеденты.

Обозначим W_t весовой *вектор* после t -й итерации, а x_t – *прецедент*, предъявляемый на t -й итерации.

Основной шаг алгоритма состоит в предъявлении очередного прецедента x_{t+1}:

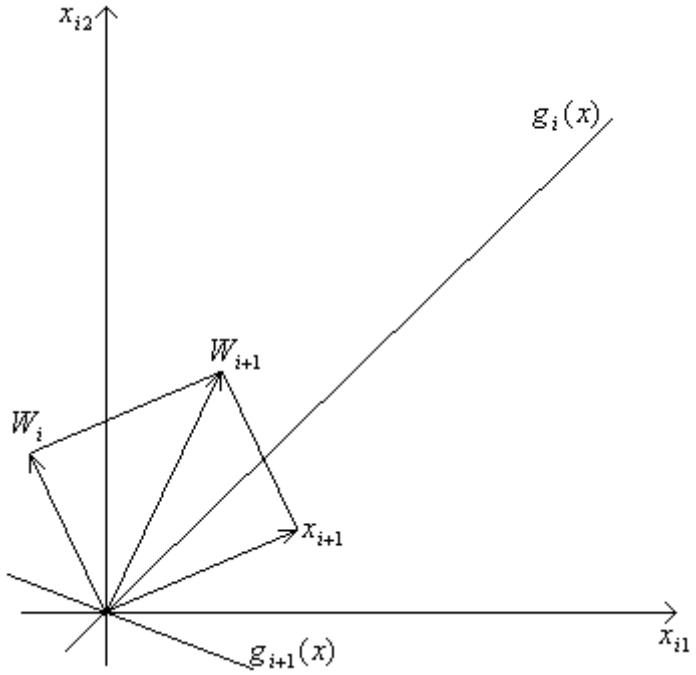
Если $x_{t+1} \in \Omega_1$ и $W_t x_{t+1} > 0$, то $W_{t+1} = W_t$;

Если $x_{t+1} \in \Omega_1$ и $W_t x_{t+1} \leq; 0$, то $W_{t+1} = W_t + x_{t+1}$;

Если $x_{t+1} \in \Omega_2$ и $W_t x_{t+1} < 0$, то $W_{t+1} = W_t$;

Если $x_{t+1} \in \Omega_2$ и $W_t x_{t+1} \geq; 0$, то $W_{t+1} = W_t + x_{t+1}$.

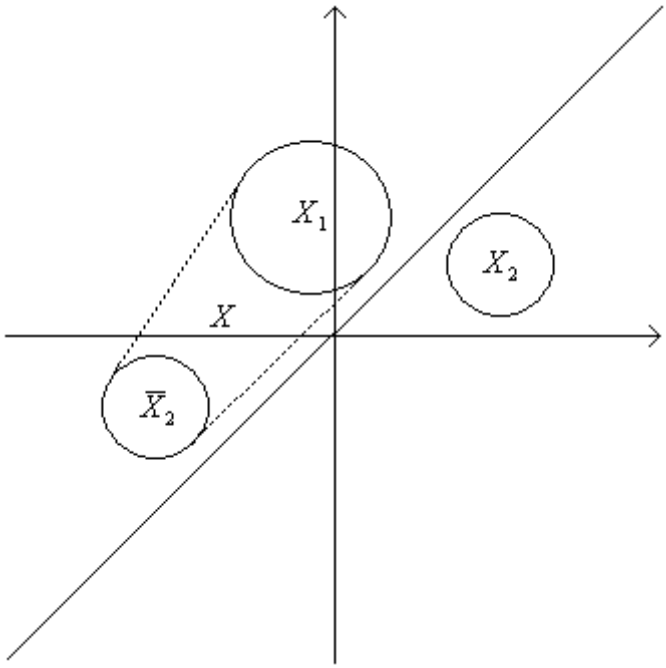
На данном рисунке $g_t(x)$ – дискриминантная *функция* после t -го шага алгоритма; W_t – весовой *вектор* после t -го шага алгоритма.



Сходимость алгоритма персептрона.

Основной вопрос, связанный с алгоритмом персептрона связан с его сходимостью. Конечен ли построенный итерационный процесс обучения?

Теорема Новикова. *Пусть* $\{x_i\}$ *– бесконечная последовательность векторов из двух непересекающихся замкнутых множеств* X_1 и X_2 ; *и пусть существует гиперплоскость, проходящая через начало координат и разделяющая* X_1 и X_2 *(не имеет с ними общих точек). Тогда при использовании алгоритма персептрона число коррекций весового вектора конечно*.



Доказательство. Пусть W^* - *направляющий вектор* разделяющей гиперплоскости (которая существует по условию). Не нарушая общности, будем считать, что он является единичным.

Пусть $X = conv\left(X_1 \bigcup \overline{X}_2\right)$, $\overline{X}_2$ в – симметричное к X_2 множество; $\rho = \rho(0, X)$, где ρ – евклидово *расстояние*. Согласноутверждению3.3 $(W^*, X) \geq \rho_0 > 0 \quad \forall x \in X$.

Оценим (W_t, W^*).

Пусть W^* – *единичный вектор* нормали, разделяющий X_1 и X_2.

$$(W^*, X) \geq \rho_0 \text{ при } x \in X_1$$
$$(W^*, X) \leq -\rho_0 \text{ при } x \in X_2$$

Пусть W_t – весовой *вектор* после предъявления вектора x_t ; $W_0 = 0$ – начальная *итерация* весового вектора $(|W^*| - 1)$. Тогда, если $(W_t, x_{t+1}) > 0$, то коррекции не происходит. Иначе, если $(W_t, x_{t+1}) \leq 0$, то *коррекция*: $W_{t+1} = W_t + x_{t+1}$

$$|W_{t+1}|^2 = |W_t|^2 + 2(x_{t+1}, W_t) + |x_{t+1}^2 \leq |W_T|^2 + D^2,$$
т.к. $(x_{t+1}, W_t) \leq 0$ и $|x_{t+1}| \leq \sup_{x \in X} |x| = D$

Таким образом, к моменту t происходит k коррекций, то

$$|W_t|^2 \leq k \cdot D^2, \text{ т.к. } |W_0| = 0 \quad (3.1)$$

В начальный момент времени $(W_0, W^*) = 0$. Если в момент $i+1$ произошла *коррекция*, то

$$(W_{t+1}, W^*) = (W_0, W^*) + (x_{t+1}, W^*) \geq (W_t, W^*) + \rho_0$$
Если *коррекция* непроисходит, то
$(W_{t+1}, W^*) = (W_t, W^*)$
Если к моменту t произошло k коррекций, то
$(W_t, W^*) \geq k\rho_0$
С другойстороны
$(W_t, W^*) \leq |W_t| \cdot |W^*| = |W_t|$
Поэтому

$$|W_t| \geq k\rho_0 \quad (3.2)$$

Изнеравенств 3.1 и 3.2 следует:

$$k^2\rho_0 \leq |W_t|^2 \leq kD^2 \Rightarrow k\rho_0 \leq D^2 \Rightarrow k \leq \frac{D^2}{\rho_0}$$

Таким образом, число коррекций k не превосходит $\lfloor \frac{D^2}{\rho_0} \rfloor$.

Оптимизационная интерпретация. Рассмотрим непрерывную кусочно-линейную функцию $J(W)$:

$$J(W) = \sum_{x \in Y} \delta_x(W, x), \text{ где } \delta_x = \begin{cases} -1, x \in X_1 \\ 1, x \in X_2 \end{cases};$$

Y – множество векторов неправильно классифицированных гиперплоскостью W. Тогда $J(W) \geq 0$ и $J(W) = 0 \Leftrightarrow Y = \varnothing$. Задачасостоит в минимизацииэтойфункции:

$$J(W) = \sum_{x \in Y} \delta_x(W, x) \to \min$$

Построим минимизацию по схеме *градиентного спуска*:

$$W_{t+1} = W_t - \rho_t \frac{dJ(W)}{dW}$$

Т.к. $\frac{dJ(W)}{dW} = \sum_{x \in Y} \delta_x x$, то $W_t - \rho_t \sum_{x \in Y} \delta_x x$

Таким образом, *алгоритм* персептрона представляет собой вариант алгоритма *градиентного спуска*. Выборпоследовательностивеличин ρ_t дляобычноосуществляетсятак, чтобы:

$$\sum_{t=0}^{\infty} |\rho_t| > \infty \quad \text{и} \quad \sum_{t=0}^{\infty} \rho_t^2 < \infty$$

Схема Кеслера. Идея построения линейного классификатора естественно обобщается на случай классификации с числом классов больше двух. Рассмотрим задачу классификации по M классам. Для каждого класса необходимо определить линейную дискриминантную функцию $W_i,\ i = 1, 2, \ldots, M$. Пусть – $x - (l+1)$-мерный *вектор* в расширенном пространстве. *Вектор* x относится к классу $|Omega_i$, если

$$W_i x > W_j x,\ \forall i \neq j$$

Схема Кеслера позволяет применить *алгоритм* персептрона для решения этой задачи.

Для каждого вектора-прецедента из Ω_i строим $(M-1)$ векторов x_{ij} размерности $(l+1)M$:

$$x_{ij} = (\underbrace{0, \ldots, 0}_{1}, \underbrace{0, \ldots, 0}_{2}, \ldots, \underbrace{x_1, \ldots, x_M}_{i}, \underbrace{0, \ldots, 0}_{i+1}, \ldots, \underbrace{-x_1, \ldots, -x_M}_{j}, \ldots$$

и *вектор* $W = (W_1, W_2, \ldots, W_M)^T$, где W_i – весовой *вектор* i -ой дискриминантной функции.

Пусть $x = (x_1, x_2, \ldots, x_M)$, тогда *вектор* x_{ij} можно записать в виде:

$$x_{ij} = (\underset{1}{0}, \underset{2}{0}, \ldots, \underset{i-1}{0}, \underset{i}{x}, \underset{i+1}{0}, \ldots, \underset{j-1}{0}, \underset{j}{-x}, \underset{j+1}{0}, \ldots, \underset{M}{0})$$

Если x относится к классу Ω_1, то $Wx_{ij} > 0 \quad \forall j = 1, 2, \ldots, M,\ i \neq j$, т.к. $W_i x > W_j x$ и $Wx_{ij} = W_i x - W_J x > 0$.

Таким образом, задача заключается в построении линейного классификатора в $(l+1)M$ -мерном пространстве так, чтобы каждый из $(M-1)N$ векторов-прецедентов лежал в положительном полупространстве. Если вектора в исходной задаче разделимы, то это можно сделать с помощью алгоритма персептрона.

Метод стохастического градиента

Пусть задана обучающая выборка $X^\ell = \{(x_i, y_i)\}_{i=1}^{\ell},\ x_i \in \mathbb{R}^n,\ y_i \in \{-1, 1\}$. Требуется найти вектор весов $w \in \mathrm{R}^n$, при котором достигается минимум аппроксимированного эмпирического риска:

$$Q(w, X^\ell) = \sum_{i=1}^{\ell} \mathscr{L}\big(\langle w, x_i\rangle\, y_i\big) \to \min_w. \tag{4.6}$$

Применим для минимизации $Q(w)$ метод градиентного спуска. В этом методе выбирается некоторое начальное приближение для вектора весов w, затем запускается итерационный процесс, на каждом шаге которого вектор w изменяется в направлении наиболее быстрого убывания функционала Q. Это направление противоположно вектору градиента $Q'(w) = \left(\frac{\partial Q(w)}{\partial w_j}\right)_{j=1}^{n}$:

$w := w - \eta Q'(w)$,

где $\eta > 0$ — величина шага в направлении антиградиента, называемая также темпом обучения (learningrate). Предполагая, что функция потерь L дифференцируема, распишем градиент:

$$w := w - \eta \sum_{i=1}^{\ell} \mathscr{L}'\big(\langle w, x_i\rangle\, y_i\big) x_i y_i. \tag{4.7}$$

Каждый прецедент (x_i, y_i) вносит аддитивный вклад в изменение вектора w, но вектор w изменяется только после перебора всех ℓ объектов. Сходимость итерационного процесса можно улучшить, если выбирать прецеденты (x_i, y_i) по одному, для каждого делать градиентный шаг и сразу обновлять вектор весов:

$$w := w - \eta \mathscr{L}'_a\big(\langle w, x_i\rangle\, y_i\big) x_i y_i. \tag{4.8}$$

В методе стохастического градиента (stochasticgradient, SG) прецеденты перебираются в случайном порядке, см.Алгоритм4.1. Если же объекты предъявлять в некотором фиксированном порядке, процесс может зациклиться или разойтись.

Инициализация весов может производиться различными способами. Стандартная рекомендация — взять небольшие случайные значения, $-\frac{1}{2n}, \frac{1}{2n}\big)$ w_j:= random. Иногда веса инициализируют нулём. Иногда берут оценки

$$w_j := \frac{\langle y, f_j \rangle}{\langle f_j, f_j \rangle}, \quad j = 1, \ldots, n, \tag{4.9}$$

где $f_j = \big(f_j(x_i)\big)_{i=1}^{\ell}$ — вектор значений j-го признака, $y = (y_i)_{i=1}^{\ell}$ — вектор ответов. Эти оценки являются точными в одном нереалистичном частном случае — когда функция потерь квадратична и признаки статистически независимы.

Критерий останова в Алгоритме 4.1 основан на приблизительной оценке функционала Qметодом экспоненциальной скользящей средней. Вычисление точного значения по всем ℓ объектам слишком вычислительно трудоёмко. Когда градиентный метод подходит к окрестности минимума, оценка скользящего среднего стабилизируется и приближается к точному значению функционала. Параметр λможно положить равным $1/\ell$. В случае избыточно длинной выборки его рекомендуется увеличивать.

Алгоритм 4.1.Метод стохастического градиента.

Вход:
X^ℓ — обучающая выборка; η— темп обучения; λ— параметр сглаживания.
Выход:
Синаптические веса $w_1,\ldots,w_n$;

1: инициализировать веса w_j, j= 1,...,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_iиз X^ℓ (например, случайным образом);
5: вычислить выходное значение алгоритма $a(x_i,w)$ и ошибку: $\varepsilon_i := \mathscr{L}\big(\langle w, x_i \rangle\, y_i\big)$;
6: сделать шаг градиентного спуска: $w := w - \eta \mathscr{L}'\big(\langle w, x_i \rangle\, y_i\big) x_i y_i$;
7: оценить значение функционала: Q:= $(1-\lambda)Q + \lambda\varepsilon_i$;
8: пока значение Qне стабилизируется и/или веса wне перестанут изменяться;

Преимущества метода SG.

- Метод легко реализуется и легко обобщается на нелинейные классификаторы и на нейронные сети — суперпозиции линейных классификаторов.
- Метод подходит для динамического обучения, когда обучающие объекты поступают потоком, и вектор весов обновляется при появлении каждого объекта.
- Метод позволяет настраивать веса на избыточно больших выборках, за счёт того, что случайной подвыборки может оказаться достаточно для обучения.

Недостатким етода SG.

- Функционал Q, как правило, многоэкстремальный, и процесс может сходиться к локальному минимуму, сходиться очень медленно или не сходиться вовсе.
- При большой размерности пространства nили малой длине выборки ℓ возможно переобучение. При этом резко возрастает норма вектора весов, появляются большие по абсолютной величине положительные и отрицательные веса, классификация становится неустойчивой — малые изменения обучающей выборки, начального приближения, порядка предъявления объектов или параметров алгоритма η, λмогут сильно изменить результирующий вектор весов, увеличивается вероятность ошибочной классификации новых объектов.
- Если функция потерь имеет горизонтальные асимптоты, то процесс может попасть в состояние «паралича». Чем больше значение скалярного произведения hw,x_ii, тем ближе значение производной L′ к нулю, тем меньше приращение весов в (4.8). Если веса w_jпопали в область больших значений, то у них практически не остаётся шансов выбраться из этой «мёртвой зоны».

Классические частные случаи

Адаптивный линейный элемент. Рассмотрим случай, когда функция потерь квадратична, L(M) = (M− 1)^2. Тогда правило обновления весов на каждой итерации метода стохастического градиента примет вид

$$w := w - \eta\big(\langle w, x_i\rangle - y_i\big)x_i. \tag{4.10}$$

Это правило предложено Видроу и Хоффом в 1960 году и называется дельтаправилом (delta-rule), а сам линейный нейрон — адаптивным линейным элементом или ADALINE [66]. Это правило подходит также и для решения задач линейной регрессии, когда Y= R, $a(x)$ = hw,$x$ii функция потерь имеет вид $\big(\langle w, x_i\rangle - y_i\big)^2$.

Персептрон Розенблатта. В 1957 году Розенблатт предложил эвристический алгоритм обучения нейрона, основанный на принципах нейрофизиологии. Экспериментально было установлено, что при синхронном возбуждении двух связанных нервных клеток синаптическая связь между ними усиливается. Чем чаще синапс угадывает правильный

ответ, тем сильнее становится связь. Своеобразная тренировка связи приводит к постепенному запоминанию информации. Если же синапс начинает часто ошибаться или вообще перестаёт использоваться, связь ослабевает, информация начинается забываться. Таким образом, память реализуется в синапсах. В математической модели нейрона роль памяти играет вектор синаптических весов w.

Данное правило обучения нетрудно формализовать. Признаки будем пока полагать бинарными, $f_j(x) \in \{0,1\}$. Ответы также принимают только два значения, $y_i \in \{-1,1\}$. Допустим, что сразу после получения классификации объекта $a(x_i)$ становится известен правильный ответ y_i. Возможны три случая.

1. Если ответ $a(x_i)$ совпадает с y_i, то вектор весов изменять не нужно.
2. Если $a(x_i) = -1$ и $y_i = 1$, то вектор весов wувеличивается. Увеличивать имеет смысл только те веса w_j, для которых $f_j(x_i) = 06$; изменение других компонент не повлияет на результат. Положим $w := w + \eta x_i$, где$\eta > 0$ — темп обучения.
3. Если $a(x_i) = 1$ и $y_i = -1$, то вектор весов уменьшается: $w := w - \eta x_i$. Эти три случая объединяются в так называемое правило Хэбба :

   Если $\langle w, x_i\rangle y_i < 0$ то$w := w + \eta x_i y_i$. (4.11)

Легко проверить, что оно в точности соответствует градиентному шагу (4.8), если взять кусочно-линейную функцию потерь $\mathrm{L}(M) = (-M)_+$. Однако формула (4.8) верна для произвольных признаков, не обязательно бинарных.

Для правила Хэбба доказана теорема сходимости, которая также справедлива для произвольных действительных признаков.

Теорема 4.1 (Новиков, 1962 [55]). *Пусть $X = \mathbb{R}^n$, $Y = \{-1, 1\}$, и выборка X^ℓ линейно разделима — существует вектор $\tilde{w}$ и положительное число δ такие, что $\langle \tilde{w}, x_i\rangle y_i > \delta$ для всех $i = 1, \ldots, \ell$. Тогда Алгоритм 4.1 с правилом Хэбба (4.11) за конечное число исправлений находит вектор весов, разделяющий обучающую выборку без ошибок, причём из любого начального приближения w^0, при любом $\eta > 0$, независимо от порядка предъявления объектов. Если $w^0 = 0$, то достаточное число исправлений вектора весов не превосходит*

$$t_{\max} = \left(\frac{D}{\delta}\right)^2, \quad \text{где } D = \max_{x \in X^\ell} \|x\|.$$

Доказательство. Запишем выражение для косинуса угла между вектором $\tilde{w}$ и вектором весов после t-го исправления w^t, полагая без ограничения общности $\|\tilde{w}\| = 1$:

$$\cos(\widehat{\tilde{w}, w^t}) = \frac{\langle \tilde{w}, w^t \rangle}{\|w^t\|}.$$

При t-м исправлении нейрону с вектором весов w^{t-1} предъявляется обучающий объект x, правильный ответ y, и при этом нейрон совершает ошибку: $\langle x, w^{t-1} \rangle y < 0$. Согласно правилу Хэбба (4.11) в этом случае происходит модификация весов. В силу условия линейной разделимости, справедлива оценка снизу:

$$\langle \tilde{w}, w^t \rangle = \langle \tilde{w}, w^{t-1} \rangle + \eta \langle \tilde{w}, x \rangle y > \langle \tilde{w}, w^{t-1} \rangle + \eta\delta > \langle \tilde{w}, w^0 \rangle + t\eta\delta.$$

В силу ограниченности выборки, $\|x\| < D$, справедлива оценка сверху:

$$\|w^t\|^2 = \|w^{t-1}\|^2 + \eta^2\|x\|^2 + 2\eta \langle x, w^{t-1} \rangle y < \|w^{t-1}\|^2 + \eta^2 D^2 < \|w^0\|^2 + t\eta^2 D^2.$$

Подставим полученные соотношения в выражение для косинуса:

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

Косинус не может превышать единицы. Следовательно, при некотором достаточно большом t не найдётся ни одного $x \in X^\ell$ такого, что $\langle x, w^t \rangle y < 0$, то есть выборка окажется поделенной безошибочно.

Если $w^0 = 0$, то нетрудно оценить сверху достаточное число исправлений. Из условия $\cos \leqslant 1$ следует $\sqrt{t}\delta/D \leqslant 1$, откуда $t_{\max} = (D/\delta)^2$. ■

На практике линейная разделимость выборки является скорее исключением, чем правилом. Если условия теоремы Новикова не выполнены, то процесс обучения может оказаться расходящимся.

Эвристики для улучшения градиентных методов обучения

В этом разделе рассматриваются эвристические приёмы и рекомендации, компенсирующие недостатки градиентных методов обучения. Все они в полной мере относятся к обучению нейронных сетей, включая широко известный метод обратного распространения ошибок. Различных тонкостей настолько много, что применение градиентного обучения по праву считается искусством.

Нормализация данных. Градиентный метод чувствителен к масштабу измерения признаков. Если норма вектора объекта kx_ikпринимает большие значения, а функция потерь имеет горизонтальные асимптоты, то итерационный процесс может оказаться «парализованным». Поэтому общей практикой является предварительная нормализация признаков:

$$x^j := \frac{x^j - x^j_{\min}}{x^j_{\max} - x^j_{\min}}, \quad \text{либо} \quad x^j := \frac{x^j - x^j_{\text{ср}}}{x^j_{\text{ско}}}, \quad j = 1, \ldots, n,$$

где $x^j_{\min}$, $x^j_{\max}$, $x^j_{\text{ср}}$, $x^j_{\text{ско}}$— соответственно минимальное, максимальное, среднее значения и среднеквадратичное отклонение j-го признака.

Порядок предъявления объектов. Кроме стандартной рекомендации брать объекты в случайном порядке, имеются ещё следующие соображения.

1. Наибольшее смещение весов ожидается для того объекта, который наименее похож на объекты, предъявленные до него. В общем случае довольно трудно найти объект, максимально информативный на данном шаге обучения. Простая эвристика заключается в том, чтобы попеременно предъявлять объекты из разных классов, поскольку объекты одного класса с большей вероятностью содержат схожую информацию. Эта техника называется перетасовкой объектов (shuffling).

2. Ещё одна эвристика состоит в том, чтобы чаще предъявлять те объекты, на которых была допущена ошибка. Для этого вероятность появления каждого объекта устанавливается пропорционально величине ошибки на данном объекте. Эту эвристику рекомендуется применять только в тех случаях, когда исходные данные не содержат выбросов, иначе процесс обучения может сосредоточиться на шумовых объектах, которые вообще следовало бы исключить из обучающей выборки.

3. Простая для реализации эвристика заключается в том, чтобы сравнить величину ошибки на предъявленном объекте с некоторым порогом. Если ошибка окажется меньше порога, вектор весов не модифицируется. Логика та же, что у персептрона Розенблатта: если объект неплохо классифицируется, то менять веса не нужно. При этом увеличивается и скорость настройки.

Квадратичная регуляризация в теории нейронных сетей называется также сокращением весов (weightsdecay). Чтобы ограничить рост абсолютных значений весов, к минимизируемому функционалу $Q(w)$ добавляется штрафное слагаемое:

$$Q_\tau(w) = Q(w) + \frac{\tau}{2}\|w\|^2.$$

Это приводит к появлению аддитивной поправки в градиенте: $Q'_\tau(w) = Q'(w) + \tau w$.
В результате правило обновления весов принимает вид

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

Таким образом, вся модификация сводится к появлению неотрицательного множителя $(1 - \eta\tau)$, приводящего к постоянному уменьшению весов. Регуляризация предотвращает паралич, повышает устойчивость весов в случае мультиколлинеарности, способствует повышению обобщающей способности алгоритма и снижению риска переобучения. Управляющий параметр τпозволяет найти компромисс между точностью настройки на конкретную выборку и устойчивостью весов.

Недостаток метода в том, что параметр τприходится подбирать в режиме скользящего контроля, что связано с большими вычислительными затратами.

Выбор величины шага.

1. Известно, что градиентные методы сходятся к локальному минимуму, если величину шага η уменьшать с числом итераций t. Точнее, сходимость гарантируется при $\eta_t \to 0$, $\sum_{t=1}^{\infty} \eta_t = \infty$, $\sum_{t=1}^{\infty} \eta_t^2 < \infty$, в частности можно положить $\eta_t = 1/t$.

2. В методе скорейшего градиентного спуска выбирается адаптивный шаг η, который является решением одномерной задачи $Q\big(w - \eta Q'(w)\big) \to \min_{\eta}$. Во многих случаях эту задачу удаётся решить аналитически [8]. В частности, для алгоритма ADALINE с квадратичной функцией потерь $\eta = \|x_i\|^{-2}$.

Выбивание из локальных минимумов необходимо для предотвращения сходимости к недостаточно хорошим локальным решениям. Один из простейших способов заключается в том, чтобы при каждой стабилизации функционала производить случайные модификации вектора весов в довольно большой окрестности текущего значения и запускать процесс градиентного спуска из новых точек. Этот приём называют встряхиванием коэффициентов (jogofweights). По сути дела, он является симбиозом градиентного метода и случайного локального поиска (stochasticlocalsearch).

Ранний останов. Чрезмерная оптимизация может вести к переобучению. Узкий глобальный минимум функционала $Q(w, X^\ell)$ хуже более широкого, но устойчивого локального минимума. Для предотвращения попадания в такие «расщелины» применяется ранний останова (earlystopping): в ходе итераций вычисляется какой-нибудь внешний критерий (стр. ??), например, средняя потеря на независимой контрольной выборке, и если он начинает возрастать, процесс настройки прекращается.

Логистическая регрессия

Метод логистической регрессии основан на довольно сильных вероятностных предположениях, которые имеют сразу несколько интересных последствий. Вопервых, линейный алгоритм классификации оказывается оптимальным байесовским классификатором. Во-вторых, однозначно определяется функция потерь. В-третьих, возникает интересная дополнительная возможность наряду с классификацией объекта получать численные оценки вероятности его принадлежности каждому из классов.

Обоснование логистической регрессии

В нормальном дискриминантном анализе доказывается, что если плотности классов нормальны и имеют равные матрицы ковариации, то оптимальный байесовский классификатор линеен. Возникает вопрос: а только ли в этом случае? Оказывается, нет — он остаётся линейным при менее жёстких предположениях.

Базовые предположения. Пусть классов два, $Y = \{-1, +1\}$, объекты описываются nчисловыми признаками $f_j: X \to \mathbb{R}$, $j = 1,\ldots,n$. Будем полагать $X = \mathbb{R}^n$, отождествляя объекты с их признаковыми описаниями: $x \equiv (f_1(x),\ldots,f_n(x))$.

Гипотеза 4.1. Множество прецедентов $X \times Y$ является вероятностным пространством. Выборка прецедентов $X^\ell = (x_i, y_i)_{i=1}^\ell$ получена случайно и независимо согласно вероятностному распределению с плотностью $p(x,y) = P_y p_y(x) = P(y|x)p(x)$, где P_y — априорные вероятности, $p_y(x)$ — функции правдоподобия, $P(y|x)$ — апостериорные вероятности классов $y \in Y$.

Опр. 4.2. Плотность распределения $p(x)$, $x \in R^n$ называется экспонентной, если $p(x) = \exp\big(c(\delta)\,\langle \theta, x\rangle + b(\delta,\theta) + d(x,\delta)\big)$, где параметр $\theta \in R^n$ называется сдвигом, параметр δ называется разбросом, b,c,d — произвольные числовые функции.

Класс экспонентных распределений очень широк. К нему относятся многие непрерывные и дискретные распределения: равномерное, нормальное, гипергеометрическое, пуассоновское, биномиальное, Г-распределение, и другие.

Пример 4.1. Многомерное нормальное распределение с вектором матожидания $\mu \in \mathbb{R}^n$ и ковариационной матрицей $\Sigma \in \mathbb{R}^{n\times n}$ является экспонентным с параметром сдвига $\theta = \Sigma^{-1}\mu$ и параметром разброса $\delta = \Sigma$:

$$\mathcal{N}(x;\mu,\Sigma) = (2\pi)^{-\frac{n}{2}}|\Sigma|^{-\frac{1}{2}}\exp\bigl(-\tfrac{1}{2}(x-\mu)^\intercal\Sigma^{-1}(x-\mu)\bigr) =$$
$$= \exp\Bigl(\underbrace{\mu^\intercal\Sigma^{-1}x}_{\langle\theta,x\rangle} - \underbrace{\tfrac{1}{2}\mu^\intercal\Sigma^{-1}\Sigma\Sigma^{-1}\mu}_{b(\delta,\theta)} - \underbrace{\tfrac{1}{2}x^\intercal\Sigma^{-1}x - \tfrac{n}{2}\ln(2\pi) - \tfrac{1}{2}\ln|\Sigma|}_{d(x,\delta)}\Bigr).$$

Гипотеза 4.2. *Функции правдоподобия классов $p_y(x)$ принадлежат экспонентному семейству плотностей, имеют равные значения параметров d и δ, но отличаются значениями параметра сдвига θ_y.*

Основная теорема. Напомним, что оптимальный байесовский классификатор имеет вид $a(x) = \arg\max\limits_{y\in Y}\lambda_y \mathsf{P}(y|x)$, где λ_y — штраф за ошибку на объектах класса y. В случае двух классов

$$a(x) = \operatorname{sign}\bigl(\lambda_+\mathsf{P}(+1|x) - \lambda_-\mathsf{P}(-1|x)\bigr) = \operatorname{sign}\left(\frac{\mathsf{P}(+1|x)}{\mathsf{P}(-1|x)} - \frac{\lambda_-}{\lambda_+}\right).$$

Теорема 4.2. *Если справедливы гипотезы 4.1, 4.2, и среди признаков $f_1(x),\ldots,f_n(x)$ есть константа, то:*

1) байесовский классификатор является линейным: $a(x) = \operatorname{sign}\bigl(\langle w,x\rangle - w_0\bigr)$, где $w_0 = \ln(\lambda_-/\lambda_+)$, а вектор w не зависит от штрафов λ_-, λ_+;

2) апостериорная вероятность принадлежности произвольного объекта $x\in X$ классу $y\in\{-1,+1\}$ может быть вычислена по значению дискриминантной функции: $\mathsf{P}(y|x) = \sigma\bigl(\langle w,x\rangle\, y\bigr)$, где $\sigma(z) = \frac{1}{1+e^{-z}}$ — сигмоидная функция.

Доказательство. Рассмотрим отношение апостериорных вероятностей классов и воспользуемся тем, что $p_y(x)$ — экспонентные плотности с параметрами θ_y и δ:

$$\frac{\mathsf{P}(+1|x)}{\mathsf{P}(-1|x)} = \frac{P_+p_+(x)}{P_-p_-(x)} = \exp\Bigl(\bigl\langle\underbrace{(c_+(\delta)\theta_+ - c_-(\delta)\theta_-)}_{w=\mathrm{const}(x)}, x\bigr\rangle + \underbrace{b_+(\delta,\theta_+) - b_-(\delta,\theta_-) + \ln\tfrac{P_+}{P_-}}_{\mathrm{const}(x)}\Bigr).$$

Здесь вектор w не зависит от x и является вектором свободных коэффициентов при признаках. Все слагаемые под экспонентой, не зависящие от x, можно считать аддитивной добавкой к коэффициенту при константном признаке. Поскольку свободные коэффициенты настраиваются по обучающей выборке, вычислять эту аддитивную добавку нет никакого смысла, и её можно включить в $\langle w,x\rangle$.

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

Используя формулу полной вероятности $\mathsf{P}(-1|x) + \mathsf{P}(+1|x) = 1$, нетрудно выразить апостериорные вероятности $\mathsf{P}(-1|x)$ и $\mathsf{P}(+1|x)$ через $\langle w, x\rangle$:

$$\mathsf{P}(+1|x) = \sigma\big(+\langle w,x\rangle\big); \quad \mathsf{P}(-1|x) = \sigma\big(-\langle w,x\rangle\big).$$

Объединяя эти два равенства в одно, получаем требуемое: $\mathsf{P}(y|x) = \sigma\big(\langle w,x\rangle\, y\big)$.

Разделяющая поверхность в байесовском решающем правиле определяется уравнением $\lambda_- \mathsf{P}(-1|x) = \lambda_+ \mathsf{P}(+1|x)$, которое равносильно $\langle w,x\rangle - \ln\frac{\lambda_-}{\lambda_+} = 0$, следовательно, разделяющая поверхность линейна. ■

Объединяя эти два равенства в одно, получаем требуемое: $\mathsf{P}(y|x) = \sigma\big(\langle w,x\rangle\, y\big)$.

Разделяющая поверхность в байесовском решающем правиле определяется уравнением λ_- P(−1|x) = λ_+ P(+1|x), которое равносильно $\langle w,x\rangle - \ln\frac{\lambda_-}{\lambda_+} = 0$, следовательно, разделяющая поверхность линейна.

Рис. 10. Логарифмическая функция потерь $\log_2(1+e^{-M_i})$ и её наклонная асимптота.

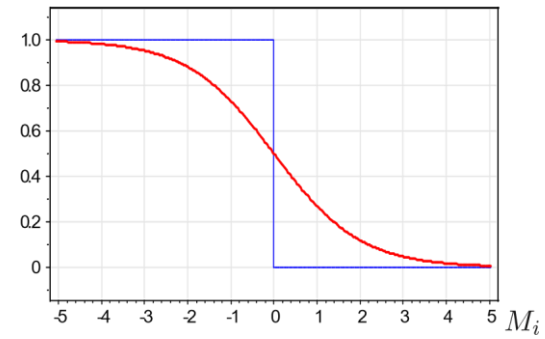


Рис. 11. Правило Хэбба: пороговое $[M_i<0]$ и сглаженное $\sigma(-M_i)$.

Принцип максимума правдоподобия. Для настройки вектора весов w по обучающей выборке X^ℓ будем максимизировать логарифм правдоподобия выборки:

$$L(w, X^\ell) = \log_2 \prod_{i=1}^{\ell} p(x_i, y_i) \to \max_w .$$

Согласно определению условной вероятности, $p(x, y) = \mathsf{P}(y|x)p(x)$, где плотности распределения объектов $p(x)$ не зависят от вектора параметров w. Апостериорные вероятности выражаются согласно Теореме 4.2 через линейную дискриминантную функцию: $\mathsf{P}(y|x) = \sigma\big(\langle w, x\rangle\, y\big)$. Таким образом,

$$L(w, X^\ell) = \sum_{i=1}^{\ell} \log_2 \sigma\big(\langle w, x_i\rangle\, y_i\big) + \text{const}(w) \to \max_w .$$

Максимизация правдоподобия $L(w, X^\ell)$ эквивалентна минимизации функционала $\widetilde{Q}(w, X^\ell)$, гладко аппроксимирующего эмпирический риск (4.1):

$$\widetilde{Q}(w, X^\ell) = \sum_{i=1}^{\ell} \log_2\big(1 + \exp(-\langle w, x_i\rangle\, y_i)\big) \to \min_w . \qquad (4.12)$$

Таким образом, логистическая функция потерь $\mathscr{L}(M) = \log_2\big(1+e^{-M}\big)$ является следствием экспонентности классов и принципа максимума правдоподобия.

Градиентный шаг. Запишем градиент функционала $\widetilde{Q}(w)$, воспользовавшись выражением для производной сигмоидной функции $\sigma'(z) = \sigma(z)\big(1-\sigma(z)\big) = \sigma(z)\sigma(-z)$, и получим *логистическое правило* обновления весов для градиентного шага в методе стохастического градиента:

$$w := w + \eta\, y_i x_i\, \sigma\big(-\langle w, x_i\rangle\, y_i\big), \qquad (4.13)$$

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

Аналогия с правилом Хэбба. Логистическая функция потерь является сглаженным вариантом кусочно-линейной функции потерь, соответствующей правилу Хэбба, Рис. 10. Поэтому и логистическое правило (4.13) оказывается, в свою очередь, сглаженным вариантом правила Хэбба (4.11), Рис. 11:

$$w := w + \eta\, y_i x_i \big[\langle w, x_i\rangle\, y_i < 0\big].$$

В правиле Хэбба смещение весов происходит только когда на объекте x_iдопускается ошибка. В логистическом правиле смещение тем больше, чем меньше отступ $Mi(w)$ = hw,xiiyi, то есть чем серьёзнее ошибка. Даже если ошибки нет, но объект близок к границе классов, веса модифицируются так, чтобы граница прошла как можно дальше от объекта. Тем самым градиентная минимизация реализует стратегию увеличения зазора (margin)

между обучающими объектами и разделяющей поверхностью, что способствует

улучшению обобщающей способности.

О методах второго порядка. Для оптимизации весов wчаще используется метод Ньютона-Рафсона, описанный в главе о регрессионном анализе, см. 5.5.5. Он имеет более высокую скорость сходимости в окрестности локального оптимума, но требует решения линейной регрессионной задачи (следовательно, обращения ковариационной матрицы размера $n \times n$) на каждой итерации. Отсюда и название метода — логистическая регрессия. Это регрессия ещё и потому, что для классифицируемого объекта xоценивается вещественная величина — апостериорная вероятность $P(+1|x)$.

Сравнение с линейным дискриминантом Фишера. Линейный дискриминант Фишера (ЛДФ) и логистическая регрессия исходят из байесовского решающего правила и принципа максимума правдоподобия, однако результат получается разный. В ЛДФ приходится оценивать $n|Y| + n(n+1)/2$ параметров (векторы средних для каждогоnкласса и общую ковариационную матрицу), в логистической регрессии — только (вектор весов w). Почему? Дело в том, что ЛДФ решает вспомогательную задачу восстановления плотностей распределения классов, предполагая, что плотности нормальны. Логистическая регрессия опирается на более слабые предположения о виде плотностей. С точки зрения философского принципа Оккама «не плодить сущности без необходимости» логистическая регрессия явно предпочтительнее, поскольку ЛДФ вводит избыточную сущность и сводит задачу классификации к более сложной задаче восстановления плотностей.

Достоинства логистической регрессии.

- с линейным дискриминантом Фишера логистическая регрессия даёт лучшие результаты по сравнению с дельта-правилом и правилом Хэбба (поскольку она использует «более правильную» функцию потерь).
- Возможность оценивать апостериорные вероятности и риски.

Недостатки логистической регрессии.

- Градиентный метод обучения логистической регрессии наследует все недостатки метода стохастического градиента. Практичная реализация должна предусматривать стандартизацию данных, отсев выбросов, регуляризацию (сокращение весов), отбор признаков, и другие эвристики для улучшения сходимости. Возможно применение метода второго порядка, но он требует обращения $n \times n$ матриц на каждом шаге и также не застрахован от плохой сходимости.

Скоринг и оценивание апостериорных вероятностей

Скоринг. В случае бинарных признаков, $X = \{0,1\}^n$, можно полагать, что функции правдоподобия классов $p_y(x)$ описываются биномиальными распределениями, следовательно, являются экспонентными. Это соображение служит дополнительным

обоснованием бинаризации признаков, когда каждый не бинарный исходный признак заменяется одним или несколькими бинарными.

В бинарном случае вычисление линейной дискриминантной функции удобно рассматривать как подсчёт баллов (score): если $f_j(x) = 1$, то есть признак f_j наблюдается у объекта x, то к сумме баллов добавляется вес w_j. Классификация производится путём сравнения набранной суммы баллов с пороговым значением w_0.

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

Рис. 12. Фрагмент скоринговой карты для задачи принятия кредитных решений.

Благодарясвой простоте подсчёт баллов или скоринг (scoring) пользуется большой популярностью в таких областях, как медицина, геология, банковское дело, социология, маркетинг, и др. Абсолютное значение веса w_j можно интерпретировать как степень важности признака f_j, а знак $\text{sign}(w_j)$ показывает, в пользу какого класса свидетельствует наличие данного признака. Это важная дополнительная информация о признаках, помогающая экспертам лучше понимать и задачу, и классификатор.

После бинаризации признаков классификатор представляется в виде так называемой скоринговой карты (scorecard), в которой перечисляются все исходные признаки, для каждого исходного — все построенные по нему бинарные признаки, для каждого бинарного — его вес. Имея такую карту, классификацию можно проводить с помощью стандартной электронной таблицы или даже вручную. Рис.12.

Вероятностный выход и оценивание рисков Логистическая функция σ переводит значение линейной дискриминантной функции $\langle w, x\rangle$ в оценку апостериорной вероятности того, что объект x принадлежит классу $+1$: $\mathsf{P}(+1|x) = \sigma\langle w, x\rangle$. Это свойство используется в тех приложениях, где наряду с классификацией объекта x требуется оценить связанный с ним *риск* как математическое ожидание потерь:

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

где λ_y — величина потери при ошибочной классификации объекта класса y.

В практических ситуациях к оценкам апостериорной вероятности следует относиться с осторожностью. Теорема 4.2 гарантирует, что $\mathsf{P}(y|x) = \sigma\big(\langle w, x\rangle\, y\big)$ только для экспонентных классов с равными параметрами разброса. В остальных случаях оценка вероятности носит эвристический характер. На практике экспонентность редко когда проверяется, а гарантировать равенство разброса вообще не представляется возможным.

Вероятностная калибровка позволяет пересчитать значение дискриминантной функции в оценку апостериорной вероятности, когда условия теоремы 4.2 не выполняются, и даже когда классификатор $a(x) = \text{sign} f(x,w)$ не является линейным. Предполагается, что

апостериорная вероятность P(+1|*x*) монотонно зависит от значения дискриминантной функции *f*(*x*,*w*). Существует много способов ввести модель этой зависимости, но мы рассмотрим только один — калибровку Платта, основанную на линейно-сигмоидальной модели:

$$\mathsf{P}(+1|x) = \sigma\big(\alpha f(x,w) + \beta\big).$$

Для настройки неизвестных параметров правдоподобия(α,β) $\in$ R^2 решается задача максимизации

$$\sum_{i=1}^{\ell} \log \mathsf{P}(y_i|x_i) = \sum_{i=1}^{\ell} \log \sigma\big(\alpha y_i f(x_i,w) + \beta y_i\big) \;\to\; \max_{\alpha,\beta}$$

любым стандартными численными методами оптимизации.

Метод опорных векторов

В 60–70-е годы коллективом советских математиков под руководством В.Н. Вапника был разработан метод обобщённого портрета, основанный на построении оптимальной разделяющей гиперплоскости. Требование оптимальности заключалось в том, что обучающие объекты должны быть удалены от разделяющей поверхности настолько далеко, насколько это возможно. На первый взгляд принцип оптимальности существенно отличается от методов минимизации эмпирического риска или максимизации правдоподобия, применяемых в других линейных классификаторах — персептроне, дискриминанте Фишера, логистической регрессии.

В 90-е годы метод получил широкую мировую известность и после некоторой переработки и серии обобщений стал называться машиной опорных векторов (supportvectormachine, SVM) . В настоящее время он считается одним из лучших методов классификации.

Метод SVM обладает несколькими замечательными свойствами. Во-первых, обучение SVM сводится к задаче квадратичного программирования, имеющей единственное решение, которое вычисляется достаточно эффективно даже на выборках в сотни тысяч объектов. Во-вторых, решение обладает свойством разреженности: положение оптимальной разделяющей гиперплоскости зависит лишь от небольшой доли обучающих объектов. Они и называются опорными векторами; остальные объекты фактически не задействуются. Наконец, с помощью изящного математического приёма — введения функции ядра — метод обобщается на случай нелинейных разделяющих поверхностей. Вопрос о выборе ядра, оптимального для данной прикладной задачи, до сих пор остаётся открытой теоретической проблемой.

Линейно разделимая выборка

Рассмотрим задачу классификации на два непересекающихся класса, в которой объекты описываются *n*-мерными вещественными векторами: $X = R^n$, $Y = \{-1,+1\}$. Будем строить линейный пороговый классификатор:

$$a(x) = \operatorname{sign}\left(\sum_{j=1}^{n} w_j x^j - w_0\right) = \operatorname{sign}\big(\langle w,x\rangle - w_0\big), \qquad (4.14)$$

где $x = (x^1,\ldots,x^n)$ — признаковое описание объекта x; вектор $w = (w^1,\ldots,w^n) \in \mathbb{R}^n$ и скалярный порог $w_0 \in \mathbb{R}$ являются параметрами алгоритма. Уравнение $\langle w,x\rangle = w_0$ описывает гиперплоскость, разделяющую классы в пространстве $\mathbb{R}^n$.

Предположим, что выборка $X^\ell = (x_i, y_i)_{i=1}^\ell$ линейно разделима и существуют значения параметров w, w_0, при которых функционал числа ошибок

$$Q(w, w_0) = \sum_{i=1}^{\ell} \big[y_i(\langle w, x_i\rangle - w_0) \leqslant 0\big]$$

принимает нулевое значение. Но тогда разделяющая гиперплоскость не единственна. Можно выбрать другие её положения, реализующие такое же разбиение выборки на два класса. Идея метода заключается в том, чтобы разумным образом распорядиться этой свободой выбора.

Оптимальная разделяющая гиперплоскость. Потребуем, чтобы разделяющая гиперплоскость максимально далеко отстояла от ближайших к ней точек обоих классов. Первоначально данный принцип классификации возник из эвристических соображений: вполне естественно полагать, что максимизация зазора (margin) между классами должна способствовать более надёжной классификации.

Нормировка. Заметим, что параметры линейного порогового классификатора определены с точностью до нормировки: алгоритм $a(x)$ не изменится, если w и w_0 одновременно умножить на одну и ту же положительную константу. Удобно выбрать эту константу таким образом, чтобы выполнялось условие

$$\text{Множество точек } x \text{ описывает полосу, разделяющую} \quad : \; -1 \leqslant \langle w, x\rangle - w_0 \leqslant 1\big\} \tag{4.15}$$

классы, см. Рис. 13. Ни один из объектов обучающей выборки не попадает внутрь этой полосы. Границами полосы служат две параллельные гиперплоскости с вектором нормали w. Разделяющая гиперплоскость проходит ровно по середине между ними. Объекты, ближайшие к разделяющей гиперплоскости, лежат на границах полосы, и именно на них достигается минимум (4.15). В каждом из классов имеется хотя бы один такой объект, в противном случае разделяющую полосу можно было бы ещё немного расширить и нарушался бы принцип максимального зазора.

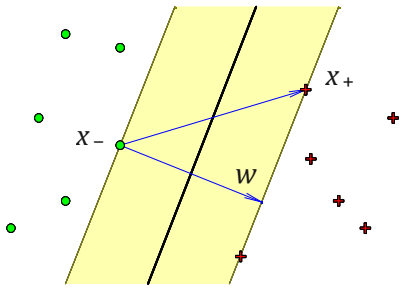


Рис. 13. Линейно разделимая выборка. Обучающие объекты x_- и x_+ находятся на границе разделяющей полосы. Вектор нормали w к разделяющей гиперплоскости определяет ширину полосы.

Ширина разделяющей полосы. Чтобы разделяющая гиперплоскость как можно дальше отстояла от точек выборки, ширина полосы должна быть максимальной. Пусть x_- и x_+ —

два обучающих объекта классов −1 и +1 соответственно, лежащие на границе полосы. Тогда ширина полосы есть

$$\left\langle (x_+ - x_-), \frac{w}{\|w\|} \right\rangle = \frac{\langle w, x_+ \rangle - \langle w, x_- \rangle}{\|w\|} = \frac{(w_0 + 1) - (w_0 - 1)}{\|w\|} = \frac{2}{\|w\|}.$$

Ширина полосы максимальна, когда норма вектора *w* минимальна.

Итак, в случае линейно разделимой выборки получаем задачу квадратичного программирования: требуется найти значения параметров *w* и w_0, при которых выполняются ℓ ограничений-неравенств и норма вектора *w* минимальна:

$$\begin{cases} \langle w, w \rangle \to \min; \\ y_i\big(\langle w, x_i \rangle - w_0\big) \geqslant 1, \quad i = 1, \ldots, \ell. \end{cases} \tag{4.16}$$

Линейная дискриминантная функция

Рассмотрим задачу построения линейной разделяющей гиперповерхности. Главным достоинством линейного классификатора является его простота и вычислительная эффективность.

Рассмотрим линейную дискриминантную функцию: $g(x) = W^T x + W_0$, где $W^T = (W_1, W_2, \ldots, W_l)^T$ – весовой *вектор,* W_0 – порог. Поведение решения задается уравнением $g(x) = 0$. Пусть X_1 и X_2 – два конечных *множества* векторов признаков в евклидовом пространстве, относящихся к классу Ω_1 и Ω_2 соответственно, т.е X_1 принадлежит классу Ω_1 при $g(x) > 0$, а X_2 принадлежит классу Ω_2 при $g(x) < 0$.

Задача состоит в том, чтобы:

- установить разделимость этих множеств;
- найти разделяющую гиперплоскость.

Рассмотрим сначала в качестве примера двумерную задачу, когда образы представляются точками на плоскости.

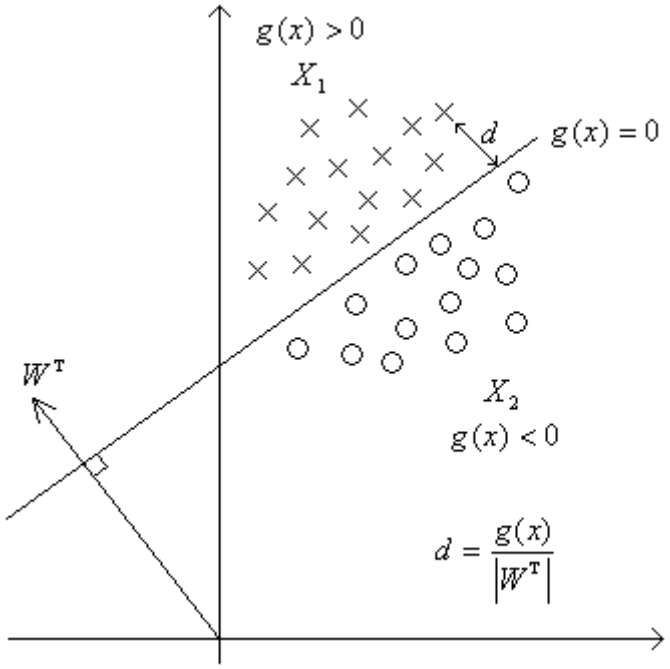
Определение. *Множество, содержащее отрезок, соединяющий две произвольные внутренние точки, называется выпуклым*.

Определение. *Выпуклая оболочка – это минимальное выпуклое множество, содержащее данное*.

Утверждение . *Два множества на плоскости линейно разделимы тогда и только тогда, когда их выпуклые оболочки не пересекаются*.

Из этого утверждения получаем следующее *правило проверки* разделимости множеств на плоскости:

1. Построить *выпуклые оболочки*.
2. Проверить пересечение *выпуклых оболочек*. Если они не пересекаются, то множества разделимы.



Очевидно и правило, по которому можно найти разделяющую прямую:

1. Найти ближайшую пару точек в выпуклых оболочках обоих множеств.
2. Построить срединный перпендикуляр к отрезку, соединяющему эти точки. Этот перпендикуляр и будет разделяющей прямой.

Пусть *размерность* вектора признаков X и вектора коэффициентов W равна l. Рассмотрим "пополненные" вектора X', W' следующего вида: $(W')^T = (W^T, W_0)$ – пополненный весовой *вектор*, $(X')^T = (X^T, 1)$ – пополненный *вектор* признаков. Рассмотрим также в $(l+1)$ -мерном пространстве однородную линейную функцию $g'(x) = \left((W')^T, (X')^T\right) = \sum_{i=0}^{l} W_i \cdot x_i$.

Очевидно следующее

Утверждение. *Множества* X_1 и X_2 *линейно разделимы* в пространстве R^l дискриминантной функцией $g(x) = W^T x + W_0$ тогда и только тогда, когда они разделимы в пополненном пространстве $R^l + 1$ *однородной дискриминантной функцией* $g'(x) = \left((W')^T, (X')^T\right) = \sum_{i=0}^{l} W_i \cdot x_i$.

Далее будем рассматривать дискриминантные функции и вектора в пополненном пространстве.

Определение. *Множество* $\overline{X} = -X$ *называется симметричным множеством к множеству* X.

Утверждение. *Два замкнутых множества* X_1 и X_2 *разделимы тогда и только тогда, когда выпуклая оболочка множества* $X_1 \bigcup \overline{X}_2$ *не содержит начала координат*.

Доказательство. Пусть *множества* X_1 и X_2 разделимы. Тогда существует *линейная функция* $g(x)$ такая, что $g(x) > 0$ при $x \in X_1$ и $g(x) < 0$ при $x \in X_2$. Рассмотрим множество $X = X_1 \bigcup \overline{X}_2$, тогда $g(x) > 0$ при $x \in X$. Следовательно, $g(x) > 0$ для выпуклой линейной комбинации из X, а это означает, что $O \notin convX$, т.к. X – замкнутое. Здесь O обозначает начало координат.

Пусть $O \notin convX$, и пусть $\widetilde{x}$ – ближайшая к началу координат O точка из $convX$. *Плоскость* $(W, x) = 0$ с *направляющим вектором* $W = \widetilde{x}$ не пересекает $convX$, а, значит, $(W, x) > 0$ на $x \in X$. Следовательно, $(W, x) < 0$ на $x \in X_2$.

Линейно неразделимая выборка

Чтобы обобщить постановку задачи на случай линейно неразделимой выборки, позволим алгоритму допускать ошибки на обучающих объектах, но при этом постараемся, чтобы ошибок было поменьше. Введём дополнительные переменные ξ_i>0, характеризующие величину ошибки на объектах x_i, i= 1,...,ℓ. Ослабим в (4.16) ограничения-неравенства и одновременно введём в минимизируемый функционал штраф за суммарную ошибку:

$$\begin{cases} \dfrac{1}{2}\langle w, w\rangle + C\sum\limits_{i=1}^{\ell}\xi_i \to \min\limits_{w, w_0, \xi}; \\ y_i\big(\langle w, x_i\rangle - w_0\big) \geqslant 1 - \xi_i, \quad i = 1, \ldots, \ell; \\ \xi_i \geqslant 0, \quad i = 1, \ldots, \ell. \end{cases} \tag{4.17}$$

Положительная константа *C* является управляющим параметром метода и позволяет находить компромисс между максимизацией ширины разделяющей полосы и минимизацией суммарной ошибки.

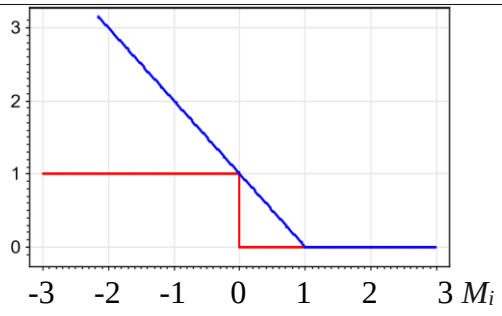


Рис. 14. Кусочно-линейная аппроксимация пороговой функции потерь: $[M_i<0] \le (1-M_i)_+$.

Регуляризация эмпирического риска. В задачах с двумя классами $Y = \{-1,+1\}$ отступом (margin) объекта x_i от границы классов называется величина

$$M_i(w, w_0) = y_i\big(\langle w, x_i\rangle - w_0\big).$$

Алгоритм (4.14) допускает ошибку на объекте x_i тогда и только тогда, когда отступ M_i отрицателен. Если $M_i \in (-1,+1)$, то объект x_i попадает внутрь разделяющей полосы. Если $M_i>1$, то объект x_i классифицируется правильно, и находится на некотором удалении от разделяющей полосы.

Согласно (4.17) ошибка ξ_i выражается через отступ M_i. Действительно, из ограничений-неравенств следует, что $\xi_i>0$ и $\xi_i>1-M_i$. В силу требования минимизации суммы $P i\xi_i$ одно из этих неравенств обязательно должно обратиться в равенство.− Следовательно, $\xi_i = (1\ M_i)_+$. Таким образом, задача (4.17) оказывается эквивалентной безусловной минимизации функционала Q, не зависящего от переменных ξ_i:

$$Q(w, w_0) = \sum_{i=1}^{\ell}\big(1 - M_i(w, w_0)\big)_+ + \frac{1}{2C}\|w\|^2 \to \min_{w,w_0}. \qquad (4.18)$$

В силу неравенства $[M_i<0] \le (1 - M_i)_+$, рис. 14, функционал (4.18) можно рассматривать как верхнюю оценку эмпирического риска (числа ошибочных классификаций объектов обучающей выборки), к которому добавлен регуляризатор $\|w\|^2$, умноженный на параметр регуляризации $\frac{1}{2C}$.

Замена пороговой функции потерь $[M<0]$ кусочно-линейной верхней оценкой $L(M) = (1 - M)_+$ делает функцию потерь чувствительной к величине ошибки. Функция потерь $L(M)$ штрафует объекты за приближение к границе классов.

Введение регуляризатора повышает устойчивость решения w. В случаях, когда минимум эмпирического риска достигается на множестве векторов w, регуляризация выбирает из них вектор с минимальной нормой. Тем самым устраняется проблема мультиколлинеарности, повышается устойчивость алгоритма, улучшается его обобщающая способность. Таким образом, принцип оптимальной разделяющей гиперплоскости или максимизации ширины разделяющей полосы тесно связан с регуляризацией некорректно поставленных задач по А.Н.Тихонову [23].

Задача (4.18) соответствует принципу максимума совместного правдоподобия (4.3), если принять модель плотности

$$p(x_i, y_i|w) = z_1 \exp\bigl(-(1 - M_i(w, w_0))_+\bigr),$$

гауссовскую модель априорного распределения вектора параметров w

$$p(w; C) = z_2 \exp\left(-\frac{\|w\|^2}{2C}\right),$$

и не накладывать никаких ограничений на параметр w_0. Здесь z_1, z_2 — нормировочные константы, C— гиперпараметр.

Понимание роли функции потерь и регуляризатора необходимо для того, чтобы более вольно обращаться с постановкой задачи, при необходимости модифицировать её, получая методы, похожие на SVM, но обладающие требуемыми свойствами.

Двойственная задача. Запишем функцию Лагранжа задачи (4.17):

$$\mathscr{L}(w, w_0, \xi; \lambda, \eta) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{\ell}\xi_i - \sum_{i=1}^{\ell}\lambda_i\bigl(M_i(w, w_0) - 1 + \xi_i\bigr) - \sum_{i=1}^{\ell}\xi_i\eta_i =$$

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

где $\lambda = (\lambda_1, \ldots, \lambda_\ell)$ — вектор переменных, двойственных к w; $\eta = (\eta_1, \ldots, \eta_\ell)$ — вектор переменных, двойственных к $\xi = (\xi_1, \ldots, \xi_\ell)$.

Согласно теореме Куна-Таккера задача (4.17) эквивалентна двойственной задаче поиска седловой точки функции Лагранжа:

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

В последних двух строках записаны *условия дополняющей нежёсткости*.

Необходимым условием седловой точки функции Лагранжа является равенство нулю её производных. Отсюда получаются три полезных соотношения:

$$\frac{\partial \mathscr{L}}{\partial w} = w - \sum_{i=1}^{\ell}\lambda_i y_i x_i = 0 \quad \Longrightarrow \quad w = \sum_{i=1}^{\ell}\lambda_i y_i x_i; \tag{4.19}$$

$$\frac{\partial \mathscr{L}}{\partial w_0} = -\sum_{i=1}^{\ell}\lambda_i y_i = 0 \quad \Longrightarrow \quad \sum_{i=1}^{\ell}\lambda_i y_i = 0; \tag{4.20}$$

$$\frac{\partial \mathscr{L}}{\partial \xi_i} = -\lambda_i - \eta_i + C = 0 \quad \Longrightarrow \quad \eta_i + \lambda_i = C, \quad i = 1, \ldots, \ell. \tag{4.21}$$

Из (4.19) следует, что искомый вектор весов wявляется линейной комбинацией

векторов обучающей выборки x_i, причём только тех, для которых $\lambda_i>0$.

Опр. 4.3. Если $\lambda_i>0$, то объект обучающей выборки x_iназывается опорным вектором (supportvector).

Из третьего соотношения (4.21) и неравенства $\eta_i>0$ следует $0 \leq \lambda_i \leq C$. Отсюда, и из условий дополняющей нежёсткости вытекает, что возможны только три допустимых сочетания значений переменных ξ_i, λ_i, η_iи отступов M_i.

Соответственно, все объекты x_i, $i= 1,\dots,\ell$ делятся на следующие три типа:

1. $\lambda_i= 0$; $\eta_i= C$; $\xi_i= 0$; $M_i>1$.

Объект x_iклассифицируется правильно и не влияет на решение w. Такие объекты будем называть периферийными или неинформативными.

2. $0 <\lambda_i< C$; $0 <\eta_i< C$; $\xi_i= 0$; $M_i= 1$.

Объект x_iклассифицируется правильно и лежит в точности на границе разделяющей полосы. Такиеобъектыбудемназыватьопорнымиграничными.

3. $\lambda_i= C$; $\eta_i= 0$; $\xi_i>0$; $M_i<1$.

Объект x_iлибо лежит внутри разделяющей полосы, но классифицируется правильно ($0 <\xi_i<1$, $0 <M_i<1$), либо попадает на границу классов ($\xi_i= 1$, $M_i= 0$), либо вообще относится к чужому классу ($\xi_i>1$, $M_i<0$). Во всех этих случаях объект x_iбудем называть опорным нарушителем.

В силу соотношения (4.21) в лагранжиане обнуляются все члены, содержащие переменные ξ_iи η_i, и он выражается только через двойственные переменные λ_i.

$$\begin{cases} -\mathscr{L}(\lambda) = -\sum\limits_{i}^{\ell} \lambda_i + \dfrac{1}{2}\sum\limits_{i=1}^{\ell}\sum\limits_{j=1}^{\ell} \lambda_i\lambda_j y_i y_j \langle x_i, x_j\rangle \;\to\; \min\limits_{\lambda}; \\ 0 \leqslant \lambda_i \leqslant C, \quad i = 1,\dots,\ell \\ \sum\limits_{i=1} \lambda_i y_i = 0 \end{cases}; \tag{4.22}$$

Здесь минимизируется квадратичный функционал, имеющий неотрицательно определённую квадратичную форму, следовательно, выпуклый. Область, определяемая ограничениями неравенствами и одним равенством, также выпуклая. Следовательно, данная двойственная задача имеет единственное решение.

Допустим, мы решили эту задачу. Тогда вектор wвычисляется по формуле (4.19). Для определения порога w_0 достаточно взять произвольный опорный граничный вектор x_iи выразить w_0 из равенства w_0 = hw,x_ii − y_i. На практике для повышения численной устойчивости рекомендуется брать медиану множества значений w_0, вычисленных по всем граничным опорным векторам:

$$w_0 = \operatorname{med}\{\langle w, x_i\rangle - y_i : \lambda_i > 0,\ M_i = 1,\ i = 1,\dots,\ell\}. \tag{4.23}$$

В итоге алгоритм классификации представляется в следующем виде:

$$a(x) = \operatorname{sign}\left(\sum_{i=1}^{\ell} \lambda_i y_i \langle x_i, x\rangle - w_0\right). \qquad (4.24)$$

Обратим внимание, что суммирование идёт не по всей выборке, а только по опорным векторам, для которых $\lambda_i = 0$. Классификатор $a(x)$ не изменится, если все остальные объекты исключить из выборки. Это свойство называют разреженностью (sparsity); именно оно и отличает SVM от других линейных классификаторов — дискриминанта Фишера, логистической регрессии и однослойного персептрона.

Ненулевыми λ_i обладают не только граничные опорные объекты, но и объекты-нарушители. Это говорит о недостаточной робастности (устойчивости к шуму) SVM. Нарушителями могут быть объекты, появившиеся в результате ошибочных наблюдений; их надо было бы исключить из выборки, а не строить по ним решение.

О подборе параметра регуляризации. Константу C обычно выбирают по критерию скользящего контроля. Это трудоёмкий способ, так как задачу приходится решать заново при каждом значении C. Хорошо то, что решение, как правило, не очень чувствительно к выбору C, и слишком точная его оптимизация не требуется.

Если есть основания полагать, что выборка почти линейно разделима, и лишь объекты-выбросы классифицируются неверно, то можно применить фильтрацию выбросов. Сначала задача решается при некотором C, и из выборки удаляется небольшая доля объектов, имеющих наибольшую величину ошибки ξ_i. После этого задача решается заново по усечённой выборке. Возможно, придётся проделать несколько таких итераций, пока оставшиеся объекты не окажутся линейно разделимыми.

Ядра и спрямляющие пространства

Существует ещё один подход к решению проблемы линейной неразделимости. Это переход от исходного пространства признаковых описаний объектов X к новому пространству H с помощью некоторого преобразования $\psi: X \to H$. Если пространство H имеет достаточно высокую размерность, то можно надеяться, что в нём выборка окажется линейно разделимой (легко показать, что если выборка X^ℓ не противоречива, то всегда найдётся пространство размерности не более ℓ, в котором она будет линейно разделима). Пространство H называют спрямляющим.

Если предположить, что признаковыми описаниями объектов являются векторы $\psi(x_i)$, а не векторы x_i, то построение SVM проводится точно так же, как и ранее.

Единственное отличие состоит в том, что скалярное произведение в пространстве X всюду заменяется скалярным произведением $\langle \psi(x), \psi($должно быть наделено $x')$ в пространстве $\langle x, x'\rangle$ в простран-H.

Отсюда вытекает естественное требование: пространство i

скалярным произведением, в частности, подойдёт любое евклидово, а в общем случае и гильбертово, пространство.

Опр. 4.4. Функция K: $X\times X\psi(\to x),\psi$R(называется$x'$)іпри некотором отображениидром

(kernelfunction), если онаψ: $X\to H$, представима в виде $K(x,x')$ = hгде H— пространство со скалярным произведением.

Постановка двойственной задачи (4.22), и сам алгоритм классификации (4.24) зависят только от скалярных произведений объектов, но не от самих признаковых описаний. Это означает, что скалярное произведение hx,x'іможно формально заменить ядром $K(x,x')$. Поскольку ядро в общем случае нелинейно, такая замена приводит к существенному расширению множества реализуемых алгоритмов a: $X\to Y$.

Более того, можно вообще не строить спрямляющее пространство Hв явном виде, и вместо подбора отображения ψзаниматься непосредственно подбором ядра.

Можно пойти ещё дальше, и вовсе отказаться от признаковых описаний объектов. Во многих практических задачах объекты изначально задаются информацией об их попарном взаимоотношении, например, отношении сходства. Если эта информация допускает представление в виде двуместной функции $K(x,x')$, удовлетворяющей аксиомам скалярного произведения, то задача может решаться методом SVM. Для такого подхода недавно был придуман термин беспризнаковое распознавание (featurelessrecognition), хотя многие давно известные метрические алгоритмы классификации (kNN, RBF и др.) также не требуют задания признаковых описаний.

Теорема Мерсера. Любая ли функция двух аргументов $K(x,x')$ может исполнять роль ядра? Следующая теорема даёт исчерпывающий ответ на этот вопрос и показывает, что класс допустимых ядер достаточно широк.

Теорема 4.3 (Мерсер, 1909 [53]). *Функция $K(x,x')$ является ядром тогда и только тогда, когда она симметрична, $K(x,x')=K(x',x)$, и неотрицательно определена: $\int_X\int_X K(x,x')g(x)g(x')dxdx'\geqslant 0$ для любой функции $g\colon X\to\mathbb{R}$.*

Существует эквивалентное определение неотрицательной определённости.

Опр. 4.5. *Функция $K(x,x')$ неотрицательно определена, если для любой конечной выборки $X^p=(x_1,\ldots,x_p)$ из X матрица $K=\|K(x_i,x_j)\|$ размера $p\times p$ неотрицательно определена: $z^{\intercal}Kz\geqslant 0$ для любого $z\in\mathbb{R}^p$.*

Проверка неотрицательной определённости функции в практических ситуациях может оказаться делом нетривиальным. Часто ограничиваются перебором конечного числа функций, про которые известно, что они являются ядрами. Среди них выбирается лучшая, как правило, по критерию скользящего контроля. Очевидно, что это не оптимальное решение. На сегодняшний день проблема выбора ядра, оптимального для данной конкретной задачи, остаётся открытой.

Конструктивные способы построения ядер. Следующие правила порождения позволяют строить ядра в практических задачах.

1. Произвольное скалярное произведение $K(x,x') = \langle x, x' \rangle$ является ядром.
2. Константа $K(x,x') = 1$ является ядром.
3. Произведение ядер $K(x,x') = K_1(x,x')K_2(x,x')$ является ядром.
4. Для любой функции $\psi\colon X \to \mathbb{R}$ произведение $K(x,x') = \psi(x)\psi(x')$ — ядро.
5. Линейная комбинация ядер с неотрицательными коэффициентами $K(x,x') = = \alpha_1 K_1(x,x') + \alpha_2 K_2(x,x')$ является ядром.
6. Композиция произвольной функции $\varphi\colon X \to X$ и произвольного ядра K_0 является ядром: $K(x,x') = K_0(\varphi(x), \varphi(x'))$.
7. Если $s\colon X \times X \to \mathbb{R}$ — произвольная симметричная интегрируемая функция, то $K(x,x') = \int_X s(x,z)s(x',z)\,dz$ является ядром.
8. Функция вида $K(x,x') = k(x - x')$ является ядром тогда и только тогда, когда Фурье-образ $F[k](\omega) = (2\pi)^{\frac{n}{2}} \int_X e^{-i\langle \omega, x\rangle} k(x)\,dx$ неотрицателен.
9. Предел локально-равномерно сходящейся последовательности ядер — ядро.
10. Композиция произвольного ядра K_0 и произвольной функции $f\colon \mathbb{R} \to \mathbb{R}$, представимой в виде сходящегося степенного ряда с неотрицательными коэффициентами $K(x,x') = f\big(K_0(x,x')\big)$, является ядром. В частности, функции $f(z) = e^z$ и $f(z) = \frac{1}{1-z}$ от ядра являются ядрами.

Примеры ядер. Существует несколько «стандартных» ядер, которые при ближайшем рассмотрении приводят к уже известным алгоритмам: полиномиальным разделяющим поверхностям, двухслойным нейронным сетям, потенциальным функциям (RBF-сетям), и другим. Таким образом, ядра претендуют на роль универсального языка для описания широкого класса алгоритмов обучения по прецедентам.

Наблюдается парадоксальная ситуация. С одной стороны, ядра — одно из самых красивых изобретений в машинном обучении. С другой стороны, до сих пор не найдено эффективного общего подхода к их подбору в конкретных задачах.

Пример 4.2. Возьмём $X = \mathrm{R}^2$ и рассмотрим ядро $K(u,v) = \mathrm{h}u,v\mathrm{i}^2$, где $u = (u_1,u_2)$, $v = (v_1,v_2)$. Попробуем понять, какое спрямляющее пространство и преобразование ψему соответствуют. Разложим квадрат скалярного произведения:

$$\begin{aligned} K(u,v) = \langle u,v\rangle^2 &= \langle (u_1,u_2),(v_1,v_2)\rangle^2 = \\ &= (u_1v_1 + u_2v_2)^2 = u_1^2v_1^2 + u_2^2v_2^2 + 2u_1v_1u_2v_2 = \\ &= \left\langle \left(u_1^2, u_2^2, \sqrt{2}u_1u_2\right), \left(v_1^2, v_2^2, \sqrt{2}v_1v_2\right)\right\rangle. \end{aligned}$$

Ядро Kпредставляется в виде скалярного произведения в пространстве $H = \mathrm{R}^3$. Преобразование ψ: $\mathrm{R}^2 \to \mathrm{R}^3$ имеет вид $\psi\colon (u_1,u_2) \mapsto (u_1^2, u_2^2, \sqrt{2}u_1u_2)$. Линейной поверхности в пространстве Hсоответствует квадратичная поверхность в исходном

пространстве X. Данное ядро позволяет разделить внутреннюю и наружную часть произвольного эллипса, что невозможно в исходном двумерном пространстве.

Пример 4.3. Усложним ситуацию. Пусть теперь $X = \mathbb{R}^n$,

$$K(u,v) = \langle u,v\rangle^d.$$

Тогда компонентами вектора $\psi(u)$ являются различные произведения $(u_1)^{d_1}\cdots(u_n)^{d_n}$ при всевозможных целых неотрицательных $d_1,\ldots,d_n$, удовлетворяющих условию $d_1+\cdots+d_n = d$. Число таких мономов, а следовательно и размерность пространства H, равно C^d_{n+d-1}. Пространство H изоморфно пространству всех полиномов, состоящих из мономов степени d от переменных $u_1,\ldots,u_n$.

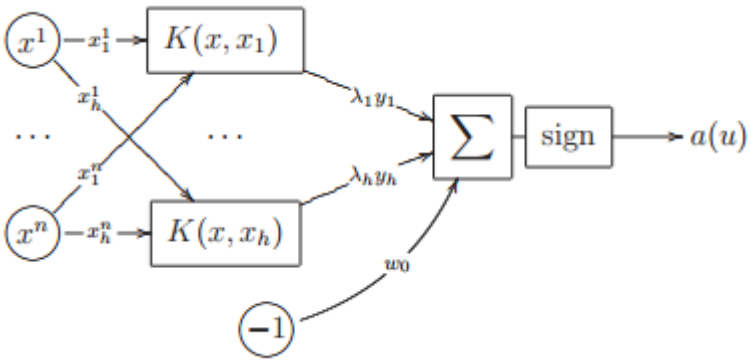
Пример 4.4. Если $X = \mathbb{R}^n$,

$$K(u,v) = \big(\langle u,v\rangle + 1\big)^d,$$

то H — пространство всех мономов степени *не выше* d от переменных $u_1,\ldots,u_n$. В этом случае пространство H изоморфно пространству всех полиномов степени d. Линейная разделимость множеств в этом пространстве эквивалентна полиномиальной разделимости множеств в исходном пространстве X.

SVM как двухслойная нейронная сеть. Рассмотрим структуру алгоритма $a(x)$ после замены в (4.24) скалярного произведения $\langle x_i,x\rangle$ ядром $K(x_i,x)$. Перенумеруем объекты так, чтобы первые h объектов оказались опорными. Поскольку $\lambda_i = 0$ для всех неопорных объектов, $i = h+1,\ldots,\ell$, алгоритм $a(x)$ примет вид

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



Если $X = \mathrm{R}^n$, то алгоритм $a(x)$ можно рассматривать как суперпозицию, называемую двухслойной нейронной сетью. Первый слой образуют ядра, второй слой — собственно

линейный классификатор. Такая суперпозиция, построенная методом SVM, имеет несколько замечательных особенностей.

Во-первых, число нейронов первого слоя определяется автоматически, тогда как в нейронных сетях определение числа нейронов является отдельной проблемой.

Во-вторых, проясняется смысл двойственных переменных: λ_i— это степень важности ядра $K(x,x_i)$, фактически, важности или «опорности» объекта x_i.

Пример 4.5. Классическая нейронная сеть с сигмоидными функциями активации получится, если в качестве ядра взять функцию

$$K(u,v) = \operatorname{th}\big(k_0 + k_1 \langle u,v\rangle\big).$$

Данная функция удовлетворяет условиям Мерсера не при всех значениях параметров k_0 и k_1. В частности, она им не удовлетворяет при $k_0 < 0$ или $k_1 < 0$. Однако это не препятствует её успешному практическому применению. Вместо гиперболического тангенса thzчасто используют также логистическую функцию $\sigma(z) = \frac{1}{1+e^{-z}}$.

Квадратичная форма утратит свойство неотрицательной определённости, минимизируемый функционал уже не будет выпуклым, и решение может оказаться не единственным. Самое неприятное то, что на границах гиперпараллелепипеда$0 \le \lambda_i \le C$возникнет огромное количество локальных минимумов, и поиск решения среди них в общем случае потребует полного перебора. В этой ситуации многие методы квадратичного программирования будут выдавать какой-то локальный минимум, совсем не обязательно хороший.

Пример 4.6. Нейронная сеть с радиальными базисными функциями (radialbasisfunctions, RBF) получится, если взять гауссовское ядро

$$K(u,v) = \exp\big(-\beta\|u-v\|^2\big),$$

где β— параметр. Ядро $K(x_i,x)$ вычисляет оценку близости объекта xк опорному объекту x_i. Чем ближе объекты, тем больше значение ядра. Выходной нейрон складывает все эти оценки, умножая их на коэффициенты $\lambda_i y_i$. При этом близости к опорным объектам класса +1 суммируются с положительными весами, а к объектам класса −1 — с отрицательными. Выходной нейрон производит голосование,xк обоим классам.сравнивая суммарные близости распознаваемого объекта

В альтернативном методе обучения RBF-сетей, основанном на EM-алгоритме гауссовские ядра играли роль компонент смеси вероятностных распределений. Центры ядер размещались не в опорных объектах, а в местах локальных сгущений плотности объектов. В этом и заключается основное отличие SVM-RBF от EM-RBF. Метод SVM сдвигает центры гауссианов ближе к границе классов, в результате форма разделяющей поверхности описывается более чётко. Таким образом, SVM-RBF лучше подходит для описания классов с границами сложной формы. С другой стороны, EM-RBF более устойчив к выбросам и предпочтителен в задачах с «размытыми» границами классов.

Преимущества SVM.

- Задача квадратичного программирования имеет единственное решение, для нахождения которого разработаны достаточно эффективные методы.
- Автоматически определяется сложность суперпозиции — число нейронов первого слоя, равное числу опорных векторов.
- Максимизация зазора между классами улучшает обобщающую способность.

Недостатки SVM.

- Неустойчивость к шуму в исходных данных. Объекты-выбросы являются опорными и существенно влияют на результат обучения.
- До сих пор не разработаны общие методы подбора ядер под конкретную задачу. На практике «вполне разумные» ядра, построенные с учётом специфики задачи, могут и не обладать свойством положительной определённости.
- Подбор параметра C требует многократного решения задачи.

Метод релевантных векторов (RVM). Ещё одна нетривиальная идея регуляризации заключается в том, что может быть указан вид функциональной зависимости вектора параметров модели w от обучающей выборки и каких-то новых параметров λ. Тогда априорное распределение можно задавать не для w, а для λ. Поясним эту конструкцию на примере метода релевантных векторов (relevancevectormachine, RVM).

Напомним, что в методе опорных векторов (SVM) вектор параметров wявляется линейной комбинацией опорных векторов x_i:

$$w = \sum_{i=1}^{\ell} \lambda_i y_i x_i, \qquad (4.25)$$

где λ_i— неотрицательные двойственные переменные, не равные нулю только для опорных векторов x_i. Один из недостатков SVM состоит в том, что опорными векторами становятся не только пограничные объекты, но и объекты-нарушители, в том числе шумовые выбросы. Метод RVM был предложен как альтернатива, призванная устранить данный дефект.

За основу в RVM берётся формула (4.25), и ставится задача определить, какие из коэффициентов λ_iможно положить равными нулю. Иными словами, делается попытка оптимизировать множество опорных объектов, сохранив свойство разреженности SVM. Для этого предполагается, что λ_i— независимые нормально распределённые случайные величины с неравными дисперсиями α_i:

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

Теперь параметры априорного распределения связываются с объектами, а не с признаками. В результате вместо отбора признаков получаем отбор объектов, однако не

такой, как в SVM, поэтому здесь опорные объекты называют релевантными. Эксперименты показали, что решение получается ещё более разреженным, чем в SVM, то есть релевантных объектов, как правило, существенно меньше, чем опорных. К сожалению, далеко не во всех задачах это действительно приводит к улучшению качества классификации.

ROC-кривая и оптимизация порога решающего правила

$a(x,w) = \text{sign}\big(f(x,w) - w_0\big)$, где $w_0 \in \mathbb{R}$ Рассмотрим задачу классификации на два класса, $Y = \{-1,+1\}$, и модель алгоритмов— аддитивный параметр дискриминантной функции. В теории нейронных сетей его называют порогом активации.

Согласно Теореме 4.2 в случае линейной дискриминантной функции параметр w_0 определяется отношением потерь: $w_0 = \ln \frac{\lambda_-}{\lambda_+}$, где λ_+ и λ_- — величина потери при ошибке на объекте класса «+1» и «−1» соответственно.

На практике отношение потерь может многократно пересматриваться. Поэтому вводится специальная характеристика — ROC-кривая, которая показывает, что происходит с числом ошибок обоих типов, если изменяется отношение потерь.

Термин операционная характеристика приёмника (receiveroperatingcharacteristic, ROCcurve) пришёл из теории обработки сигналов. Эту характеристику впервые ввели во время II мировой войны, после поражения американского военного флота в ПёрлХарборе в 1941 году, когда была осознана проблема повышения точности распознавания самолётов противника по радиолокационному сигналу. Позже нашлись и другие применения: медицинская диагностика, приёмочный контроль качества, кредитный скоринг, предсказание лояльности клиентов, и т.д.

Каждая точка на ROC-кривой соответствует некоторому алгоритму. В общем случае это даже не обязательно кривая — дискретное множество алгоритмов может быть отображено в тех же координатах в виде точечного графика.

По оси X откладывается доля ошибочных положительных классификаций (false positive rate, FPR):

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

Алгоритм 4.2. Эффективный алгоритм построения ROC-кривой

Вход:
обучающая выборка X^ℓ; $f(x) = \langle w, x \rangle$ — дискриминантная функция;

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

1: $\ell_- := \sum_{i=1}^{\ell} [y_i = -1]$ — число объектов класса -1;
$\ell_+ := \sum_{i=1}^{\ell} [y_i = +1]$ — число объектов класса $+1$;
2: упорядочить выборку X^ℓ по убыванию значений $f(x_i)$;
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**
6: сместиться на один шаг вправо:
$\mathrm{FPR}_i := \mathrm{FPR}_{i-1} + \frac{1}{\ell_-}$; $\mathrm{TPR}_i := \mathrm{TPR}_{i-1}$;
$\mathrm{AUC} := \mathrm{AUC} + \frac{1}{\ell_-}\mathrm{TPR}_i$;
7: **иначе**
8: сместиться на один шаг вверх:

Величина 1 − FPR(a) равна доле правильных отрицательных классификаций (truenegativerate, TNR) и называется специфичностью алгоритма a. Поэтому на горизонтальной оси иногда пишут «1 − специфичность».

По оси Y откладывается доля правильных положительных классификаций (truepositiverate, TPR), называемая также чувствительностью алгоритма a:

$$\mathrm{TPR}(a, X^\ell) = \frac{\sum_{i=1}^{\ell} [y_i = +1][a(x_i) = +1]}{\sum_{i=1}^{\ell} [y_i = +1]}.$$

Каждая точка ROC-кривой соответствует определённому значению параметра w_0. ROC-кривая монотонно не убывает и проходит из точки (0,0) в точку (1,1).

Для построения ROC-кривой нет необходимости вычислять FPR и TPR суммированием по всей выборке при каждом w_0. Более эффективный Алгоритм 4.2 основан на простой идее, что в качестве значений порога w_0 достаточно перебрать только

ℓ значений дискриминантной функции $f(x_i)$ = hw,x_ii, которые она принимает на объектах выборки.

Чем выше проходит ROC-кривая, тем выше качество классификации. Идеальная ROC-кривая проходит через левый верхний угол — точку (0,1). Наихудший алгоритм соответствует диагональной прямой, соединяющей точки (0,0) и (1,1); её также изображают на графике как ориентир.

В роли общей характеристики качества классификации, не зависящей от конъюнктурного параметра w_0, выступает площадь под ROC-кривой (areaundercurve, AUC).

Список литературы

[1] Нейроинформатика / А. Н. Горбань, В. Л. Дунин-Барковский, А. Н. Кирдин, Е. М. Миркес, А. Ю. Новоходько, Д. А. Россиев, С. А. Терехов и др. — Новосибирск: Наука, 1998. — 296 с.

[2] Орлов А. И. Нечисловая статистика. — М.: МЗ-Пресс, 2004.

[3] Тихонов А. Н., Арсенин В. Я. Методы решения некорректных задач. — М.: Наука, 1986.

[4] Уиллиамс У. Т., Ланс Д. Н. Методы иерархической классификации // Статистические методы для ЭВМ / Под ред. М. Б. Малютов. — М.: Наука, 1986. — С. 269–301.

[5] Хардле В. Прикладная непараметрическая регрессия. — М.: Мир, 1993.

[6] Шлезингер М., Главач В. Десять лекций по статистическому и структурному распознаванию. — Киев: Наукова думка, 2004.

[7] Шлезингер М. И. О самопроизвольном различении образов // Читающие автоматы. — Киев, Наукова думка, 1965. — Pp. 38–45.

[8] Шурыгин А. М. Прикладная стохастика: робастность, оценивание, прогноз. — М.: Финансы и статистика, 2000.

[9] Яблонский С. В. Введение в дискретную математику. — М.: Наука, 1986.

---

[10] Asuncion A., Newman D. UCI machine learning repository: Tech. rep.: University of California, Irvine, School of Information and Computer Sciences, 2007.
http://www.ics.uci.edu/~mlearn/MLRepository.html.

[11] Bartlett P. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network // IEEE Transactions on Information Theory. — 1998. — Vol. 44, no. 2. — Pp. 525–536. http://discus.anu.edu.au/~bartlett.

[12] Bartlett P., Shawe-Taylor J. Generalization performance of support vector machines and other pattern classifiers // Advances in Kernel Methods. — MIT Press, Cambridge, USA, 1999. — Pp. 43–54. http://citeseer.ist.psu.edu/bartlett98generalization.html.

[13] Bandalos, D. L., and M. R. Boehm-Kaufman. 2009. “Four Common Misconceptions in Exploratory Factor Analysis.” In Statistical and Methodological Myths and Urban Legends, edited by C. E. Lance and R. J. Vandenberg, 61–87. New York: Routledge.

[14] Bates, D. 2005. “Fitting Linear Mixed Models in R.” R News 5 (1). www. r-project.org/doc/Rnews/Rnews_2005-1.pdf.

[15] Breslow, N., and D. Clayton. 1993. “Approximate Inference in Generalized Linear Mixed Models.” Journal of the American Statistical Association 88:9–25.

[16] Bretz, F., T. Hothorn, and P. Westfall. 2010. Multiple Comparisons Using R. Boca Raton, FL: Chapman & Hall.

[17] Canty, A. J. 2002. “Resampling Methods in R: The boot Package.” http:// cran.r-project.org/doc/Rnews/Rnews_2002-3.pdf.

[18] Chambers, J. M. 2008. Software for Data Analysis: Programming with R. New York: Springer.

[19] Cleveland, W. 1981. “LOWESS: A Program for Smoothing Scatter Plots by Robust Locally Weighted Regression.” The American Statistician 35:54.