

# Отчет по заданию «Метод опорных векторов»

Скачков Н.А. 317 группа

20 декабря 2017 г.

## 1 Введение

В данном отчете приведено сравнение различных стратегий поиска разделяющей поверхности методом опорных векторов. Для построения линейных поверхностей решались:

- прямая задача SVM — методом внутренней точки;
- двойственная задача SVM без ядрового перехода — методом внутренней точки;
- прямая задача SVM — методом полного и стохастического градиентных спусков, а также методом PEGASOS.

Для построения нелинейных разделяющих поверхностей решались:

- двойственная задача SVM с полиномиальным ядром — методом внутренней точки;
- двойственная задача SVM с RBF ядром — методом внутренней точки;

Во второй части отчета будет проведено сравнение предложенных методов с использованием параметров по умолчанию. Для сравнения будут использоваться различные по сложности и структуре сгенерированные выборки.

В третьей части отчета будет проведен анализ влияния настройки параметров на качество классификации и форму разделяющей поверхности.

## 2 Сравнение различных стратегий

Проведем сравнение времени работы алгоритмов в зависимости от количества объектов в выборке (Таблица 1). Сравнение проводилось на линейно неразделимой выборке с 500 признаками, чтобы алгоритмы не сходились слишком быстро.

Как видно, из методов внутренней точки самыми долгими являются решения прямой задачи и двойственной с полиномиальным ядром. Для прямой задачи это может быть связано с тем, что число оптимизируемых параметров значительно больше чем в двойственных методах. Для двойственной с полиномиальным ядром это может быть связано с длительностью операции поэлементного возведения в степень матрицы размера число объектов на число объектов.

Алгоритм	500	1000	1500	2000
Метод внутренней точки (прямая задача)	0.8	2.1	4.5	10.3
Метод внутренней точки (двойственная, без ядра)	0.2	1.2	2.9	6.7
Метод внутренней точки (двойственная, RBF ядро)	0.3	1.4	2.2	5.2
Метод внутренней точки (двойственная, полином. ядро)	0.4	1.8	4.4	9.8
Полный субградиентный спуск	7.4	25.7	39.4	62.0
Стохастический субградиентный спуск	7.1	13.5	18.0	22.4
Метод PEGASOS	5.2	12.9	19.1	20.9

Таблица 1: Время работы различных стратегий в зависимости от количества объектов, (с)

Для субградиентных методов было взято одинаковое количество итераций. Видно, что полный градиентный спуск работает значительно дольше других методов, что объясняется тем, что время выполнения поиска субградиента на каждой итерации производится по всей выборке.

Сравнение времени работы алгоритмов для различного числа признаков приведено в Таблице 2. Как видно, из методов внутренней точки дольше всех работает решение прямой задачи. Это связано с тем, что только в ней количество параметров в ней пропорционально количеству признаков.

Для всех субградиентных методов виден схожий рост времени работы алгоритмов, что объясняется тем, что время подсчета градиента для всех этих методов растет одинаково, пропорционально числу признаков.

Алгоритм	100	500	1000	1500
Метод внутренней точки (прямая задача)	2.0	3.1	5.1	9.0
Метод внутренней точки (двойственная, без ядра)	1.5	1.6	1.6	1.5
Метод внутренней точки (двойственная, RBF ядро)	0.8	0.9	1.5	1.6
Метод внутренней точки (двойственная, полином. ядро)	1.9	2.0	2.0	2.0
Полный субградиентный спуск	2.0	11.6	24.3	44.6
Стохастический субградиентный спуск	3.0	13.9	23.3	37.3
Метод PEGASOS	2.2	12.5	26.8	37.7

Таблица 2: Время работы различных стратегий в зависимости от количества признаков, (с)

Значения целевой функции, к которым сошлись различные методы, можно увидеть в Таблице 3.

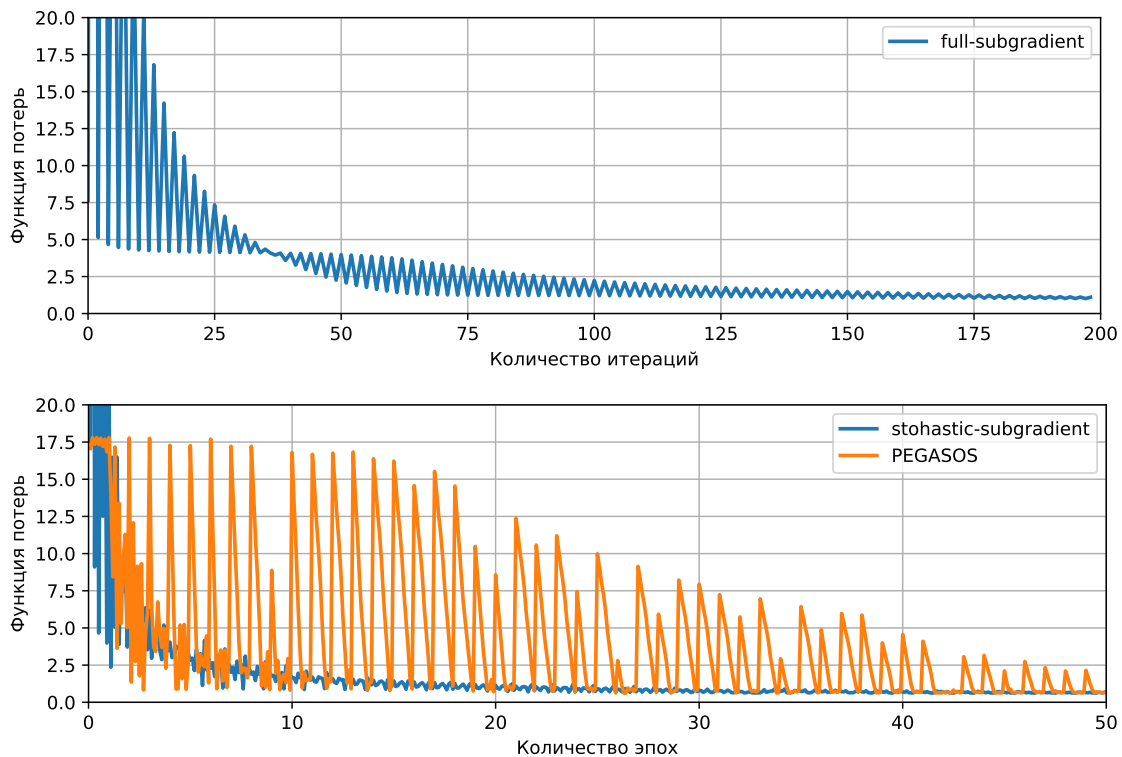
Алгоритм	Значение функции потерь
Метод внутренней точки (прямая задача)	0.62
Метод внутренней точки (двойственная, без ядра)	-0.62
Метод внутренней точки (двойственная, RBF ядро)	-0.98
Метод внутренней точки (двойственная, полином. ядро)	-0.33
Полный субградиентный спуск	0.062
Стохастический субградиентный спуск	0.62
Метод PEGASOS	0.62

Таблица 3: Значение целевой функции для различных методов

Двойственные задачи имеют, как и ожидалось, отрицательные значения, так как изначально решается задача максимизации. Можно заметить, что для всех линейных методов получились одинаковые итоговые значения, а для нелинейных — другие, так как в них оптимизировался другой функционал.

Характер сходимости можно увидеть на Рисунке 1. Как видно, значение функционала во всех методах колеблется, и сходимость присутствует только за счет наличия затухания шага во всех субградиентных методах. Также у полного градиента можно увидеть точку, в которой значение функционала практически перестает меняться (30 итерация), однако после этого функционал продолжает уменьшаться, увеличив колебания. Это означает, что есть риск ранней остановки, и лучше делать больше итераций в алгоритмах субградиентного спуска, так как возможно резкое улучшение после нахождения на «плато» — области с близким к 0 значением субградиента.

Рис. 1: Сходимость функций потерь субградиентных методов



Рассмотрим значения точности различных моделей на искусственных датасетах, обладающих различными свойствами. (Таблица 4). Были взяты следующие датасеты:

- `lin-split` — линейно разделимая выборка;
- `moons` — линейно неразделимая выборка, состоящая из непересекающихся классов, сгенерированных в форме лун;
- `non-split` — выборка, состоящая из частично пересекающихся классов;
- `outliers` — линейно разделимая выборка с добавлением 30% выбросов;
- `imbalanced` — выборка с соотношением классов 2 к 8.

Алгоритм	lin-split	moons	non-split	outliers	imbalanced
SVM, прямая задача	1.000	0.796	0.910	0.857	1.000
SVM, двойственная, без ядра	1.000	0.783	0.842	0.857	1.000
SVM, двойственная, RBF ядро	1.000	0.988	0.994	0.983	1.000
SVM, двойственная, полином. ядро	1.000	0.873	0.924	0.857	1.000
Полный субградиентный спуск	1.000	0.798	0.916	0.857	0.931
Стох. субградиентный спуск	1.000	0.796	0.908	0.857	0.970
Метод PEGASOS	1.000	0.799	0.924	0.857	1.000

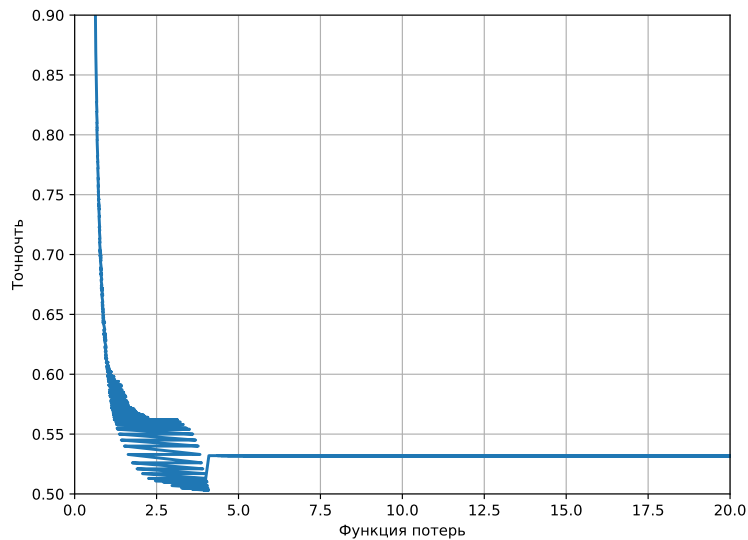
Таблица 4: Точность разных моделей на различных датасетах

Можно увидеть, что нелинейные алгоритмы имеют намного более высокие значения точности, что объясняется их большей сложностью. При этом не исключено переобучение таких моделей.

Для линейных моделей можно заметить, что полный субградиентный метод намного сильнее остальных ошибается на несбалансированных классах, что меньше проявляется для стохастического метода. Также видно, что PEGASOS дает наилучшие из всех линейных методов результаты.

На примере стохастического субградиентного метода, проанализируем связь между точностью на обучающей выборке и значением функции потерь (Рис. 2).

Рис. 2: Связь функции потерь и точности



Как можно увидеть, низкие значения функционала связаны с высокой точностью, также как высокие значения — с низкой точностью. В остальных случаях зависимости не наблюдается. Можно сделать вывод, что для определения сходимости достаточно рассматривать лишь один из этих функционалов.

## 3 Настройка параметров

### 3.1 Настройка параметров SVM

Настроим параметр  $C$ , общий для всех моделей, решая линейную двойственную задачу. Оценивать модели будем по кроссвалидации с 5 фолдами по выборке с пересекающимися классами. Значения по всем фолдам будем усреднять.

$C$	1	5	10	25	<b>50</b>	100
Точность	0.763	0.836	0.851	0.864	<b>0.868</b>	0.856

Таблица 5: Значение точности на кроссвалидации для разных  $C$

Как видно, оптимальное значение достигается при  $C = 50$ , после чего точность начинает падать.

Попробуем подобрать  $C$  обучаясь на линейно разделимой выборке (Таблица 6). Тут также использовалась кроссвалидация с 5 фолдами.

$C$	1	5	10	25	50	100
Точность	1.000	1.000	1.000	1.000	1.000	0.998

Таблица 6: Значение точности на кроссвалидации для разных  $C$  для линейно разделимой выборки

Как можно заметить, подбор параметра не дал никаких результатов, так как для любых  $C$  классификация идеальна или близка к идеальной.

Подберем параметр  $\gamma$  для RBF ядра. На этот раз кроссвалидацию будем проводить только на выборке с пересекающимися классами (Таблица 7).

$\gamma$	1	<b>5</b>	10	15
Точность	0.876	<b>0.877</b>	0.873	0.870

Таблица 7: Значение точности на кроссвалидации для разных  $\gamma$

Как видно, оптимальное значение параметра ширины  $\gamma$  равняется 5.

Подберем теперь степень полинома  $d$  для полиномиального ядра, используя кроссвалидацию на том же датасете (Таблица 8).

Степень	1	2	<b>3</b>	4	5
Точность	0.925	0.921	<b>0.937</b>	0.935	0.935

Таблица 8: Значение точности на кроссвалидации для разных  $d$

Заметим, что RBF ядро дает несколько более плохие значения после настройки параметров, что связано с тем, что оно строит более сложную поверхность. В итоге эта модель становится более чувствительной к выбросам и ее обобщающая способность падает.

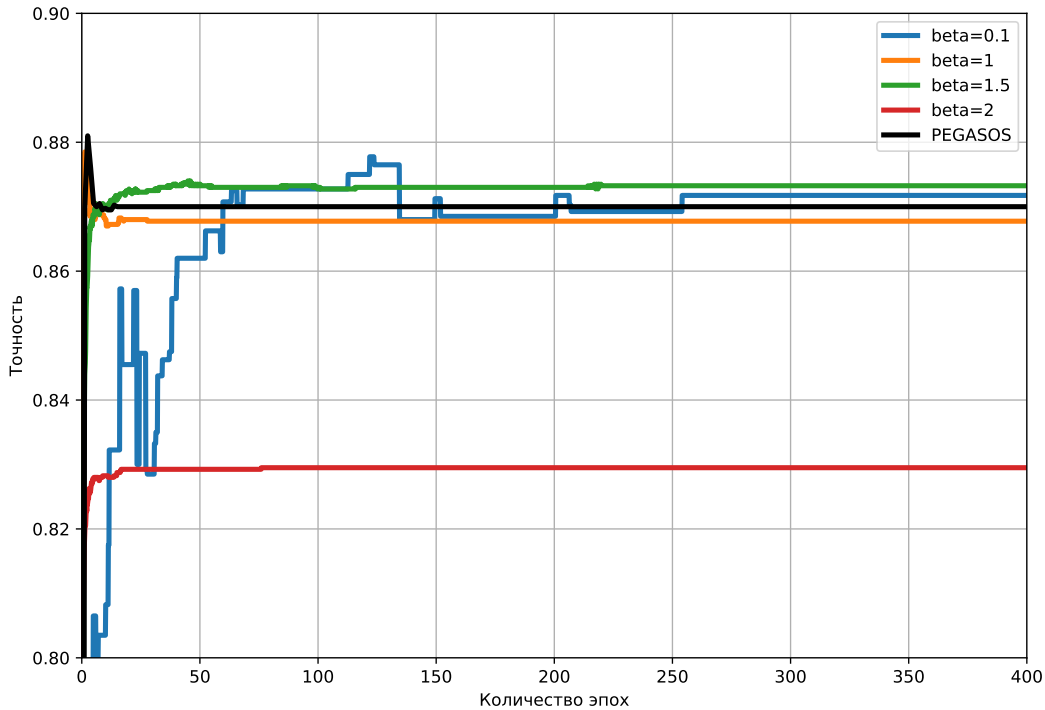
Также заметим, что хоть выборка и состоит из пересекающихся классов, линейная модель дает достаточно неплохие результаты, хоть и строит очень простую разделяющую поверхность.

### 3.2 Настройка параметров субградиентных методов

Различные параметры субградиентных методов не меняют функционал качества, но влияют на скорость сходимости этих методов, то к чему они сходятся, и сходятся ли они вообще.

Рассмотрим влияние параметра угасания шага субградиентного спуска  $\beta$  (Рис. 3). Здесь и далее оценка параметра производится на выборке с пересекающимися классами, с кроссвалидацией по 5 фолдам. Значения истории функционала точности для всех батчей усредняются.

Рис. 3: Сходимость стох. субградиентного метода для разных значений параметра  $\beta$



Как видно, наилучшим оказалось значение  $\beta = 0.1$ , однако метод PEGASOS имеет более высокую точность и достигает ее намного быстрее.

Значение  $\beta = 0$ , так же было рассмотрено, однако максимальная точность на нем была порядка 0.4 и не была отображена на графике. Это можно объяснить тем, что в теории субградиентный метод не обязан сходиться, и чтобы метод обладал этим свойством необходимо, чтобы величина шага уменьшалась с итерациями.

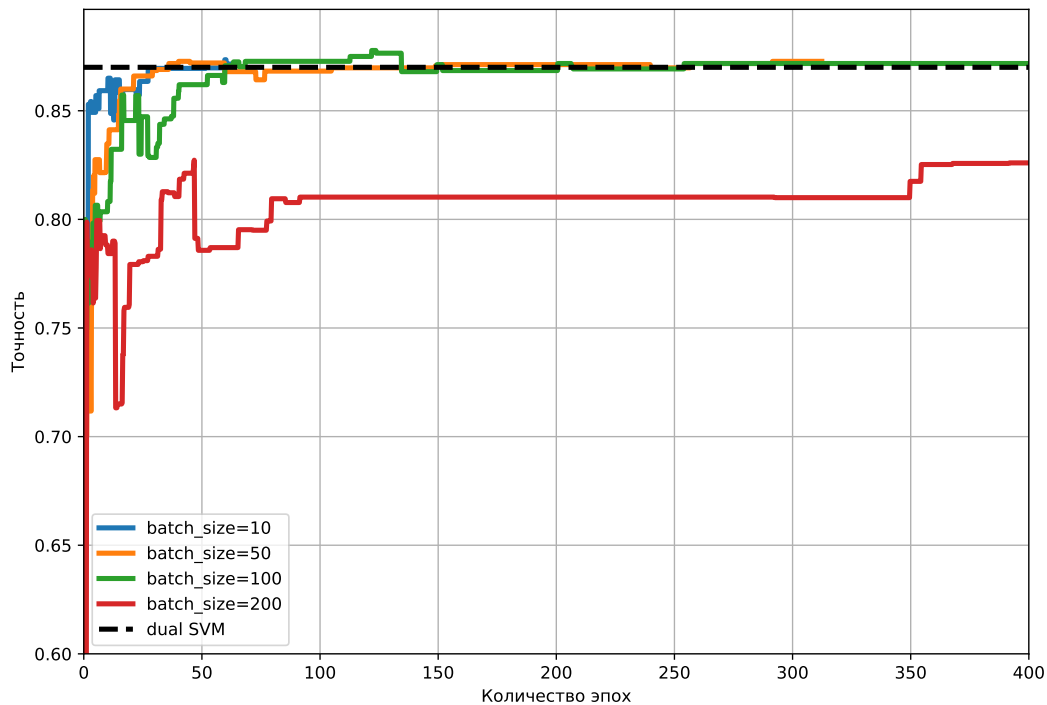
В целом можно сказать, что сходимость субградиентного метода неустойчива и при попадании в оптимальное значение алгоритм может легко из него уйти.

Рассмотрим теперь характер сходимости для различных размеров батчей (Рис. 4). «Базовое» значение точности, полученное с помощью решения двойственной задачи методом внутренней точки, показано на графике пунктиром.

Можно увидеть, что увеличение числа объектов в батче ухудшает качество классификации. Это может объясняться тем, что алгоритм достаточно нестабилен, и при малых значениях батча больше вероятность того, что метод случайно попадет ближе к оптимуму.

Заметим также, что при наилучшем параметре размера батча, равном 100, алгоритм дал лучший результат, чем решение методом внутренней точки.

Рис. 4: Сходимость стохастического субградиентного метода для различных размеров батчей



Сравним результаты подбора параметров с результатами, полученными для градиентных методов в задаче логистической регрессии.

В градиентных методах увеличение размера батча зачастую приводило к улучшению качества модели, увеличивая лишь время работы метода. В субградиентных методах большая величина батча ухудшает итоговое качество, так как уменьшается вероятность попадания в оптимум.

Параметр угасания шага  $\beta$  в градиентных методах влиял только на скорость сходимости методов и лишь немного улучшал качество. В субградиентных методах этот параметр является обязательным, так как без него метод вообще не сходится.

Характер сходимости у градиентных и субградиентных методов также различается. Градиентные методы сходятся монотонно, без скачков. В субградиентных, наоборот, требуется смотреть лучшее значение, полученное за время работы метода. Сходимость немонотонная, присутствуют резкие скачки функционалов качества.

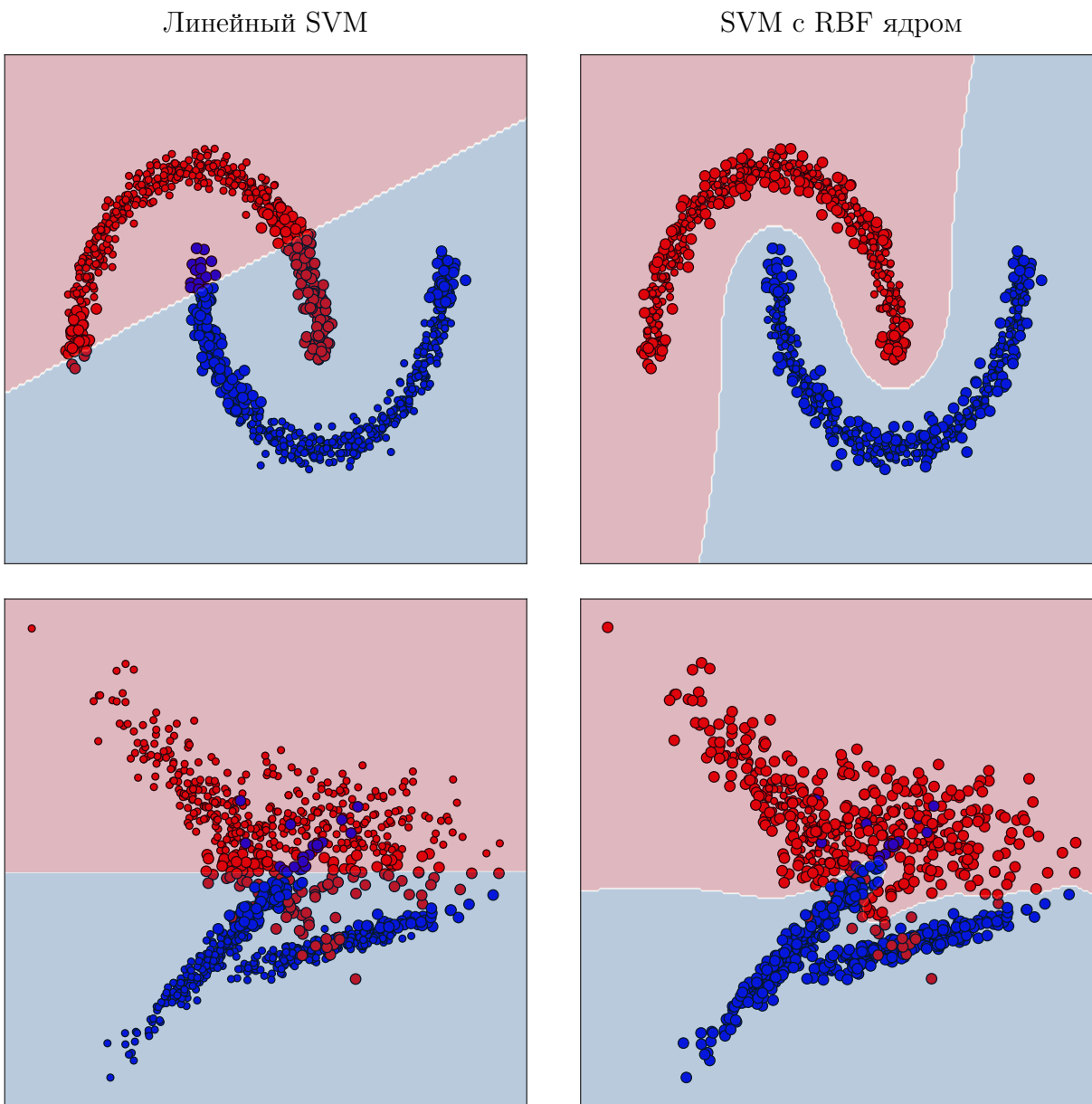
Субградиентный метод стоит использовать, в случае, если требуется решить задачу классификации быстро. Однако есть вероятность того, что найденное решение не будет оптимальным, так как субградиентный метод, может «не попасть» в оптимум.

## 4 Визуализация разделяющих поверхностей SVM

Построим разделяющую поверхность для двумерных выборок линейным SVM и SVM с RBF ядром (Рис. 5). На картинках опорные объекты отмечены крупнее. Для построения поверхностей были взяты найденные оптимальные  $C = 50$  и  $\gamma = 5$ .

Заметим, что опорные объекты для линейной поверхности лежат рядом с границей. Для RBF опорными объектами являются все, граничащие с внешней областью, а не опорными являются внутренние точки.

Рис. 5: Разделяющая поверхность линейного SVM и SVM с ядром



Теперь построим разделяющую поверхность RBF в зависимости от параметра  $\gamma$  (Рис. 6). Параметр  $C$  был взят оптимальным, равным 50.

Видно, что при значениях  $\gamma$ , больших оптимального, выбросы начинают оказывать большое влияние на разделяющую поверхность, что уменьшает обобщающую способность алгоритма. При сверхбольших значениях параметра, граница плотнее прилегает к одному из классов, окольцовывая все его точки, а все выбросы «дырявят» область противоположного класса. Точность классификации на объектах обучающей выборки в этом случае становится равна 1, но обобщающая способность такого алгоритма невелика.



Рис. 6: Разделяющая поверхность линейного SVM и SVM с ядром

