ТЕХНОСФЕРА

Лекция 9
Support Vector Machine

Кристина Федоренко

28 ноября 2016 г.

План занятия

Линейная разделимость

SVM

Функции ядра

SGD, Метод Ньютона

Постановка задачи

Дано. Признаковые описания N объектов $\mathbf{x} = (x_1, \ldots, x_m) \in \mathcal{X}$, образующие тренировочный набор данных X, и значения целевой переменной $y = f(\mathbf{x}) \in \mathcal{Y}$ для каждого объекта из X.

Найти. Для семейства параметрических функций

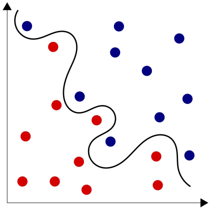
$$H = \{h(\mathbf{x}, \theta) = y : \mathcal{X} \times \Theta \to \mathcal{Y}\},$$

найти значение вектора параметров θ^*, такое что $h^*(\mathbf{x}) = h(\mathbf{x}, \theta^*)$ наилучшим образом приближает целевую функцию.

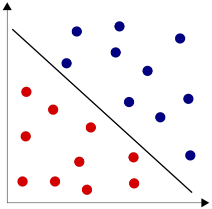
$$\begin{aligned} Y &\in \{C_1, C_2, \ldots, C_K\} \text{ – задача классификации} \\ Y &\in [a, b] \subset \mathcal{R} \text{ – задача регрессии} \end{aligned}$$

Определение

Два множества точек в n-мерном пространстве линейно разделимы, если они могут быть отделены $(n-1)$-мерной гиперплоскостью.

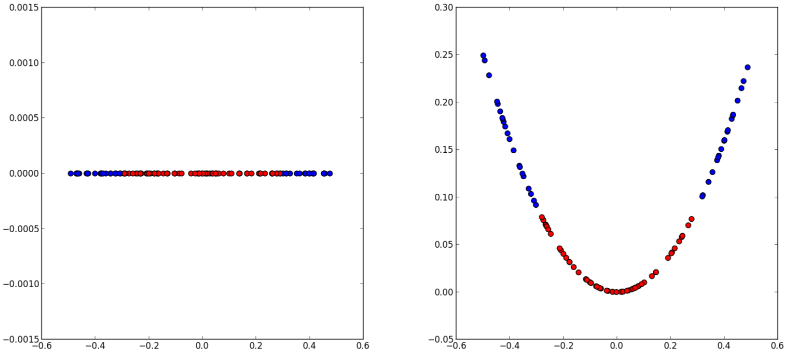


(a) Линейно неразделимые данные



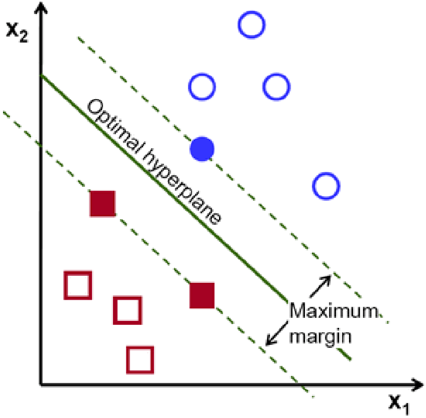
(b) Линейно разделимые данные

Мотивация


0.0015
0.0010
0.0005
0.0000
−0.0005
−0.0010
−0.0015
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6

Support Vector Machine

Интуиция



Постановка задачи

Дана обучающая выборка $\mathbf{x} = (x_1, \ldots, x_m)$, $x_i \in R^N$, $y_i \in \{-1, 1\}$.

Гипотеза

$$h(\mathbf{x}, \theta) = sign(\theta^\top \phi(\mathbf{x}) + \theta_0),$$

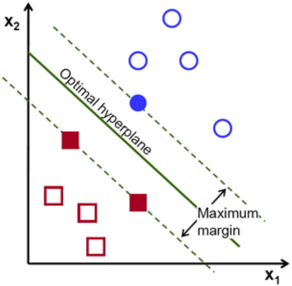
где $\theta = (\theta_1, \ldots, \theta_m)$,
ϕ – функция преобразования исходных признаков.

Максимальный зазор

Margin – наименьшее расстояние между разделяющей плоскостью (РП) и обучающим объектом.

$$d_j = \frac{|\theta^T \phi(\mathbf{x}) + \theta_0|}{\|\theta\|} =$$

$$= \frac{y_j(\theta^T \phi(\mathbf{x}) + \theta_0)}{\|\theta\|}$$



Оптимальная РП

$$\arg\max_{\theta,\theta_0} \left[\frac{1}{\|\theta\|} \min_j y_j(\theta^\top \phi(\mathbf{x}_j) + \theta_0) \right]$$

Задача оптимизации

Расстояние от точки x_j до РП

$$d_j = \frac{y_j(\theta^\top \phi(\mathbf{x}_j) + \theta_0)}{\|\theta\|}$$

Для точки x_j, лежащей на минимальном расстоянии от РП положим

$$y_j(\theta^\top \phi(\mathbf{x}_j) + \theta_0) = 1$$

Задача оптимизации

$$\frac{1}{2}\|\theta\|^2 \to \min_{\theta, \theta_0}$$

при условиях

$$y_j(\theta^\top \phi(\mathbf{x}_j) + \theta_0) \geq 1, \quad \forall j \in 1, \dots, N$$

Метод множителей Лагранжа

Задача
$f(x, y) \to \max_{x,y}$
При ограничениях
$g(x, y) = c.$

Решение
$\mathcal{L}(x, y, a) = f(x, y) - a(g(x, y) - c)$, a – **множитель Лагранжа**, $\mathcal{L}$ – **функция лагранжа или лагранжиан**.
Составим систему уравнений, приравняв к нулю $\frac{\partial \mathcal{L}}{\partial x}$, $\frac{\partial \mathcal{L}}{\partial y}$, $\frac{\partial \mathcal{L}}{\partial a}$.
Решение системы может быть условным экстремумом, то есть решением исходной задачи.

Упражнение

$f(x, y) = x + y \to \max_{x,y}$
При ограничениях
$x^2 + y^2 = 1$

Решите задачу методом Лагранжа.

Метод множителей Лагранжа $\mathbf{a} = (a_1, \ldots, a_N)^\top, \; a_i \geq 0$.

$$\mathcal{L}(\theta, \theta_0, \mathbf{a}) = \frac{1}{2}\|\theta\|^2 - \sum_{j=1}^{N} a_j [y_j(\theta^\top \phi(\mathbf{x}_j) + \theta_0) - 1]$$

Дифференцируем по θ и θ_0

$$\theta = \sum_{j=1}^{N} a_j y_j \phi(\mathbf{x}_j), \quad 0 = \sum_{j=1}^{N} a_j y_j$$

Подставляем θ и θ_0 в лагранжиан

Сопряженная задача

Сопряженная задача

$$\tilde{\mathcal{L}}(\mathbf{a}) = \sum_{j=1}^{N} a_j - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j y_i y_j \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) \to \max_{\mathbf{a}}$$

при условиях

$$a_j \geq 0, \quad \forall j \in 1, \dots, N$$

$$\sum_{j=1}^{N} a_j y_j = 0$$

Наблюдения

- $k(x_i, x_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$ – неотрицательно-определенная функция
- лагранжиан $\tilde{\mathcal{L}}(\mathbf{a})$ – выпуклая и ограниченная сверху функция

Классификация

Функция принятия решения

$$h(x,\theta,\theta_0) = \theta^\top \phi(\mathbf{x}) + \theta_0 = \sum_{j=1}^{N} a_j y_j \phi(\mathbf{x}_j)^\top \phi(\mathbf{x}) + \theta_0 = \sum_{j=1}^{N} a_j y_j k(\mathbf{x}_j, \mathbf{x}) + \theta_0$$

Условия Karush-Kuhn-Tucker

$$\begin{aligned} a_j &\geq 0 \\ y_j(\theta^\top \phi(x_j) + \theta_0) - 1 &\geq 0 \\ a_j\{y_j(\theta^\top \phi(x_j) + \theta_0) - 1\} &= 0 \end{aligned}$$

Опорным векторам $\mathbf{x}_j \in S$ соответствуют $a_j > 0$

$$\theta_0 = \frac{1}{N_s} \sum_{i \in S} \left(y_i - \sum_{j \in S} a_j y_j k(\mathbf{x_i}, \mathbf{x_j}) \right)$$

Линейно-разделимый случай

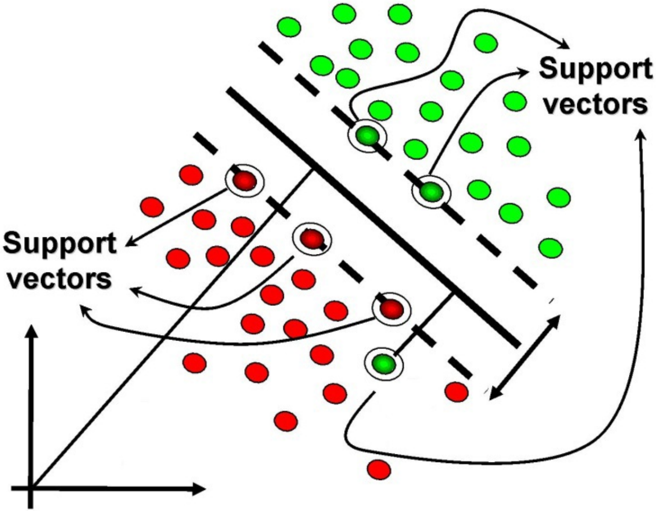
Задача

Дана обучающая выборка

	x_1	x_2	t
$\mathbf{x}_1$	1	-2	1
$\mathbf{x}_2$	1	2	-1

Найти оптимальную разделяющую плоскость, используя сопряженную задачу оптимизации

Линейно-неразделимый случай



Смягчение ограничений

Переменные $\xi_j \geq 0$ (slacks):

$$\xi_j = \begin{cases} 0, & \text{если } h(x, \theta, \theta_0) y_j \geq 1 \\ |y_j - h(x, \theta, \theta_0)|, & \text{иначе} \end{cases}$$

Задача оптимизации

$$C \sum_{j=1}^{N} \xi_j + \frac{1}{2} \|\theta\|^2 \to \min_{\theta, \theta_0}$$

при условиях

$$y_j h(x, \theta, \theta_0) \geq 1 - \xi_j, \;\; \xi_j \geq 0$$

Сопряженная задача

Сопряженная задача

$$\tilde{\mathcal{L}}(\mathbf{a}) = \sum_{j=1}^{N} a_j - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j y_i y_j \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) \to \max_{\mathbf{a}}$$

при условиях

$$0 \leq a_j \leq C, \quad \forall j \in 1, \ldots, N$$

$$\sum_{j=1}^{N} a_j y_j = 0$$

- $a_j = 0$ – правильно проклассифицированные объекты
- $a_j = C$ – опорные векторы внутри отступа
- $0 < a_j < C$ – опорные векторы на границе

Классификация

Функция принятия решения

$$h(\mathbf{x}, \theta, \theta_0) = \sum_{j=1}^{N} a_j y_j k(\mathbf{x}_j, \mathbf{x}) + \theta_0$$

Константа b

$$\theta_0 = \frac{1}{N_{\mathcal{M}}} \sum_{i \in \mathcal{M}} \left(y_i - \sum_{j \in S} a_j y_j k(\mathbf{x_i}, \mathbf{x_j}) \right)$$

Функции ядра

Функции ядра

$\phi(\mathbf{x})$ – функция преобразования $\mathbf{x}$ из исходного пространства в спрямляющее пространство

Проблема: количество признаков может быть очень велико

Идея Kernel Trick

В процессе тренировки и применения SVM исходные векторы $\mathbf{x}$ используются только как аргументы в скалярном произведении $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x_j})$. Но в этом случае можно избежать вычисления $\varphi(\mathbf{x})$!

Теорема Мерсера

Теорема

Функция $k(\mathbf{x}, \mathbf{z})$ является ядром тогда и только тогда, когда она

- симметрична
$$k(\mathbf{x}, \mathbf{z}) = k(\mathbf{z}, \mathbf{x})$$
- неотрицательно определена
$$\int_{\mathbf{x} \in \mathbf{X}} \int_{\mathbf{z} \in \mathbf{X}} k(\mathbf{x}, \mathbf{z}) g(\mathbf{x}) g(\mathbf{z}) d\mathbf{x} d\mathbf{z} \geqslant 0, \quad \forall g(\mathbf{x}) : \mathbf{X} \to R$$

Упражнение

Пусть $\mathbf{x} \in R^2$, а преобразование $\phi(\mathbf{x})$

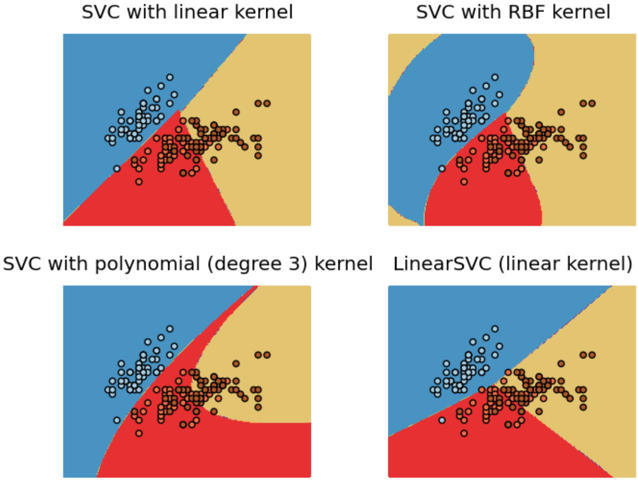
$$\phi(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2).$$

Проверить, что функция $k(\mathbf{x}, \mathbf{z}) = (1 + \mathbf{x}^\top \mathbf{z})^2$ является функцией ядра для данного преобразования.

Некоторые стандартные функции ядра

- Линейное ядро
$$k(\mathbf{x}, \mathbf{z}) = \mathbf{x}^\top \mathbf{z}$$
- Полиномиальное ядро степени d
$$k(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^\top \mathbf{z} + r)^d$$
- Radial Basis Function
$$k(\mathbf{x}, \mathbf{z}) = e^{-\gamma |\mathbf{x} - \mathbf{z}|^2}$$
- Sigmoid
$$k(\mathbf{x}, \mathbf{z}) = \tanh(\gamma \mathbf{x}^\top \mathbf{z} + r)$$

Ирисы и SVM


SVC with linear kernel
SVC with RBF kernel
SVC with polynomial (degree 3) kernel
LinearSVC (linear kernel)

Связь с линейными моделями

Задача оптимизации

$$C\sum_{j=1}^{N}\xi_j + \frac{1}{2}\|\theta\|^2 \sim \sum_{j=1}^{N} E(h(\mathbf{x}_j), y_j) + \lambda\|\theta\|^2 \to \min_{\theta,\theta_0}$$

Hinge loss

$$E(h(\mathbf{x}_j), y_j) = \begin{cases} 1 - h(\mathbf{x}_j)y_j, & \text{если } h(\mathbf{x}_j)y_j < 1 \\ 0, & \text{иначе} \end{cases}$$

Stochastic Gradient Descent

Stochastic Gradient Descent

```
function gd(J, alpha, epsilon):
    initialise theta
    do:
        randomly shuffle examples in the training set
        theta = new_theta
        for i=1,2,...,n do
            new_theta = theta - alpha*grad_i(theta)
    until dist(new_theta, theta) < epsilon
    return theta
```

SGD tips

- Использовать SGD, когда обучение модели занимает слишком много времени
- Перемешать тренировочную выборку
- Следить за training error и **validation error**
- Подобрать α на небольшой выборке

Метод Ньютона

$$J(\theta) \approx J(\theta_k) + \nabla J(\theta_k)^T(\theta - \theta_k) + \frac{1}{2}(\theta - \theta_k)^T \nabla^2 J(\theta_k)(\theta - \theta_k) \to \min_\theta$$

$$\theta = \theta_k - \nabla^2 J(\theta_k)^{-1} \nabla J(\theta_k)$$

```
function newton(grad, hessian, a0, epsilon, alpha):
    initialise eta(k)
    k = 0
    a = a0
    do:
        k = k + 1
        g = grad(a)
        H = hessian(a)
        d = solve(H * d = −g) # find d = − inv(H) * g
        a = a + alpha d
    until convergence
    return a
```

BFGS – использовать приближение $\nabla^2 J(\mathbf{a}_k)$ или $\nabla^2 J(\mathbf{a}_k)^{-1}$

SVM – итоги

- \+ Нелинейная разделяющая поверхность
- \+ Глобальная оптимизация
- \+ Разреженное решение
- \+ Хорошая обобщающая способность
- \- Не возвращает вероятность
- \- Чувствительность к выбросам
- \- Нет алгоритма выбора ядра
- \- Медленное обучение

Вопросы