Введение в Data Science
Занятие 5. Машины Опорных Векторов

Николай Анохин Михаил Фирулик

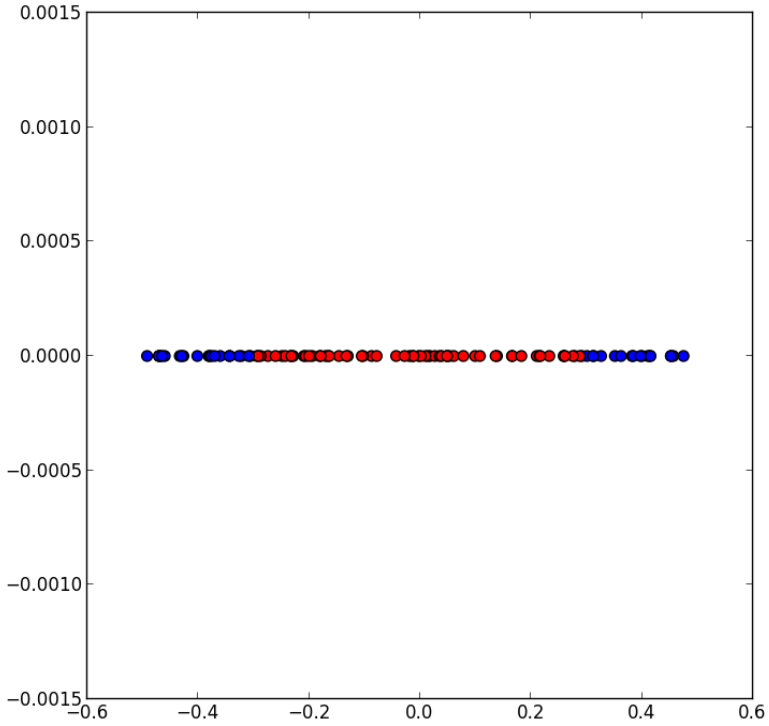
29 марта 2014 г.

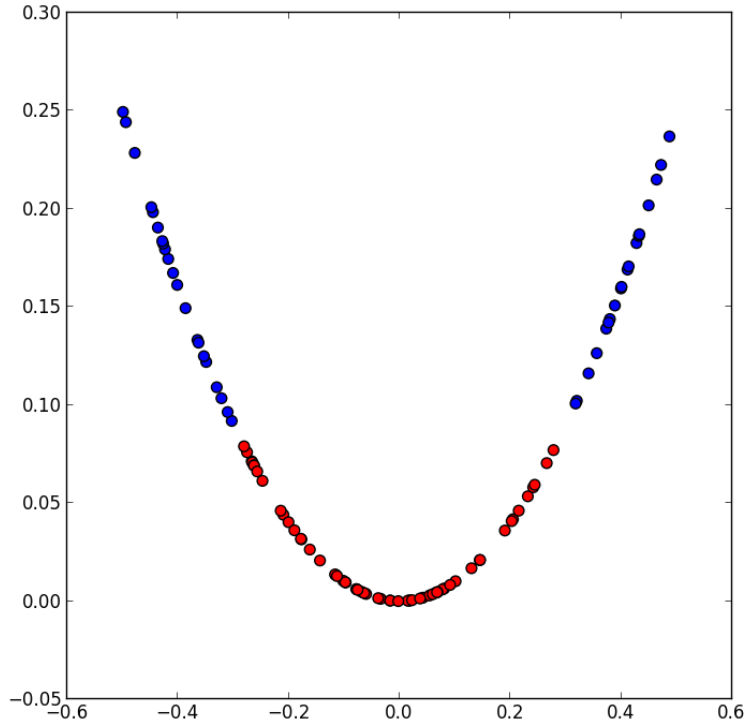
План занятия

Идея maximum margin

Функции ядра

Мотивация





Обобщенные линейные модели

Обучающая выборка

$$\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$$

Значения целевой переменной

$$t_1, \ldots, t_N; \quad t_j \in \{-1, +1\}$$

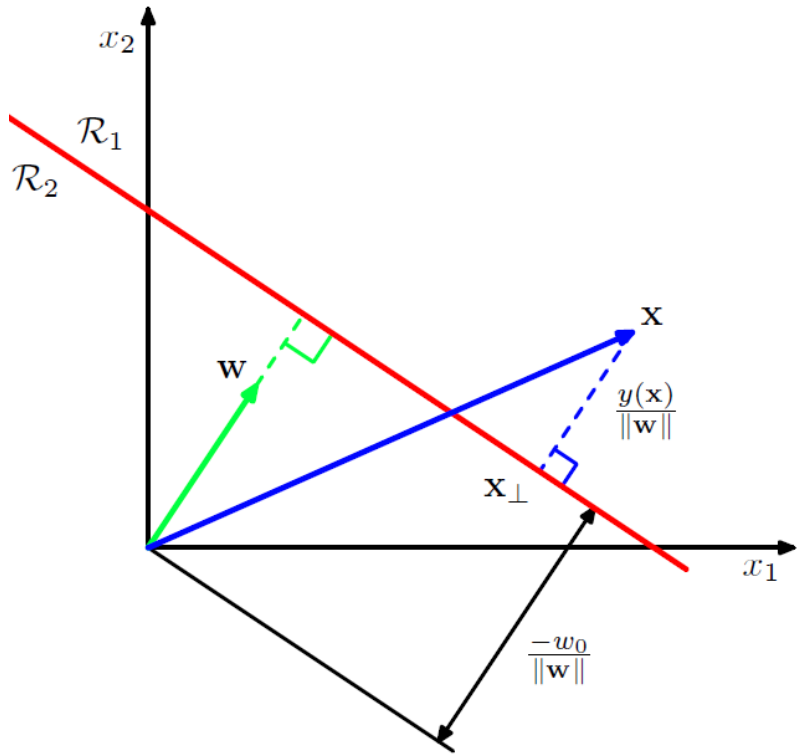
Функция принятия решения

$$y(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b$$

Разделяющая плоскость (РП)

$$y_i(\mathbf{x_i}) t_i > 0$$

Решений много: как выбрать?



Максимальный зазор

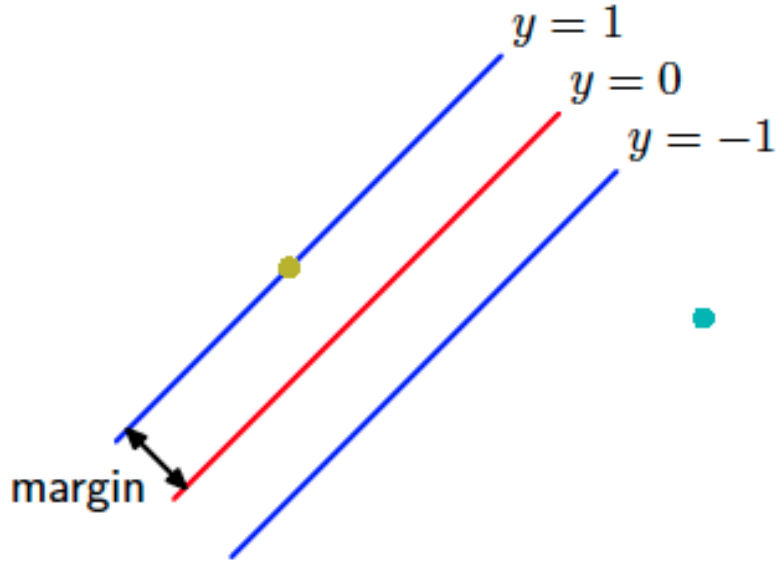
Margin – наименьшее расстояние между РП и обучающим объектом.

$$d_j = \frac{|y(\mathbf{x}_j)|}{\|\mathbf{w}\|} = \frac{t_j y(\mathbf{x}_j)}{\|\mathbf{w}\|} =$$

$$= \frac{t_j(\mathbf{w}^\top \phi(\mathbf{x}_j) + b)}{\|\mathbf{w}\|}$$

Оптимальная РП

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



Задача оптимизации

Расстояние от точки x_j до РП

$$d_j = \frac{t_j(\mathbf{w}^\top \phi(\mathbf{x}_j) + b)}{\|\mathbf{w}\|}$$

Для точки x_j, лежащей на минимальном расстоянии от РП положим

$$t_j(\mathbf{w}^\top \phi(\mathbf{x}_j) + b) = 1$$

Задача оптимизации

$$\frac{1}{2}\|\mathbf{w}\|^2 \to \min_{\mathbf{w},b}$$

при условиях

$$t_j(\mathbf{w}^\top \phi(\mathbf{x}_j) + b) \geq 1, \quad \forall j \in 1, \ldots, N$$

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

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

Дифференцируем по $\mathbf{w}$ и b

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

Подставляем $\mathbf{w}$ и b в лагранжиан

Сопряженная задача

Сорпяженная задача

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

при условиях

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

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

Наблюдения

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

Классификация

Функция принятия решения

$$y(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b = \sum_{j=1}^{N} a_j t_j \phi(\mathbf{x}_j)^\top \phi(\mathbf{x}) + b = \sum_{j=1}^{N} a_j t_j k(\mathbf{x}_j, \mathbf{x}) + b$$

Условия Karush-Kuhn-Tucker

$$\begin{aligned} a_j &\geq 0 \\ t_j y(\mathbf{x_j}) - 1 &\geq 0 \\ a_j \{t_j y(\mathbf{x_j}) - 1\} &= 0 \end{aligned}$$

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

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

Линейно-разделимый случай

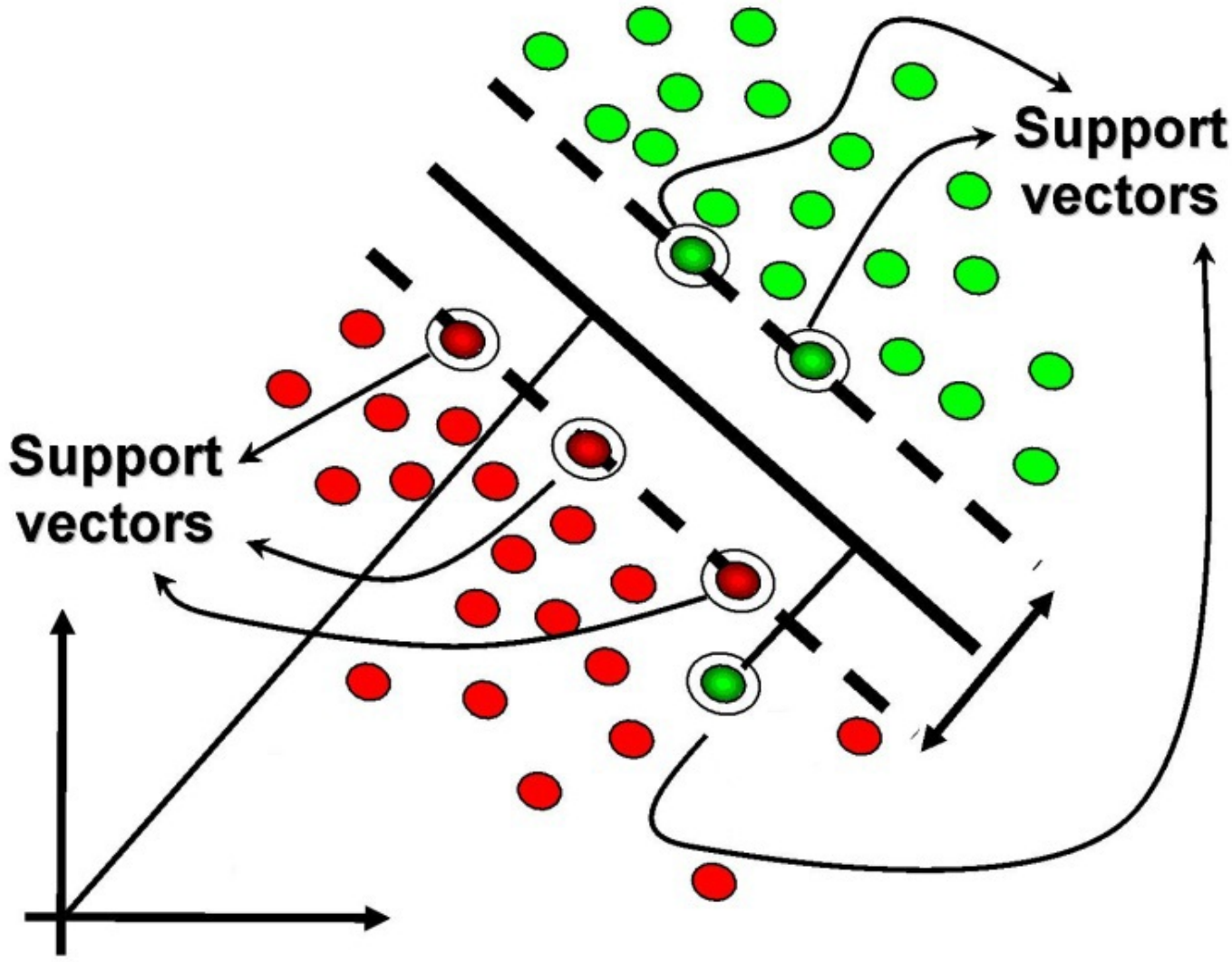
Задача

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



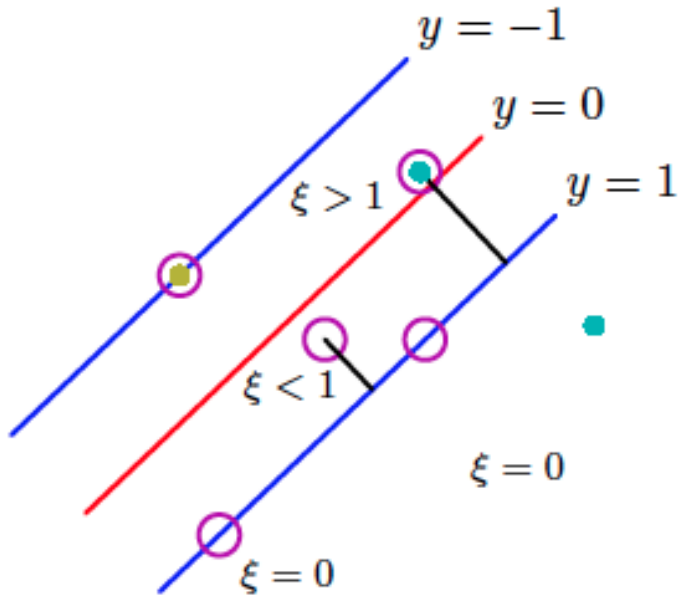
Смягчение ограничений

Переменные $\xi_j \geq 0$ (slacks):

$$\xi_j = \begin{cases} 0, & \text{если } y(\mathbf{x_j})t_j \geq 1 \\ |t_j - y(\mathbf{x}_j)|, & \text{иначе} \end{cases}$$

Задача оптимизации

$$C\sum_{j=1}^{N}\xi_j + \frac{1}{2}\|\mathbf{w}\|^2 \to \min_{\mathbf{w},b}$$



Сопряженная задача

Сорпяженная задача

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

при условиях

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

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

Наблюдения

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

Классификация

Функция принятия решения

$$y(\mathbf{x}) = \sum_{j=1}^{N} a_j t_j k(\mathbf{x}_j, \mathbf{x}) + b$$

Константа b

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

Связь с линейными моделями

Задача оптимизации

$$\sum_{j=1}^{N} \xi_j + \frac{1}{2}\|w\|^2 \sim \sum_{j=1}^{N} E(y(\mathbf{x}_j), t_j) + \lambda\|w\|^2 \to \min_{\mathbf{w}, b}$$

Hinge loss

$$E(y_j, t_j) = \begin{cases} 1 - y_j t_j, & \text{если } y_j t_j > 1 \\ 0, & \text{иначе} \end{cases}$$

Численные методы оптимизации

- Chunking (Vapnik, 1982)
- Decomposition (Osuna, 1996)
- Sequential Minimal Optimization (Platt, 1999)

Задача регрессии

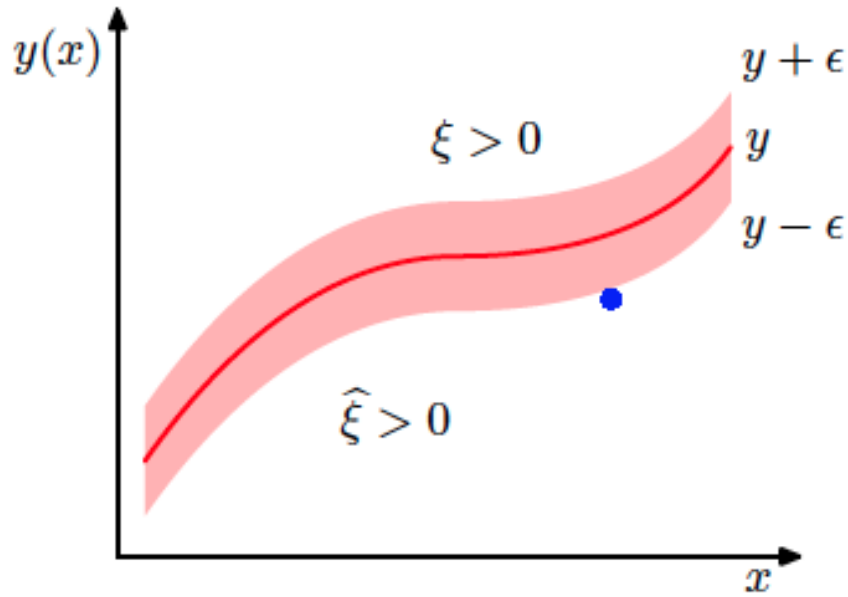
Переменные $\xi_j \geq 0$, $\hat{\xi}_j \geq 0$ (slacks):

$$t_j \leq y(\mathbf{x}_j) + \epsilon + \xi_n$$

$$t_j \geq y(\mathbf{x}_j) - \epsilon - \hat{\xi}_n$$

Задача оптимизации

$$C \sum_{j=1}^{N} (\hat{\xi}_j + \xi_j) + \frac{1}{2}\|\mathbf{w}\|^2 \to \min_{\mathbf{w}, b}$$



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

$$\int_{\mathbf{x} \in \mathbf{X}} \int_{\mathbf{z} \in \mathbf{X}} k(\mathbf{x}, \mathbf{z}) g(\mathbf{x}) g(\mathbf{z}) d\mathbf{x} d\mathbf{z}, \quad \forall g(\mathbf{x}) : \mathbf{X} \to R$$

Задача

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

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

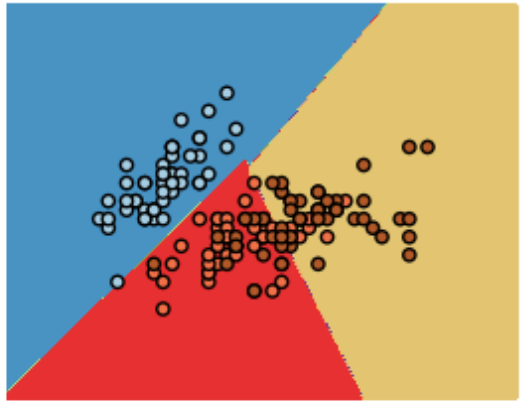
Проверить, что функция $k(\mathbf{x}, \mathbf{z}) = (1 + \mathbf{x}^\top \mathbf{z})^2$ является функцией ядра для данного преобразования.

Некоторые стандартные функции ядра

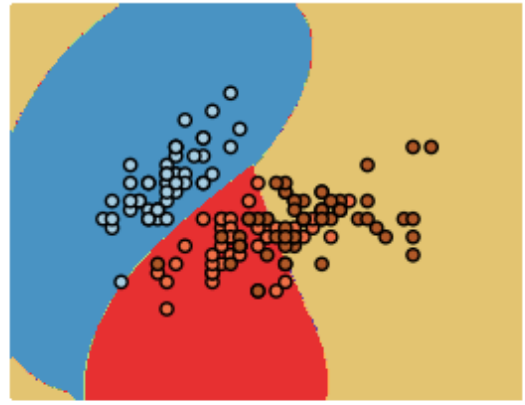
- Линейное ядро
$$k(\mathbf{x}, \mathbf{z}) = \mathbf{x}^\top \mathbf{z}$$
- Полиномиальное ядро степени d
$$k(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^\top \mathbf{z} + r)^d$$
- Radial Basis Function
$$k(\mathbf{x}, \mathbf{z}) = e^{-\gamma|\mathbf{x}-\mathbf{z}|^2}$$
- Sigmoid
$$k(\mathbf{x}, \mathbf{z})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

Опять ирисы

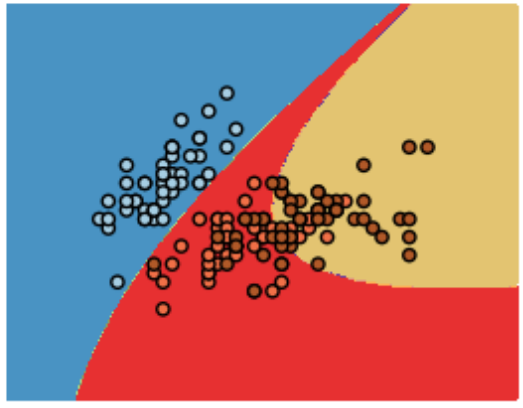
SVC with linear kernel



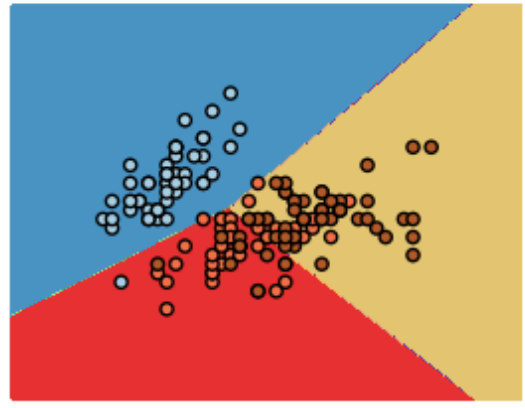
SVC with RBF kernel



SVC with polynomial (degree 3) kernel



LinearSVC (linear kernel)



SVM. Итоги

+ Нелинейная разделяющая поверхность
+ Глобальая оптимизация
+ Разреженное решение
+ Хорошая обобщающая способность
- Не поддерживает $p(C_k|\mathbf{x})$
- Чувствительность к выбросам
- Нет алгоритма выбора ядра
- Медленное обучение

Эксперименты над SVM

В задаче рассматриваются 4 алгоритма генерации обучающих выборок. Требуется разработать функцию, автоматически подбирающую параметры модели SVM в зависимости от полученной выборки.

Инструкция по выполнению задания

1. Выкачать в локальный репо код из ветки svm
2. Запустить хелп, при желании изучить

```
$ python svm.py -h
$ python svm.py cc -h
```

3. Запустить какой-нибудь пример, разобраться с картинкой

```
$ python svm.py gauss
```

4. Работать над кодом (функция `select_model`)

Домашнее задание 4

Машина опорных векторов

Использовать *готовую* имплементацию

- алгоритма SVM для задачи классификации
- алгоритма SVM для задачи регрессии

Ключевые даты

- До 2014/04/05 00.00 предоставить решение задания

Спасибо!

Обратная связь