SVM. От спрямляющих пространств к ядерным функциям

Подготовили: Зимин И Жилин Андреи

Введение

о SVM пару слов.

Функции потерь для градиентных методов:

Hinge loss

$$F(M) = max(0, 1 - M)$$

Ошибка перцептрона

$$F(M) = max(0, -M)$$

Градиент?

Молодцы

Действительно важный вопрос, что такое M

$$M = y_i < w, x_i >$$

это расстояние до разделяющей гиперплоскости, просто переобозначили для удобства записи

Введение

о SVM пару слов.

Функции потерь для градиентных методов:

Hinge loss
$$F(M) = max(0, 1 - M)$$

Ошибка перцептрона
$$F(M) = max(0, -M)$$

Градиент

$$\nabla L(w) = \begin{cases} 0, y_i < w, x_i > \leq 0 \\ 1, y_i < w, x_i >> 0 \end{cases}$$

Какое ядро у SVM?

Загадка……..

Какое ядро у SVM?

На самом деле неизвестно

Какое ядро у SVM?

На самом деле неизвестно

Ядро линейное, но оно подается в виде комбинации признаков, пользуясь трюком множество ядер можно представить в виде линейного

Математически

$$X^* = (x_1^*, x_2^*, \dots, x_n^*)$$ количество начальных признаков.

$$x_i^* = (f_1(x_1), f_2(x_2), \dots, f(x_r))$$ r - количество подаваемых признаков

Спрямляющее пространство

По факту спрямляющее пространство это оператор перехода

$$\psi : X \to H, K(x, y) =< \psi(x), \psi(y) >$$

Соответственно, мы наше пространство пытаемся представить в виде такого набора признаков, с помощью которых мы сможем линейно разделить наши данные.

$$K(x, y) =< x, y >^2, x, y \in \mathbb{R}^2, K^*(x^*, y^*) =< x^*, y^* >, x*, y* \in \mathbb{R}^d$$

$$d-?, \psi-?$$

Спрямляющее пространство

По факту спрямляющее пространство это оператор перехода

$$\psi : X \to H, K(x, y) =< \psi(x), \psi(y) >$$

Соответственно, мы наше пространство пытаемся представить в виде такого набора признаков, с помощью которых мы сможем линейно разделить наши данные.

Примеры перехода для R^2

$$K(x, y) =< x, y >^2, x, y \in \mathbb{R}^2, K^*(x^*, y^*) =< x^*, y^* >, x*, y* \in \mathbb{R}^d$$

$$H = \mathbb{R}^3, \psi : (x_1, x_2) \to (x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

Ядра и их спрямляющие пространства на примере двумерных пространств

$$K(x,y) = < x, y >, x, y \in \mathbb{R}^d$$

какое ядро?

$$K(x,y) = < x, y >^2, x, y \in \mathbb{R}^d$$

какое ядро?

$$K(x,y) = (< x, y > + r)^n, r \geq 0, n \geq 1, x, y \in \mathbb{R}^d$$

какое ядро?

$$K(x,y) = exp(-\frac{||x-y||^2}{2\sigma^2}), x, y \in \mathbb{R}^d, \sigma > 0$$

Ядра и их спрямляющие пространства на примере двумерных пространств

$$K(x,y) =< x,y >, x,y \in \mathbb{R}^d$$

$$\psi : (x_1, x_2) \to (x_1, x_2)$$

$$K(x,y) =< x,y >^2, x,y \in \mathbb{R}^d$$

$$\psi : (x_1, x_2) \to (x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

$$K(x,y) = (< x,y > +r)^n, r \geq 0, n \geq 1, x,y \in \mathbb{R}^d$$

$$K(x,y) = exp(-\frac{||x-y||^2}{2\sigma^2}), x,y \in \mathbb{R}^d, \sigma > 0$$

1. $K(x,x') = \langle x,x'\rangle^2$
   — квадратичное ядро;
2. $K(x,x') = \langle x,x'\rangle^d$
   — полиномиальное ядро с мономами степени d;
3. $K(x,x') = \bigl(\langle x,x'\rangle + 1\bigr)^d$
   — полиномиальное ядро с мономами степени $\leqslant d$;
4. $K(x,x') = \sigma\bigl(\langle x,x'\rangle\bigr)$
   — нейросеть с заданной функцией активации $\sigma(z)$
   (не при всех σ является ядром);
5. $K(x,x') = \mathrm{th}\bigl(k_1\langle x,x'\rangle - k_0\bigr),\ \ k_0, k_1 \geqslant 0$
   — нейросеть с сигмоидными функциями активации;
6. $K(x,x') = \exp\bigl(-\beta\|x - x'\|^2\bigr)$
   — сеть радиальных базисных функций (RBF ядро);

Чем плохо?

Для параболических функций в R^2 это не столь очевидно, потому что у нас было два признака, а стало три.

Две явных проблемы присутствует:

1) Сильно сложные ядра, дают сильный рост количества признаков, особенно, если большая изначальная размерность

Однако для размерности d в случае

$$K(x, y) = < x, y >^2, x, y \in \mathbb{R}^d$$

размерность будет d + d! Очень легко это доказать

Чем плохо?

Две явных проблемы присутствует:

1) Сильно сложные ядра, дают сильный рост количества признаков, особенно, если большая изначальная размерность
2) Не все ядра представимы в виде конечного набора признаков в спрямляющем пространстве

Например гауссово ядра Формула для R^2

$$H = \psi : (x_1, x_2) \to C * (1, x_1, \frac{x_1^2}{2!}, \ldots, \frac{x_1^r}{r!})^T (1, x_2, \frac{x_2^2}{2!}, \ldots, \frac{x_2^r}{r!})$$

$$C = \exp(-\frac{||x||^2 + ||y||^2}{2})$$

Спрямляющее пространство для Гаусса

Заметно, что RBF-ядро не представимо в виде конечномерного пространства, потому приходится выкручиваться используя аппроксимацию рядом Тейлора, что вообще очень плохо, можно сказать, что мы возвращаемся таким образом к первой проблеме. Проблеме большого количества размерностей.

Этому есть решение в задаче квадратичного программирования

Постановка задачи

$$\max_{\lambda} W(\lambda) = \sum_{i=1}^{n} \lambda_i - 0.5 \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j k(x_i, x_j)$$

$$\forall i \; 0 \leq \lambda_i \leq C$$

$$\sum_{i=1}^{n} \lambda_i y_i = 0$$

Методы оптимизации

Градиентный спуск

Покоординатный спуск

$$\max_{\lambda} W(\lambda) = \sum_{i=1}^{n} \lambda_i - 0.5 \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j k(x_i, x_j)$$

$$\forall i \; 0 \leq \lambda_i \leq C$$

$$\sum_{i=1}^{n} \lambda_i y_i = 0$$

Методы оптимизации

Градиентный спуск

Покоординатный спуск

Sequential Minimal Optimization (SMO)

$$\max_{\lambda} W(\lambda) = \sum_{i=1}^{n} \lambda_i - 0.5 \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j k(x_i, x_j)$$

$$\forall i \; 0 \leq \lambda_i \leq C$$

$$\sum_{i=1}^{n} \lambda_i y_i = 0$$

Fast Training of Support Vector Machines using Sequential Minimal Optimization

Идея алгоритма: разбить задачу на подзадачи, при этом каждую подзадачу решать аналитически

Из-за ограничений мы можем разбивать задачи так, чтобы там было больше одного λi

$$\sum_{i=1}^{n} \lambda_i y_i = 0$$

Рассмотрим подзадачу для двух λ0 λ1

Fast Training of Support Vector Machines using Sequential Minimal Optimization

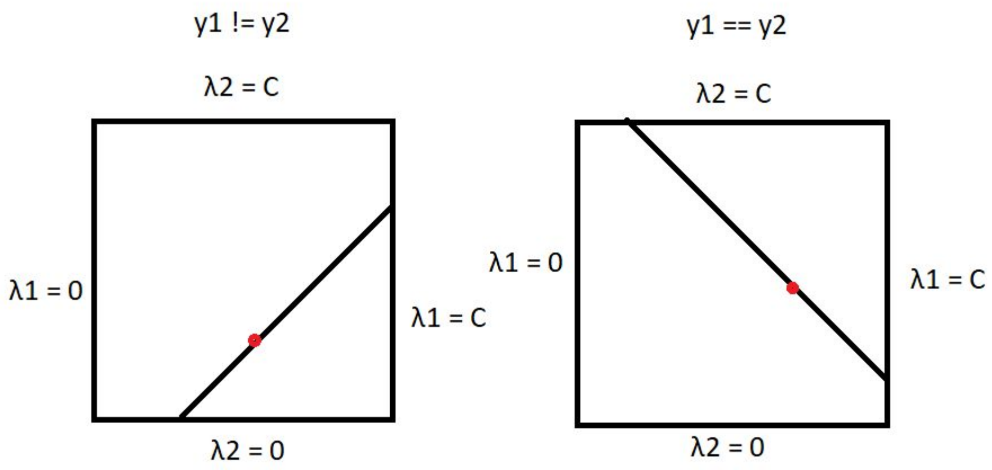
Выбираем произвольные λ0 λ1

$$\max_{\lambda} Q(\lambda_1, \lambda_2) = \lambda_1 + \lambda_2 - \lambda_1 \lambda_2 y_1 y_2 k(x_1, x_2)$$

$$0 \le \lambda_1, \lambda_2 \le C$$

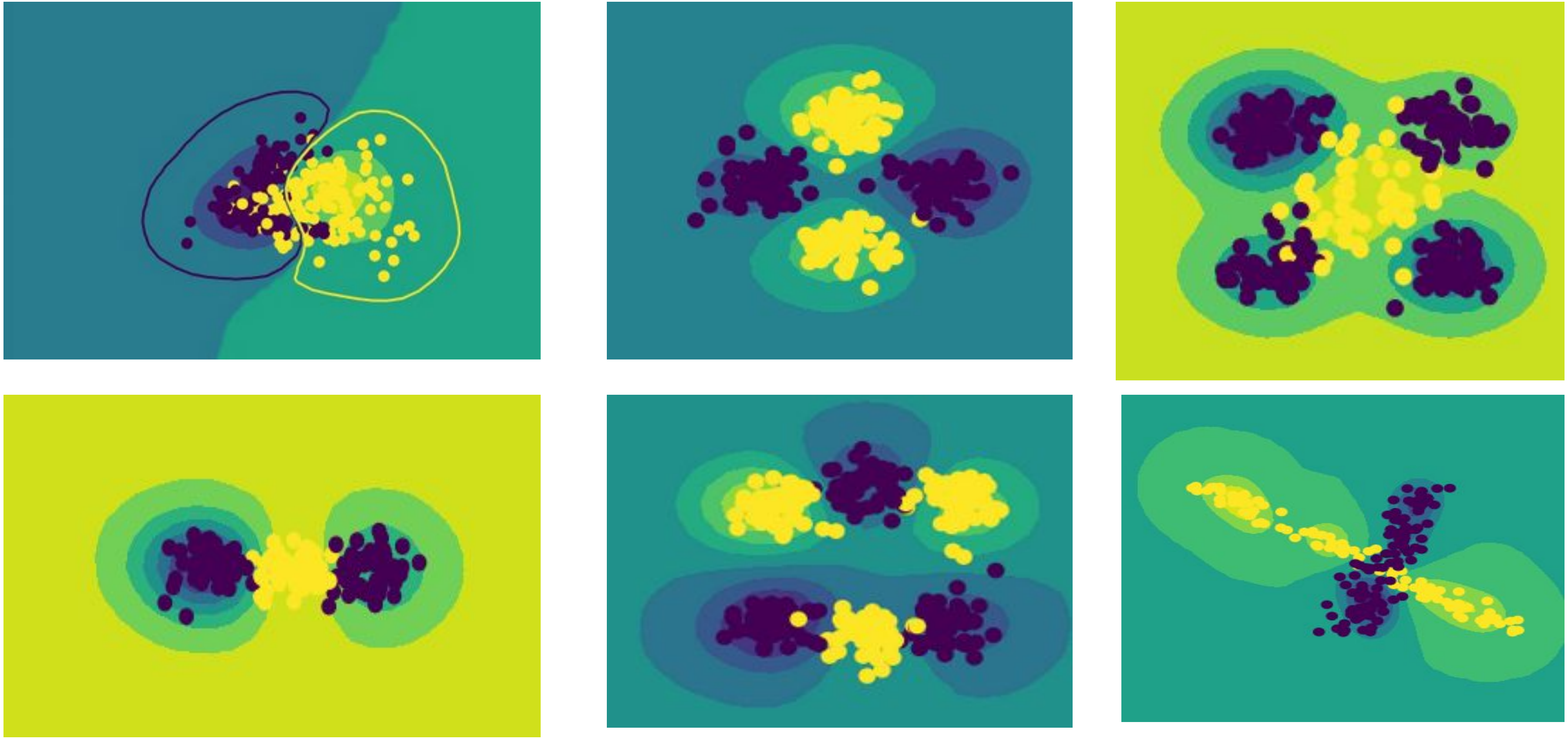
$$\lambda_1 y_1 + \lambda_2 y_2 = t$$

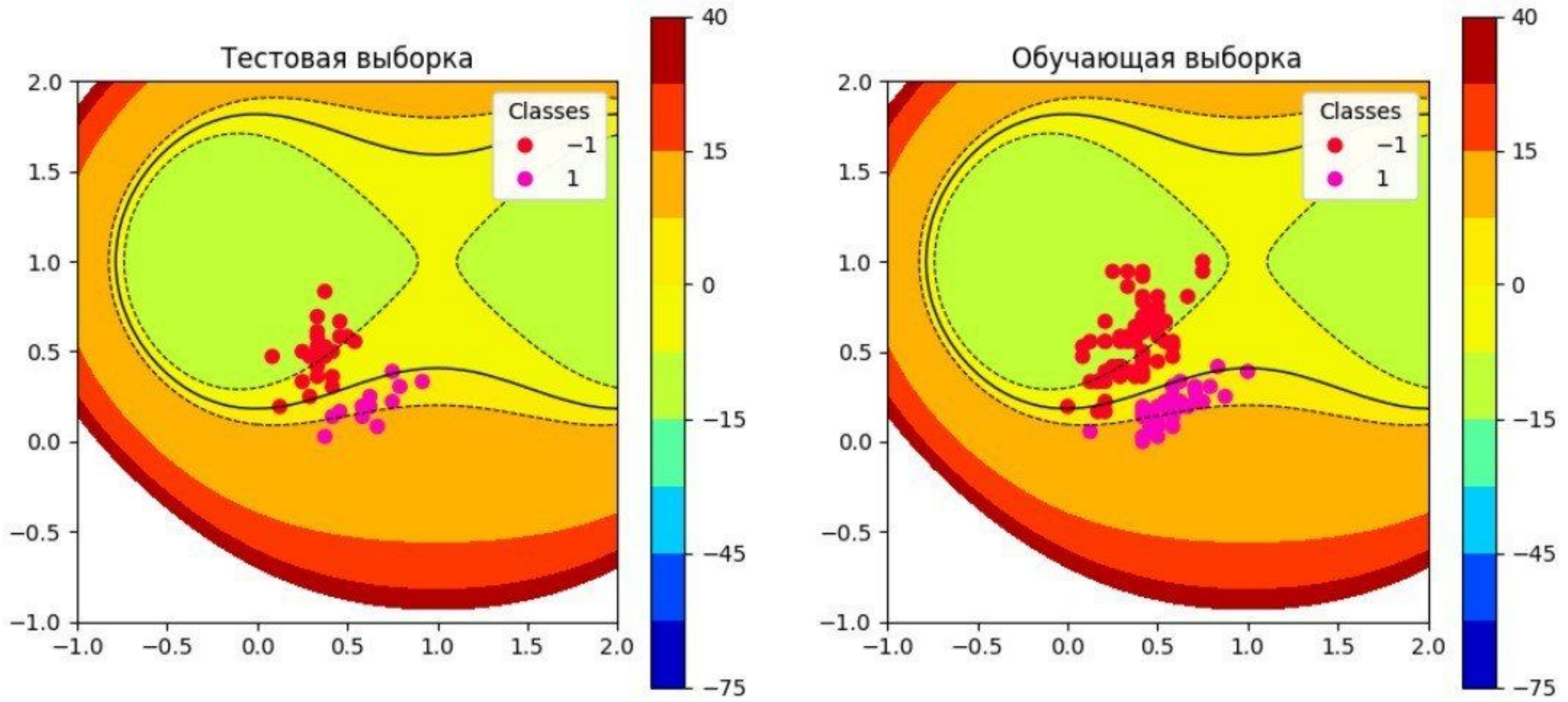
$$t = -\sum_{i=3}^{n} \lambda_i y_i$$



Сходимость обеспечивает теорема Каруша-Куна-Таккера (ККТ)*

Результаты нашей реализации SVM на SMO




Тестовая выборка
Обучающая выборка
Classes
−1
1
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
40
15
0
−15
−45
−75

Литература

Линейные методы классификации и регрессии: метод опорных векторов, лекция Воронцова Константина Вячеславовича : http://www.machinelearning.ru/wiki/images/archive/a/a0/20150316172222%21Voron-ML-Lin-SVM.pdf

Интуитивное понимание пространств и ядер в машинном обучении: https://habr.com/ru/articles/814343/

Лекции по методу опорных векторов К.В. Воронцов:

http://www.ccas.ru/voron/download/SVM.pdf

Спасибо всем