Научная статья
УДК 004.855.5
DOI: 10.14529/ctcr240202

ПЕРЕОБУЧЕНИЕ В МАШИННОМ ОБУЧЕНИИ: ПРОБЛЕМЫ И РЕШЕНИЯ

В.А. Парасич[1], *pva16@yandex.ru, https://orcid.org/0000-0003-3593-2345*
И.В. Парасич[1], *parasichiv@mail.ru, https:/orcid.org/0000-0003-1965-8737*
Г.И. Волович[2], *g_volovich@mail.ru, https://orcid.org/0000-0002-3572-1823*
С.Г. Некрасов[1], *nekrasovsg@susu.ru*
А.В. Парасич[3], *parasichav@yandex.ru, https://orcid.org/0000-0003-2728-0893*
[1] *Южно-Уральский государственный университет, Челябинск, Россия*
[2] *ООО «Челэнергоприбор», Челябинск, Россия*
[3] *ООО «ТРИДИВИ», Челябинск, Россия*

Аннотация. Переобучение является одним из важнейших факторов, влияющих на качество работы алгоритмов машинного обучения. При решении задач машинного обучения важно уметь эффективно решать проблему переобучения. **Цель исследования.** Цель данной статьи – изучить проблему переобучения в задачах машинного обучения. В статье рассматриваются эффективные приёмы обучения, направленные на предотвращение переобучения. **Материалы и методы.** Основное внимание в статье уделяется различным важным с практической точки зрения нестандартным вопросам, связанным с переобучением. Рассматриваются различные причины переобучения, его последствия и методы борьбы с переобучением. Изучается зависимость переобучения и обобщающей способности от качества признаков и свойств обучающей выборки. Особое внимание уделяется особенностям обучения и формирования обучающей выборки в многомерных пространствах признаков. Рассматривается вопрос правильного формирования обучающей выборки и правильного добавления данных в обучающую выборку с точки зрения предотвращения переобучения, а также влияние неправильного распределения целевой переменной на переобучение. Объясняется, почему методы добавления в обучающую выборку некорректных данных, такие как MixUp и CutMix, могут повысить качество обучения. Рассматривается проблема уверенности алгоритма в своих предсказаниях, а также проблема overconfidence алгоритма в неправильных предсказаниях, характерная в том числе для ChatGPT. Рассматривается проблема оценки качества работы алгоритма. Показано, почему нормализация может помочь избежать переобучения. **Результаты.** Предложен алгоритм обучения деревьев решений Random Samples Mix-Up, предназначенный для борьбы с переобучением, который позволяет улучшить качество обучения деревьев решений. Проводится сравнительный анализ качества моделей до и после применения данного метода борьбы с переобучением. Эксперименты на реальных данных подтверждают эффективность данного метода. **Заключение.** Результаты исследования могут быть полезны при разработке новых алгоритмов машинного обучения и повышении эффективности существующих. Результаты исследования могут быть полезны для разработчиков алгоритмов машинного обучения и специалистов в области искусственного интеллекта.

Ключевые слова: машинное обучение, переобучение, глубокое обучение, деревья решений, обучение метрик, обучающая выборка

Для цитирования: Переобучение в машинном обучении: проблемы и решения / В.А. Парасич, И.В. Парасич, Г.И. Волович и др. // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». 2024. Т. 24, № 2. С. 18–27. DOI: 10.14529/ctcr240202

Original article
DOI: 10.14529/ctcr240202

OVERFITTING IN MACHINE LEARNING: PROBLEMS AND SOLUTIONS

V.A. Parasich[1], *pva16@yandex.ru, https://orcid.org/0000-0003-3593-2345*
I.V. Parasich[1], *parasichiv@mail.ru, https:/orcid.org/0000-0003-1965-8737*
G.I. Volovich[2], *g_volovich@mail.ru, https://orcid.org/0000-0002-3572-1823*
S.G. Nekrasov[1], *nekrasovsg@susu.ru*
A.V. Parasich[3], *parasichav@yandex.ru, https://orcid.org/0000-0003-2728-0893*
[1] *South Ural State University, Chelyabinsk, Russia*
[2] *LLC Chelenergopribor, Chelyabinsk, Russia*
[3] *LLC TRIDIVI, Chelyabinsk, Russia*

Abstract. Overfitting is one of the most important factors affecting the performance of machine learning algorithms. When solving machine learning problems, it is important to be able to effectively solve the problem of overfitting. **The research objective.** The purpose of this article is to study the problem of overfitting in machine learning tasks. The article discusses effective learning methods aimed at preventing overfitting. **Material and methods.** The focus of the article is on various non-standard issues related to overfitting that are important from a practical point of view. Various causes of overfitting, its consequences and methods of combating overfitting are considered. The dependence of overfitting and generalizing ability on the quality of features and properties of the training set is studied. Particular attention is paid to the features of training and the formation of a training sample in multidimensional feature spaces. The question of the correct formation of the training set and the correct addition of data to the training set from the point of view of overfitting prevention, as well as the impact of incorrect distribution of the target variable on overfitting, is considered. It is explained why the methods of adding incorrect data to the training set, such as MixUp and CutMix, can improve the quality of training. The problem of the algorithm's confidence in its predictions is considered, as well as the problem of algorithm overconfidence in incorrect predictions, which is also typical for ChatGPT. The problem of assessing the quality of the algorithm is considered. It is shown why normalization can help avoid overfitting. **Results.** An algorithm for training decision trees Random Samples Mix-Up is proposed to combat overfitting, which improves the quality of training decision trees. A comparative analysis of the quality of models before and after the application of this method of combating overfitting is carried out. Experiments on real data confirm effectiveness of this method. **Conclusion.** The results of the study can be useful in developing new machine learning algorithms and improving the efficiency of existing ones. The results of the study can be useful for developers of machine learning algorithms and specialists in the field of artificial intelligence.

Keywords: machine learning, overfitting, deep learning, decision trees, metric learning, training set

For citation: Parasich V.A., Parasich I.V., Volovich G.I., Nekrasov S.G., Parasich A.V. Overfitting in machine learning: problems and solutions. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronics*. 2024;24(2):18–27. (In Russ.) DOI: 10.14529/ctcr240202

Введение

Переобучение – это широко распространённая проблема в машинном обучении, которая сильно влияет на качество обучения. Основной теорией машинного обучения (и переобучения) на сегодняшний день является теория Вапника – Червоненкиса [1]. Одно из основных положений этой теории – обобщающая способность алгоритма зависит от сложности модели. Рассмотрим, от чего ещё может зависеть переобучение.

Почему у алгоритмов машинного обучения возникает способность правильно работать на данных, которых нет в обучающей выборке, и почему эти алгоритмы оказываются способными к обобщению на те данные, которые не участвовали в процессе обучения и про которые алгоритм ничего не знает? По сути, обобщающая способность является следствием того, что похожие объекты имеют похожие значения признаков. Иначе объекты одного класса могут оказаться произвольно разбросанными по признаковому пространству и обучение будет крайне затруднено. Похожей концепцией является требование низкого *variance* модели в *bias-variance tradeoff* [2].

Поэтому рассмотрим далее утверждение о том, что обобщающая способность зависит от устойчивости и стабильности признаков, входящих в модель.

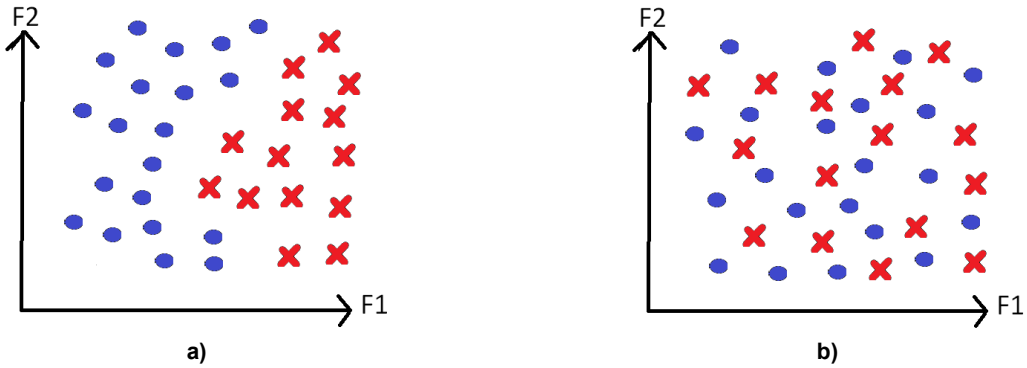

Рис. 1. Пример пространства признаков в случае устойчивых (a) и неустойчивых (b) признаков
Fig. 1. An example of a feature space in the case of stable (a) and unstable (b) features

Представим для наглядности двумерное пространство признаков. Если признаки объектов не будут обладать свойством стабильности, то объекты окажутся раскиданными вперемешку по этому признаковому пространству (рис. 1) и обобщение будет, по сути, невозможно, а возможно будет только заучивание конкретных обучающих примеров. Похожей концепцией является гипотеза компактности [3, 4].

1. Эксперимент

Самыми неустойчивыми и нестабильными из вычислительных функций с точки зрения variance являются криптографические хэш-функции [5] – они построены таким образом, чтобы малому изменению входного значения функции соответствовало большое изменение выходного значения (для такой функции сложно подобрать обратную функцию). Поэтому проведём следующий эксперимент – заменим значения всех признаков модели на значения криптографической хэш-функции от этих признаков. Для эксперимента был выбран стандартный набор данных «Ирисы Фишера» [6]. В этом наборе данных 150 объектов, для каждого известны 4 числовых признака – petal_length, petal_width, sepal_length, sepal_width. Прохэшируем эти признаки (использовался алгоритм sha256 [7]) и обучим на них SVM (SVC) [8]. В результате признаки похожих объектов перестали быть похожими, а это ключевое условие для того, чтобы алгоритм мог обобщаться на данные, которых нет в обучающей выборке. В эксперименте 105 объектов включены в обучающую выборку, оставшиеся 45 – в тестовую.

Сравним результаты обучения на исходных и на модифицированных признаках. При обучении на исходных признаках качество на обучающей выборке составило 98,1 %, на тестовой – 97,8 % (табл. 1). При обучении на хэшированных признаках качество на обучающей выборке составило 100 %, на тестовой – 26,7 %, что примерно соответствует качеству случайного угадывания. То есть при обучении на хэшированных признаках произошло сильное переобучение из-за того, что нарушилось свойство стабильности признаков (признаки похожих объектов перестали быть похожими).

Таблица 1

Качество классификации SVC на датасете Fisher's Iris в зависимости от используемых признаков

Table 1

The quality of SVC classification on the Fisher's Iris dataset depending on the features used

	Обучение на исходных признаках	Обучение на криптографических хэш-функциях от признаков
Качество на обучающей выборке, %	98,1	100
Качество на тестовой выборке, %	97,8	26,7

Напрашивающийся отсюда вывод: в задачах машинного обучения следует использовать устойчивые признаки с низким *variance* (то есть те признаки, которые максимально не похожи на криптографические хэш-функции).

Примером неустойчивой функции является деление со знаменателем, близким к нулю. Также неустойчивыми являются операции взятия максимума и минимума, лучше вместо них использовать перцентили. Примером неустойчивого преобразования из области геометрии является восстановление прямой по двум близко лежащим точкам. В этом случае при небольших изменениях в положении точек восстановленная по ним прямая будет изменяться очень сильно.

Другой проблемой являются признаки с очень редкими значениями. Например, если в задаче классификации для некоторого значения V некоторого категориального признака F в обучающей выборке существует только один пример x_i с таким значением признака $F(x) = V$ (при этом целевая переменная для объекта x_1 принимает значение $C(x_i)$), то алгоритм может выучить тривиальную закономерность $(F(x) = V) => (y(x) = C(x_i))$. В данном случае признак F не является устойчивым с точки зрения *variance*.

2. Проблема уверенности (confidence) алгоритма в своих предсказаниях

Часто при использовании моделей машинного обучения возникает необходимость определить уверенность модели в своих предсказаниях. Эту информацию можно использовать, например, для того, чтобы отбрасывать те предсказания, в которых модель не уверена, а учитывать только те, в которых она уверена.

Однако здесь стоит учитывать следующее. Модель, полученная с помощью машинного обучения, как правило, хуже всего работает на тех типах данных, которых не было в обучающей выборке. А раз этих данных не было в обучающей выборке, то у модели при обучении не было возможности правильно настроить выдачу уверенности для таких данных. Поэтому часто возникает проблема излишней уверенности (*overconfidence*) модели в неправильных предсказаниях. На практике определение уверенности алгоритма в его предсказаниях обычно работает также достаточно плохо.

Проблема *overconfidence* актуальна [9] и для современных больших языковых моделей (LLM), таких как GPT-3 [10] и ChatGPT. ChatGPT, например, иногда пишет правдоподобно звучащие, но неправильные или бессмысленные ответы. При этом сама ChatGPT не может определить, является её ответ правильным или нет, что сильно осложняет её использование в реальных задачах.

3. Проблема оценки качества работы алгоритма

Причём всё то же самое (что и в случае с уверенностью модели в своих предсказаниях) можно сказать и про измерение качества работы алгоритма на тестовой выборке. Алгоритм хуже всего работает на тех данных, которых не было или было мало в обучающей выборке. А если их не было в обучающей выборке и тестовая выборка взята из того же источника, что и обучающая (например получена с помощью классического случайного разделения одной выборки на две части), то, значит, и в тестовой выборке таких данных не было или было мало.

Это касается в том числе и обучения деревьев решений. Самые проблемные с точки обучения вершины – это те вершины, в которые при обучении попало мало данных (в таких вершинах высок риск переобучения). Но если тестовая выборка взята из того же распределения, что и обучающая, то на этапе тестирования в эти вершины попадёт ещё меньше данных, чем на этапе обучения (так как обучающая выборка обычно в несколько раз больше тестовой). Таким образом, метрики качества работы этих вершин будут ещё более неточными, чем ответы в этих вершинах. Можно бороться с этим с помощью того, что делать тестовую выборку в несколько раз больше обучающей и получать более надёжные оценки качества работы, а потом проводить обучение на полной обучающей выборке, используя подобранные таким образом гиперпараметры. Но тогда гиперпараметры, подобранные при обучении на такой уменьшенной обучающей выборке, могут быть далеки от гиперпараметров, оптимальных для обучения на полноразмерной обучающей выборке.

Поэтому рекомендуется тестировать качество модели не только на тестовых выборках, полученных с помощью случайного *train_test_spilt* некоторого датасета, но также тестировать *Cross-dataset Generalization*, то есть тестировать качество в том числе на отдельных датасетах, частей которых не было в обучении.

4. Особенности обучения в высокоразмерных пространствах признаков

Чаще всего в задачах машинного обучения приходится иметь дело с пространствами признаков высокой размерности. При использовании табличных данных у каждого элемента выборки могут быть десятки или сотни признаков. Нейронные сети, применяющиеся в задачах обработки изображений, имеют миллионы параметров, и если использовать число их возможных внутренних состояний в качестве размерности пространства признаков, то и вовсе получится несколько миллионов измерений.

Главной особенность таких пространств является то, что даже при использовании больших обучающих выборок почти всё пространство окажется пустым и не будет заполнено никакими данными. Если мы возьмём пространство размерности 100 и будем рассматривать только его углы (каждый из признаков может иметь по два возможных значения), то получится, что пространство состоит из 2^{100} точек, что составляет величину порядка 10^{30}. Обучающие выборки, как правило, состоят в лучшем случае из нескольких миллионов элементов. Если же признаков будет не 100, а 1000, то всё становится ещё хуже.

Таким образом, результат работы обученного алгоритма в большинстве точек пространства будет или плохо обусловлен, или вообще являться величиной, зависящей в основном от случайных факторов (рис. 2), потому что в окрестности этих точек нет никаких данных. Также может наблюдаться сильное изменение качества работы алгоритма в ходе обучения, или при небольшом изменении гиперпараметров, или при добавлении небольшого количества данных.

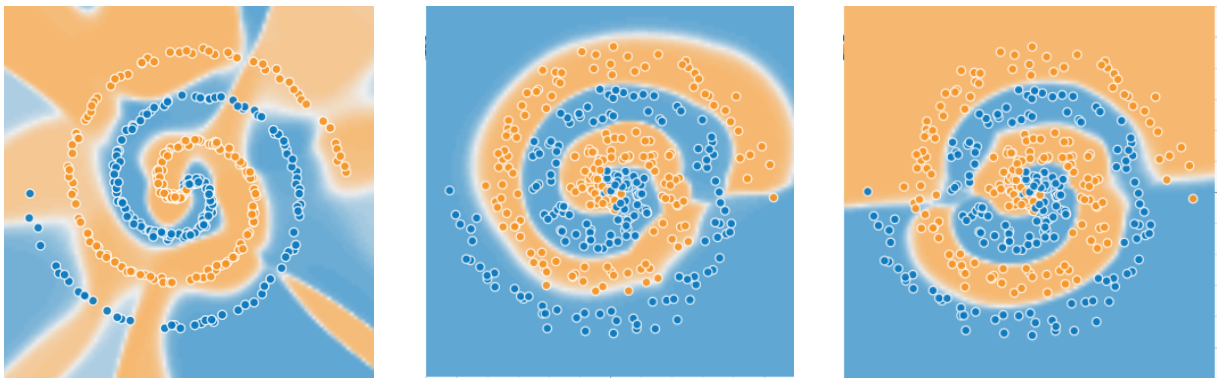

Рис. 2. Примеры разделяющей поверхности обученного алгоритма. В незаполненных областях пространства признаков ответ алгоритма во многом зависит от случайных факторов. В многомерных признаковых пространствах таких незаполненных областей будет гораздо больше
Fig. 2. Examples of the separating surface of the trained algorithm. In unfilled areas of the feature space, the algorithm's response largely depends on random factors. In multidimensional feature spaces, there will be much more such unfilled areas

Один из возможных методов решения данной проблемы – заполнить пустующие области, каким-то образом интерполируя (модифицируя) данные, уже имеющиеся в обучающей выборке. Для задач обработки изображений существуют такие методы, как *MixUp* [11] и *CutMix* [12]. Эти методы позволяют повысить качество обучения. При этом полученные изображения могут выглядеть некорректно (то есть в реальности таких изображений никогда не возникает), например, может получиться кошка с головой собаки. Однако лучше настроить ответ алгоритма в этих областях на основании полукорректных данных, чем оставить его полностью случайным. Другим возможным методом для заполнения пустых областей пространства признаков является *Pseudo-Labelling* [13].

5. Влияние неправильного распределения целевой переменной в обучающей выборке на переобучение

Что такое ошибка на тестовой выборке? Это разница между распределением ответов решающего правила и распределением тестовой выборки в точках, соответствующих элементам тестовой выборки. Однако распределение тестовой выборки не обязательно в точности отражает распределение генеральной совокупности. Некоторые данные в тестовой выборке могут отсутствовать, также может иметь место неправильный баланс классов относительно генеральной сово-

купности (во всей выборке либо на определённом участке пространства признаков). То есть может существовать разница в распределениях обучающей и тестовой выборок. Таким образом, ошибка на тестовой выборке состоит из двух слагаемых: разница распределения ответов алгоритма и генеральной совокупности и разница распределения генеральной совокупности и распределения тестовой выборки.

В самом простом случае в тестовой выборке могут быть неправильно сбалансированы классы относительно генеральной совокупности, причём в обучающей выборке они могут быть сбалансированы более правильно, чем в тестовой. В таком случае, подстраивая баланс ответов под баланс тестовой выборки, мы улучшаем качество на тестовой выборке, но ухудшаем качество на генеральной совокупности. При этом на самом деле не происходит повышения обобщающей способности алгоритма, хотя качество на тестовой выборке растёт. Также не является настоящим переобучением обратный случай, когда мы знаем, что в обучающей выборке распределение классов более правильное, чем в тестовой, и двигаем баланс в сторону от тестовой выборки к обучающей (хотя на графиках обучения это будет выглядеть как переобучение – ошибка на обучающей выборке будет уменьшаться, а на тестовой – увеличиваться). Поэтому качество работы алгоритма на тестовой выборке нельзя считать абсолютно точной мерой качества работы алгоритма и его обобщающей способности. Поэтому для повышения надёжности измерения качества рекомендуется применять различные методы кросс-валидации, в том числе *k-Fold Cross-Validation* [14], а также тестировать *Cross-Dataset Generalization*.

Для простоты дальнейших рассуждений будем считать, что у нас есть тестовая выборка, точно отражающая генеральную совокупность (хотя в большинстве случаев это недостижимо).

Как говорилось выше, ошибка на тестовой выборке – это разница между распределением ответов модели и распределением тестовой выборки в точках, соответствующих элементам тестовой выборки. Поэтому причиной переобучения может быть не только излишняя сложность модели или нехватка обучающих данных, но и искривление распределения (неправильное распределение) целевой переменной в обучающей выборке (или в определённых участках признакового пространства в обучающей выборке) относительно тестовой выборки (потому что распределение ответов алгоритма, как правило, отражает распределение целевой переменной в обучающей выборке).

Такое искривление может возникнуть, например, из-за недосэмплирования (рис. 3). Из-за особенностей сбора обучающей выборки в некоторой подобласти пространства признаков может не оказаться данных определённого класса, из-за чего обученная модель будет возвращать в этой области пространства признаков неправильный ответ.

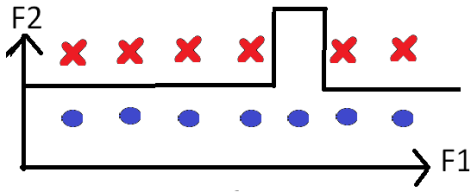

Рис. 3. Пример заучивания ложной закономерности из-за недосэмплирования
Fig. 3. An example of learning a false pattern due to undersampling

Также может возникнуть ситуация пересэмплирования – однотипные (или одинаковые) данные включены в выборку слишком большое число раз, из-за чего сильно искажается распределение в их окрестности. Всё это может негативным образом сказаться на качестве обучения. Примером данной проблемы являются покупки оптовиков в задачах, связанных с предсказанием спроса.

Одним из самых эффективных приёмов улучшения качества обучения является добавление в обучающую выборку сложных обучающих примеров (похожих на те, которые распознаются плохо). Однако при таком добавлении есть большой риск создать искривление распределения в той области, в которую добавляются данные, особенно если мы добавляем примеры только одного класса. Это может привести к тому, что ухудшится качество работы на объектах других

классов. При этом может показаться, что это происходит из-за ограниченной ёмкости модели (новые данные «не входят» в модель и поэтому улучшить качество работы на них нельзя). Однако на самом деле стоит просто исправить искривление распределения. Поэтому при добавлении данных в обучающую выборку надо следить за тем, чтобы не создать дополнительных искривлений распределения, при этом не рекомендуется добавлять примеры только одного класса.

При этом в датасетах часто распределение целевой переменной искривлено по тем или иным параметрам. Достаточно сложно обеспечить, чтобы распределение было изначально правильным по всем возможным параметрам. Выравнивание распределения может стать эффективным приёмом повышения качества.

Факторы, влияющие на наблюдаемые признаки объекта, можно разделить на внутренние и внешние (например, условия освещения в задачах обработки изображений). Таким образом, один и тот же объект может оказаться в разных точках пространства признаков в зависимости от внешних условий (разброс этих точек будет зависеть от устойчивости признаков к изменению внешних условий).

При этом, если при изменении внешних условий (например освещения) данные попадают в существенно разные участки пространства признаков, нам надо обеспечить, чтобы в каждом из этих участков пространства признаков распределение целевой переменной было правильным (не возникало искривления распределений). Например, если в задаче классификации собак и кошек (в задаче отличить собаку от кошки) окажется, что в обучающей выборке в условиях очень яркого освещения есть только изображения собак (или их в несколько раз больше, чем изображений кошек), то алгоритм выучит соответствующую ложную закономерность.

Одним из возможных решений данной проблемы может быть нормализация. В случае с разной освещённостью мы можем привести все изображения к одинаковой средней яркости. Тогда данные будут не так сильно разбросаны по разным участкам пространства признаков и будет проще обеспечить правильное распределение целевой переменной во всех случаях (не надо будет отдельно обеспечивать, чтобы было правильное распределение классов и при сильном, и при слабом освещении).

6. Random Samples Mix-Up

Одна из основных причин переобучения при обучении деревьев решений [15] – при обучении (выборе параметров) некоторых вершин используется малое число обучающих примеров. Так происходит потому, что при обучении каждой вершины множество обучающих данных делится на две части, соответствующие левому и правому сыновьям данной вершины, и далее левый и правый сыновья вершины обучаются на уменьшенных подмножествах оригинального множества данных. Таким образом, при обучении нижних уровней дерева для обучения конкретной вершины используется гораздо меньше данных, чем было в исходном множестве данных или при обучении вершин на верхних уровнях дерева. В лучшем случае при полностью равномерном разделении выборки на две части в каждой из вершин минимальное число примеров для обучения вершины на уровне D будет равно $N/2^D$, где N – размер обучающей выборки, D – глубина вершины. Однако в реальности разделение выборки в каждой из вершин происходит неравномерно, поэтому будут вершины, в которых попадёт ещё меньше данных.

Естественное решение этой проблемы – увеличить число данных, которые используются при обучении вершины. Рассмотрим способы сделать это без расширения обучающей выборки. Мы можем использовать при обучении вершины те обучающие примеры, которые должны были использоваться при обучении других вершин. Чтобы не выделять дополнительную память, мы можем в процессе обучения каждой из вершин для каждого отдельного обучающего примера с некоторой вероятностью p перераспределять его не в ту вершину, в которую он должен был попасть. Легко убедиться, что при такой процедуре у тех вершин, в которых было много данных, будет отобрано много данных, а у тех вершин, в которых было мало данных, будет отобрано мало данных (но в них будет добавлено много данных). То есть произойдёт перераспределение обучающей выборки в пользу тех вершин, в которых было мало данных.

Возможное обобщение данного метода – перемешивать данные между вершинами не случайно, а в зависимости от значений признаков объектов. То есть таким образом, чтобы в другую

вершину попадали те объекты, которые имеют более высокую вероятность попасть в эту вершину, если на значения их признаков повлияет шум (или это будет объект, похожий на данный, но с немного другими значениями признаков). Для этого на этапе выбора признака в вершину будем случайно модифицировать значения данного признака для некоторых объектов, вследствие чего часть объектов обучающей выборки в процессе обучения попадёт не в свою вершину.

В наших экспериментах этот метод сам по себе не дал прироста в качестве обучения (хотя это может быть не так в других задачах). Однако если скомбинировать его с базовой версией метода (то есть для половины объектов обучающей выборки использовать базовую схему с полностью случайным перемешиванием объектов, а для другой половины объектов – случайную модификацию признаков), то получится добиться улучшения качества обучения. Результаты экспериментов представлены в табл. 2.

Таблица 2

Качество классификации на тестовой выборке при разных значениях вероятности p_m изменения вершины в дереве для обучающего примера

Table 2

The quality of classification on the test sample for different values of the probability p_m of changing the node in the tree for training sample

Вероятность изменения вершины для обучающего примера	$p_m = 0$	$p_m = 0{,}1$	$p_m = 0{,}04$	$p_m = 0{,}02$	$p_m = 0{,}01$	$p_m = 0{,}005$
Качество классификации случайное перемешивание примеров, %	67,66	68,24	68,39	68,22	67,96	67,82
Качество классификации, случайная модификация признаков, %	67,66	68,15	68.21	68,27	67,85	67,76
Качество классификации, комбинированный алгоритм, %	67,66	68,55	68,78	68,47	68,03	67,95

Выводы

Сформулируем кратко основные выводы.

– Обобщающая способность алгоритма зависит от устойчивости признаков. Похожие объекты должны иметь похожие значения признаков, в противном случае возникает переобучение. Поэтому при решении задач машинного обучения следует использовать устойчивые признаки.

– Переобучение также может быть следствием неправильного распределения целевой переменной в пространстве признаков.

– При добавлении данных в обучающую выборку важно не создавать искривление распределения целевой переменной в пространстве признаков.

– Наличие в выборке большого количества однотипных данных (или данных из одного источника) может привести к искривлению распределения целевой переменной в пространстве признаков и, следовательно, к ухудшению качества обучения.

– Устранение неправильного распределения целевой переменной в обучающей выборке может привести к повышению качества.

– В многомерных пространствах признаков многие области не заполнены никакими объектами обучающей выборки, поэтому ответ обученного алгоритма в таких областях будет плохо обусловленным или случайным. Поэтому имеет смысл применять аугментации, производящие некорректные изображения (например, MixUp или CutMix), чтобы хоть как-то заполнить пустующие области и сделать ответ алгоритма в них менее случайным.

Список литературы

1. Вапник В.Н., Червоненкис А.Я. Теория распознавания образов. Статистические проблемы обучения. М.: Наука, 1974. 416 с.
2. Reconciling modern machine-learning practice and the classical bias–variance trade-off / M. Belkin, D. Hsu, S. Ma, S. Mandal // Proceedings of the National Academy of Sciences. 2019. Vol. 116, no. 32. P. 15849–15854. DOI: 10.1073/pnas.1903070116
3. Аркадьев А.Г., Браверман Э.М. Обучение машины распознаванию образов. М.: Наука, 1964. 110 с.
4. Загоруйко Н.Г. Гипотезы компактности и λ-компактности в методах анализа данных // Сибирский журнал индустриальной математики. 1998. Т. 1, № 1, С. 114–126.
5. Augot D., Finiasz M., Sendrier N. A fast provably secure cryptographic hash function // Cryptology ePrint Archive. 2003. No. 230. P. 3–4.
6. Fisher R.A. The use of multiple measurements in taxonomic problems // Annals of eugenics. 1936. Vol. 7, no. 2. P. 179–188. DOI: 10.1111/j.1469-1809.1936.tb02137.x
7. Dang Q. Secure Hash Standard. Federal Inf. Process. Stds. (NIST FIPS), National Institute of Standards and Technology, Gaithersburg, MD, 2015. DOI: 10.6028/NIST.FIPS.180-4
8. Вапник В.Н. Восстановление зависимостей по эмпирическим данным. М.: Наука, 1979. 448 с.
9. Xiao Y., Wang W.Y. On hallucination and predictive uncertainty in conditional language generation // arXiv preprint arXiv:2103.15025. 2021. DOI: 10.48550/arXiv.2103.15025
10. Language models are few-shot learners / T. Brown et al. // Advances in neural information processing systems. 2020. Vol. 33. P. 1877–1901.
11. mixup: Beyond empirical risk minimization / H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz // arXiv preprint arXiv:1710.09412. 2017. DOI: 10.48550/arXiv.1710.09412
12. Cutmix: Regularization strategy to train strong classifiers with localizable features / S. Yun, D. Han, S.J. Oh et al. // Proceedings of the IEEE/CVF international conference on computer vision. 2019. P. 6023–6032. DOI: 10.1109/ICCV.2019.00612
13. Pseudo-labeling and confirmation bias in deep semi-supervised learning / E. Arazo, D. Ortego, P. Albert et al. // 2020 International Joint Conference on Neural Networks (IJCNN). IEEE. 2020. P. 1–8. DOI: 10.1109/IJCNN48605.2020.9207304
14. The 'K' in K-fold Cross Validation / D. Anguita, L. Ghelardoni, A. Ghio et al. // Conference: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN). 2012. P. 441–446.
15. Magee J.F. Decision trees for decision making // Harvard Business Review. 1964. Vol. 42, no. 4. P. 126–138.

References

1. Vapnik V.N., Chervonenkis A.Ya. *Teoriya raspoznavaniya obrazov. Statisticheskie problemy obucheniya* [Theory of pattern recognition. Statistical learning problems]. Moscow: Nauka; 1974. 416 p. (In Russ.)
2. Belkin M., Hsu D., Ma S., Mandal S. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*. 2019;116(32):15849–15854. DOI: 10.1073/pnas.1903070116
3. Arcadiev A.G., Braverman E.M. *Obuchenie mashiny raspoznavaniyu obrazov* [Teaching a Machine to Pattern Recognition]. Moscow: Nauka; 1964. 110 p. (In Russ.)
4. Zagoruiko N.G. [Hypotheses of compactness and λ-compactness in methods of data analysis]. *Sibirskiy zhurnal industrial'noy matematiki*. 1998;1(1):114–126. (In Russ.)
5. Augot D., Finiasz M., Sendrier N. A fast provably secure cryptographic hash function. *Cryptology ePrint Archive*. 2003;(230):3–4.
6. Fisher R.A. The use of multiple measurements in taxonomic problems. *Annals of eugenics*. 1936;7(2):179–188. DOI: 10.1111/j.1469-1809.1936.tb02137.x
7. Dang Q. *Secure Hash Standard*. Federal Inf. Process. Stds. (NIST FIPS), National Institute of Standards and Technology, Gaithersburg, MD; 2015. DOI: 10.6028/NIST.FIPS.180-4
8. Vapnik V.N. *Vosstanovlenie zavisimostey po empiricheskim dannym* [Recovery of dependences on empirical data]. Moscow: Nauka; 1979. 448 p. (In Russ.)

9. Xiao Y., Wang W.Y. On hallucination and predictive uncertainty in conditional language generation. *arXiv preprint arXiv:2103.15025*. 2021. DOI: 10.48550/arXiv.2103.15025

10. Brown T. et al. Language models are few-shot learners. *Advances in neural information processing systems*. 2020;33:1877–1901.

11. Zhang H., Cisse M., Dauphin Y.N., Lopez-Paz D. mixup: Beyond empirical risk minimization. *arXiv preprint. arXiv:1710.09412*. 2017. DOI: 10.48550/arXiv.1710.09412

12. Yun S., Han, D., Oh S.J., Chun S., Choe J., Yoo Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019. P. 6023–6032. DOI: 10.1109/ICCV.2019.00612

13. Arazo E., Ortego D., Albert P., O'Connor N.E., McGuinness K. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In: *2020 International Joint Conference on Neural Networks (IJCNN). IEEE*. 2020. P. 1–8. DOI: 10.1109/IJCNN48605.2020.9207304

14. Anguita D., Ghelardoni L., Ghio A., Oneto L., Ridella S. The 'K' in K-fold Cross Validation. In: *Conference: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN).* 2012. P. 441–446.

15. Magee J.F. Decision trees for decision making. *Harvard Business Review.* 1964;42(4):126–138.

Информация об авторах

Парасич Виктор Александрович, канд. техн. наук, доц., доц. кафедры электронных вычислительных машин, Южно-Уральский государственный университет, Челябинск, Россия; pva16@yandex.ru.

Парасич Ирина Васильевна, канд. техн. наук, доц. кафедры математического и компьютерного моделирования, Южно-Уральский государственный университет, Челябинск, Россия; parasichiv@mail.ru.

Волович Георгий Иосифович, д-р техн. наук, проф., директор, ООО «Челэнергоприбор», Челябинск, Россия; g_volovich@mail.ru.

Некрасов Сергей Геннадьевич, д-р техн. наук, проф. кафедры информационно-измерительной техники, Южно-Уральский государственный университет, Челябинск, Россия; nekrasovsg@susu.ru.

Парасич Андрей Викторович, инженер-программист, ООО «ТРИДИВИ», Челябинск, Россия; parasichav@yandex.ru.

Information about the authors

Victor A. Parasich, Cand. Sci. (Eng.), Ass. Prof., Ass. Prof. of the Department of Electronic Computing Machines, South Ural State University, Chelyabinsk, Russia; pva16@yandex.ru.

Irina V. Parasich, Cand. Sci. (Eng.), Ass. Prof. of the Department of Mathematical and Computer Modeling, South Ural State University, Chelyabinsk, Russia; parasichiv@mail.ru.

Georgiy I. Volovich, Dr. Sci. (Eng.), Prof., Director, LLC Chelenergopribor, Chelyabinsk, Russia; g_volovich@mail.ru.

Sergey G. Nekrasov, Dr. Sci. (Eng.), Prof. of the Department of Information and Measuring Technology, South Ural State University, Chelyabinsk, Russia; nekrasovsg@susu.ru.

Andrey V. Parasich, Software engineer, LLC TRIDIVI, Chelyabinsk, Russia; parasichav@yandex.ru.

Статья поступила в редакцию 16.05.2023
The article was submitted 16.05.2023