Введение в искусственный интеллект. Машинное обучение

Тема: Машины опорных векторов – SVM

Бабин Д.Н., Иванов И.Е.

кафедра Математической Теории Интеллектуальных Систем

План лекции

1. Случай линейной разделимости

План лекции

1. Случай линейной разделимости
2. Случай линейной неразделимости

План лекции

1. Случай линейной разделимости
2. Случай линейной неразделимости
3. Решение с помощью двойственной задачи

План лекции

1. Случай линейной разделимости
2. Случай линейной неразделимости
3. Решение с помощью двойственной задачи
4. Обобщение SVM с помощью ядрового трюка

План лекции

1. Случай линейной разделимости
2. Случай линейной неразделимости
3. Решение с помощью двойственной задачи
4. Обобщение SVM с помощью ядрового трюка
5. Многоклассовый SVM

Вспомним прошлую лекцию

Рассмотрим задачу бинарной классификации: $X \to Y$, $X = \mathbb{R}^n$, $Y = \{+1, -1\}$ на обучающей выборке $X^m = (x_i, y_i)_{i=1}^m$.
Линейный классификатор $a(x; w, w_0) = \mathrm{sign}(\langle w, x\rangle - w_0)$.

Вспомним прошлую лекцию

Рассмотрим задачу бинарной классификации: $X \to Y$, $X = \mathbb{R}^n$, $Y = \{+1, -1\}$ на обучающей выборке $X^m = (x_i, y_i)_{i=1}^m$.

Линейный классификатор $a(x; w, w_0) = \text{sign}(\langle w, x\rangle - w_0)$.

Минимизация эмпирического риска в данном случае:

$$R(w, w_0, X^m) = \sum_{i=1}^{m}[a(x_i; w, w_0) \neq y_i] = \sum_{i=1}^{m}[(\langle w, x_i\rangle - w_0)y_i < 0],$$

Вспомним прошлую лекцию

Рассмотрим задачу бинарной классификации: $X \to Y$, $X = \mathbb{R}^n$, $Y = \{+1, -1\}$ на обучающей выборке $X^m = (x_i, y_i)_{i=1}^m$.
Линейный классификатор $a(x; w, w_0) = \operatorname{sign}(\langle w, x\rangle - w_0)$.
Минимизация эмпирического риска в данном случае:

$$R(w, w_0, X^m) = \sum_{i=1}^{m} [a(x_i; w, w_0) \neq y_i] = \sum_{i=1}^{m} [(\langle w, x_i\rangle - w_0) y_i < 0],$$

Добавим аппроксимацию и L_2 регуляризацию:

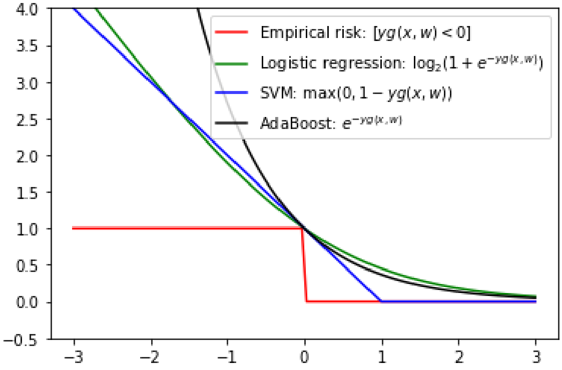
$$R(w, w_0, X^m) \leq \sum_{i=1}^{m} \max(0, 1 - (\langle w, x_i\rangle - w_0) y_i) + \frac{1}{2C} ||w||^2$$

Аппроксимация и регуляризация

- Аппроксимация штрафует за приближение к границе классов: $(\langle w, x_i \rangle - w_0) y_i = 1$

Аппроксимация и регуляризация

- Аппроксимация штрафует за приближение к границе классов: $(\langle w, x_i \rangle - w_0)y_i = 1$
- Регуляризация штрафует неустойчивые решения



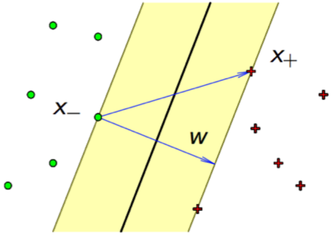
Случай линейной разделимости[1]

Линейная разделимость

$\exists w, w_0$ т.ч. $(\langle w, x_i \rangle - w_0) y_i > 0$ для всех $i = 1, \dots m$.

Очевидно, что можно перенормировать вектор w, т.ч. $\min_i (\langle w, x_i \rangle - w_0) y_i = 1$.

Разделяющая полоса: $-1 \leq \langle w, x_i \rangle - w_0 \leq +1$.
Разделяющая гиперплоскость (посередине):
$\langle w, x_i \rangle - w_0 = 0$.
Можем добиться того, что существует по крайней мере одна точка на каждой из границ (**Упражнение**: доказать): $\exists x_\pm : \langle w, x_\pm \rangle - w_0 = \pm 1$.
Ширина полосы: $\frac{\langle x_+ - x_-, w \rangle}{||w||} = \frac{2}{||w||} \to \max_w$



[1]Вапник, В. Н., Червоненкис, А. Я. (1964). Об одном классе персептронов.

Случай линейной разделимости – вывод

Т.о., в случае линейной разделимости можно оптимизационную задачу записать как:

$$\begin{cases} \frac{1}{2}||w||^2 \to \min_w, \\ y_i(\langle w, x_i \rangle - w_0) \geq 1, \quad i = 1, \ldots, m \end{cases}$$

Случай линейной неразделимости

Обобщим[2] задачу на этот случай: алгоритм может допускать ошибки на обучающих объектах.

[2] Cortes, C., and Vapnik, V. (1995). Support-vector networks.

Обобщим[2] задачу на этот случай: алгоритм может допускать ошибки на обучающих объектах.
Ограничение: таких ошибок должно быть поменьше.

[2] Cortes, C., and Vapnik, V. (1995). Support-vector networks.

Случай линейной неразделимости

Обобщим[2] задачу на этот случай: алгоритм может допускать ошибки на обучающих объектах.

Ограничение: таких ошибок должно быть поменьше.

Решение: введение дополнительных переменных $\xi_i \geq 0$, характеризующих величину ошибки на объектах x_i.

[2] Cortes, C., and Vapnik, V. (1995). Support-vector networks.

Обобщим[2] задачу на этот случай: алгоритм может допускать ошибки на обучающих объектах.
Ограничение: таких ошибок должно быть поменьше.
Решение: введение дополнительных переменных $\xi_i \geq 0$, характеризующих величину ошибки на объектах x_i.

$$\begin{cases} \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i \to \min_{w,w_0,\xi}, & \\ y_i(\langle w, x_i\rangle - w_0) \geq 1 - \xi_i, & i = 1,\ldots,m, \\ \xi_i \geq 0 & i = 1,\ldots,m. \end{cases}$$

[2] Cortes, C., and Vapnik, V. (1995). Support-vector networks.

Случай линейной неразделимости

Обобщим[2] задачу на этот случай: алгоритм может допускать ошибки на обучающих объектах.
Ограничение: таких ошибок должно быть поменьше.
Решение: введение дополнительных переменных $\xi_i \geq 0$, характеризующих величину ошибки на объектах x_i.

$$\begin{cases} \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i \to \min_{w,w_0,\xi}, & \\ y_i(\langle w, x_i\rangle - w_0) \geq 1 - \xi_i, & i = 1,\dots,m, \\ \xi_i \geq 0 & i = 1,\dots,m. \end{cases}$$

Замечание. Положительная константа C определяет компромисс между максимизацией ширины разделяющей полосы и минимизацией суммарной ошибки.

[2] Cortes, C., and Vapnik, V. (1995). Support-vector networks.

О константе C

Поставленная выше оптимизационная задача эквивалентна минимизации аппроксимированного э.р. с регуляризатором:

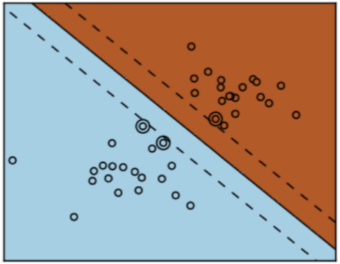
$$\sum_{i=1}^{m} \max(0, 1 - y_i(\langle w, x_i \rangle - w_0)) + \frac{1}{2C}||w||^2 \to min_{w,w_0}$$

О константе C

Поставленная выше оптимизационная задача эквивалентна минимизации аппроксимированного э.р. с регуляризатором:

$$\sum_{i=1}^{m} \max(0, 1 - y_i(\langle w, x_i \rangle - w_0)) + \frac{1}{2C}||w||^2 \to min_{w,w_0}$$

Большое значение C: узкая полоса, мало ошибок

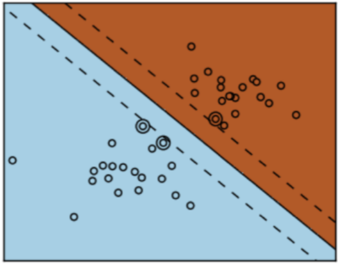


О константе C

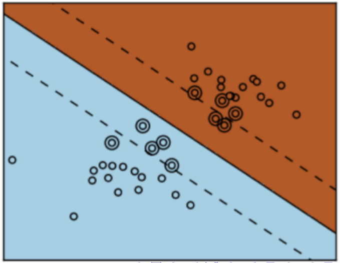
Поставленная выше оптимизационная задача эквивалентна минимизации аппроксимированного э.р. с регуляризатором:

$$\sum_{i=1}^{m} \max(0, 1 - y_i(\langle w, x_i \rangle - w_0)) + \frac{1}{2C}||w||^2 \to min_{w,w_0}$$

Большое значение C: узкая полоса, мало ошибок



Маленькое значение C: широкая полоса, много ошибок



Время для вопросов

Условия Каруша-Куна-Таккера (KKT)[3,4]

Условия ККТ – это **необходимые** условия решения задачи нелинейного программирования (обобщение метода множителей Лагранжа).
Задача нелинейного программирования:

$$\begin{cases} f(x) \to \min_x, \\ g_i(x) \leq 0, \qquad i = 1, \dots, k, \\ h_j(x) = 0, \qquad j = 1, \dots, \ell. \end{cases}$$

[3] W. Karush (1939). Minima of Functions of Several Variables with Inequalities as Side Constraints.
[4] Kuhn, H. W.; Tucker, A. W. (1951). Nonlinear programming.

Условия Каруша-Куна-Таккера (ККТ)[3,4]

Условия ККТ – это **необходимые** условия решения задачи нелинейного программирования (обобщение метода множителей Лагранжа).
Задача нелинейного программирования:

$$\begin{cases} f(x) \to \min_x, \\ g_i(x) \le 0, & i = 1, \ldots, k, \\ h_j(x) = 0, & j = 1, \ldots, \ell. \end{cases}$$

Необходимые условия: если x - точка локального минимума, то $\exists$ множители μ_i, λ_j, т.ч.

$$\begin{cases} \frac{\partial L}{\partial x} = 0; L(x, \mu, \lambda) = f(x) + \sum_{i=1}^{k} \mu_i g_i(x) + \sum_{j=1}^{\ell} \lambda_j h_j(x) & \text{(функция Лагранжа)} \\ g_i(x) \le 0, h_j(x) = 0 & \text{(исходные ограничения)} \\ \mu_i \ge 0 & \text{(двойственные ограничения)} \\ \mu_i g_i(x) = 0 & \text{(дополняющая нежесткость)} \end{cases}$$

[3] W. Karush (1939). Minima of Functions of Several Variables with Inequalities as Side Constraints.
[4] Kuhn, H. W.; Tucker, A. W. (1951). Nonlinear programming.

Условия ККТ и SVM

Функция Лагранжа для SVM

$L(w, w_0, \xi; \lambda, \eta) = \frac{1}{2}||w||^2 - \sum_{i=1}^{m} \lambda_i((\langle w, x_i \rangle - w_0)y_i - 1) - \sum_{i=1}^{m} \xi_i(\lambda_i + \eta_i - C)$, где:
λ_i – переменные, двойственные к ограничениям $(\langle w, x_i \rangle - w_0)y_i \geq 1 - \xi_i$,
η_i – переменные, двойственные к ограничениям $\xi_i \geq 0$.

Условия ККТ и SVM

Функция Лагранжа для SVM

$L(w, w_0, \xi; \lambda, \eta) = \frac{1}{2}||w||^2 - \sum_{i=1}^{m} \lambda_i((\langle w, x_i\rangle - w_0)y_i - 1) - \sum_{i=1}^{m} \xi_i(\lambda_i + \eta_i - C)$, где:

λ_i – переменные, двойственные к ограничениям $(\langle w, x_i\rangle - w_0)y_i \geq 1 - \xi_i$,

η_i – переменные, двойственные к ограничениям $\xi_i \geq 0$.

Необходимые условия примут вид:

$$\begin{cases} \frac{\partial L}{\partial w} = 0; \frac{\partial L}{\partial w_0} = 0; \frac{\partial L}{\partial \xi} = 0 & \\ \xi_i \geq 0, \lambda_i \geq 0, \eta_i \geq 0 & i = 1, \ldots, m \\ \lambda_i = 0 \text{ либо } (\langle w, x_i\rangle - w_0)y_i = 1 - \xi_i & i = 1, \ldots, m \\ \eta_i = 0 \text{ либо } \xi_i = 0 & i = 1, \ldots, m \end{cases}$$

Дифференцируем функцию Лагранжа

$$L(w, w_0, \xi; \lambda, \eta) = \frac{1}{2}||w||^2 - \sum_{i=1}^m \lambda_i((\langle w, x_i\rangle - w_0)y_i - 1) - \sum_{i=1}^m \xi_i(\lambda_i + \eta_i - C).$$

$$\begin{cases} \frac{\partial L}{\partial w} = w - \sum_{i=1}^m \lambda_i y_i x_i = 0 & \Rightarrow w = \sum_{i=1}^m \lambda_i y_i x_i \\ \frac{\partial L}{\partial w_0} = -\sum_{i=1}^m \lambda_i y_i = 0 & \Rightarrow \sum_{i=1}^m \lambda_i y_i = 0 \\ \frac{\partial L}{\partial \xi_i} = -\lambda_i - \eta_i + C = 0 & \Rightarrow \lambda_i + \eta_i = C \end{cases}$$

Дифференцируем функцию Лагранжа

$$L(w, w_0, \xi; \lambda, \eta) = \frac{1}{2}||w||^2 - \sum_{i=1}^m \lambda_i((\langle w, x_i\rangle - w_0)y_i - 1) - \sum_{i=1}^m \xi_i(\lambda_i + \eta_i - C).$$

$$\begin{cases} \frac{\partial L}{\partial w} = w - \sum_{i=1}^m \lambda_i y_i x_i = 0 & \Rightarrow w = \sum_{i=1}^m \lambda_i y_i x_i \\ \frac{\partial L}{\partial w_0} = -\sum_{i=1}^m \lambda_i y_i = 0 & \Rightarrow \sum_{i=1}^m \lambda_i y_i = 0 \\ \frac{\partial L}{\partial \xi_i} = -\lambda_i - \eta_i + C = 0 & \Rightarrow \lambda_i + \eta_i = C \end{cases}$$

1. $\lambda_i = 0, \eta_i = C, \xi_i = 0, (\langle w, x_i\rangle - w_0)y_i \geq 1$: периферийные объекты

Дифференцируем функцию Лагранжа

$$L(w, w_0, \xi; \lambda, \eta) = \frac{1}{2}||w||^2 - \sum_{i=1}^m \lambda_i((\langle w, x_i \rangle - w_0)y_i - 1) - \sum_{i=1}^m \xi_i(\lambda_i + \eta_i - C).$$

$$\begin{cases} \frac{\partial L}{\partial w} = w - \sum_{i=1}^m \lambda_i y_i x_i = 0 & \Rightarrow w = \sum_{i=1}^m \lambda_i y_i x_i \\ \frac{\partial L}{\partial w_0} = -\sum_{i=1}^m \lambda_i y_i = 0 & \Rightarrow \sum_{i=1}^m \lambda_i y_i = 0 \\ \frac{\partial L}{\partial \xi_i} = -\lambda_i - \eta_i + C = 0 & \Rightarrow \lambda_i + \eta_i = C \end{cases}$$

1. $\lambda_i = 0, \eta_i = C, \xi_i = 0, (\langle w, x_i \rangle - w_0)y_i \geq 1$: периферийные объекты
2. $0 < \lambda_i < C, 0 < \eta_i < C, \xi_i = 0, (\langle w, x_i \rangle - w_0)y_i = 1$: опорные объекты на границе

Дифференцируем функцию Лагранжа

$$L(w, w_0, \xi; \lambda, \eta) = \frac{1}{2}||w||^2 - \sum_{i=1}^m \lambda_i((\langle w, x_i \rangle - w_0)y_i - 1) - \sum_{i=1}^m \xi_i(\lambda_i + \eta_i - C).$$

$$\begin{cases} \frac{\partial L}{\partial w} = w - \sum_{i=1}^m \lambda_i y_i x_i = 0 & \Rightarrow w = \sum_{i=1}^m \lambda_i y_i x_i \\ \frac{\partial L}{\partial w_0} = -\sum_{i=1}^m \lambda_i y_i = 0 & \Rightarrow \sum_{i=1}^m \lambda_i y_i = 0 \\ \frac{\partial L}{\partial \xi_i} = -\lambda_i - \eta_i + C = 0 & \Rightarrow \lambda_i + \eta_i = C \end{cases}$$

1. $\lambda_i = 0, \eta_i = C, \xi_i = 0, (\langle w, x_i \rangle - w_0)y_i \geq 1$: периферийные объекты
2. $0 < \lambda_i < C, 0 < \eta_i < C, \xi_i = 0, (\langle w, x_i \rangle - w_0)y_i = 1$: опорные объекты на границе
3. $\lambda_i = C, \eta_i = 0, \xi_i > 0, (\langle w, x_i \rangle - w_0)y_i < 1$: опорные объекты-ошибки

О двойственных задачах

Прямая задача:

$$\begin{cases} f(x) \to \min_x, \\ g_i(x) \le 0, & i = 1, \dots, k, \\ h_j(x) = 0, & j = 1, \dots, \ell. \end{cases}$$

Функция Лагранжа: $L(x, \mu, \lambda) = f(x) + \sum_{i=1}^{k} \mu_i g_i(x) + \sum_{j=1}^{\ell} \lambda_j h_j(x) \to \min_x$.

[5] Wolfe, P. (1961). A duality theorem for non-linear programming.

О двойственных задачах

Прямая задача:

$$\begin{cases} f(x) \to \min_x, & \\ g_i(x) \leq 0, & i = 1, \dots, k, \\ h_j(x) = 0, & j = 1, \dots, \ell. \end{cases}$$

Функция Лагранжа: $L(x, \mu, \lambda) = f(x) + \sum_{i=1}^{k} \mu_i g_i(x) + \sum_{j=1}^{\ell} \lambda_j h_j(x) \to \min_x$.
Двойственная функция: $Q(\mu, \lambda) = \min_x L(x, \mu, \lambda)$.

[5] Wolfe, P. (1961). A duality theorem for non-linear programming.

О двойственных задачах

Прямая задача:

$$\begin{cases} f(x) \to \min_x, \\ g_i(x) \le 0, & i = 1, \dots, k, \\ h_j(x) = 0, & j = 1, \dots, \ell. \end{cases}$$

Функция Лагранжа: $L(x, \mu, \lambda) = f(x) + \sum_{i=1}^{k} \mu_i g_i(x) + \sum_{j=1}^{\ell} \lambda_j h_j(x) \to \min_x$.
Двойственная функция: $Q(\mu, \lambda) = \min_x L(x, \mu, \lambda)$.
Двойственная задача:

$$\begin{cases} Q(\mu, \lambda) \to \max_{\mu,\lambda}, \\ \mu_i \ge 0, & i = 1, \dots, k \end{cases}$$

[5] Wolfe, P. (1961). A duality theorem for non-linear programming.

О двойственных задачах

Прямая задача:

$$\begin{cases} f(x) \to \min_x, \\ g_i(x) \le 0, & i = 1, \dots, k, \\ h_j(x) = 0, & j = 1, \dots, \ell. \end{cases}$$

Функция Лагранжа: $L(x, \mu, \lambda) = f(x) + \sum_{i=1}^{k} \mu_i g_i(x) + \sum_{j=1}^{\ell} \lambda_j h_j(x) \to \min_x$.
Двойственная функция: $Q(\mu, \lambda) = \min_x L(x, \mu, \lambda)$.
Двойственная задача:

$$\begin{cases} Q(\mu, \lambda) \to \max_{\mu, \lambda}, \\ \mu_i \ge 0, & i = 1, \dots, k \end{cases}$$

Теорема (Дуальность Вулфа[5])

Если $f(x), g_i(x), h_j(x)$ – выпуклые функции, x^* – решение прямой задачи, а (μ^*, λ^*) – решение двойственной задачи, то $Q(\mu^*, \lambda^*) = f(x^*)$.

[5] Wolfe, P. (1961). A duality theorem for non-linear programming.

Двойственная задача для SVM

Подставим решения прямой задачи

$$w = \sum_{i=1}^{m} \lambda_i y_i x_i, \sum_{i=1}^{m} \lambda_i y_i = 0, \lambda_i + \eta_i = C$$

в функцию Лагранжа

$L(w, w_0, \xi; \lambda, \eta) = \frac{1}{2}||w||^2 - \sum_{i=1}^{m} \lambda_i(y_i(\langle w, x_i\rangle - w_0) - 1) - \sum_{i=1}^{m} \xi_i(\lambda_i + \eta_i - C)$.

Двойственная задача для SVM

Подставим решения прямой задачи

$$w = \sum_{i=1}^{m} \lambda_i y_i x_i, \sum_{i=1}^{m} \lambda_i y_i = 0, \lambda_i + \eta_i = C$$

в функцию Лагранжа
$L(w, w_0, \xi; \lambda, \eta) = \frac{1}{2}||w||^2 - \sum_{i=1}^{m} \lambda_i (y_i(\langle w, x_i \rangle - w_0) - 1) - \sum_{i=1}^{m} \xi_i (\lambda_i + \eta_i - C).$
Получим:

$$Q(\lambda) = \frac{1}{2} \sum_{i=1}^{m} \lambda_i y_i x_i \sum_{j=1}^{m} \lambda_j y_j x_j - \sum_{i=1}^{m} \lambda_i y_i \left\langle \sum_{j=1}^{m} \lambda_j y_j x_j, x_i \right\rangle + w_0 \sum_{i=1}^{m} \lambda_i y_i + \sum_{i=1}^{m} \lambda_i - \sum_{i=1}^{m} \xi_i (C - C) =$$

$$= -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i \lambda_j y_i y_j \langle x_i, x_j \rangle + \sum_{i=1}^{m} \lambda_i$$

.

Двойственная задача для SVM

Объединяя, получаем формулировку двойственной задачи:

$$\begin{cases} -Q(\lambda) = \frac{1}{2}\sum_{i=1}^m \sum_{j=1}^m \lambda_i \lambda_j y_i y_j \langle x_i, x_j \rangle - \sum_{i=1}^m \lambda_i \to \min_\lambda, \\ 0 \leq \lambda_i \leq C, \\ \sum_{i=1}^m \lambda_i y_i = 0. \end{cases}$$

[6] https://en.wikipedia.org/wiki/Quadratic_programming

Объединяя, получаем формулировку двойственной задачи:

$$\begin{cases} -Q(\lambda) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\lambda_i\lambda_j y_i y_j\langle x_i, x_j\rangle - \sum_{i=1}^{m}\lambda_i \to \min_{\lambda}, \\ 0 \leq \lambda_i \leq C, \\ \sum_{i=1}^{m}\lambda_i y_i = 0. \end{cases}$$

Минимизируется квадратичный функционал $Q(\lambda)$, имеющий неотрицательно определённую квадратичную форму $\Rightarrow$ этот функционал – выпуклый.

[6] https://en.wikipedia.org/wiki/Quadratic_programming

Двойственная задача для SVM

Объединяя, получаем формулировку двойственной задачи:

$$\begin{cases} -Q(\lambda) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\lambda_i\lambda_j y_i y_j\langle x_i, x_j\rangle - \sum_{i=1}^{m}\lambda_i \to \min_\lambda, \\ 0 \le \lambda_i \le C, \\ \sum_{i=1}^{m}\lambda_i y_i = 0. \end{cases}$$

Минимизируется квадратичный функционал $Q(\lambda)$, имеющий неотрицательно определённую квадратичную форму $\Rightarrow$ этот функционал – выпуклый. Область, определяемая линейными ограничениями, также выпуклая.

[6] https://en.wikipedia.org/wiki/Quadratic_programming

Двойственная задача для SVM

Объединяя, получаем формулировку двойственной задачи:

$$\begin{cases} -Q(\lambda) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\lambda_i\lambda_j y_i y_j\langle x_i, x_j\rangle - \sum_{i=1}^{m}\lambda_i \to \min_{\lambda}, \\ 0 \le \lambda_i \le C, \\ \sum_{i=1}^{m}\lambda_i y_i = 0. \end{cases}$$

Минимизируется квадратичный функционал $Q(\lambda)$, имеющий неотрицательно определённую квадратичную форму $\Rightarrow$ этот функционал – выпуклый.
Область, определяемая линейными ограничениями, также выпуклая.
Следовательно, данная двойственная задача имеет **единственное** решение.

[6] https://en.wikipedia.org/wiki/Quadratic_programming

Двойственная задача для SVM

Объединяя, получаем формулировку двойственной задачи:

$$\begin{cases} -Q(\lambda) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\lambda_i\lambda_j y_i y_j \langle x_i, x_j\rangle - \sum_{i=1}^{m}\lambda_i \to \min_{\lambda}, \\ 0 \le \lambda_i \le C, \\ \sum_{i=1}^{m}\lambda_i y_i = 0. \end{cases}$$

Минимизируется квадратичный функционал $Q(\lambda)$, имеющий неотрицательно определённую квадратичную форму $\Rightarrow$ этот функционал – выпуклый.
Область, определяемая линейными ограничениями, также выпуклая.
Следовательно, данная двойственная задача имеет **единственное** решение.
Способ решения – методами **квадратичного** программирования (например, можно использовать метод внутренней точки[6]).

[6] https://en.wikipedia.org/wiki/Quadratic_programming

Решение прямой задачи для SVM

Пусть единственное решение двойственной задачи $(\lambda_i)_{i=1}^m$. Тогда решение прямой задачи выражается через решение двойственной как:

$$\begin{cases} w = \sum_{\lambda_i \neq 0} \lambda_i y_i x_i, & \text{суммируем только по опорным векторам } \lambda_i \neq 0 \\ w_0 = \langle w, x_j \rangle - y_j & \text{для опорного вектора на границе } 0 < \lambda_j < C \end{cases}$$

Решение прямой задачи для SVM

Пусть единственное решение двойственной задачи $(\lambda_i)_{i=1}^m$. Тогда решение прямой задачи выражается через решение двойственной как:

$$\begin{cases} w = \sum_{\lambda_i \neq 0} \lambda_i y_i x_i, & \text{суммируем только по опорным векторам } \lambda_i \neq 0 \\ w_0 = \langle w, x_j \rangle - y_j & \text{для опорного вектора на границе } 0 < \lambda_j < C \end{cases}$$

При этом сам линейный классификатор примет вид

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

что можно понимать как линейность в пространстве $\mathbb{R}^m$ с признаками $f_i = \langle x_i, x \rangle$.

Время для вопросов

SVM и регуляризаторы[7]

Напомним задачу минимизации аппроксимированного э.р. (в данном случае – Hinge loss) с L_2-регуляризатором:

$$\sum_{i=1}^{m} \max(0, 1 - y_i(\langle w, x_i \rangle - w_0)) + \frac{1}{2C}||w||^2 \to min_{w,w_0}$$

[7] Lopez-Martinez, D. (2017). Regularization approaches for support vector machines with applications to biomedical data.

SVM и регуляризаторы[7]

Напомним задачу минимизации аппроксимированного э.р. (в данном случае – Hinge loss) с L_2-регуляризатором:

$$\sum_{i=1}^{m} \max(0, 1 - y_i(\langle w, x_i \rangle - w_0)) + \frac{1}{2C}||w||^2 \to min_{w,w_0}$$

А что если добавить и/или заменить на L_1-регуляризацию?

[7] Lopez-Martinez, D. (2017). Regularization approaches for support vector machines with applications to biomedical data.

SVM и регуляризаторы[7]

Напомним задачу минимизации аппроксимированного э.р. (в данном случае – Hinge loss) с L_2-регуляризатором:

$$\sum_{i=1}^{m} \max(0, 1 - y_i(\langle w, x_i \rangle - w_0)) + \frac{1}{2C}||w||^2 \to min_{w,w_0}$$

А что если добавить и/или заменить на L_1-регуляризацию?
Оказывается, обычная SVM (которая соответствует гребневой регрессии) преобразуется в LASSO либо ElasticNet SVM, и при этом все равно возможно сведение к квадратичной задаче.

[7] Lopez-Martinez, D. (2017). Regularization approaches for support vector machines with applications to biomedical data.

LASSO SVM

Аппроксимированный э.р. с L_1регуляризатором:

$$\sum_{i=1}^{m} \max(0, 1 - y_i(\langle w, x_i \rangle - w_0)) + \frac{1}{C} \sum_{i=1}^{n} |w_i| \to min_{w, w_0}$$

LASSO SVM

Аппроксимированный э.р. с L_1регуляризатором:

$$\sum_{i=1}^{m} \max(0, 1 - y_i(\langle w, x_i \rangle - w_0)) + \frac{1}{C}\sum_{i=1}^{n} |w_i| \to min_{w,w_0}$$

Введем переменные $u_i = \frac{|w_i|+w_i}{2}$, $v_i = \frac{|w_i|-w_i}{2}$, $u_i \geq 0$, $v_i \geq 0$.

LASSO SVM

Аппроксимированный э.р. с L_1регуляризатором:

$$\sum_{i=1}^{m} \max(0, 1 - y_i(\langle w, x_i \rangle - w_0)) + \frac{1}{C} \sum_{i=1}^{n} |w_i| \to min_{w,w_0}$$

Введем переменные $u_i = \frac{|w_i|+w_i}{2}$, $v_i = \frac{|w_i|-w_i}{2}$, $u_i \geq 0$, $v_i \geq 0$.
Тогда замена переменных: $w_i = u_i - v_i, |w_i| = u_i + v_i$, и задача выше сводится к:

$$\begin{cases} \sum_{i=1}^{n} u_i + \sum_{i=1}^{n} v_i + C \sum_{i=1}^{m} \xi_i \to \min_{u,v,w_0,\xi}, & \\ y_i(\langle u, x_i \rangle - \langle v, x_i \rangle - w_0) \geq 1 - \xi_i, & i = 1, \dots, m, \\ \xi_i \geq 0 & i = 1, \dots, m, \\ u_i \geq 0, v_i \geq 0 & i = 1, \dots, n. \end{cases}$$

которая аналогична рассмотренной до этого.

Ядро и неотрицательность

Если в исходном пространстве сложно разделить выборку, то попробуем перейти в пространство большей размерности[8] $\varphi : X \to H$.

[8] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers.

Ядро и неотрицательность

Если в исходном пространстве сложно разделить выборку, то попробуем перейти в пространство большей размерности[8] $\varphi : X \to H$.

Определение ядра

Ядро – функция $K : X \times X \to \mathbb{R}$, т.ч. $K(x_1, x_2) = \langle \varphi(x_1), \varphi(x_2) \rangle$ при некотором $\varphi : X \to H$, где H – гильбертово пространство.

[8] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers.

Ядро и неотрицательность

Если в исходном пространстве сложно разделить выборку, то попробуем перейти в пространство большей размерности[8] $\varphi : X \to H$.

Определение ядра

Ядро – функция $K : X \times X \to \mathbb{R}$, т.ч. $K(x_1, x_2) = \langle \varphi(x_1), \varphi(x_2) \rangle$ при некотором $\varphi : X \to H$, где H – гильбертово пространство.

Неотрицательная определенность

Следующие два определения эквивалентны для проверки $K(x_1, x_2)$

- $\int_X \int_X K(x_1, x_2) f(x_1) f(x_2) dx_1 dx_2 \geq 0$ для любой функции $f : X \to \mathbb{R}$
- Для любой конечной выборки $X^m = (x_1, \dots, x_m)$ из X матрица $K = \|K(x_i, x_j)\|$ размера $m \times m$ неотрицательно определена: $z^T K z \geq 0$ для любого $z \in \mathbb{R}^m$.

[8] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers.

Проверка на ядро

Теорема Мерсера[9]

Функция $K(x_1, x_2)$ является ядром тогда и только тогда, когда:

- $K(x_1, x_2)$ симметрична: $K(x_1, x_2) = K(x_2, x_1)$, и
- $K(x_1, x_2)$ неотрицательно определена.

[9] Mercer, J. (1909). Functions of positive and negative type, and their connection the theory of integral equations.

Проверка на ядро

Теорема Мерсера[9]

Функция $K(x_1, x_2)$ является ядром тогда и только тогда, когда:

- $K(x_1, x_2)$ симметрична: $K(x_1, x_2) = K(x_2, x_1)$, и
- $K(x_1, x_2)$ неотрицательно определена.

Замечание. Если K не удовлетворяет указанным выше условиям, то минимизируемый функционал для классификатора уже не будет выпуклым, и решение может оказаться не единственным!

[9] Mercer, J. (1909). Functions of positive and negative type, and their connection the theory of integral equations.

Операции над ядрами

- Скалярное произведение $K(x_1, x_2) = \langle x_1, x_2 \rangle$ – ядро

Операции над ядрами

- Скалярное произведение $K(x_1, x_2) = \langle x_1, x_2 \rangle$ – ядро
- Константа $K(x_1, x_2) = c$ – ядро

Операции над ядрами

- Скалярное произведение $K(x_1, x_2) = \langle x_1, x_2 \rangle$ – ядро
- Константа $K(x_1, x_2) = c$ – ядро
- Произведение ядер $K(x_1, x_2) = K_1(x_1, x_2) K_2(x_1, x_2)$ – ядро

Операции над ядрами

- Скалярное произведение $K(x_1, x_2) = \langle x_1, x_2 \rangle$ – ядро
- Константа $K(x_1, x_2) = c$ – ядро
- Произведение ядер $K(x_1, x_2) = K_1(x_1, x_2)K_2(x_1, x_2)$ – ядро

Операции над ядрами

- Скалярное произведение $K(x_1, x_2) = \langle x_1, x_2 \rangle$ – ядро
- Константа $K(x_1, x_2) = c$ – ядро
- Произведение ядер $K(x_1, x_2) = K_1(x_1, x_2)K_2(x_1, x_2)$ – ядро
- Линейная $K(x_1, x_2) = \alpha_1 K_1(x_1, x_2) + \alpha_2 K_2(x_1, x_2)$ – ядро при $\alpha_1, \alpha_2 > 0$, K_1, K_2 - ядрах

Операции над ядрами

- Скалярное произведение $K(x_1, x_2) = \langle x_1, x_2 \rangle$ – ядро
- Константа $K(x_1, x_2) = c$ – ядро
- Произведение ядер $K(x_1, x_2) = K_1(x_1, x_2)K_2(x_1, x_2)$ – ядро
- Линейная $K(x_1, x_2) = \alpha_1 K_1(x_1, x_2) + \alpha_2 K_2(x_1, x_2)$ – ядро при $\alpha_1, \alpha_2 > 0$, K_1, K_2 - ядрах
- Для любой $\varphi : X \to X$ подстановка $K(x_1, x_2) = K_1(\varphi(x_1), \varphi(x_2))$ – ядро при K_1 - ядро

Операции над ядрами

- Скалярное произведение $K(x_1, x_2) = \langle x_1, x_2 \rangle$ – ядро
- Константа $K(x_1, x_2) = c$ – ядро
- Произведение ядер $K(x_1, x_2) = K_1(x_1, x_2)K_2(x_1, x_2)$ – ядро
- Линейная $K(x_1, x_2) = \alpha_1 K_1(x_1, x_2) + \alpha_2 K_2(x_1, x_2)$ – ядро при $\alpha_1, \alpha_2 > 0$, K_1, K_2 - ядрах
- Для любой $\varphi : X \to X$ подстановка $K(x_1, x_2) = K_1(\varphi(x_1), \varphi(x_2))$ – ядро при K_1 - ядро
- $K(x_1, x_2) = k(x_1 - x_2)$ – ядро $\Leftrightarrow$ Фурье-образ $F[k](\omega) = (2\pi)^{\frac{n}{2}} \int_X e^{-i\langle \omega, x\rangle} k(x)dx$ неотрицателен

Операции над ядрами

- Скалярное произведение $K(x_1, x_2) = \langle x_1, x_2 \rangle$ – ядро
- Константа $K(x_1, x_2) = c$ – ядро
- Произведение ядер $K(x_1, x_2) = K_1(x_1, x_2)K_2(x_1, x_2)$ – ядро
- Линейная $K(x_1, x_2) = \alpha_1 K_1(x_1, x_2) + \alpha_2 K_2(x_1, x_2)$ – ядро при $\alpha_1, \alpha_2 > 0$, K_1, K_2 - ядрах
- Для любой $\varphi : X \to X$ подстановка $K(x_1, x_2) = K_1(\varphi(x_1), \varphi(x_2))$ – ядро при K_1 - ядро
- $K(x_1, x_2) = k(x_1 - x_2)$ – ядро $\Leftrightarrow$ Фурье-образ $F[k](\omega) = (2\pi)^{\frac{n}{2}} \int_X e^{-i\langle\omega, x\rangle} k(x)dx$ неотрицателен
- Композиция произвольного ядра K_1 и произвольной функции $f : \mathbb{R} \to \mathbb{R}$, представимой в виде сходящегося степенного ряда с неотрицательными коэффициентами $K(x_1, x_2) = f(K_1(x_1, x_2))$ – ядро

SVM с другими ядрами

Изначально наша двойственная задача была сформулирована в терминах линейного ядра:

$$\begin{cases} -Q(\lambda) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\lambda_i\lambda_j y_i y_j\langle x_i, x_j\rangle - \sum_{i=1}^{m}\lambda_i \to \min_\lambda, \\ 0 \le \lambda_i \le C, \\ \sum_{i=1}^{m}\lambda_i y_i = 0. \end{cases}$$

SVM с другими ядрами

Изначально наша двойственная задача была сформулирована в терминах линейного ядра:

$$\begin{cases} -Q(\lambda) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\lambda_i\lambda_j y_i y_j\langle x_i, x_j\rangle - \sum_{i=1}^{m}\lambda_i \to \min_\lambda, \\ 0 \le \lambda_i \le C, \\ \sum_{i=1}^{m}\lambda_i y_i = 0. \end{cases}$$

Когда мы меняем ядро с $\langle x_i, x_j\rangle$ на $K(x_i, x_j)$, задача преобразуется в:

$$\begin{cases} -Q(\lambda) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\lambda_i\lambda_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{m}\lambda_i \to \min_\lambda, \\ 0 \le \lambda_i \le C, \\ \sum_{i=1}^{m}\lambda_i y_i = 0. \end{cases}$$

SVM с другими ядрами

Изначально наша двойственная задача была сформулирована в терминах линейного ядра:

$$\begin{cases} -Q(\lambda) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\lambda_i\lambda_j y_i y_j\langle x_i, x_j\rangle - \sum_{i=1}^{m}\lambda_i \to \min_\lambda, \\ 0 \le \lambda_i \le C, \\ \sum_{i=1}^{m}\lambda_i y_i = 0. \end{cases}$$

Когда мы меняем ядро с $\langle x_i, x_j\rangle$ на $K(x_i, x_j)$, задача преобразуется в:

$$\begin{cases} -Q(\lambda) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\lambda_i\lambda_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{m}\lambda_i \to \min_\lambda, \\ 0 \le \lambda_i \le C, \\ \sum_{i=1}^{m}\lambda_i y_i = 0. \end{cases}$$

При этом сам линейный классификатор принимает вид (x_j - опорный вектор на границе):

$$a(x) = \mathrm{sign}\left(\sum_{i=1}^{m}\lambda_i y_i K(x_i, x) - w_0\right), w_0 = \sum_{i=1}^{m}\lambda_i y_i K(x_i, x_j) - y_j$$

Пример нахождения пространства H

Пусть $X = \mathbb{R}^2, K(u, v) = \langle u, v\rangle^2$ при $u = (u_1, u_2), v = (v_1, v_2)$.
Хотим найти H и $\varphi : X \to H$, т.ч. $K(x_1, x_2) = \langle\varphi(x_1), \varphi(x_2)\rangle$.

Пример нахождения пространства H

Пусть $X = \mathbb{R}^2, K(u, v) = \langle u, v\rangle^2$ при $u = (u_1, u_2), v = (v_1, v_2)$.
Хотим найти H и $\varphi : X \to H$, т.ч. $K(x_1, x_2) = \langle\varphi(x_1), \varphi(x_2)\rangle$.
Сделаем эквивалентные преобразования:
$K(u, v) = \langle u, v\rangle^2 = \langle(u_1, u_2), (v_1, v_2)\rangle^2 = (u_1 v_1 + u_2 v_2)^2 = u_1^2 v_1^2 + u_2^2 v_2^2 + 2u_1 v_1 u_2 v_2$

Пример нахождения пространства H

Пусть $X = \mathbb{R}^2, K(u, v) = \langle u, v\rangle^2$ при $u = (u_1, u_2), v = (v_1, v_2)$.
Хотим найти H и $\varphi : X \to H$, т.ч. $K(x_1, x_2) = \langle\varphi(x_1), \varphi(x_2)\rangle$.
Сделаем эквивалентные преобразования:
$K(u, v) = \langle u, v\rangle^2 = \langle(u_1, u_2), (v_1, v_2)\rangle^2 = (u_1 v_1 + u_2 v_2)^2 = u_1^2 v_1^2 + u_2^2 v_2^2 + 2u_1 v_1 u_2 v_2$
$K(u, v) = \langle(u_1^2, u_2^2, \sqrt{2}u_1 u_2), (v_1^2, v_2^2, \sqrt{2}v_1 v_2)\rangle$

Пример нахождения пространства H

Пусть $X = \mathbb{R}^2, K(u,v) = \langle u, v\rangle^2$ при $u = (u_1, u_2), v = (v_1, v_2)$.
Хотим найти H и $\varphi : X \to H$, т.ч. $K(x_1, x_2) = \langle \varphi(x_1), \varphi(x_2)\rangle$.
Сделаем эквивалентные преобразования:
$K(u,v) = \langle u, v\rangle^2 = \langle (u_1, u_2), (v_1, v_2)\rangle^2 = (u_1 v_1 + u_2 v_2)^2 = u_1^2 v_1^2 + u_2^2 v_2^2 + 2 u_1 v_1 u_2 v_2$
$K(u,v) = \langle (u_1^2, u_2^2, \sqrt{2} u_1 u_2), (v_1^2, v_2^2, \sqrt{2} v_1 v_2)\rangle$
Т.о., $H = \mathbb{R}^3$ и $\varphi : (u_1, u_2) \mapsto (u_1^2, u_2^2, \sqrt{2} u_1 u_2)$.

Пример нахождения пространства H

Пусть $X = \mathbb{R}^2, K(u, v) = \langle u, v\rangle^2$ при $u = (u_1, u_2), v = (v_1, v_2)$.
Хотим найти H и $\varphi : X \to H$, т.ч. $K(x_1, x_2) = \langle \varphi(x_1), \varphi(x_2)\rangle$.
Сделаем эквивалентные преобразования:
$K(u, v) = \langle u, v\rangle^2 = \langle (u_1, u_2), (v_1, v_2)\rangle^2 = (u_1 v_1 + u_2 v_2)^2 = u_1^2 v_1^2 + u_2^2 v_2^2 + 2u_1 v_1 u_2 v_2$
$K(u, v) = \langle (u_1^2, u_2^2, \sqrt{2}u_1 u_2), (v_1^2, v_2^2, \sqrt{2}v_1 v_2)\rangle$
Т.о., $H = \mathbb{R}^3$ и $\varphi : (u_1, u_2) \mapsto (u_1^2, u_2^2, \sqrt{2}u_1 u_2)$.
Линейной поверхности в H будет соответствовать квадратичная поверхность в X.

Примеры ядер для SVM

- Полиномиальное ядро с мономами степени d: $K(x_1, x_2) = \langle x_1, x_2 \rangle^d$

Примеры ядер для SVM

- Полиномиальное ядро с мономами степени d: $K(x_1, x_2) = \langle x_1, x_2 \rangle^d$
- Полиномиальное ядро с мономами степени $\leq d$: $K(x_1, x_2) = (\langle x_1, x_2 \rangle + 1)^d$

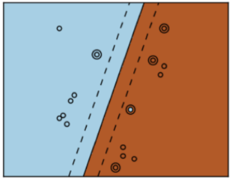
Примеры ядер для SVM

- Полиномиальное ядро с мономами степени d: $K(x_1, x_2) = \langle x_1, x_2 \rangle^d$
- Полиномиальное ядро с мономами степени $\leq d$: $K(x_1, x_2) = (\langle x_1, x_2 \rangle + 1)^d$
- Радиальное ядро (RBF): $K(x_1, x_2) = \exp(-\gamma ||x_1 - x_2||^2)$ (наиболее универсальное)

Примеры ядер для SVM

- Полиномиальное ядро с мономами степени d: $K(x_1, x_2) = \langle x_1, x_2 \rangle^d$
- Полиномиальное ядро с мономами степени $\leq d$: $K(x_1, x_2) = (\langle x_1, x_2 \rangle + 1)^d$
- Радиальное ядро (RBF): $K(x_1, x_2) = \exp(-\gamma ||x_1 - x_2||^2)$ (наиболее универсальное)

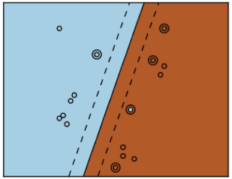
Линейное ядро
$K(x_1, x_2) = \langle x_1, x_2 \rangle$



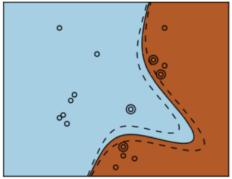
Примеры ядер для SVM

- Полиномиальное ядро с мономами степени d: $K(x_1, x_2) = \langle x_1, x_2 \rangle^d$
- Полиномиальное ядро с мономами степени $\leq d$: $K(x_1, x_2) = (\langle x_1, x_2 \rangle + 1)^d$
- Радиальное ядро (RBF): $K(x_1, x_2) = \exp(-\gamma ||x_1 - x_2||^2)$ (наиболее универсальное)

Линейное ядро
$K(x_1, x_2) = \langle x_1, x_2 \rangle$

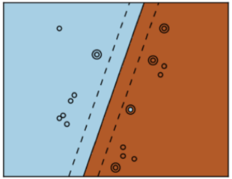
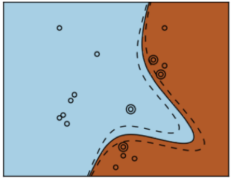
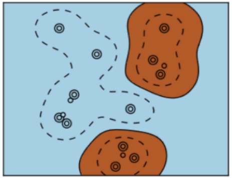


Полиномиальное ядро
$K(x_1, x_2) = (\langle x_1, x_2 \rangle + 1)^3$



Примеры ядер для SVM

- Полиномиальное ядро с мономами степени d: $K(x_1, x_2) = \langle x_1, x_2 \rangle^d$
- Полиномиальное ядро с мономами степени $\leq d$: $K(x_1, x_2) = (\langle x_1, x_2 \rangle + 1)^d$
- Радиальное ядро (RBF): $K(x_1, x_2) = \exp(-\gamma ||x_1 - x_2||^2)$ (наиболее универсальное)

Линейное ядро	Полиномиальное ядро	Радиальное ядро
$K(x_1, x_2) = \langle x_1, x_2 \rangle$	$K(x_1, x_2) = (\langle x_1, x_2 \rangle + 1)^3$	$K(x_1, x_2) = \exp(-||x_1 - x_2||^2)$
		

Многоклассовый SVM

Предположим, что нужно построить классификатор методом опорных векторов для задачи классификации с количеством классов $|Y| = N > 2$. Тогда возможны 2 варианта:

Многоклассовый SVM

Предположим, что нужно построить классификатор методом опорных векторов для задачи классификации с количеством классов $|Y| = N > 2$. Тогда возможны 2 варианта:

Стратегия “один-против-всех”

Обучаем N бинарных SVM-классификаторов, каждый из которых отделяет некоторый класс от остальных $N-1$ классов.
Затем в качестве класса берем $a(x, w, w_0) = \arg\max_{c \in Y}(\langle w^c, x\rangle - w_0^c)$

Многоклассовый SVM

Предположим, что нужно построить классификатор методом опорных векторов для задачи классификации с количеством классов $|Y| = N > 2$. Тогда возможны 2 варианта:

Стратегия “один-против-всех”

Обучаем N бинарных SVM-классификаторов, каждый из которых отделяет некоторый класс от остальных $N - 1$ классов.

Затем в качестве класса берем $a(x, w, w_0) = \arg\max_{c \in Y}(\langle w^c, x\rangle - w_0^c)$

Стратегия “каждый-против-каждого”

Обучаем $\frac{N(N-1)}{2}$ бинарных SVM-классификаторов, каждый из которых отделяет между собой некоторую пару классов.

В результате применения классификаторов получаем $\frac{N(N-1)}{2}$ доминирующих классов.

Итоговый класс выбирается большинством голосов.

Плюсы и минусы SVM

Плюсы

- Наглядная оптимизационная модель
- Задача имеет единственное решение
- Легко обобщается для нелинейной классификации

Плюсы и минусы SVM

Плюсы

- Наглядная оптимизационная модель
- Задача имеет единственное решение
- Легко обобщается для нелинейной классификации

Минусы

- Непонятно, как подбирать ядро в конкретном случае
- Подбор константы C
- Решение задачи квадратичного программирования, особенно с экзотическими ядрами, может занять много времени

Время для вопросов