Л - линейные классификаторы)

Машинное обучение, 20!7

Спасибо К. В. Воронцову, МФТИ, Data Factory Яндекса, O.D.S. и кофеину.

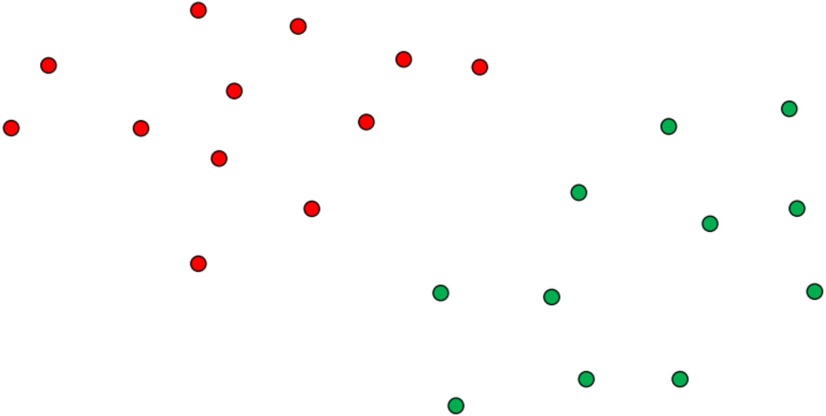
Малютин Е. А.

Содержание

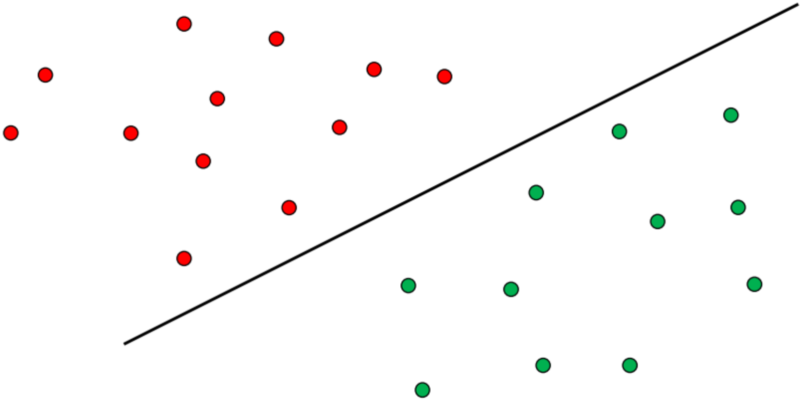
Планчик

- Линейный классификатор
- Линейные классификаторы
- SVM – support vector machine
- Just another kernel trick
- Много SVM – не мало

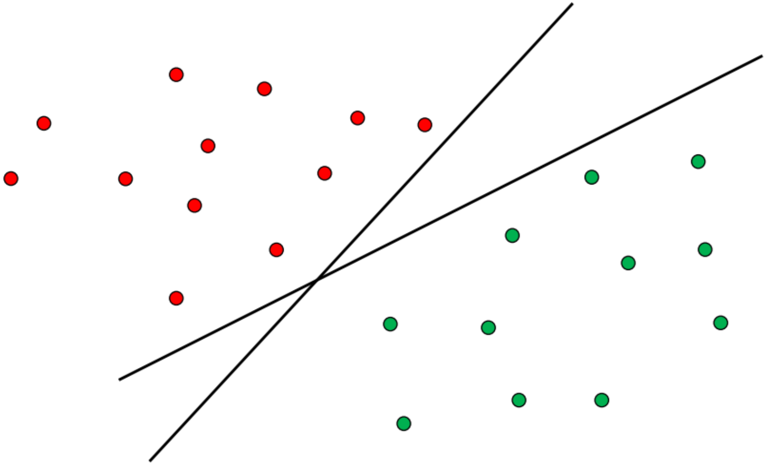
Что такое линейный классификатор?



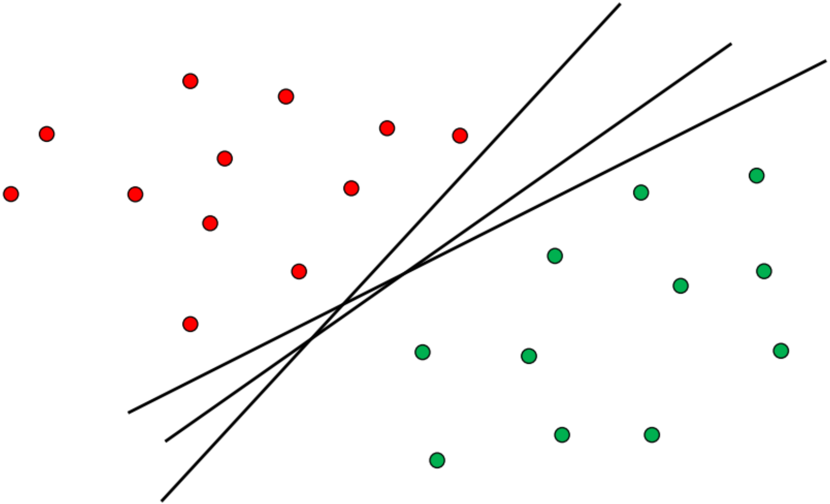
Что такое линейный классификатор?



Что такое линейный классификатор?



Что такое линейный классификатор?



Что такое линейный классификатор?

Рис.: А теперь я объясню мемас

Задача построения разделяющей поверхности

- Задача классификации с двумя классами: -1 и +1
  - Задана обучающая выборка $X^l = (x_i, y_i)_{i=1}^n$
  - $f(x, w)$ – разделяющая (дискриминантная) функция, где w – вектор параметров
  - Будем сторить алгоритм классификации след. образом: $a(x_i) = sign(f(x_i, w))$
- Тогда можно ввести понятие отступа (margin)
  - $M_i(x) = y_i \times f(x_i, w)$
  - $M_i(w) < 0 \rightarrow a(x_i, w)$ ошибается на объекте w_i

Задача минимизации эмпирического...

Правильно, риска!

- Функционал эмпирического риска:

$$Q(w) = \sum_{i=1}^{l} [M_i(w) < 0]$$

Задача минимизации эмпирического...

Правильно, риска!

- Функционал эмпирического риска:

$$Q(w) = \sum_{i=1}^{l} [M_i(w) < 0]$$

- Но решать такую задачу неудобно, вводится аппроксимация:

$$Q(w) = \sum_{i=1}^{l} [M_i(w) < 0] \leq Q(w) = \sum_{i=1}^{n} \mathcal{L}(M_i(w)) \to \min_{w}$$

Задача минимизации эмпирического...

Правильно, риска!

- Функционал эмпирического риска:

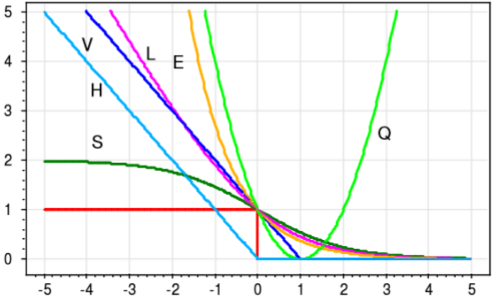
$$Q(w) = \sum_{i=1}^{l} [M_i(w) < 0]$$

- Но решать такую задачу неудобно, вводится аппроксимация:

$$Q(w) = \sum_{i=1}^{l} [M_i(w) < 0] \leq Q(w) = \sum_{i=1}^{n} \mathcal{L}(M_i(w)) \to \min_{w}$$

- Функция потерь $\mathcal{L}$ невозрастающая, неотрицательная

Часто-используемые функции



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

Support Vector Machine

- Будем искать алгоритм в след. виде:

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

Support Vector Machine

- Будем искать алгоритм в след. виде:

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

- Нормируем w, w_0 так: $\langle w_i, x_i \rangle - w_0 = y_i$ – для ближайших к разделяющей гиперплоскости объектов

Support Vector Machine

- Будем искать алгоритм в след. виде:

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

- Нормируем w, w_0 так: $\langle w_i, x_i \rangle - w_0 = y_i$ – для ближайших к разделяющей гиперплоскости объектов
- Тогда условие $-1 < \langle w, x_i \rangle - w_0 < 1$ задаёт полосу, разделяющую классы.

Support Vector Machine

- Будем искать алгоритм в след. виде:

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

- Нормируем w, w_0 так: $\langle w_i, x_i\rangle - w_0 = y_i$ – для ближайших к разделяющей гиперплоскости объектов
- Тогда условие $-1 < \langle w, x_i\rangle - w_0 < 1$ задаёт полосу, разделяющую классы.
- Возьмём две точки x_+, x_- на границе, тогда ширина разделяющей полосы:

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

Support Vector Machine

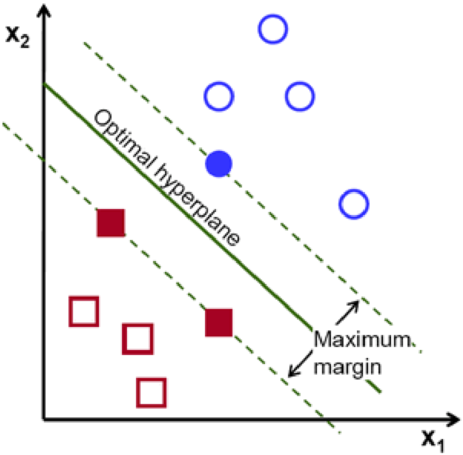


Рис.: Разделяющая полоса

Минимизируем квадратичную форму:

$$\begin{cases} (w, w) \to min \\ y_i(\langle w, x_i \rangle - w_0) \geq 1, i = 1...l \end{cases}$$

По теореме Куна-Таккера эта задача эквивалентна двойственной задаче поиска седловой точки функции Лагранжа:

$$\begin{cases} (\mathcal{L}(w, w_0; X) = \frac{1}{2}(w, w) - \sum\limits_{i=1}^{l} \lambda_i(y_i(\langle w_i, x_i \rangle - w_0) - 1) \to \min\limits_{w_0, w} \max\limits_{\lambda} \\ \lambda_i \geq 0, i = 1..l \\ \lambda_i = 0 \longleftrightarrow (\langle w, x_i \rangle - w_0) = y_i \end{cases}$$

Support Vector Machine

Линейно-разделимая выборка

Необходимым условием седловой точки является равенство нулю производных Лагранжиана. Отсюда немедленно вытекают два полезных соотношения:

$$\frac{\partial \mathcal{L}}{\partial w} = w - \sum_{i}^{l} \lambda_i y_i x_i = 0 \implies w = \sum \lambda_i x_i y_i$$
$$\frac{\partial \mathcal{L}}{\partial w_0} = -\sum \lambda_i y_i \implies \sum \lambda_i y_i = 0$$

Искомый вектор весов w является линейной комбинацией векторов обучающей выборки, причём только тех, для которых $\lambda_i \neq 0$; **Эти вектора - опорные**

Предположим мы решили задачу, как найти w_0? Любой вектор и...

$w_9 = \langle w_i, x_i \rangle - y_i$

А сам алгоритм: $a(x) = sign(\sum_{i=1}^{l} \lambda_i \langle x_i, x_i' \rangle - w0)$

Перепишем квадратичную форму в виде:

$$\begin{cases} \frac{1}{2}(w, w) + C\sum \xi_i \to min \\ y_i(\langle w, x_i \rangle - w_0) \geq 1 - \xi_i, i = 1 \ldots l \\ \xi_i \geq 0 \end{cases}$$

Или в терминах отступов и регуляризации функционал качества:

$$Q(w, X^l) = \sum (1 - m_i)_+ \tau ||w||^2 \to \min_{w, w_0}$$

Support Vector Machine

Линейно-неразделимая выборка

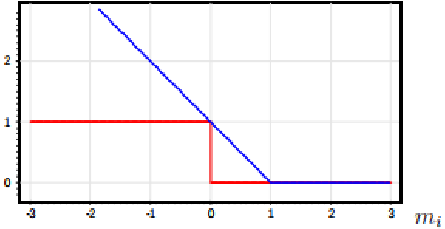


Рис. 1. Кусочно-линейная аппроксимация пороговой функции потерь: $[m_i < 0] \leqslant (1 - m_i)_+$.

Рис.: В случае $Q(w, X^l) = \sum (1 - m_i)_+ \tau ||w||^2 \to \min\limits_{w, w_0}$

Support Vector Machine

Линейно-неразделимая выборка

Давайте подставим всё это в лагран... аналогичные соотношения при допускаемых ошибках:

$$\frac{\partial \mathcal{L}}{\partial w} = w - \sum_{i}^{l} \lambda_i y_i x_i = 0 \Longrightarrow w = \sum \lambda_i x_i y_i$$

$$\frac{\partial \mathcal{L}}{\partial w_0} = -\sum \lambda_i y_i \Longrightarrow \sum \lambda_i y_i = 0$$

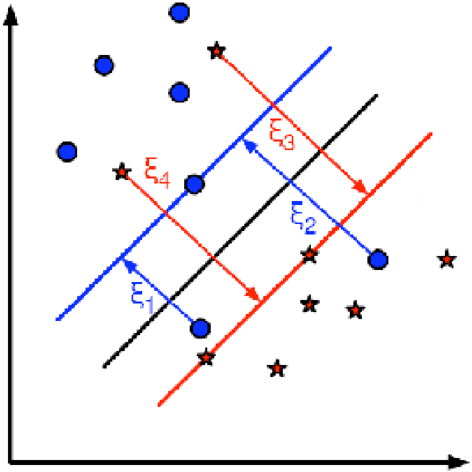
$$\frac{\partial \mathcal{L}}{\partial \xi_i} = -\lambda_i - \eta_i + C = 0 \Longrightarrow \eta_i + \lambda_i = C$$

Отсюда, и из условий дополняющей нежёсткости вытекает, что возможны только три допустимых сочетания значений переменных ξ_i, λ_i, η_i и отступов m_i.

1. $\lambda_i = 0; \eta_i = C; \xi_i = 0; m_i > 1$:
   Объект x_i классифицируется правильно и находится далеко от разделяющей полосы. Такие объекты будем называть *периферийными*.
2. $0 < \lambda_i < C; 0 < \eta_i < C; \xi_i = 0; m_i = 1$:
   Объект x_i классифицируется правильно и лежит в точности на границе разделяющей полосы. Такие объекты, как и раньше, будем называть *опорными*.
3. $\lambda_i = C; \eta_i = 0; \xi_i > 0; m_i < 1$:
   Объект x_i либо 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_i будем называть *нарушителем*.

Support Vector Machine

Линейно-неразделимая выборка



Kernel Trick

$\exists\psi(x) : X \to H$, где в H определено скалярное произведение, а выборка – линейно-разделима.
Тогда всюду в алгоритме: $\langle x, x' \rangle = \langle \psi(x), \psi(x') \rangle$
Ядро: Функция $K : \ X \times X \to R(kernelfunction), K(x, x') = \langle \psi(x), \psi(x') \rangle$ при некотором отображении $\psi : X \to H$, где H — пространство со скалярным произведением.

Теорема Мерсера

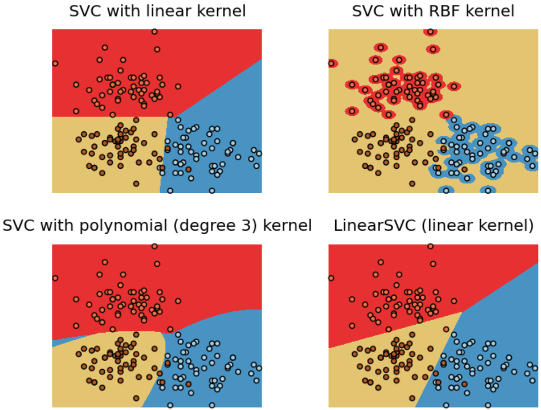
Функция $K(x, x')$ является ядром тогда и толь- ко тогда, когда она симметрична, $K(x, x') = K(x', x)$, и неотрицательно определена:
$\int_X \int_X K(x, x')g(x)g(x')dxdx' \geq 0$ для любой функции $g : X \to R$.

Kernel trick

1. Произвольное скалярное произведение $K(x, x') = \langle x, x' \rangle$ является ядром.
2. Константа $K(x, x') = 1$ – ядро
3. Произведение ядер $K(x, x') = K_1(x, x')K_2(x, x')$ является ядром.
4. Для любой функции $\psi : X \to R$ произведение $K(x, x') = \psi(x)\psi(x')$ является ядром.
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 $\psi : X \to X$ и произвольного ядра K_0 является ядром: $K(x, x') = K_0(\psi(x), \psi(x'))$
7. Если $s : X \times X \to R$ — произвольная симметричная интегрируемая функция, то $K(x, x') = \int_X s(x, z)s(x', z)dz$ является ядром.
8. Предел локально-равномерно сходящейся последовательности ядер является ядром

KKKKernel

9 Композиция произвольного ядра K_0 и произвольной функции $f : R \times R$, представимой в виде сходящегося степенного ряда с неотрицательными коэффициентами $K(x, x') = \int K_0(x, x')$, является ядром.



Резюме

Преимущества

- Вместо многоэкстремальной задачи решается задача квадратичного программирования
- Принцип оптимальной разделяющей гиперплоскости приводит к максимизации ширины разделяющей полосы между классами, следовательно, к более уверенной классификации

Недостатки

- Неустойчив к шуму
- Проблема выбора ядер
- В общем случае, когда линейная разделимость не гарантируется, приходится подбирать управляющий параметр алгоритма C.