

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
Математико-механический факультет  
Кафедра системного программирования

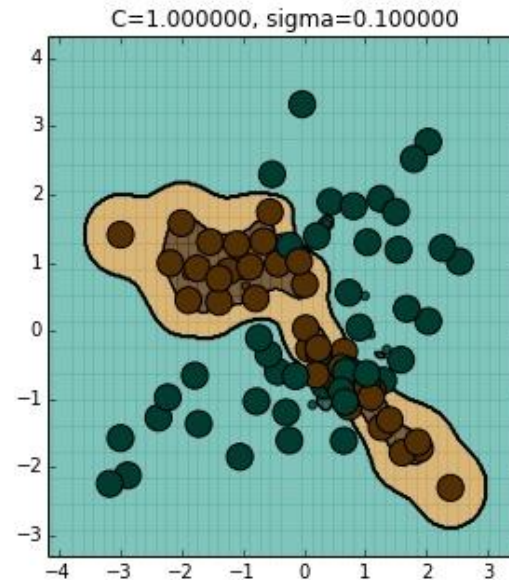
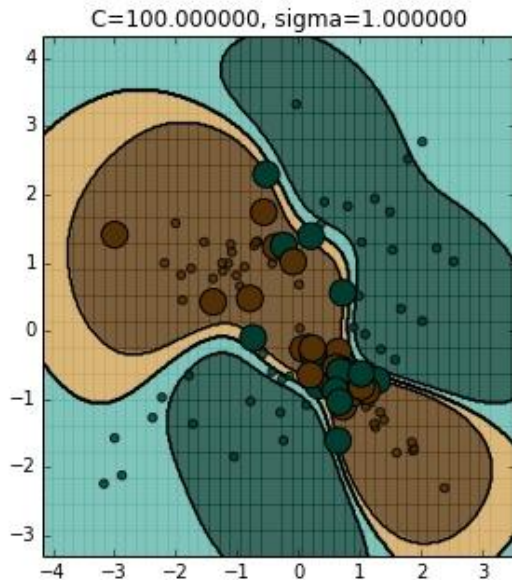
# Машина опорных векторов Метод активных множеств

Атаманова Анна, 344  
Руководитель: Невоструев Константин

# Постановка задачи

- реализация и исследование алгоритма ASM на языке Python
- сравнение алгоритма ASM с другими по скорости и качеству обучения

# Машина опорных векторов (Support Vector Machine)

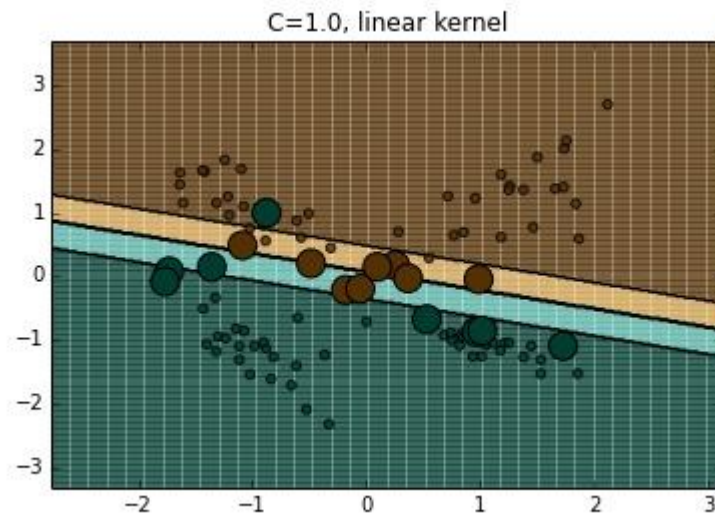


$$\begin{aligned} \frac{1}{2} \lambda^T Q \lambda + C \xi &\rightarrow \min_{\lambda, \xi} \\ \text{s.t. } -Q \lambda + b y + s + \xi &= -e \\ 0 \leq \lambda \leq C, s \geq 0, \xi &\geq 0 \end{aligned}$$

$$a(x) = \sum_i \lambda_i y_i K(x_i, x)$$

$$Q = y_i y_j K(x_i, x_j) \in R^{m \times m}, \lambda, \xi, s \in R^m$$

# Метод активных множеств (Active Set Method)



- $I_0$  – периферийные объекты
- $I_S$  – опорные объекты
- $I_C$  – объекты нарушители

$$\frac{1}{2}\lambda_S^T Q_{SS}\lambda_S + C\lambda_S^T Q_{SC} - e^T\lambda_S \rightarrow \min_{\lambda, \xi}$$
$$s.t. y_S^T\lambda_S + y_C^T C = 0$$

# Проблемы

- Накопление ошибки
  - В итоге, зацикливание
  - Частично решена: сначала пересчитываем, потом перекидываем
- Скорость
  - Python, циклы, итеративный алгоритм
  - Немногожко улучшена, но в целом не решена

# Тестирование

Набор данных Титаник (1526/673, 5)

	Precision [0,1]	Recall [0,1]	F1_score [0,1]	Time, sec
SVM	0.88	0.52	0.66	0.002
ASM	0.8	0.59	0.68	3.037

Набор данных Diabetec (575/191, 8)

	Precision [0,1]	Recall [0,1]	F1_score [0,1]	Time, sec
SVM	0.74	0.84	0.80	0,04
ASM	0.79	0.65	0.71	9,51

Набор данных Adult (300/16280, 123)

	Precision [0,1]	Recall [0,1]	F1_score [0,1]	Time, sec
SVM	0.75	0.64	0.69	0.02
ASM	0.75	0.72	0.74	95.6

# Итоги

- Изучены основные методы SVM
- Реализован алгоритм ASM на Python
- Обнаружена проблема зацикливания и накопления ошибки. Анализ ее и решение
- Проведено тестирование и сравнение со стандартным SVM