Математические модели в морфологии

Автоматическая классификация.

Алексей Сорокин

курс по выбору, ОТИПЛ МГУ,
осенний семестр 2016–2017 учебного года

Введение и постановка задачи

- Автоматическая классификация — отнесение объекта к одному из заранее известных классов на основе его признаков.

Введение и постановка задачи

- Автоматическая классификация — отнесение объекта к одному из заранее известных классов на основе его признаков.
- Примеры задач классификации:
  - Определить морфологическую категорию слова на основании его суффиксов/префиксов.
  - Определить морфологическую категорию слова на основании морфологических категорий предыдущих слов.

Введение и постановка задачи

- Автоматическая классификация — отнесение объекта к одному из заранее известных классов на основе его признаков.
- Примеры задач классификации:
  - Определить морфологическую категорию слова на основании его суффиксов/префиксов.
  - Определить морфологическую категорию слова на основании морфологических категорий предыдущих слов.
  - Определить жанр текста по частотам вхождений отдельных слов и другим признакам.

Введение и постановка задачи

- Автоматическая классификация — отнесение объекта к одному из заранее известных классов на основе его признаков.
- Примеры задач классификации:
  - Определить морфологическую категорию слова на основании его суффиксов/префиксов.
  - Определить морфологическую категорию слова на основании морфологических категорий предыдущих слов.
  - Определить жанр текста по частотам вхождений отдельных слов и другим признакам.
- Необходимые этапы:
  - Перекодировать признаки из произвольной формы в числовую.

Введение и постановка задачи

- Автоматическая классификация — отнесение объекта к одному из заранее известных классов на основе его признаков.
- Примеры задач классификации:
  - Определить морфологическую категорию слова на основании его суффиксов/префиксов.
  - Определить морфологическую категорию слова на основании морфологических категорий предыдущих слов.
  - Определить жанр текста по частотам вхождений отдельных слов и другим признакам.
- Необходимые этапы:
  - Перекодировать признаки из произвольной формы в числовую.
  - Настроить параметры алгоритма с помощью объектов, ответ на которых известен.

Введение и постановка задачи

- Автоматическая классификация — отнесение объекта к одному из заранее известных классов на основе его признаков.
- Примеры задач классификации:
  - Определить морфологическую категорию слова на основании его суффиксов/префиксов.
  - Определить морфологическую категорию слова на основании морфологических категорий предыдущих слов.
  - Определить жанр текста по частотам вхождений отдельных слов и другим признакам.
- Необходимые этапы:
  - Перекодировать признаки из произвольной формы в числовую.
  - Настроить параметры алгоритма с помощью объектов, ответ на которых известен.
  - Применить настроенный алгоритм к новым данным той же природы.

Математическая постановка задачи классификации

- V — пространство объектов,
- C — множество возможных ответов, для задачи классификации $|C| < \infty$.

Математическая постановка задачи классификации

- V — пространство объектов,
- C — множество возможных ответов, для задачи классификации $|C| < \infty$.
- $X_T \subset V$ — обучающая выборка, $X_T = \langle x_1, \ldots, x_t \rangle$.

Математическая постановка задачи классификации

- V — пространство объектов,
- C — множество возможных ответов, для задачи классификации $|C| < \infty$.
- $X_T \subset V$ — обучающая выборка, $X_T = \langle x_1, \ldots, x_t \rangle$.
- $Y_T = \langle y_1, \ldots, y_t \rangle$ — ответы на обучающей выборке, $y_i \in C$.

  Требуется найти такую функцию g из заданного семейства $\mathcal{G}$, которая лучше всего предсказывает ответы на обучающей выборке.

Математическая постановка задачи классификации

- V — пространство объектов,
- C — множество возможных ответов, для задачи классификации $|C| < \infty$.
- $X_T \subset V$ — обучающая выборка, $X_T = \langle x_1, \ldots, x_t \rangle$.
- $Y_T = \langle y_1, \ldots, y_t \rangle$ — ответы на обучающей выборке, $y_i \in C$.

Требуется найти такую функцию g из заданного семейства $\mathcal{G}$, которая лучше всего предсказывает ответы на обучающей выборке. Это позволяет надеяться, что и на других объектах алгоритм будет давать правильную классификацию.

Математическая постановка задачи классификации

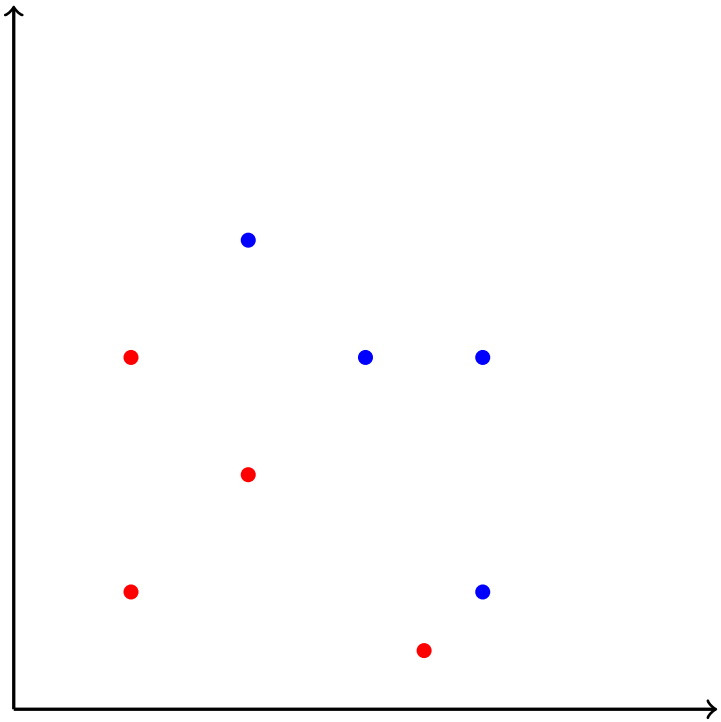
- V — пространство объектов,
- C — множество возможных ответов, для задачи классификации $|C| < \infty$.
- $X_T \subset V$ — обучающая выборка, $X_T = \langle x_1, \ldots, x_t \rangle$.
- $Y_T = \langle y_1, \ldots, y_t \rangle$ — ответы на обучающей выборке, $y_i \in C$.
  Требуется найти такую функцию g из заданного семейства $\mathcal{G}$, которая лучше всего предсказывает ответы на обучающей выборке. Это позволяет надеяться, что и на других объектах алгоритм будет давать правильную классификацию.
- Будем считать, что семейство G параметризовано некоторым параметром $\theta \in \Theta$: $G = \{g_\theta \mid \theta \in \Theta\}$.

Пример

- Пусть $X = \mathbb{R}^2$ (классифицируются точки на плоскости).

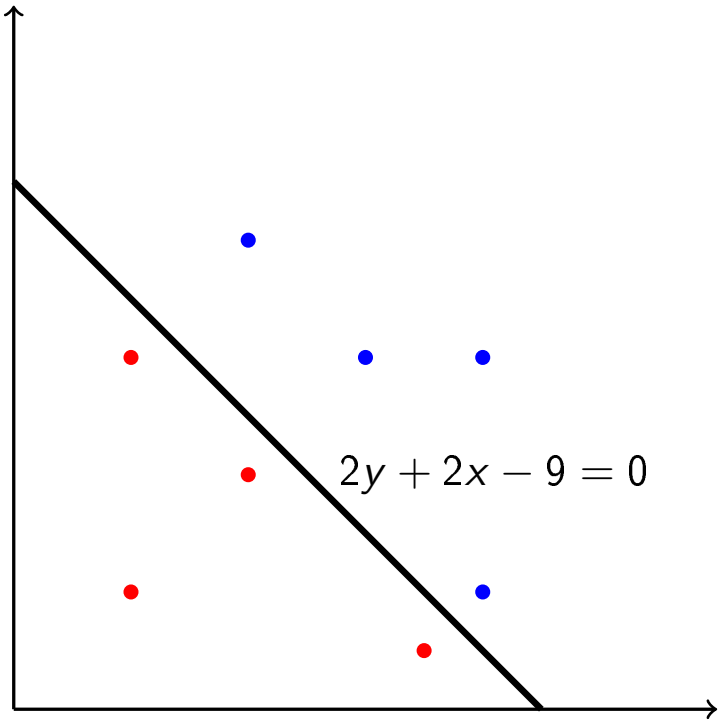
Пример

- Пусть $X = \mathbb{R}^2$ (классифицируются точки на плоскости).
- Предположим, что классы разделяются некоторой прямой:



Пример

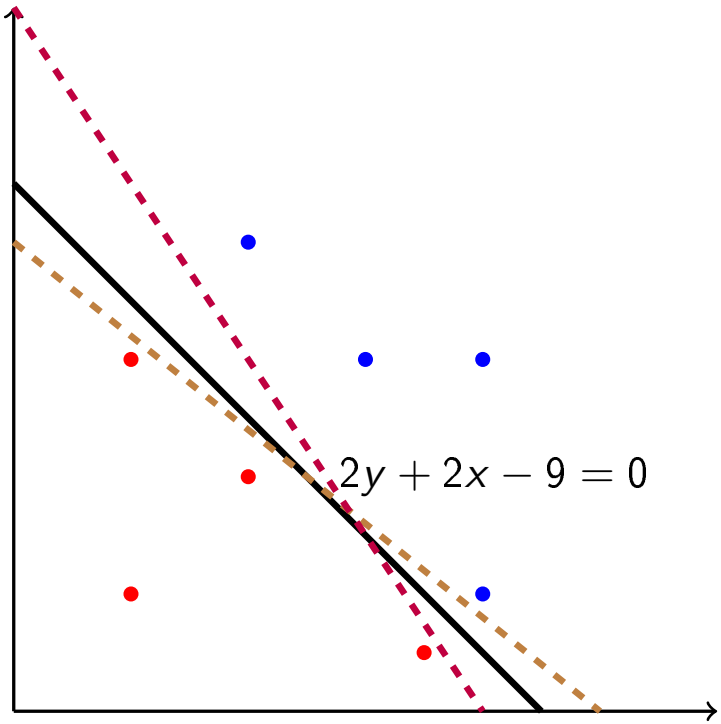
- Пусть $X = \mathbb{R}^2$ (классифицируются точки на плоскости).
- Предположим, что классы разделяются некоторой прямой:



- Тогда параметр алгоритма — уравнение этой прямой.

Пример

- Пусть $X = \mathbb{R}^2$ (классифицируются точки на плоскости).
- Предположим, что классы разделяются некоторой прямой:



- Тогда параметр алгоритма — уравнение этой прямой.

Линейный классификатор: пример

- Линейный классификатор задаёт решающую функцию

$$g(x) = \operatorname{sgn} w_1 x_1 + w_2 x_2 - w_0$$

Линейный классификатор: пример

- Линейный классификатор задаёт решающую функцию

$$g(x) = \operatorname{sgn} w_1 x_1 + w_2 x_2 - w_0$$

- Тогда $G = \{g_{(w_1,w_2,w_0)} | w_0, w_1, w_2 \in \mathbb{R}\}$.

Линейный классификатор: пример

- Линейный классификатор задаёт решающую функцию

$$g(x) = \operatorname{sgn} w_1 x_1 + w_2 x_2 - w_0$$

- Тогда $G = \{g_{(w_1,w_2,w_0)} | w_0, w_1, w_2 \in \mathbb{R}\}$.
- В общем случае линейный классификатор имеет вид

$$g(x) = \operatorname{sgn} (\langle w, x \rangle - w_0),$$

где $\langle u, v \rangle$ — скалярное произведение

- Нужно найти оптимальный вектор весов w и порог w_0.

Линейный классификатор: пример

- Линейный классификатор задаёт решающую функцию

$$g(x) = \operatorname{sgn} w_1 x_1 + w_2 x_2 - w_0$$

- Тогда $G = \{g_{(w_1,w_2,w_0)} | w_0, w_1, w_2 \in \mathbb{R}\}$.
- В общем случае линейный классификатор имеет вид

$$g(x) = \operatorname{sgn} (\langle w, x \rangle - w_0),$$

где $\langle u, v \rangle$ — скалярное произведение

- Нужно найти оптимальный вектор весов w и порог w_0.
- Как это сделать? Зависит от алгоритма!
  - Можно искать наиболее вероятный класс (наивный байесовский классификатор, логистическая регрессия, . . .),

Линейный классификатор: пример

- Линейный классификатор задаёт решающую функцию

$$g(x) = \operatorname{sgn} w_1 x_1 + w_2 x_2 - w_0$$

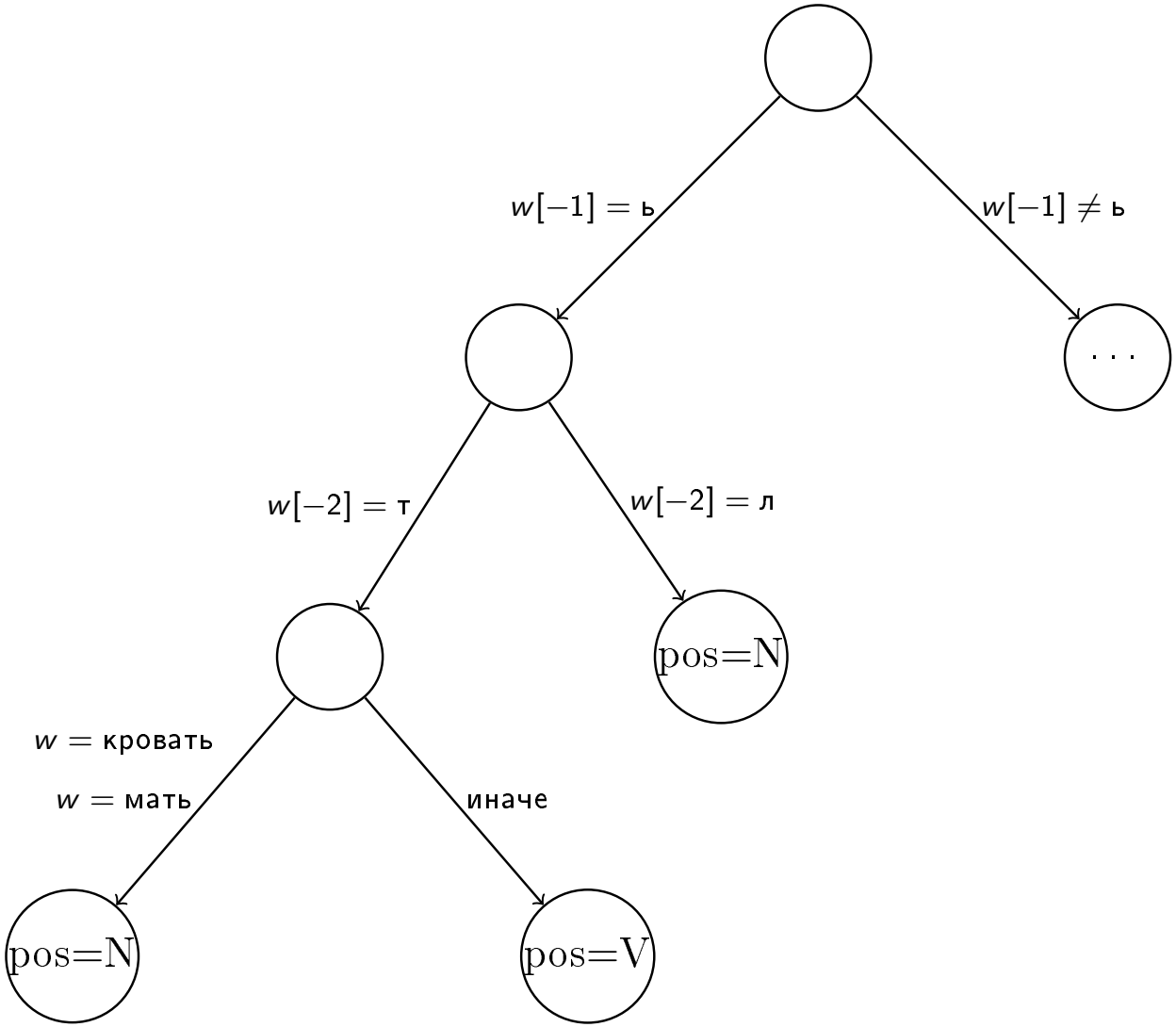
- Тогда $G = \{g_{(w_1,w_2,w_0)} | w_0, w_1, w_2 \in \mathbb{R}\}$.
- В общем случае линейный классификатор имеет вид

$$g(x) = \operatorname{sgn} (\langle w, x \rangle - w_0),$$

где $\langle u, v \rangle$ — скалярное произведение

- Нужно найти оптимальный вектор весов w и порог w_0.
- Как это сделать? Зависит от алгоритма!
  - Можно искать наиболее вероятный класс (наивный байесовский классификатор, логистическая регрессия, . . .),
  - Можно минимизировать функционал качества (например, число ошибок или суммарную ошибку).

Деревья решений



Кодирование признаков

- В большинстве алгоритмов признаки находятся в числовом виде.

Кодирование признаков

- В большинстве алгоритмов признаки находятся в числовом виде.
- Как закодировать, например, суффиксы длины до 5:

Кодирование признаков

- В большинстве алгоритмов признаки находятся в числовом виде.
- Как закодировать, например, суффиксы длины до 5:
- Стандартное решение – индикаторная схема:

	\$a	\$к	\$ка	\$ла	\$ик	\$рка	...
арка	1	0	1	0	0	1	...
школа	1	0	0	1	0	0	...
блик	0	1	0	0	1	0	...
...	...	...	...	...	...	...	...

Кодирование признаков

- В большинстве алгоритмов признаки находятся в числовом виде.
- Как закодировать, например, суффиксы длины до 5:
- Стандартное решение – индикаторная схема:

	$a	$к	$ка	$ла	$ик	$рка	...
арка	1	0	1	0	0	1	...
школа	1	0	0	1	0	0	...
блик	0	1	0	0	1	0	...
...	...	...	...	...	...	...	...

- Каждому возможному суффиксу соответствует один бинарный признак.

Кодирование признаков

- В большинстве алгоритмов признаки находятся в числовом виде.
- Как закодировать, например, суффиксы длины до 5:
- Стандартное решение – индикаторная схема:

	\$а	\$к	\$ка	\$ла	\$ик	\$рка	...
арка	1	0	1	0	0	1	...
школа	1	0	0	1	0	0	...
блик	0	1	0	0	1	0	...
...	...	...	...	...	...	...	...

- Каждому возможному суффиксу соответствует один бинарный признак.
- Признаки с конечным числом значений перекодируются в бинарные: k возможных значений последней буквы передаётся с помощью k индикаторов $[\![w[-1] = \text{а}]\!]$.

Признаки при морфологической классификации

- В морфологической классификации тоже используются индикаторные признаки.
- В случае триграммных моделей признаки зависят от меток и значений двух предыдущих слов:

Признаки при морфологической классификации

- В морфологической классификации тоже используются индикаторные признаки.
- В случае триграммных моделей признаки зависят от меток и значений двух предыдущих слов:
- Возможные признаки:
  - $w_{-1} = \text{that}$: предыдущее слово *that*.

Признаки при морфологической классификации

- В морфологической классификации тоже используются индикаторные признаки.
- В случае триграммных моделей признаки зависят от меток и значений двух предыдущих слов:
- Возможные признаки:
  - $w_{-1} = \text{that}$: предыдущее слово *that*.
  - $\text{POS}_{-1} = \text{N}$: предыдущее слово – существительное.

Признаки при морфологической классификации

- В морфологической классификации тоже используются индикаторные признаки.
- В случае триграммных моделей признаки зависят от меток и значений двух предыдущих слов:
- Возможные признаки:
  - $w_{-1} = \text{that}$: предыдущее слово *that*.
  - $\text{POS}_{-1} = \text{N}$: предыдущее слово – существительное.
  - $\text{POS}_{-2} = \text{PREP}\ \&\&\ \text{POS}_{-1} = \text{ADJ}\ \&\&\ \text{CASE}_{-2} = \text{DAT}$: предыдущее слово — прилагательное, перед ним предлог, требующий дательный падеж.

Признаки при морфологической классификации

- В морфологической классификации тоже используются индикаторные признаки.
- В случае триграммных моделей признаки зависят от меток и значений двух предыдущих слов:
- Возможные признаки:
  - $w_{-1} = \text{that}$: предыдущее слово *that*.
  - $\text{POS}_{-1} = \text{N}$: предыдущее слово – существительное.
  - $\text{POS}_{-2} = \text{PREP}\ \&\&\ \text{POS}_{-1} = \text{ADJ}\ \&\&\ \text{CASE}_{-2} = \text{DAT}$: предыдущее слово — прилагательное, перед ним предлог, требующий дательный падеж.
- Признаков может получиться очень много...

Признаки при морфологической классификации

- В морфологической классификации тоже используются индикаторные признаки.
- В случае триграммных моделей признаки зависят от меток и значений двух предыдущих слов:
- Возможные признаки:
  - $w_{-1} = \text{that}$: предыдущее слово *that*.
  - $\text{POS}_{-1} = \text{N}$: предыдущее слово – существительное.
  - $\text{POS}_{-2} = \text{PREP}$ && $\text{POS}_{-1} = \text{ADJ}$ && $\text{CASE}_{-2} = \text{DAT}$: предыдущее слово — прилагательное, перед ним предлог, требующий дательный падеж.
- Признаков может получиться очень много...
- Выход: либо задавать признаки с учётом морфологии языка, либо ограничивать учитываемые показатели.

Постановка задачи

- Пусть $X \subseteq \mathbb{R}^n$ — множество объектов,
  $C = [c_1, \ldots, c_m]$ — множество возможных классов.
- Будем находить наиболее вероятный класс:

$$g(x) = \text{argmax}_j \, p(c_j|x)$$

.

Постановка задачи

- Пусть $X \subseteq \mathbb{R}^n$ — множество объектов,
  $C = [c_1, \ldots, c_m]$ — множество возможных классов.
- Будем находить наиболее вероятный класс:

$$g(x) = \text{argmax}_j \, p(c_j|x)$$

.

- Применим правило Байеса

$$\begin{aligned} g(x) &= \text{argmax}_j \, p(c_j|x) = \text{argmax}_j \, \frac{p(x|c_j)p(c_j)}{p(x)} \\ &= \text{argmax}_j \, p(x|c_j)p(c_j) \end{aligned}$$

Постановка задачи

- Пусть $X \subseteq \mathbb{R}^n$ — множество объектов,
  $C = [c_1, \ldots, c_m]$ — множество возможных классов.
- Будем находить наиболее вероятный класс:

$$g(x) = \text{argmax}_j \, p(c_j|x)$$

.

- Применим правило Байеса

$$\begin{aligned} g(x) &= \text{argmax}_j \, p(c_j|x) = \text{argmax}_j \, \frac{p(x|c_j)p(c_j)}{p(x)} \\ &= \text{argmax}_j \, p(x|c_j)p(c_j) \end{aligned}$$

- Будем считать, что все признаки $x_1, \ldots, x_n$ объекта x независимы.

Постановка задачи

- Пусть $X \subseteq \mathbb{R}^n$ — множество объектов,
  $C = [c_1, \ldots, c_m]$ — множество возможных классов.
- Будем находить наиболее вероятный класс:

$$g(x) = \text{argmax}_j \, p(c_j|x)$$

.

- Применим правило Байеса

$$\begin{aligned} g(x) &= \text{argmax}_j \, p(c_j|x) = \text{argmax}_j \, \frac{p(x|c_j)p(c_j)}{p(x)} \\ &= \text{argmax}_j \, p(x|c_j)p(c_j) \end{aligned}$$

- Будем считать, что все признаки $x_1, \ldots, x_n$ объекта x независимы.
- То есть, например, независимы вхождения различных ключевых слов в текст, символов в текст и т. д.

Постановка задачи

- Пусть $X \subseteq \mathbb{R}^n$ — множество объектов,
  $C = [c_1, \ldots, c_m]$ — множество возможных классов.
- Будем находить наиболее вероятный класс:

$$g(x) = \text{argmax}_j \, p(c_j|x)$$

.

- Применим правило Байеса

$$\begin{aligned} g(x) &= \text{argmax}_j \, p(c_j|x) = \text{argmax}_j \, \frac{p(x|c_j)p(c_j)}{p(x)} \\ &= \text{argmax}_j \, p(x|c_j)p(c_j) \end{aligned}$$

- Будем считать, что все признаки $x_1, \ldots, x_n$
  объекта x независимы.
- То есть, например, независимы вхождения различных ключевых слов в текст, символов в текст и т. д.
- Тогда получим $g(x) = \text{argmax}_j(\prod_i p(x_i|c_j))p(c_j)$
- То есть нужно найти вероятности значений признаков для текущего класса.

Признаки в наивном байесовском классификаторе

- Нужно оценить вероятности $p(x_i|c_j)$.
- Для этого нужно понять природу самих признаков x_i.

Признаки в наивном байесовском классификаторе

- Нужно оценить вероятности $p(x_i|c_j)$.
- Для этого нужно понять природу самих признаков x_i.
- Многомерный байесовский классификатор:
  - $x_i \in [0, 1]$ — индикатор вхождения слова w_i (i-го слова в словаре) в текст x.

Признаки в наивном байесовском классификаторе

- Нужно оценить вероятности $p(x_i|c_j)$.
- Для этого нужно понять природу самих признаков x_i.
- Многомерный байесовский классификатор:
  - $x_i \in [0,1]$ — индикатор вхождения слова w_i (i-го слова в словаре) в текст x.
  - $p(x_i = 1|c_j) \approx \frac{N_{ij}}{N_j}$, где

    N_{ij} — количество текстов, содержащих слово w_i, среди текстов класса c_j,

    N_j — количество текстов класса c_j.

Признаки в наивном байесовском классификаторе

- Нужно оценить вероятности $p(x_i|c_j)$.
- Для этого нужно понять природу самих признаков x_i.
- Многомерный байесовский классификатор:
  - $x_i \in [0,1]$ — индикатор вхождения слова w_i (i-го слова в словаре) в текст x.
  - $p(x_i = 1|c_j) \approx \frac{N_{ij}}{N_j}$, где

    N_{ij} — количество текстов, содержащих слово w_i, среди текстов класса c_j,

    N_j — количество текстов класса c_j.
- Нужно ввести поправку на нулевые вероятности.

Многомерный байесовский классификатор

- Уточнённая формула:

$$p(x_i = 1|c_j) = \frac{N_{ij} + \alpha}{N_j + 2\alpha}$$

$$p(x_i = 0|c_j) = \frac{N_{\bar{i}j} + \alpha}{N_j + 2\alpha}$$

Многомерный байесовский классификатор

- Уточнённая формула:

$$p(x_i = 1|c_j) = \frac{N_{ij} + \alpha}{N_j + 2\alpha}$$

$$p(x_i = 0|c_j) = \frac{N_{\bar{i}j} + \alpha}{N_j + 2\alpha}$$

- $p(c_j)$ оценивается аналогично:

$$p(c_j) = \frac{N_j}{N}$$

N_j — количество текстов класса c_j,

N — суммарное количество текстов

Многомерный байесовский классификатор

- Уточнённая формула:

$$p(x_i = 1|c_j) = \frac{N_{ij} + \alpha}{N_j + 2\alpha}$$

$$p(x_i = 0|c_j) = \frac{N_{\bar{i}j} + \alpha}{N_j + 2\alpha}$$

- $p(c_j)$ оценивается аналогично:

$$p(c_j) = \frac{N_j}{N}$$

N_j — количество текстов класса c_j,

N — суммарное количество текстов

- Недостатки наивного байесовского классификатора:
  - Не учитывается число вхождений отдельного слова.

Многомерный байесовский классификатор

- Уточнённая формула:

$$p(x_i = 1|c_j) = \frac{N_{ij} + \alpha}{N_j + 2\alpha}$$

$$p(x_i = 0|c_j) = \frac{N_{\bar{i}j} + \alpha}{N_j + 2\alpha}$$

- $p(c_j)$ оценивается аналогично:

$$p(c_j) = \frac{N_j}{N}$$

N_j — количество текстов класса c_j,

N — суммарное количество текстов

- Недостатки наивного байесовского классификатора:
  - Не учитывается число вхождений отдельного слова.
  - Негативные вероятности могут “перевесить” положительные (большинство слов словаря в текст не входят, их вероятности “невхождения” сильнее повлияют на произведение).

Мультиномиальный байесовский классификатор

- Мультиномиальный байесовский классификатор:

Мультиномиальный байесовский классификатор

- Мультиномиальный байесовский классификатор:
  $x_i \in \mathbb{N}$ — число вхождений слова w_i в текст x.

Мультиномиальный байесовский классификатор

- Мультиномиальный байесовский классификатор:

$$\begin{aligned} x_i \in \mathbb{N} &\quad — \quad \text{число вхождений слова } w_i \text{ в текст } x. \\ p(x_i = n_i | c_j) &\approx p_{ij}^{n_i} \\ p_i j &= \frac{L_{ij} + \alpha}{L_i + \alpha |D|} \end{aligned}$$

Мультиномиальный байесовский классификатор

- Мультиномиальный байесовский классификатор:

$$
\begin{aligned}
x_i \in \mathbb{N} &\quad — \quad \text{число вхождений слова } w_i \text{ в текст } x. \\
p(x_i = n_i | c_j) &\quad \approx \quad p_{ij}^{n_i} \\
p_i j &\quad = \quad \frac{L_{ij} + \alpha}{L_i + \alpha |D|} \\
L_{ij} &\quad — \quad \text{число вхождений } w_i \text{ в тексты класса } c_j \\
L_j &\quad — \quad \text{суммарное число слов в текстах класса } c_j \\
|D| &\quad — \quad \text{размер словаря}
\end{aligned}
$$

Мультиномиальный байесовский классификатор

- Мультиномиальный байесовский классификатор:

$$
\begin{aligned}
x_i \in \mathbb{N} &\quad — \quad \text{число вхождений слова } w_i \text{ в текст } x. \\
p(x_i = n_i | c_j) &\approx p_{ij}^{n_i} \\
p_i j &= \frac{L_{ij} + \alpha}{L_i + \alpha |D|} \\
L_{ij} &\quad — \quad \text{число вхождений } w_i \text{ в тексты класса } c_j \\
L_j &\quad — \quad \text{суммарное число слов в текстах класса } c_j \\
|D| &\quad — \quad \text{размер словаря}
\end{aligned}
$$

- Преимущества мультиномиального классификатора:
  - Учитывается количество вхождений слова.

Мультиномиальный байесовский классификатор

- Мультиномиальный байесовский классификатор:

$$
\begin{aligned}
x_i \in \mathbb{N} &\quad — \quad \text{число вхождений слова } w_i \text{ в текст } x. \\
p(x_i = n_i | c_j) &\approx p_{ij}^{n_i} \\
p_i j &= \frac{L_{ij} + \alpha}{L_i + \alpha |D|} \\
L_{ij} &\quad — \quad \text{число вхождений } w_i \text{ в тексты класса } c_j \\
L_j &\quad — \quad \text{суммарное число слов в текстах класса } c_j \\
|D| &\quad — \quad \text{размер словаря}
\end{aligned}
$$

- Преимущества мультиномиального классификатора:
  - Учитывается количество вхождений слова.
  - Не учитывается отрицательная информация.

Байесовский классификатор и униграммная модель

- При подсчёте вероятности текста $p(x|c_j)$ мультиномиальный классификатор перемножает вероятности всех слов в тексте.

Байесовский классификатор и униграммная модель

- При подсчёте вероятности текста $p(x|c_j)$ мультиномиальный классификатор перемножает вероятности всех слов в тексте.
- То есть байесовский классификатор пытается понять, какая униграммная модель лучше описывает текст.
- Множитель $p(c_j)$ — поправка на вероятность модели (некоторые модели более вероятны, чем другие).

Недостатки байесовских методов

- Достоинства байесовских методов:
  - Простота модели и отсутствие настраиваемых параметров.
  - Быстрота обучения модели.

Недостатки байесовских методов

- Достоинства байесовских методов:
  - Простота модели и отсутствие настраиваемых параметров.
  - Быстрота обучения модели.
  - Высокие результаты при независимости признаков (и даже иногда в её отсутствие).

Недостатки байесовских методов

- Достоинства байесовских методов:
  - Простота модели и отсутствие настраиваемых параметров.
  - Быстрота обучения модели.
  - Высокие результаты при независимости признаков (и даже иногда в её отсутствие).
  - Легко обобщается на несколько классов.

Недостатки байесовских методов

- Достоинства байесовских методов:
  - Простота модели и отсутствие настраиваемых параметров.
  - Быстрота обучения модели.
  - Высокие результаты при независимости признаков (и даже иногда в её отсутствие).
  - Легко обобщается на несколько классов.
- Недостатки байесовских методов:
  - Независимых признаков не бывает!
  - Различные вхождения одного и того же слова зависят друг от друга (часто вместо признаков берут их логарифмы).

Недостатки байесовских методов

- Достоинства байесовских методов:
  - Простота модели и отсутствие настраиваемых параметров.
  - Быстрота обучения модели.
  - Высокие результаты при независимости признаков (и даже иногда в её отсутствие).
  - Легко обобщается на несколько классов.
- Недостатки байесовских методов:
  - Независимых признаков не бывает!
  - Различные вхождения одного и того же слова зависят друг от друга (часто вместо признаков берут их логарифмы).
  - Сложно комбинировать признаки различной природы.

Недостатки байесовских методов

- Достоинства байесовских методов:
  - Простота модели и отсутствие настраиваемых параметров.
  - Быстрота обучения модели.
  - Высокие результаты при независимости признаков (и даже иногда в её отсутствие).
  - Легко обобщается на несколько классов.
- Недостатки байесовских методов:
  - Независимых признаков не бывает!
  - Различные вхождения одного и того же слова зависят друг от друга (часто вместо признаков берут их логарифмы).
  - Сложно комбинировать признаки различной природы.
  - Априорные вероятности классов зависят от сбалансированности выборки (часто P_j не учитывают).

Общее описание

- Линейные методы классификации — классификаторы с решающим правилом

$$g(x) = \text{sgn}\left(\sum_i w_i x_i - w_0\right) = \text{sgn}\left(\langle w, x\rangle - w_0\right),\ w \in \mathbb{R}^n, w_0 \in \mathbb{R}$$

- Пока считаем, что классов всего два, 1 и -1.

Общее описание

- Линейные методы классификации — классификаторы с решающим правилом

$$g(x) = \text{sgn}\left(\sum_i w_i x_i - w_0\right) = \text{sgn}\left(\langle w, x\rangle - w_0\right),\ w \in \mathbb{R}^n, w_0 \in \mathbb{R}$$

- Пока считаем, что классов всего два, 1 и -1.
- Если считать, что среди признаков есть константа, то w_0 можно не добавлять.

Общее описание

- Линейные методы классификации — классификаторы с решающим правилом

$$g(x) = \text{sgn}\left(\sum_i w_i x_i - w_0\right) = \text{sgn}\left(\langle w, x\rangle - w_0\right),\ w \in \mathbb{R}^n, w_0 \in \mathbb{R}$$

- Пока считаем, что классов всего два, 1 и -1.
- Если считать, что среди признаков есть константа, то w_0 можно не добавлять.
- Отличие между алгоритмами: метод подбора оптимального вектора весов.

Общее описание

- Линейные методы классификации — классификаторы с решающим правилом

$$g(x) = \operatorname{sgn}\left(\sum_i w_i x_i - w_0\right) = \operatorname{sgn}\left(\langle w, x\rangle - w_0\right),\ w \in \mathbb{R}^n, w_0 \in \mathbb{R}$$

- Пока считаем, что классов всего два, 1 и -1.
- Если считать, что среди признаков есть константа, то w_0 можно не добавлять.
- Отличие между алгоритмами: метод подбора оптимального вектора весов.
- Возможные затруднения:
  - Большая размерность пространства признаков.

Общее описание

- Линейные методы классификации — классификаторы с решающим правилом

$$g(x) = \operatorname{sgn}\left(\sum_i w_i x_i - w_0\right) = \operatorname{sgn}\left(\langle w, x\rangle - w_0\right),\ w \in \mathbb{R}^n, w_0 \in \mathbb{R}$$

- Пока считаем, что классов всего два, 1 и -1.
- Если считать, что среди признаков есть константа, то w_0 можно не добавлять.
- Отличие между алгоритмами: метод подбора оптимального вектора весов.
- Возможные затруднения:
  - Большая размерность пространства признаков.
  - При этом большинство признаков каждого объекта — нулевые (разреженность).

Общее описание

- Линейные методы классификации — классификаторы с решающим правилом

$$g(x) = \operatorname{sgn}\left(\sum_i w_i x_i - w_0\right) = \operatorname{sgn}\left(\langle w, x\rangle - w_0\right),\ w \in \mathbb{R}^n, w_0 \in \mathbb{R}$$

- Пока считаем, что классов всего два, 1 и -1.
- Если считать, что среди признаков есть константа, то w_0 можно не добавлять.
- Отличие между алгоритмами: метод подбора оптимального вектора весов.
- Возможные затруднения:
  - Большая размерность пространства признаков.
  - При этом большинство признаков каждого объекта — нулевые (разреженность).
  - Зачастую решение ищется приближённо.

Наивный байесовский классификатор как линейный

- Наивный байесовский классификатор — тоже линейный!
- Пусть $C = \{-1, 1\}$, тогда решающее правило имеет вид:

$$
\begin{aligned}
g(x) = 1 \quad &\Leftrightarrow \quad p(1|x) \geqslant p(-1|x) \\
&\Leftrightarrow \quad \log p(1|x) \geqslant \log p(-1|x) \\
&\Leftrightarrow \quad \log \Big(\prod_i p(x_i|1)p(1)\Big) \geqslant \log \Big(\prod_i p(x_i|-1)p(-1)\Big) \\
&\Leftrightarrow \quad \log \Big(\prod_i p_{i,1}^{n_i} p(1)\Big) \geqslant \log \Big(\prod_i p_{i,-1}^{n_i} p(-1)\Big) \\
&\Leftrightarrow \quad \sum_i n_i \log p_{i,1} + \log p(1) \geqslant \sum_i n_i \log p_{i,-1} + \log p(-1) \\
&\Leftrightarrow \quad \sum_i (\log p_{i,1} - \log p_{i,-1}) n_i + (\log p(1) - \log p(-1)) \geqslant 0
\end{aligned}
$$

Наивный байесовский классификатор как линейный

- Наивный байесовский классификатор — тоже линейный!
- Пусть $C = \{-1, 1\}$, тогда решающее правило имеет вид:

$$
\begin{aligned}
g(x) = 1 \quad &\Leftrightarrow \quad p(1|x) \geqslant p(-1|x) \\
&\Leftrightarrow \quad \log p(1|x) \geqslant \log p(-1|x) \\
&\Leftrightarrow \quad \log \Big(\prod_i p(x_i|1)p(1)\Big) \geqslant \log \Big(\prod_i p(x_i|-1)p(-1)\Big) \\
&\Leftrightarrow \quad \log \Big(\prod_i p_{i,1}^{n_i} p(1)\Big) \geqslant \log \Big(\prod_i p_{i,-1}^{n_i} p(-1)\Big) \\
&\Leftrightarrow \quad \sum_i n_i \log p_{i,1} + \log p(1) \geqslant \sum_i n_i \log p_{i,-1} + \log p(-1) \\
&\Leftrightarrow \quad \sum_i (\log p_{i,1} - \log p_{i,-1}) n_i + (\log p(1) - \log p(-1)) \geqslant 0
\end{aligned}
$$

- То есть можно считать

  $w_i = \log p_{i,1} - \log p_{i,-1}, \quad w_0 = \log p(-1) - \log p(1).$

Наивный байесовский классификатор как линейный

- Наивный байесовский классификатор — тоже линейный!
- Пусть $C = \{-1, 1\}$, тогда решающее правило имеет вид:

$$
\begin{aligned}
g(x) = 1 \quad &\Leftrightarrow \quad p(1|x) \geqslant p(-1|x) \\
&\Leftrightarrow \quad \log p(1|x) \geqslant \log p(-1|x) \\
&\Leftrightarrow \quad \log \Big(\prod_i p(x_i|1)p(1)\Big) \geqslant \log \Big(\prod_i p(x_i|-1)p(-1)\Big) \\
&\Leftrightarrow \quad \log \Big(\prod_i p_{i,1}^{n_i} p(1)\Big) \geqslant \log \Big(\prod_i p_{i,-1}^{n_i} p(-1)\Big) \\
&\Leftrightarrow \quad \sum_i n_i \log p_{i,1} + \log p(1) \geqslant \sum_i n_i \log p_{i,-1} + \log p(-1) \\
&\Leftrightarrow \quad \sum_i (\log p_{i,1} - \log p_{i,-1}) n_i + (\log p(1) - \log p(-1)) \geqslant 0
\end{aligned}
$$

- То есть можно считать

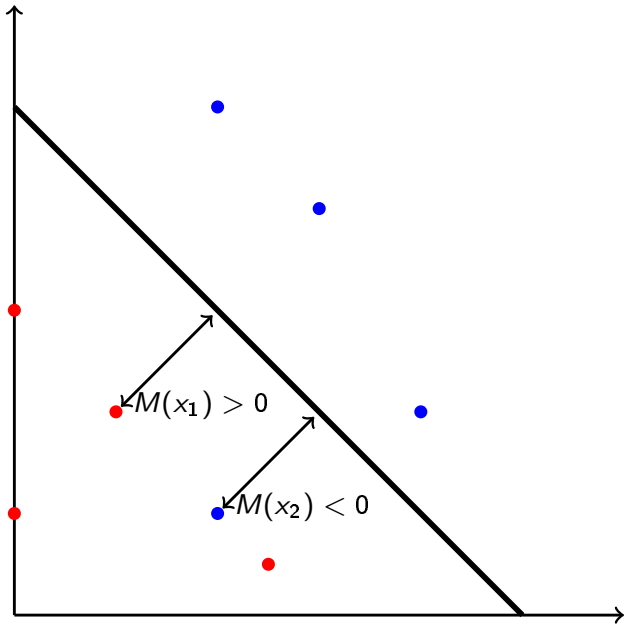
  $w_i = \log p_{i,1} - \log p_{i,-1}, \quad w_0 = \log p(-1) - \log p(1).$
- Другие алгоритмы считают веса по-другому...
- Они либо следуют другой вероятностной модели, либо пытаются

Линейная разделимость

- Пусть задан классификатор $g(x) = \text{sgn}\left(\langle w, x\rangle - w_0\right)$.

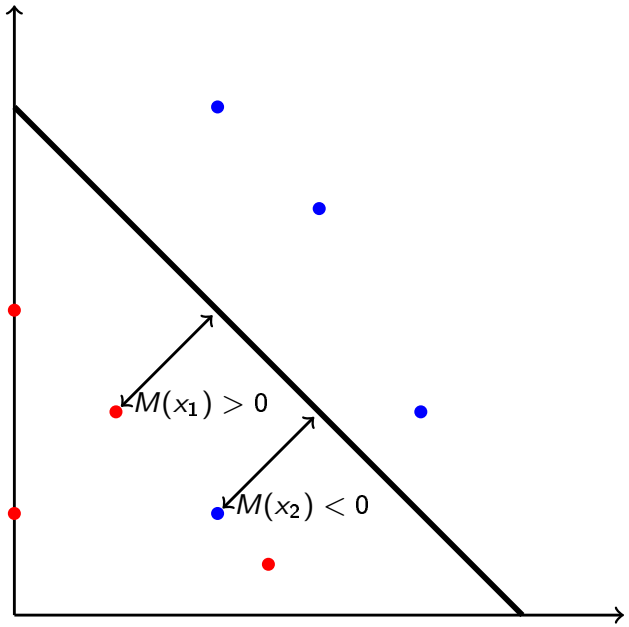
Линейная разделимость

- Пусть задан классификатор $g(x) = \text{sgn}(\langle w, x \rangle - w_0)$.
- Отступ объекта: $M(x) = g(x)y(x)$, где $y(x) \in \{-1, 1\}$ — класс объекта x. Отступ пропорционален расстоянию от разделяющей плоскости.



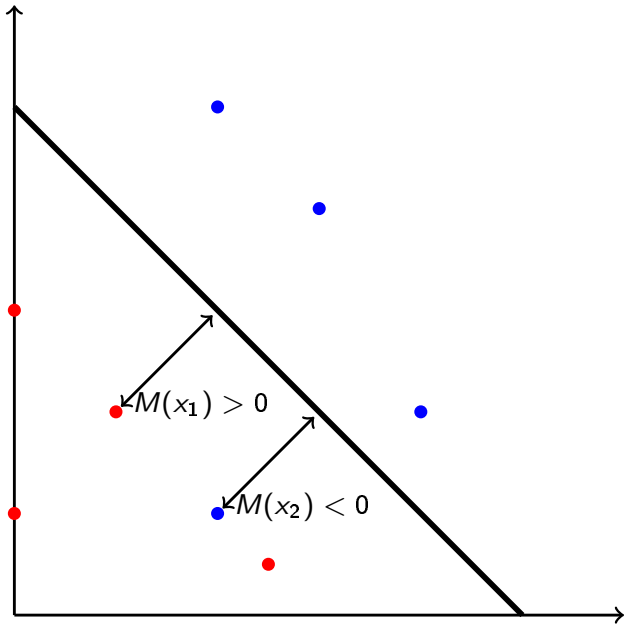
Линейная разделимость

- Пусть задан классификатор $g(x) = \text{sgn}(\langle w, x \rangle - w_0)$.
- Отступ объекта: $M(x) = g(x)y(x)$, где $y(x) \in \{-1, 1\}$ — класс объекта x. Отступ пропорционален расстоянию от разделяющей плоскости.



Линейная разделимость

- Пусть задан классификатор $g(x) = \text{sgn}(\langle w, x \rangle - w_0)$.
- Отступ объекта: $M(x) = g(x)y(x)$, где $y(x) \in \{-1, 1\}$ — класс объекта x. Отступ пропорционален расстоянию от разделяющей плоскости.



- $M(x) > 0$ — объект классифицируется правильно, $M(x) < 0$ — на объекте допущена ошибка.
- Чем меньше отступ, тем хуже классифицирован объект.

Персептрон Розенблатта

- Простейший линейный метод — персептрон Розенблатта.

Персептрон Розенблатта

- Простейший линейный метод — персептрон Розенблатта.
- Алгоритм настройки вектора весов w:

Алгоритм 2 Алгоритм обучения персептрона Розенблатта.

Вход: Обучающая выборка $X^L = \subset \mathbb{R}^n$, вектор ответов $Y^L \in \{-1, 1\}^L$,
шаг обучения $\eta > 0$, число итераций T.
Выход: Вектор весов $w \in \mathbb{R}^n$.

```
Инициализировать w случайный образом или нулями.
for t = 1, ..., T do
    for l = 1, ..., L do
        if (w, x^l) y^l ⩽ 0 then
            Положить w ← w + η x^l y^l.
        end if
    end for
    if не было ошибок на данной итерации then
        break
    end if
end for
```

Персептрон Розенблатта

Линейная разделимость

Выборка $X \subset \mathbb{R}^n$ линейно разделима с порогом δ, если
$\exists w \in \mathbb{R}^n (\|w\| = 1 \wedge (\forall l = 1, \ldots, |X| \; (\langle w, x^l \rangle y^l \geqslant \delta)))$

Персептрон Розенблатта

Линейная разделимость

Выборка $X \subset \mathbb{R}^n$ линейно разделима с порогом δ, если
$\exists w \in \mathbb{R}^n (\|w\| = 1 \wedge (\forall l = 1, \ldots, |X| \; (\langle w, x^l \rangle y^l \geqslant \delta)))$

Теорема

Если выборка линейно разделима, то персептрон находит разделяющую гиперплоскость за конечное число шагов.

Персептрон Розенблатта

Линейная разделимость

Выборка $X \subset \mathbb{R}^n$ линейно разделима с порогом δ, если
$\exists w \in \mathbb{R}^n (\|w\| = 1 \wedge (\forall l = 1, \ldots, |X| \; (\langle w, x^l \rangle y^l \geqslant \delta)))$

Теорема

Если выборка линейно разделима, то персептрон находит разделяющую гиперплоскость за конечное число шагов.

- Наличие разделимости нельзя проверить заранее.

Персептрон Розенблатта

Линейная разделимость

Выборка $X \subset \mathbb{R}^n$ линейно разделима с порогом δ, если
$\exists w \in \mathbb{R}^n (\|w\| = 1 \wedge (\forall l = 1, \ldots, |X| \; (\langle w, x^l \rangle y^l \geqslant \delta)))$

Теорема

Если выборка линейно разделима, то персептрон находит разделяющую гиперплоскость за конечное число шагов.

- Наличие разделимости нельзя проверить заранее.
- В случае неразделимости ничего не гарантируется.

Персептрон Розенблатта

Линейная разделимость

Выборка $X \subset \mathbb{R}^n$ линейно разделима с порогом δ, если
$\exists w \in \mathbb{R}^n (\|w\| = 1 \wedge (\forall l = 1, \ldots, |X| \; (\langle w, x^l \rangle y^l \geqslant \delta)))$

Теорема

Если выборка линейно разделима, то персептрон находит разделяющую гиперплоскость за конечное число шагов.

- Наличие разделимости нельзя проверить заранее.
- В случае неразделимости ничего не гарантируется.
- Число шагов может быть достаточно большим.

Персептрон Розенблатта

Линейная разделимость

Выборка $X \subset \mathbb{R}^n$ линейно разделима с порогом δ, если
$\exists w \in \mathbb{R}^n (\|w\| = 1 \wedge (\forall l = 1, \ldots, |X| \; (\langle w, x^l \rangle y^l \geqslant \delta)))$

Теорема

Если выборка линейно разделима, то персептрон находит разделяющую гиперплоскость за конечное число шагов.

- Наличие разделимости нельзя проверить заранее.
- В случае неразделимости ничего не гарантируется.
- Число шагов может быть достаточно большим.
- Разделяющая плоскость для обучающей выборки может не подойти для контрольной.

Персептрон Розенблатта

- Недостатки персептрона Розенблатта:
  - Не гарантировано качество в случае отсутствия разделимости.

Персептрон Розенблатта

- Недостатки персептрона Розенблатта:
  - Не гарантировано качество в случае отсутствия разделимости.
  - Найденная плоскость может быть неоптимальна.

Персептрон Розенблатта

- Недостатки персептрона Розенблатта:
  - Не гарантировано качество в случае отсутствия разделимости.
  - Найденная плоскость может быть неоптимальна.
  - Алгоритм склонен к переобучению.
- Преимущества персептрона Розенблатта:
  - Простота алгоритма, отсутствие параметров.

Персептрон Розенблатта

- Недостатки персептрона Розенблатта:
  - Не гарантировано качество в случае отсутствия разделимости.
  - Найденная плоскость может быть неоптимальна.
  - Алгоритм склонен к переобучению.
- Преимущества персептрона Розенблатта:
  - Простота алгоритма, отсутствие параметров.
  - Легко адаптируется под более сложные модели.

Персептрон Розенблатта

- Недостатки персептрона Розенблатта:
  - Не гарантировано качество в случае отсутствия разделимости.
  - Найденная плоскость может быть неоптимальна.
  - Алгоритм склонен к переобучению.
- Преимущества персептрона Розенблатта:
  - Простота алгоритма, отсутствие параметров.
  - Легко адаптируется под более сложные модели.
  - Существуют способы борьбы с переобучением (averaged margin perceptron):
    - Брать средний вектор весов по всем итерациям.
    - Обновлять вектор весов для всех объектов, чей отступ не превышает δ.

Персептрон Розенблатта

- Недостатки персептрона Розенблатта:
  - Не гарантировано качество в случае отсутствия разделимости.
  - Найденная плоскость может быть неоптимальна.
  - Алгоритм склонен к переобучению.
- Преимущества персептрона Розенблатта:
  - Простота алгоритма, отсутствие параметров.
  - Легко адаптируется под более сложные модели.
  - Существуют способы борьбы с переобучением (averaged margin perceptron):
    - Брать средний вектор весов по всем итерациям.
    - Обновлять вектор весов для всех объектов, чей отступ не превышает δ.
- Персептрон особенно популярен для подбора параметров в сложных моделях (скрытые марковские модели, условные случайные поля, . . .), для классификации используется реже.

Поиск разделяющей плоскости

- Персептрон может зациклиться в случае отсутствия разделяющей гиперплоскости.
- А в случае наличия — найти неоптимальным образом.

Поиск разделяющей плоскости

- Персептрон может зациклиться в случае отсутствия разделяющей гиперплоскости.
- А в случае наличия — найти неоптимальным образом.
- Кстати, а что значит оптимальная гиперплоскость?
- Можно максимизировать расстояние от плоскости до ближайшего объекта.

Поиск разделяющей плоскости

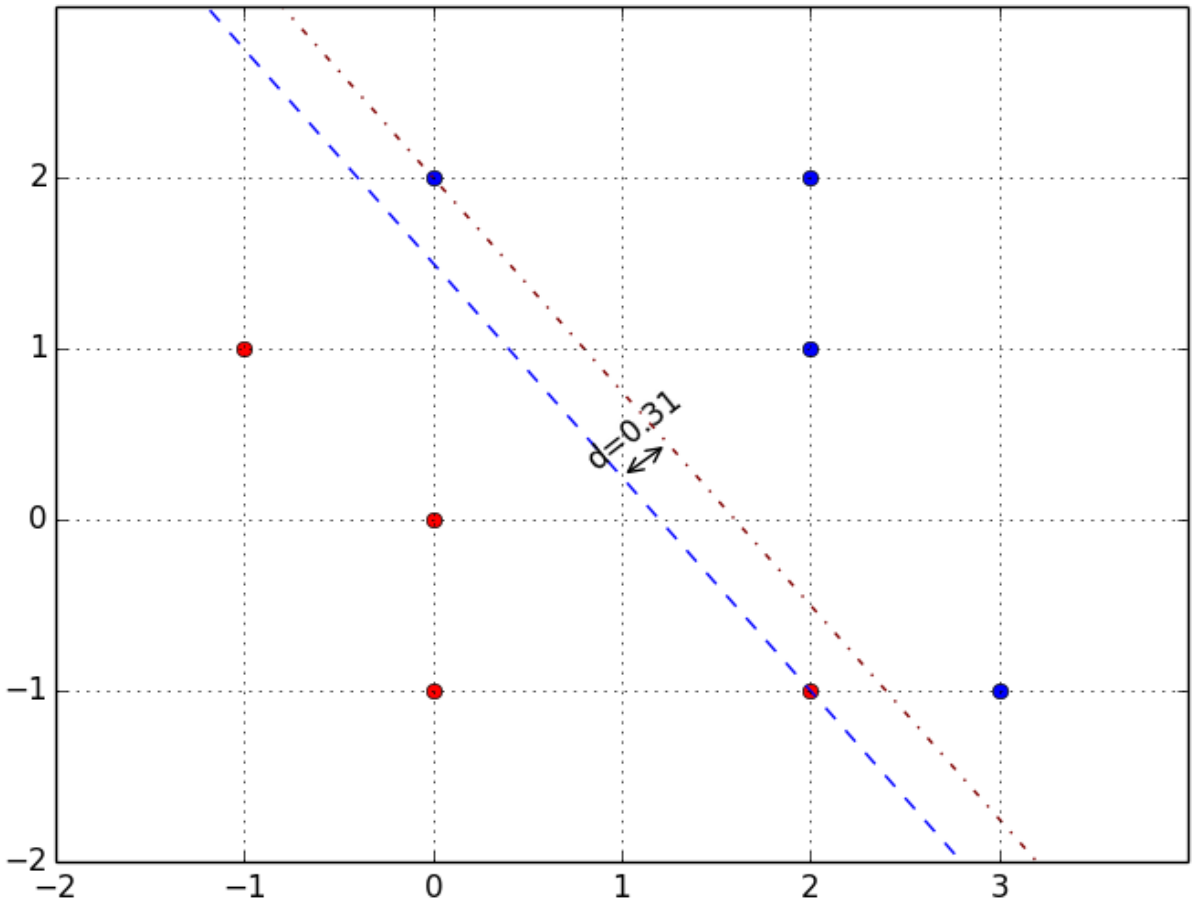
- Персептрон может зациклиться в случае отсутствия разделяющей гиперплоскости.
- А в случае наличия — найти неоптимальным образом.
- Кстати, а что значит оптимальная гиперплоскость?
- Можно максимизировать расстояние от плоскости до ближайшего объекта.
- Эквивалентно — максимизировать расстояние между классами.

Поиск разделяющей плоскости

- Персептрон может зациклиться в случае отсутствия разделяющей гиперплоскости.
- А в случае наличия — найти неоптимальным образом.
- Кстати, а что значит оптимальная гиперплоскость?
- Можно максимизировать расстояние от плоскости до ближайшего объекта.
- Эквивалентно — максимизировать расстояние между классами.
- Чем больше расстояние, тем классификатор надёжнее.

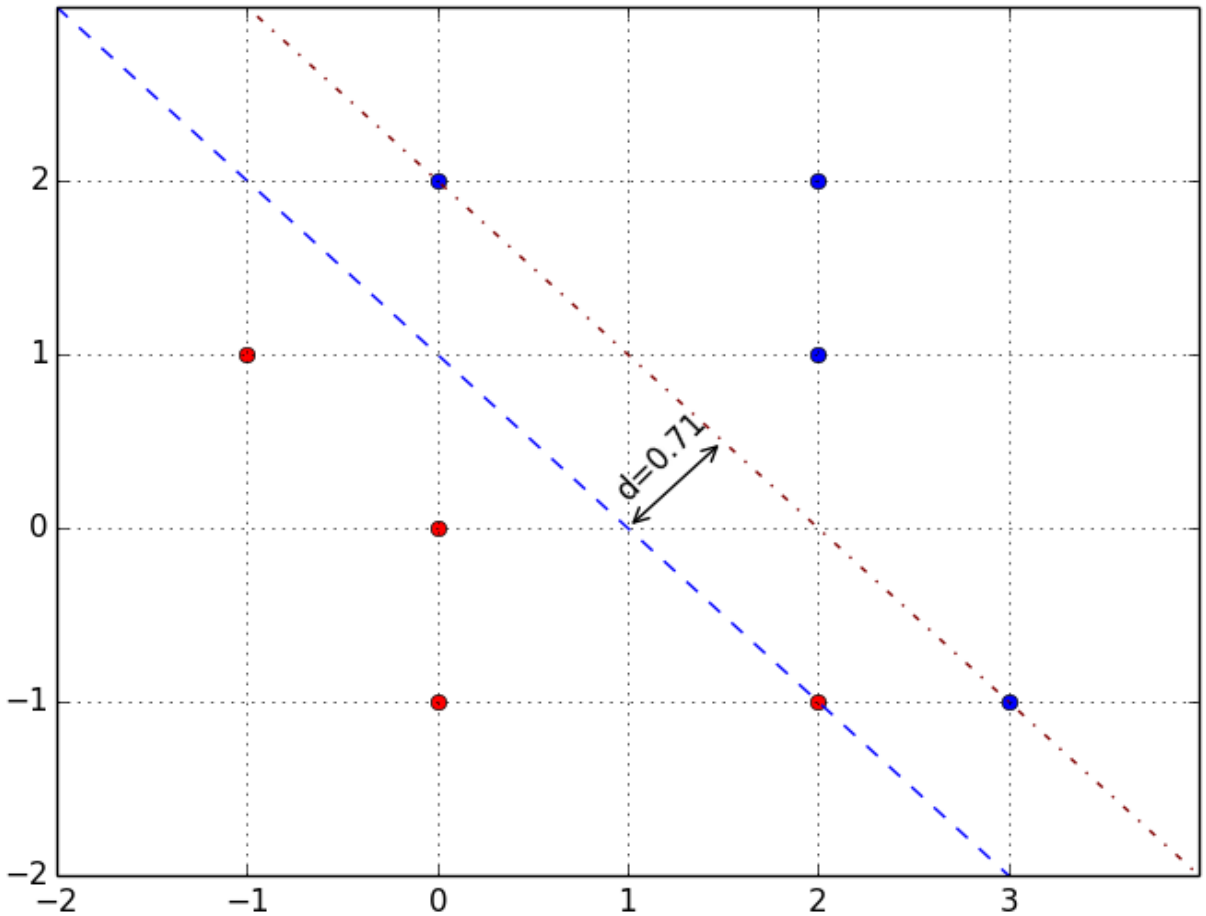
Маленькое расстояние

$$w = [7.0, 5.0, 4.0], d = 0.31$$



Большое расстояние

$$w = [3.0, 2.0, 2.0], d = 0.71$$



Постановка задачи для разделимой выборки

- Исходная постановка задачи:

$$\min_{x_i \in X^L} (\langle w, x_i \rangle - w_0) y_i \to \max, \|w\|^2 + w_0^2 = 1$$

Постановка задачи для разделимой выборки

- Исходная постановка задачи:

$$\min_{x_i \in X^L} (\langle w, x_i \rangle - w_0) y_i \to \max, \|w\|^2 + w_0^2 = 1$$

- Преобразованная постановка задачи:

$$\frac{(w, w) + (w_0, w_0)}{2} \to \min \quad \text{при условии}$$

$$y_i((w, x_i) - w_0) \geqslant 1 \text{ для всех } i = 1, \ldots, |X|.$$

- В дальнейшем мы считаем, что свободный член w_0 входит в вектор коээфициентов w (у всех x_i нулевая координата равна -1).

Постановка задачи для разделимой выборки

- Исходная постановка задачи:

$$\min_{x_i \in X^L} (\langle w, x_i \rangle - w_0) y_i \to \max, \|w\|^2 + w_0^2 = 1$$

- Преобразованная постановка задачи:

$$\frac{(w, w) + (w_0, w_0)}{2} \to \min \quad \text{при условии}$$

$$y_i((w, x_i) - w_0) \geqslant 1 \text{ для всех } i = 1, \ldots, |X|.$$

- В дальнейшем мы считаем, что свободный член w_0 входит в вектор коээфициентов w (у всех x_i нулевая координата равна -1).
- Что делать в случае неразделимой выборки?
- Можно ввести штрафы за ошибку.

Постановка задачи для неразделимой выборки

- Постановка задачи для разделимой выборки:

$$\frac{(w,w)}{2} \to \min \qquad \text{при условии}$$

$$y_i((w,x_i) - w_0) \geqslant 1 \text{ для всех } i = 1,\ldots,|X|.$$

Постановка задачи для неразделимой выборки

- Постановка задачи для разделимой выборки:

$$\frac{(w,w)}{2} \to \min \qquad \text{при условии}$$

$$y_i((w,x_i) - w_0) \geqslant 1 \text{ для всех } i = 1,\ldots,|X|.$$

- Постановка задачи для неразделимой выборки:

$$\frac{(w,w)}{2} + C\sum_i \xi_i \to \min \qquad \text{при условии}$$

$$y_i((w,x_i) - w_0) \geqslant (1 - \xi_i) \text{ для всех } i = 1,\ldots,|X|.$$

$$\xi_i \geqslant 0 \text{ для всех } i = 1,\ldots,|X|.$$

Постановка задачи для неразделимой выборки

- Постановка задачи для разделимой выборки:

$$\frac{(w,w)}{2} \to \min \qquad \text{при условии}$$

$$y_i((w,x_i) - w_0) \geqslant 1 \text{ для всех } i = 1,\ldots,|X|.$$

- Постановка задачи для неразделимой выборки:

$$\frac{(w,w)}{2} + C\sum_i \xi_i \to \min \qquad \text{при условии}$$

$$y_i((w,x_i) - w_0) \geqslant (1-\xi_i) \text{ для всех } i = 1,\ldots,|X|.$$

$$\xi_i \geqslant 0 \text{ для всех } i = 1,\ldots,|X|.$$

- ξ_i — максимально разрешённая ошибка на объекте x_i.

Постановка задачи для неразделимой выборки

- Постановка задачи для разделимой выборки:

$$\frac{(w,w)}{2} \to \min \qquad \text{при условии}$$

$$y_i((w,x_i) - w_0) \geqslant 1 \text{ для всех } i = 1,\ldots,|X|.$$

- Постановка задачи для неразделимой выборки:

$$\frac{(w,w)}{2} + C\sum_i \xi_i \to \min \qquad \text{при условии}$$

$$y_i((w,x_i) - w_0) \geqslant (1-\xi_i) \text{ для всех } i = 1,\ldots,|X|.$$

$$\xi_i \geqslant 0 \text{ для всех } i = 1,\ldots,|X|.$$

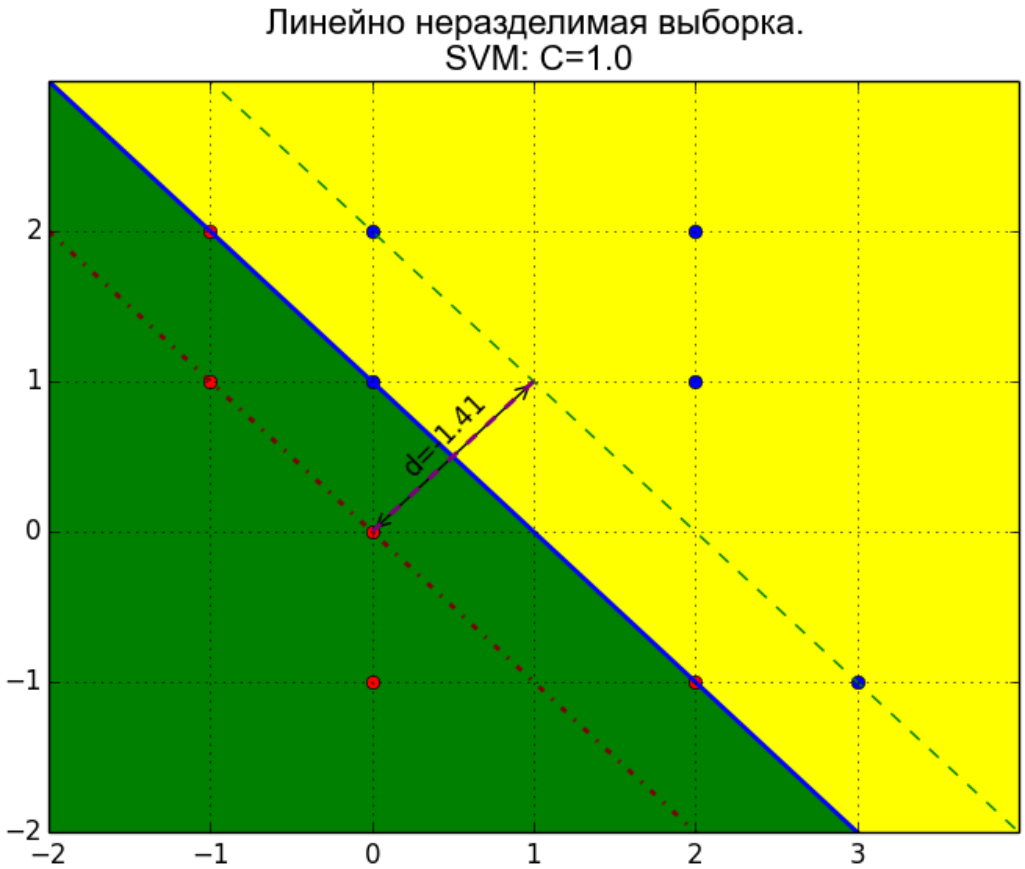
- ξ_i — максимально разрешённая ошибка на объекте x_i.
- C — параметр регуляризации (чем больше, тем строже наказываются ошибки).

Управляющий параметр

- Маленький параметр C — широкая разделяющая полоса.

Управляющий параметр

- Маленький параметр C — широкая разделяющая полоса.

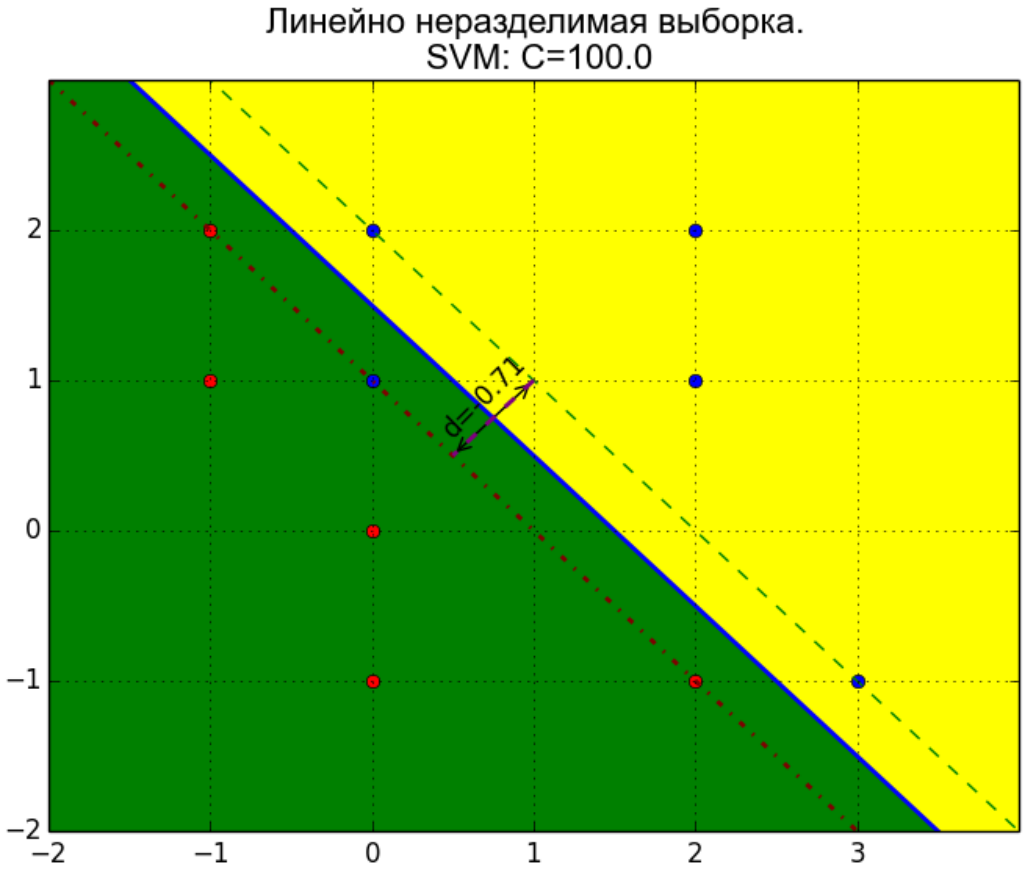


Управляющий параметр

- Маленький параметр C — широкая разделяющая полоса.
- Большой параметр C — меньше ошибок.

Управляющий параметр

- Маленький параметр C — широкая разделяющая полоса.
- Большой параметр C — меньше ошибок.



Управляющий параметр

- Маленький параметр C — широкая разделяющая полоса.
- Большой параметр C — меньше ошибок.
- Чем больше параметр C, тем точнее классификация на обучающей выборке.

Управляющий параметр

- Маленький параметр C — широкая разделяющая полоса.
- Большой параметр C — меньше ошибок.
- Чем больше параметр C, тем точнее классификация на обучающей выборке.
- Как следствие, больше чувствительность к выбросам и риск переобучения.

Управляющий параметр

- Маленький параметр C — широкая разделяющая полоса.
- Большой параметр C — меньше ошибок.
- Чем больше параметр C, тем точнее классификация на обучающей выборке.
- Как следствие, больше чувствительность к выбросам и риск переобучения.
- На практике C подбирают по скользящему контролю.

Неразделимая выборка: эквивалентная постановка

- Постановка задачи для неразделимой выборки:

$$\frac{(w,w)}{2} + C\sum_i \xi_i \to \min \qquad \text{при условии}$$

$$\xi_i \geqslant (1 - y_i((w,x_i) - w_0)) \text{ для всех } i = 1,\ldots,|X|.$$

$$\xi_i \geqslant 0 \text{ для всех } i = 1,\ldots,|X|.$$

Неразделимая выборка: эквивалентная постановка

- Постановка задачи для неразделимой выборки:

$$\frac{(w,w)}{2} + C\sum_i \xi_i \to \min \qquad \text{при условии}$$
$$\xi_i \geqslant (1 - y_i((w,x_i) - w_0)) \text{ для всех } i = 1,\ldots,|X|.$$
$$\xi_i \geqslant 0 \text{ для всех } i = 1,\ldots,|X|.$$

- Безусловная постановка задачи:

$$\frac{(w,w)}{2} + C\sum_i (1 - y_i((w,x_i) - w_0))_+ \to \min_{w,w_0}$$

Неразделимая выборка: эквивалентная постановка

- Постановка задачи для неразделимой выборки:

$$\frac{(w,w)}{2} + C\sum_i \xi_i \to \min \qquad \text{при условии}$$
$$\xi_i \geqslant (1 - y_i((w,x_i) - w_0)) \text{ для всех } i = 1,\ldots,|X|.$$
$$\xi_i \geqslant 0 \text{ для всех } i = 1,\ldots,|X|.$$

- Безусловная постановка задачи:

$$\frac{(w,w)}{2} + C\sum_i (1 - y_i((w,x_i) - w_0))_+ \to \min_{w,w_0}$$

- $(1 - M_i)_+ = \max(1 - M_i, 0)$ — штраф за ошибку, равную $-M_i$.

Неразделимая выборка: эквивалентная постановка

- Постановка задачи для неразделимой выборки:

$$\frac{(w,w)}{2} + C\sum_i \xi_i \to \min \qquad \text{при условии}$$
$$\xi_i \geqslant (1 - y_i((w,x_i) - w_0)) \text{ для всех } i = 1,\ldots,|X|.$$
$$\xi_i \geqslant 0 \text{ для всех } i = 1,\ldots,|X|.$$

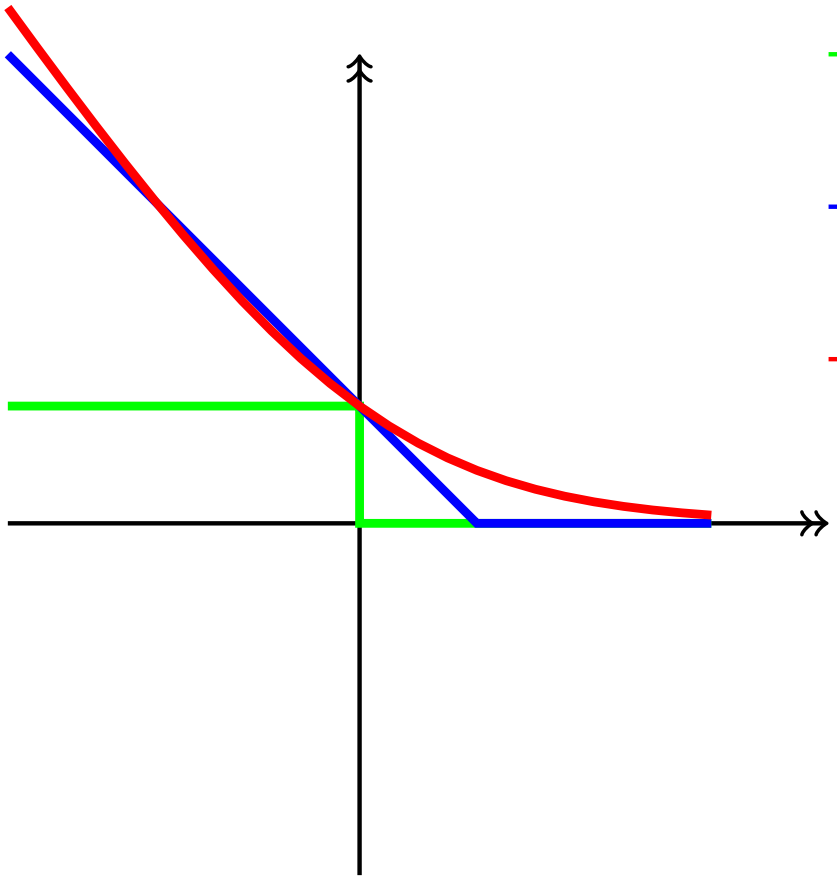
- Безусловная постановка задачи:

$$\frac{(w,w)}{2} + C\sum_i (1 - y_i((w,x_i) - w_0))_+ \to \min_{w,w_0}$$

- $(1 - M_i)_+ = \max(1 - M_i, 0)$ — штраф за ошибку, равную $-M_i$.
- $\frac{(w,w)}{2}$ — регуляризатор (препятствует переобучению).

Сравнение функций штрафа

$Q(x)$ — функция штрафа на объекте x с отступом $M(x)$.



- $Q(x) = [\![M(x) < 0]\!]$ (персептрон)
- $Q(x) = (1 - M(x))_+$ (метод опорных векторов)
- $Q(x) = \log_2(1 + \exp(-x))$ (логистическая регрессия)

Свойства метода опорных векторов

- Оптимальный вектор весов имеет вид $w = \sum_i \lambda_i x_i y_i$, где

$$\lambda_i(M_i - (1 - \xi_i)) = 0 \text{ для всех } i$$

Свойства метода опорных векторов

- Оптимальный вектор весов имеет вид $w = \sum_i \lambda_i x_i y_i$, где

$$\lambda_i(M_i - (1 - \xi_i)) = 0 \text{ для всех } i$$

- То есть если $\lambda_i \neq 0$, то $M_i \leqslant 1$.

Свойства метода опорных векторов

- Оптимальный вектор весов имеет вид $w = \sum_i \lambda_i x_i y_i$, где

$$\lambda_i(M_i - (1 - \xi_i)) = 0 \text{ для всех } i$$

- То есть если $\lambda_i \neq 0$, то $M_i \leqslant 1$.
- Вспомним, что $M_i = 1$ — уравнение границы разделяющей полосы.

Свойства метода опорных векторов

- Оптимальный вектор весов имеет вид $w = \sum_i \lambda_i x_i y_i$, где

$$\lambda_i(M_i - (1 - \xi_i)) = 0 \text{ для всех } i$$

- То есть если $\lambda_i \neq 0$, то $M_i \leqslant 1$.
- Вспомним, что $M_i = 1$ — уравнение границы разделяющей полосы.
- В формировании вектора весов участвуют либо объекты на границе разделяющей полосы (опорные вектора, $M_i = 1$), либо попавшие внутрь неё ($0 < M_i < 1$) или «на чужую сторону» ($M_i \leqslant 0$).

Свойства метода опорных векторов

- Оптимальный вектор весов имеет вид $w = \sum_i \lambda_i x_i y_i$, где

$$\lambda_i(M_i - (1 - \xi_i)) = 0 \text{ для всех } i$$

- То есть если $\lambda_i \neq 0$, то $M_i \leqslant 1$.
- Вспомним, что $M_i = 1$ — уравнение границы разделяющей полосы.
- В формировании вектора весов участвуют либо объекты на границе разделяющей полосы (опорные вектора, $M_i = 1$), либо попавшие внутрь неё ($0 < M_i < 1$) или «на чужую сторону» ($M_i \leqslant 0$).
- Это свойство называется разреженностью.