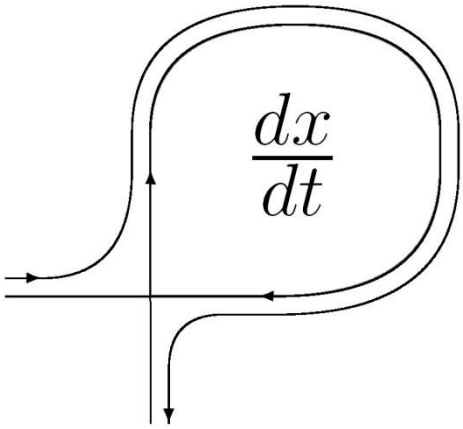

ДИФФЕРЕНЦИАЛЬНЫЕ УРАВНЕНИЯ И ПРОЦЕССЫ УПРАВЛЕНИЯ
N. 3, 2022
Электронный журнал,
рег. Эл № ФС77-39410 от 15.04.2010
ISSN 1817-2172

http://diffjournal.spbu.ru/
e-mail: jodiff@mail.ru

Машинное обучение и искусственный интеллект в процессах управления

Экспериментальное исследование реакции алгоритмов машинного обучения на ошибки разметки данных

Дюк В.А.

Институт проблем транспорта им. Н.С. Соломенко Российской академии наук

v_duke@mail.ru

Аннотация. Известны авторитетные мнения, что разметка данных сегодня является самым важным элементом в процедуре создания систем искусственного интеллекта на основе методов машинного обучения. Вместе с тем, особенно при краудсорсинге возникает серьезная проблема неточной разметки данных. Материалы данной статьи дополняют известные подходы к решению данной проблемы исследованием реакции на неточную разметку данных некоторых популярных методов машинного обучения. Это наивный байесовский классификатор, трехслойный перцептрон, метод ближайших соседей (KNN), деревья решений, случайный лес, логистическая регрессия, машина опорных векторов (SVM). Алгоритмы обучались на копиях специально сгенерированных данных с различными долями ошибок разметки и затем испытывались на данных с точной разметкой. По результатам эксперимента на данных, имитирующих простую и сложную структуру двух классов многомерных объектов, продемонстрирован феномен относительно слабой зависимости точности моделей классификации KNN и SVM от ошибок разметки обучающей выборки. Сделан вывод, что в условиях неточной разметки данных более предпочтительным является алгоритм KNN. Он менее трудоёмок, имеет меньше настраиваемых параметров, свободен от априорных предположений о структуре данных, устойчив к аномальным выбросам, интерпретируем. Кроме того, этот метод обладает существенным потенциалом дальнейшего теоретического и практического развития на основе подхода, связанного с построением контекстно-зависимых локальных метрик.

Ключевые слова: машинное обучение, искусственный интеллект, ошибки разметки данных, контекстно-зависимые локальные метрики.

1. Введение

В современном мире возрастающую роль играют системы искусственного интеллекта. При разработке таких систем всё чаще применяют машинное обучение, включающее методы создания и обучения по накопленным или сконструированным данным компьютерных моделей, корректно отражающих разнообразные отношения объектов реального или идеального мира. В настоящее время известно большое количество примеров эффективного использования машинного обучения в маркетинге, в банковском деле и страховании, здравоохранении, на транспорте, в системах безопасности, в геологоразведке, в системах распознавания изображений, понимания и синтеза речи, анализа текстов, автоматического перевода с языка на язык, в хемо- и биоинформатике и многих других сферах [1].

Для реализации методов машинного обучения на основе множества примеров, содержащих пары «известный вход – известный выход» (машинное обучение с учителем), требуется разметка (аннотирование) данных. Спрос на разметку данных сегодня велик и рынок разметки данных развивается нарастающими темпами (ежегодный рост 25-30 %). В 2021 году его сегмент достиг более 1 миллиарда долларов и ожидается, что к 2027 году превзойдёт 7 миллиардов долларов. Некоторые источники, в том числе Research and Markets, считают, что к 2028 году рынок будет стоить не менее 8,2 миллиарда [2].

Основные подходы к разметке следующие [3-5]:

Разметка внутри компании. При таком подходе процесс разметки легко контролировать и можно быть уверенным в точности и качестве работы. Однако, этот способ подходит в основном только крупным компаниям, имеющим собственную команду соответствующих экспертов.

Аутсорсинг применяется тогда, когда команда для разметки данных нужна на определенный период времени. Разместив объявление на рекрутинговых сайтах или в своих социальных сетях, формируется база потенциальных исполнителей, из которых в ходе интервью и тестирования определяются те, кто обладает требуемыми навыками. Другой вариант аутсорсинга – привлечение сторонней команды, специализирующейся в данной конкретной области.

Краудсорсинг – это способ решить одну отдельно взятую задачу при помощи большого количества исполнителей с помощью специальных платформ.

На сегодня существует более десятка различных компаний и платформ разметки данных. С их помощью аннотируют миллионы единиц данных для сотен разных проектов. К наиболее известным компаниям, разработавшим и использующим различные инструменты и платформы для аутсорсинга и краудсорсинга относятся, например, Appen (appen.com), Lionbridge AI (lionbridge.ai), Scale (scale.com), Toloka (toloka.ai), MTurk (mturk.com), Hive (thehive.ai), Webtunix AI (webtunix.com), iMerit (imerit.net), CloudFactory (cloudfactory.com), Clickworker (clickworker.com).

Высказываются мнения, что разметка данных сегодня является самым важным элементом в процедуре создания систем искусственного интеллекта. При этом обращается внимание на качество данных. Об этом говорит, например, основатель deeplearning.ai и бывший руководитель Google Brain Эндрю ЫН [6]. Вместе с тем, важное значение особенно при краудсорсинге приобретает проблема неточной разметки данных. Различные аспекты данной проблемы рассматриваются в ряде работ [7-10]. И, как следует из этих и других работ, основной причиной является человеческий фактор, который связан с неточностью формулирования задания, с ошибками восприятия и недобросовестностью исполнителя; усталостью исполнителя, сложностью объектов разметки и др.

Для решения проблемы неточной разметки данных используются два подхода. Первый подход направлен на повышение качества команды асессоров и использует обучение и экзамены исполнителей, различные виды защита от попыток обмана, борьбу со спамом и злонамеренным искажением разметки, отстранение исполнителей, которые допускают ошибки, передача работ добросовестным исполнителям и т.п.

Во втором подходе для коррекции ошибок и оптимизации процесса разметки данных применяют приемы, связанные с той или иной математической обработкой приходящих от исполнителей меток. Это, например, методы голосования, статистические методы, основанные на

предположении, что метки подвержены случайному шуму [11-13], редактирование данных с применением теории графов [14] и др. Достаточно обширный обзор современных приемов приведен в [15].

Материалы данной статьи дополняют перечисленные выше подходы исследованием реакции на неточную разметку данных некоторых популярных методов машинного обучения. Образно говоря, в статье рассмотрена проблема «плохой учитель – хороший ученик», а именно, может ли способный ученик (обучающийся алгоритм) превзойти учителя, который его тренирует и при этом ошибается в классификации тех или иных многомерных объектов.

2. Исследуемые методы

Разные разработчики систем искусственного интеллекта применяют различные алгоритмы машинного обучения от простейших типа наивного байесовского классификатора до достаточно изощренных типа машины опорных векторов и алгоритмов глубокого обучения. Проблемы, с которыми сегодня сталкиваются разработчики порождены спецификой данных и ранее нами описывались в ряде статей и монографий (например, в [1]). В сжатом виде специфика данных следующая:

1. нечеткость целевых показателей и критериев;
2. неопределенность, неточность, разнотипность и неизвестная размерность описаний;
3. гетерогенность эквифинальных состояний исследуемых систем;
4. наличие русел и джокеров разного, заранее не известного формата с неизвестной локализацией.

Попытки построения предиктивных моделей в предметных областях с подобными характеристиками породили большое количество подходов. В табл. 1 приведена статистика применяемых методов машинного обучения по результатам опросов большого количества специалистов.

Таблица 1. Статистика применяемых методов машинного обучения

N	Методы (алгоритмы)	2007 год (203 чел.) в %	2011 год (311 чел.) в %	2016 год (844 чел.) в %	2017 год (732 чел.) в %	2018/2019 (833 чел.) в %
1.	Регрессионный анализ	51	58	67	61	56
2.	Логические методы (деревья решений и др.)	63	60	55	51	48
3.	Метод ближайших соседей	13		46	39	33
4.	Ансамбли алгоритмов		28	34		30
5.	Метод опорных векторов	16	39	29	34	22
6.	Нейросети (классические)	17	27	24		
7.	Наивный байесовский классификатор	16	22	24		
8.	Нейросети (глубокое обучение)			19		25
9.	Методы поиска ассоциативных правил	26	29	15		
10	Генетические алгоритмы	11	9	9		

Примечание: Данные основаны на результатах опроса специалистов на портале www.kdnuggets.com.

Попытки ранжирования методов машинного обучения по их эффективности предпринимались неоднократно (например, обзор таких работ приводится в [16]). В одной из наиболее обширных работ [17] была изучена эффективность 179 методов классификации на 121 наборе данных. В проведенном исследовании для каждого метода классификации оценивалась общая точность предсказаний (overall accuracy) и другие показатели эффективности построенных моделей. Авторы выделили четыре группы методов, обладающих наибольшей точностью прогноза (перечислены в порядке убывания эффективности): Случайный лес (Random Forest); Машины опорных векторов (Support Vector Machines); Искусственные нейронные сети (Artificial Neural Networks) и ансамбли моделей, построенные с использованием процедуры бустинга (Boosting Ensembles).

Более свежий опрос 20000 специалистов, проведенный популярной платформой для испытания и сравнения алгоритмов машинного обучения Kaggle (https://www.kaggle.com/), показал, что профессионалы в области науки о данных наиболее часто использовали алгоритмы множественного регрессионного анализа, деревья и леса решений, и ранее упомянутые методы построения ансамблей алгоритмов с использованием процедуры градиентного бустинга [18].

Нами проведено исследование популярных алгоритмов, входящих в состав свободного программного обеспечения для анализа данных и машинного обучения, университета Вайкато (Новая Зеландия) WEKA, распространяемого по лицензии GNU GPL [19]:

1. Наивный байесовский классификатор;
2. Трехслойный перцептрон, использующий алгоритм обратного распространения ошибки;
3. Метод ближайших соседей (БС);
4. Деревья решений;
5. Случайный лес;
6. Логистическая регрессия;
7. Машина опорных векторов (SVM).

При исследовании указанных методов в основном использовались параметры, заданные по умолчанию в пакете WEKA. Вместе с тем, следует сделать некоторые уточнения. В качестве одной из возможных реализаций алгоритма машины опорных векторов применялся алгоритм SMO (Sequential Minimal Optimization), описанный в [20]. При этом задавалось полиномиальное ядро со степенями 1 и 3, что в дальнейшем обозначалось соответственно SVM (p=1) и SVM (p=3). Также нужно отметить, что для метода БС число ближайших соседей было фиксированным, задавалось вручную и составляло 10 % от общего объема анализируемых объектов. Ещё отметим, что при построении деревьев решений использовался алгоритм J48, который является аналогом на Java известного алгоритма C4.5 [21].

3. Данные для тестирования алгоритмов

Часто для сравнительного исследования эффективности различных алгоритмов используют известные наборы данных, как например это сделано в [16]. Эти наборы стараются подобрать так, чтобы в более или менее полной мере охватить разнообразие структур данных, типов которых, вообще говоря, великое множество. В нашем исследовании использовались искусственно сгенерированные данные. При их создании ставилась цель вместо большого разнообразия возможных структур рассмотреть 2 полярных случая. Схематично эти 2 случая можно охарактеризовать следующим образом:

– *простая структура данных*, когда классы объектов однородны, компактны и линейно разделимы в пространстве количественных признаков,

– *сложная структура данных*, когда классы гетерогенны (неоднородны) и их нельзя разделить гиперплоскостью в исходном пространстве признаков.

Также в нашем исследовании считалось важным обеспечить наглядность и прозрачность результатов тестирования. Поэтому рассматривалось всего 2 класса, имеющих одинаковый объём и одинаковые симметричные распределения в пространстве описания невысокой размерности.

После генерирования исходных точно размеченных выборок данных как с простой, так и со сложной структурой, из них создавалось 5 копий, для которых метки классов были случайным образом перепутаны. Доли перепутанных меток составляли 10, 20, 30, 40 и 50 процентов (от сравнительно небольших ошибок до полностью случайной разметки).

Далее алгоритмы для создания моделей классификации обучались на той или иной копии с различными долями ошибок разметки. Полученные модели проверялись на обучающей выборке с использованием приема кросс-валидации (10 блоков), и затем модели испытывались на данных с точной разметкой. Общая схема исследования приведена на рис. 1.

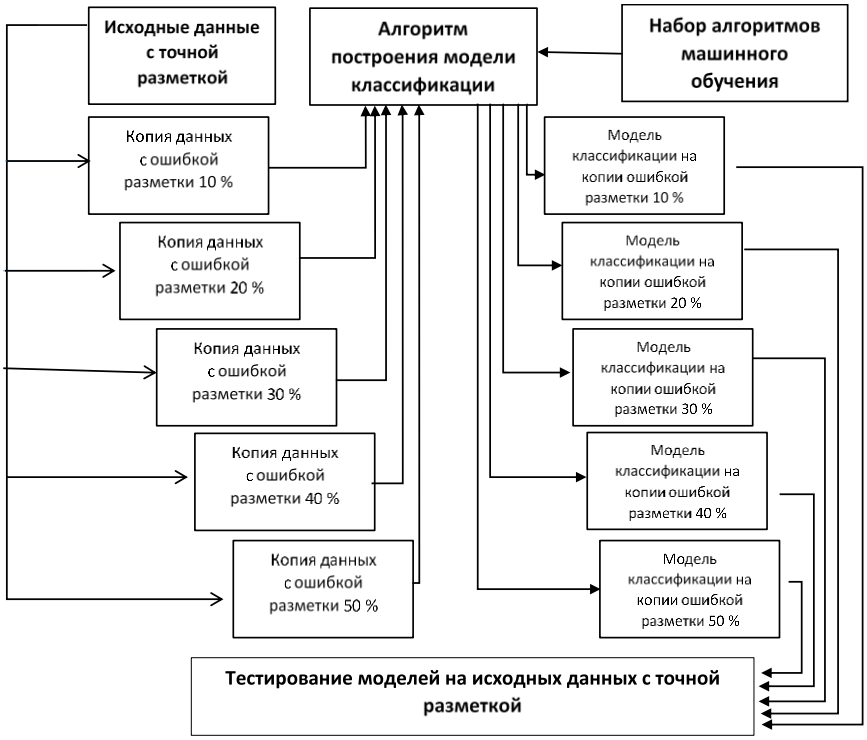

Рис. 1. Общая схема исследования алгоритмов

4. Исследование алгоритмов на простой структуре данных

В качестве простой структуры данных было рассмотрено 200 объектов (по 100 объектов в каждом из 2-х классов) в пространстве 5-ти количественных признаков. Объекты в каждом классе были распределены по нормальному закону. Средние значения признаков x_i и среднеквадратические отклонения приведены в табл. 2. В пространстве трёх главных компонент распределения объектов выглядят как слегка пересекающиеся гипершары (Рис. 2).

Таблица 2. Параметры распределения объектов

	Класс 1	Класс 2
Количество объектов	100	100
Средние значения		
x_1	0	1,5
x_2	0	1,5
x_3	0	1,5
x_4	0	1,5
x_5	0	1,5
Среднеквадратические отклонения		
x_1	1	1
x_2	1	1

x_3	1	1
x_4	1	1
x_5	1	1

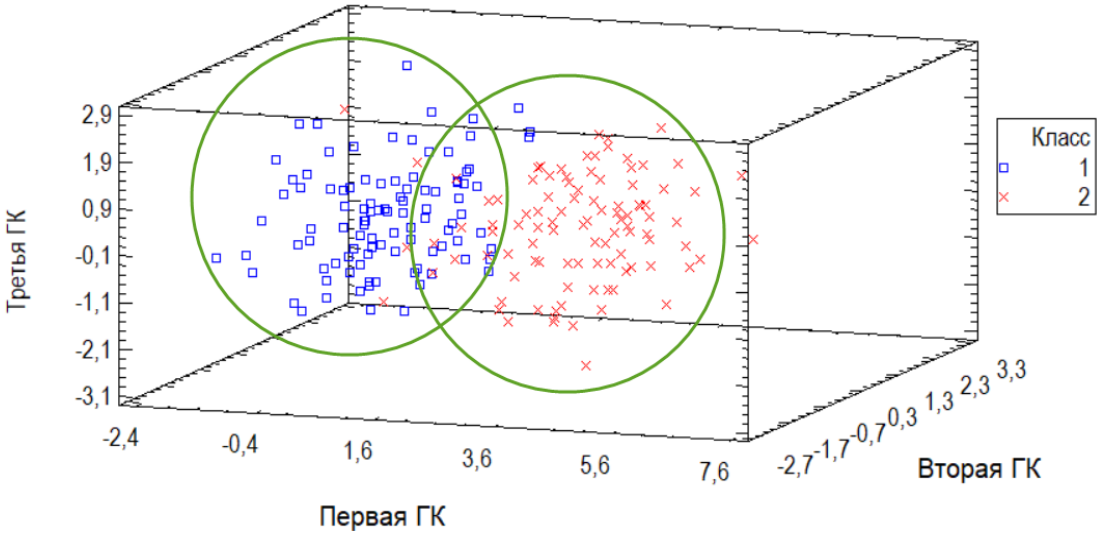

Рис. 2. Проекция объектов в пространство первых 3-х главных компонент

В табл. 3 приведены значения вероятностей правильной классификации моделей, построенных с использованием различных алгоритмов на обучающих выборках с различными ошибками разметки. На рис. 3 эти вероятности изображены графически. Как следует из таблиц и графиков, все модели имеют примерно одинаковую точность и алгоритмы для построения моделей адекватно реагируют на нарушения точности разметки данных, демонстрируя практически синхронное снижение вероятностей правильной классификации.

Таблица 3. Вероятности правильной классификации разных алгоритмов на обучающей выборке в зависимости от ошибок разметки (с использованием кросс-валидации на 10 блоков)

Ошибка разметки в %	0	10	20	30	40	50
SVM (p=1)	94,5	81,5	77,5	69	64	54
SVM (p=3)	94	82	77,5	68,5	59	53,5
БС	94,5	80	77,5	68	57,5	49
Деревья решений J48	89	71,5	73,5	63	58	51,5
Логистическая регрессия	94	81	79	68	63	51
Многослойный перцептрон	93,5	80	77	64	57,5	49,5
Наивный Байес	94,5	83	80	69	59	50,5
Случайный лес	93,5	80,5	78	65	58	52

Несколько иная картина вырисовывается при применении моделей, построенных на данных с неточной разметкой, на тестовой выборке с точной разметкой (табл. 4 и рис. 3). Как видно, несмотря на ошибки разметки при обучении, ряд алгоритмов демонстрируют в своих моделях на тестовом материале не на много меньшую точность, чем если бы они обучались на правильно размеченных данных. Это, в первую очередь, касается наивного байесовского классификатора, машины опорных векторов (при различных степенях полиномиального ядра) и алгоритма ближайшего соседа. Указанный феномен наблюдается при достаточно больших ошибках разметки

вплоть до 30-40 % ошибок. При этом для простой структуры данных наилучшие показатели у простейшего алгоритма – наивного байесовского классификатора.

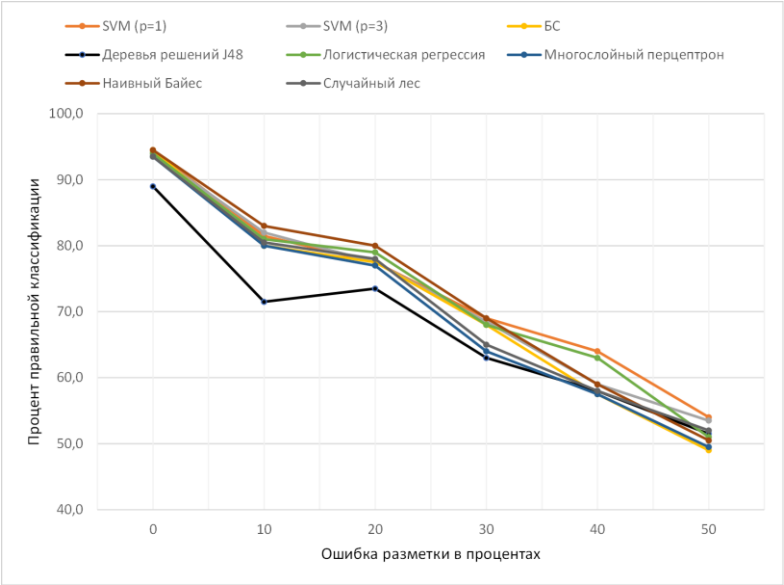

Рис. 3. Графики вероятностей правильной классификации разных алгоритмов на обучающей выборке в зависимости от ошибок разметки обучающей выборки

Таблица 4. Вероятности правильной классификации моделей на тестовой выборке с точной разметкой в зависимости от ошибок разметки обучающей выборки

Ошибка разметки в %	0	10	20	30	40	50
SVM (p=1)	94,5	94,5	94,5	91,5	78	39,5
SVM (p=3)	94	94	92	91	88	39
БС	94,5	93,5	94,5	89,5	80,5	51
Деревья решений J48	89	90	89,5	82	82,5	50
Логистическая регрессия	94	93,5	92	86,5	72,5	38
Многослойный перцептрон	93,5	96	90,5	82	75,5	47
Наивный Байес	94,5	94,5	95,5	94,5	93,5	30,5
Случайный лес	93,5	87	83	70	59,5	47,5

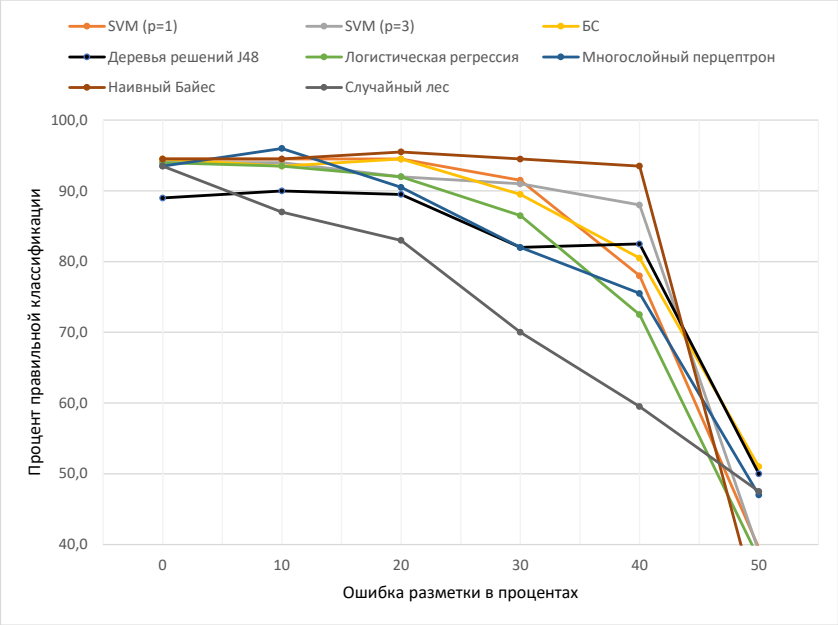

Рис. 4. Графики вероятностей правильной классификации разных алгоритмов на тестовой выборке в зависимости от ошибок разметки обучающей выборки

5. Исследование алгоритмов на сложной структуре данных

В качестве сложной структуры данных было рассмотрено 400 объектов (по 200 объектов в каждом из 2-х классов), которые распределены в вершинах куба таким образом, что разделение классов возможно только в этом трехмерном пространстве. При этом, как видно из рис. 5, каждый класс является гетерогенным (неоднородным) и состоит из четырех равномощных непересекающихся группировок.

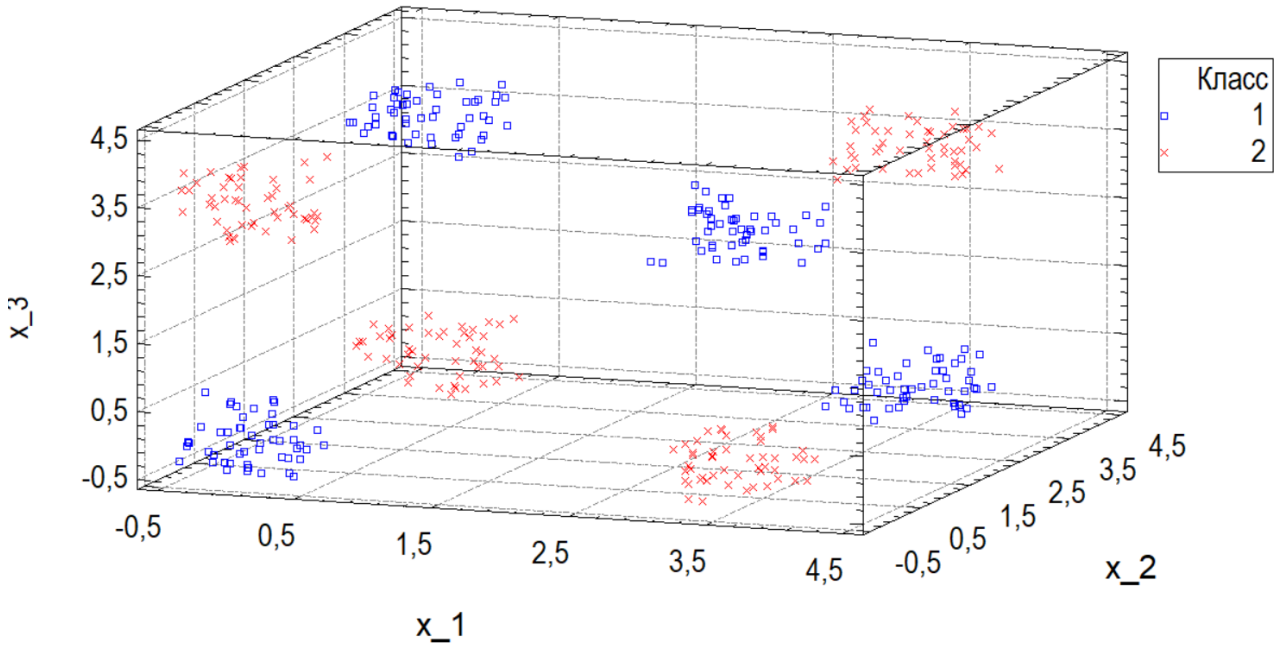

Рис. 5. Распределение объектов в трехмерном пространстве

В табл. 5 приведены значения вероятностей правильной классификации моделей, построенных с использованием различных алгоритмов на обучающих выборках с различными ошибками разметки. На рис. 6 сведения таблицы отображены графически. Как и ожидалось, наивный Байес, логистическая регрессия, деревья решений, машина опорных векторов с линейным ядром не

способны строить модели разделения классов в условиях столь неоднородной их структуры. Трехслойный перцептрон и алгоритм «случайный лес» дали высокое, но не оптимальное разделение анализируемых классов. Вместе с тем, точные результаты разделения классов при отсутствии ошибок разметки продемонстрировали модели, построенные с помощью алгоритмов ближайшего соседа и машины опорных векторов с полиномиальным ядром третьей степени. При снижении точности разметки модели указанных алгоритмов показывают релевантное снижение точности классификации на выборках с ошибками разметки.

В табл. 6 и на рис. 7 приведены данные и графики, отражающие зависимость вероятности правильной классификации моделей на тестовой выборке с точной разметкой в зависимости от ошибок разметки обучающей выборки. Исходя из результатов предыдущего испытания моделей на обучающей выборке, далее имеет смысл рассматривать только 2 модели, одна из которых строится с помощью машины опорных векторов с полиномиальным ядром третьей степени SVM (p=3), а другая с помощью метода ближайших соседей БС. И здесь мы наблюдаем феномен, когда, обучаясь на выборках с ошибками разметки вплоть до уровня ошибок разметки в 30 %, модели SVM (p=3) и БС демонстрируют практически 100 % точность классификации на тестовой выборке с безошибочной разметкой. Более того, даже при 40 % уровне ошибок алгоритм БС показывает точность классификации в 84 % на тесте.

Таблица 5. Вероятности правильной классификации разных алгоритмов на обучающей выборке в зависимости от ошибок разметки (с использованием кросс-валидации на 10 блоков)

Ошибка разметки в %	0	10	20	30	40	50
SVM (p=1)	39	44,25	49,25	51	53,5	46,5
SVM (p=3)	100	86	75,25	64,75	61,75	48,75
БС	100	90,5	77,75	69,75	57,5	46
Деревья решений J48	50	51	51,25	52,75	55,25	50,5
Логистическая регрессия	41,25	43,75	49,75	49,5	54	47,75
Многослойный перцептрон	87,25	83,25	69,75	63,25	56	49,25
Наивный Байес	42,75	42,25	49,75	49,75	54,75	47
Случайный лес	94,5	79	62,75	57,25	51,75	42,5

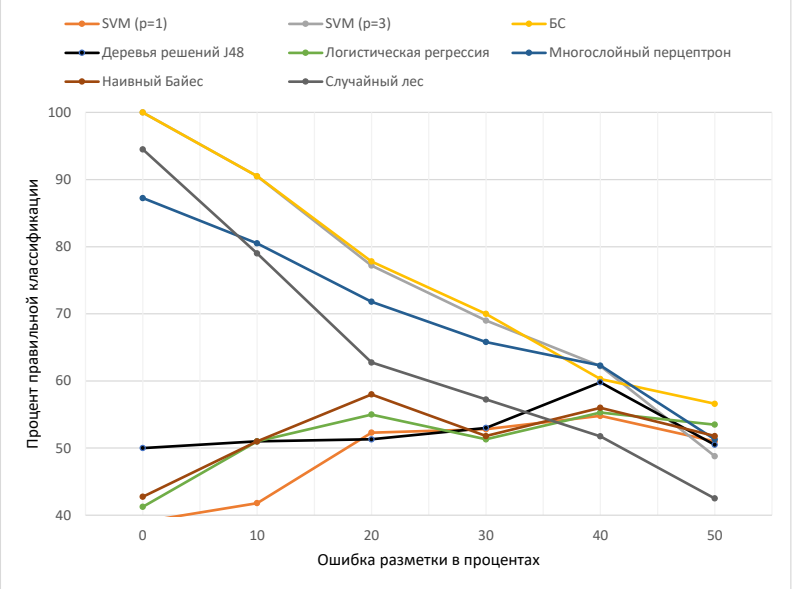

Рис. 6. Графики вероятностей правильной классификации разных алгоритмов на обучающей выборке в зависимости от ошибок разметки обучающей выборки

Таблица 6. Вероятности правильной классификации моделей на тестовой выборке с точной разметкой в зависимости от ошибок разметки обучающей выборки

Ошибка разметки в %	0	10	20	30	40	50
SVM (p=1)	39	39,8	51,5	50,0	50,0	50,0
SVM (p=3)	100	100,0	99,5	97,3	73,5	33,8
БС	100	100,0	100,0	97,5	84,0	48,5
Деревья решений J48	50	50,0	50,0	50,0	49,5	50,0
Логистическая регрессия	41,3	50,0	55,8	46,5	45,0	55,0
Многослойный перцептрон	87,3	87,5	87,5	84,0	72,5	50,3
Наивный Байес	42,8	50,0	56,7	48,5	46,3	54,8
Случайный лес	94,5	90,5	77,8	69,8	59,8	49,0

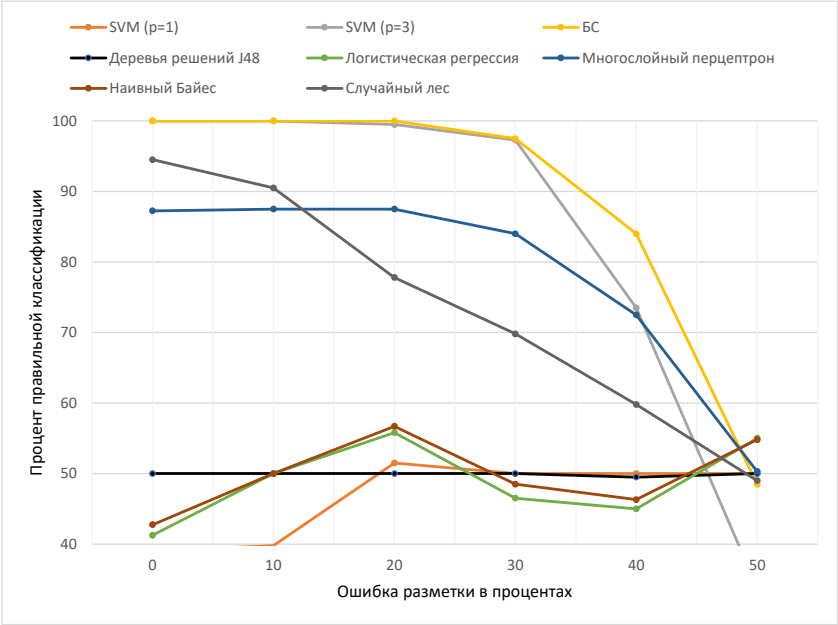

Рис. 7. Графики вероятностей правильной классификации разных алгоритмов на тестовой выборке в зависимости от ошибок разметки обучающей выборки

6. Обсуждение результатов

На основании проведенного исследование были сделаны следующие выводы.

1. Алгоритмы машинного обучения по-разному ведут себя в условиях неточной разметки данных.

2. Для простой структуры данных, когда классы объектов компактны и линейно разделимы:

Несмотря на ошибки разметки при обучении, ряд алгоритмов демонстрируют в своих моделях на тестовом материале не на много меньшую точность, чем если бы они обучались на правильно размеченных данных. Это, в первую очередь, касается наивного байесовского классификатора, машины опорных векторов (при различных степенях полиномиального ядра) и алгоритма ближайшего соседа. Указанный феномен наблюдается при достаточно больших ошибках разметки вплоть до 30-40 % ошибок. При этом для простой структуры данных наилучшие показатели у простейшего алгоритма – наивного байесовского классификатора.

3. Для структуры данных, когда классы объектов имеют сложную гетерогенную структуру:

– Алгоритмы наивный Байес, логистическая регрессия, деревья решений, машина опорных векторов с линейным ядром не способны строить модели разделения классов в условиях неоднородной их структуры.

– Точные результаты разделения классов при отсутствии ошибок разметки продемонстрировали модели, построенные с помощью алгоритмов ближайшего соседа и машины опорных векторов с полиномиальным ядром третьей степени. При снижении точности разметки модели указанных алгоритмов показывают релевантное снижение точности классификации на выборках с ошибками разметки.

– Обучаясь на выборках с ошибками разметки вплоть до уровня ошибок разметки в 30 %, модели SVM (p=3) и БС демонстрируют практически 100 % точность классификации на тестовой выборке с безошибочной разметкой. Более того, даже при 40 % уровне ошибок алгоритм БС показывает точность классификации в 84 % на тесте.

Проведенное исследование высветило ценное для практики свойство алгоритма машины опорных векторов с полиномиальным ядром и алгоритма ближайшего соседа. Несмотря на ошибки разметки данных (ошибки «учителя»), эти алгоритмы способны игнорировать такие ошибки и превосходить «учителя» по точности в задачах классификации данных.

Вместе с тем, указанные алгоритмы имеют как достоинства, так и недостатки, которые следует учитывать для определения их приоритета в задачах классификации данных с неточной разметкой.

Преимущества машины опорных векторов:

– метод сводится к решению задачи квадратичного программирования в выпуклой области, которая всегда имеет единственное решение;

– метод находит разделяющую полосу максимальной ширины, что позволяет в дальнейшем осуществлять более уверенную классификацию.

Недостатки машины опорных векторов.

– метод чувствителен к шумам и стандартизации данных;

– не существует общего подхода к автоматическому выбору ядра (и построению спрямляющего подпространства в целом) в случае линейной неразделимости классов;

– вычислительная (временная) сложность от $O(m^2 \times n)$ до $O(m^3 \times n)$, где m – количество объектов, n – число признаков.

Преимущества алгоритма k–ближайших соседей:

– свобода от априорных предположений о структуре данных;

– строго доказанная точность, близкая к теоретически достижимому пределу при неограниченном увеличении объема выборки [22, 23]. В указанных работах показано, что асимптотические вероятности ошибки для метода 1-БС превышают ошибки оптимального правила Байеса не более чем в два раза;

– результат работы алгоритма интерпретируем. Экспертам в различных областях понятна логика работы алгоритма, основанная на нахождении схожих объектов;

– устойчивость к выбросам и аномальным значениям, поскольку вероятность попадания содержащих их записей в число k-ближайших соседей мала;

– программная реализация алгоритма относительно проста.

Недостатки алгоритма k–ближайших соседей:

– считается, что для работы метода требуется хранить в памяти всю обучающую выборку;

– высокая трудоёмкость из-за необходимости вычисления расстояний до всех примеров. В случае расчёта расстояний между всеми объектами вычислительная сложность составляет $O(m^2 \times n)$, для более изощренных методов, например K-D tree или Ball Tree, $O(n \times m \times \log(m))$;

– практически все исследователи говорят о непростой проблеме выбора метрики для измерения расстояния между объектами, от которой существенным образом зависит достаточный объем выборки;

– Нет теоретических обоснований выбора оптимального числа соседей.

С учетом изложенных выше достоинств и недостатков, по нашему мнению, при решении задач классификации с неточной разметкой данных приоритет принадлежит алгоритму ближайших

соседей. Как показано в наших работах (например, [24, 25]), основная проблема использования метода *k*-ближайших соседей – необходимость хранить в памяти всю обучающую выборку – получает свое разрешение путем формирования для объектов выборки собственных контекстно-зависимых локальных метрик [26], существенно расширяющих «сферу действия» объектов, как представителей своего класса. При таком подходе модели машинного обучения, основанные на методе ближайших соседей, представляют собой ансамбли относительно небольшого круга объектов с привязанными к ним собственными оптимизированными метриками.

7. Выводы

1. На искусственных данных, имитирующих простую и сложную структуру классов многомерных объектов, продемонстрирован феномен относительно слабой зависимости точности различных моделей классификации от ошибок разметки обучающей выборки.

2. Для простой структуры классов, несмотря на ошибки разметки при обучении, практически все исследованные алгоритмы показали в своих моделях на тестовом материале с точной разметкой не на много меньшую точность, чем если бы они обучались на правильно размеченных данных. Указанный феномен наблюдается при достаточно больших ошибках разметки вплоть до 30-40 % ошибок.

3. Алгоритмы машины опорных векторов с полиномиальным ядром и k-ближайших соседей продемонстрировали способность игнорировать ошибки разметки (ошибки «учителя») и существенно превосходить «учителя» по точности в задачах классификации данных как для простой, так и для сложной (гетерогенной) структуры классов.

4. В условиях неточной разметки данных более предпочтительным зарекомендовал себя алгоритм k-ближайших соседей. Он менее трудоёмок, имеет меньше настраиваемых параметров, свободен от априорных предположений о структуре данных, устойчив к аномальным выбросам, интерпретируем. Кроме того, метод k-ближайших соседей обладает существенным потенциалом дальнейшего теоретического и практического развития на основе подхода, связанного с построением контекстно-зависимых локальных метрик. Для эффективной реализации данного подхода необходимо дополнительно решать задачи выбора критериев и способов построения контекстно-зависимых локальных метрик, методов формирования композиций объектов-прецедентов по матрицам близости с нарушением метрических отношений и разработки быстрых алгоритмов поиска ближайших объектов. Указанные задачи являются предметом наших дальнейших исследований.

8. Литература

[1] Дюк В. А. Логические методы машинного обучения (инструментальные средства и практические примеры). - СПб.: Вузиздат. - 2020. - 248 с.

[2] https://www.researchandmarkets.com/reports/5415416 (дата обращения 05.06.2022).

[3] Roh Y.; Heo G.; Whang S. A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective. IEEE Trans. Knowl. Data Eng. – 2019. - No. 33, P. 1328-1347.

[4] CloudFactory. The Ultimate Guide to Data Labeling for Machine Learning. https://www.cloudfactory.com/data-labeling-guide (дата обращения 05.06.2022).

[5] Cognilytica. Data Preparation and Labeling for AI 2020. https://www.cognilytica.com/document/data-preparation-labeling-for-ai-2020/ (дата обращения 05.06.2022).

[6] A Chat with Andrew on MLOps: From Model-centric to Data-centric AI. 2021. - https://youtu.be/06-AZXmwHjo (дата обращения 05.06.2022).

[7] Experian’s 2021 Data experience research report. https://www.edq.com/blog/experians-2021-data-experience-research-report (дата обращения 05.06.2022).

[8] Кафтанников И.Л., Парасич А.В. Проблемы формирования обучающей выборки в задачах машинного обучения // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». - 2016. - Т. 16, No. 3. - С. 15-24.

[9] Zhou Z-H. A brief introduction to weakly supervised learning. Natl Sci Rev, 2018. - Vol. 5, - No.1, - P. 44-53

[10] Adam Kilgarriff and Adam Kilgarriff. Gold standard datasets for evaluating word sense disambiguation programs. Computer Speech and Language, - 1998. - Vol. 12, - No. 3, - P. 453-472.

[11] Angluin D., Laird, P. Learning from noisy examples. Mach. Learn. 1988. - Vol. 2, - No. 4, - P. 343-370.

[12] Blum A., Kalai A., Wasserman H. Noise-tolerant learning, the parity problem, and the statistical query model. JACM 50(4), - 2003. - P. 506-519.

[13] Gao W., Wang L, Li YFet al. Risk minimization in the presence of label noise. In 30th AAAI Conference on Artificial Intelligence, Phoenix, AZ, - 2016. - P. 1575-1581.

[14] Muhlenbach, F., Lallich, S. & Zighed, D.A. Identifying and Handling Mislabelled Instances. Journal of Intelligent Information Systems, -2004. - No. 22. - P. 89-109.

[15] Гилязев Р. А., Турдаков Д. Ю. Активное обучение и краудсорсинг: обзор методов оптимизации разметки данных. Труды ИСП РАН, том 30, вып. 2, - 2018. - С. 215-250.

[16] Noyunsan C., Katanyukul T., Saikaew K. Performance evaluation of supervised learning algorithms with various training data sizes and missing attributes. Engineering and Applied Science Research. – 2018. - No. 45(3), - P. 221-229.

[17] Fernández-Delgado M., Cernadas E., Barro S., Amorim D. Do we need hundreds of classifiers to solve real world classification problems? J Mach Learn Res. – 2014. - No. 15(1). - P. 3133-3181.

[18] Hayes B. Top Machine Learning Algorithms, Frameworks, Tools and Products Used by Data Scientists. July 24, 2020. https://customerthink.com/top-machine-learning-algorithms-frameworks-tools-and-products-used-by-data-scientists/ (дата обращения 05.06.2022).

[19] Eibe Frank, Mark A. Hall, and Ian H. Witten. The WEKA Workbench. Online Appendix for «Data Mining: Practical Machine Learning Tools and Techniques», Morgan Kaufmann, Fourth Edition, - 2016.

[20] Platt C. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines, In: Advances in Kernel Methods - Support Vector Learning, ed. by B. Schölkopf and C. J. C. Burges and A. J. Smola, Cambridge, MA, MIT Press. – 1999. - P. 185-208.

[21] Куинлан, Дж. Р. C4.5: Программы для машинного обучения. Издательство Морган Кауфманн, 1993.

[22] Cover T., Hart P. Nearest neighbour pattern classification. IEEE Trans. Inform. Theory, - Vol. IT 13. - 1967. - P. 21-27.

[23] Duda R.O., Hart P. E. Pattern classification and scene analysis, Wiley, New York. - 1973.

[24] Дюк В.А., Брюс Ф.О., Богданов А.В. Перспектива экстенсиональных методов машинного обучения // Информация и космос. - No. 2. - 2020. - С. 69-76.

[25] Дюк В.А., Михов О.М., Брюс Ф.О. Экстенсиональные методы машинного обучения // В сб. «Транспорт России: проблемы и перспективы - 2019». Материалы международной-научно-практической конференции. - 2019. - С. 198-202.

[26] Dyuk V.A. Context-dependent local metrics and geometrical approach to the problem of knowledge formation. Journal of Computer and Systems Sciences International. - 1996. - Vol. 35. – No. 5. - P. 715-722.

An Experimental Study of the Machine Learning Algorithms Response to Data Labeling Errors

Diuk V.A.

Institute of Transport Problems after N.S. Solomenko of the Russian Academy of Sciences

v_duke@mail.ru

Abstract. There are authoritative opinions that data labeling is today the most important element in the procedure for creating AI systems based on machine learning methods. At the same time, in particular with crowdsourcing, there is a serious problem of inaccurate data labeling. The materials of this article complement the well-known approaches to solving this problem by studying the reaction to inaccurate data labeling of some popular machine learning methods. These are naive Bayesian classifier, three-layer perceptron, nearest neighbor method (KNN), decision trees, random forest, logistic regression, support vector machine (SVM). We trained algorithms on copies of specially generated data with different proportions of labeling errors and then tested them on data with accurate labeling. Based on the results of the experiment on data simulating a simple and complex structure of two classes of multidimensional objects, the phenomenon of a relatively weak dependence of the accuracy of the KNN and SVM classification models on the labeling errors of the training sample was demonstrated. In conditions of inaccurate data labeling, the KNN algorithm is more preferable. It is less complicated, has fewer adjustable parameters, is free from a priori assumptions about the data structure, is resistant to anomalous outliers, and is interpretable. In addition, this method has significant potential for further theoretical and practical development based on the approach associated with the construction of context-dependent local metrics.

Keywords: machine learning, artificial intelligence, data labeling errors, context-dependent local metrics.