Выбор объектов для обучения в условиях сильной несбалансированности классов

Анна Кузьмишкина, Ольга Баринова, Антон Конушин
факультет Вычислительной Математики и Кибернетики
Московский Государственный Университет им. Ломоносова, Москва, Россия
akuzmishkina@gmail.com, obarinova@graphics.cs.msu.ru, ktosh@graphics.cs.msu.ru

АННОТАЦИЯ

В данной работе рассматривается проблема несбалансированных классов (imbalanced data) в задачах классификации, характерная для области компьютерного зрения. В статье приводится обзор и сравнение существующих методов настройки классификаторов в условиях несбалансированных классов. Предлагается итеративный алгоритм выбора прецедентов для обучения, на каждой итерации которого строится сбалансированная подвыборка данных. В статье приведены результаты сравнительных экспериментов на данных из UCI [1] и реальных данных из задач компьютерного зрения.

Keywords: *классификация**,** несбалансированные классы.*

1. ВВЕДЕНИЕ

Алгоритмы машинного обучения широко применяется в работах по компьютерному зрению. Примером таких работ служат системы поиска лиц на изображениях [2] или системы поиска горизонтальных линий на фотографиях городских сцен [3]. Большинство алгоритмов классификации направлено на минимизацию общей ошибки обучения. Однако в задачах компьютерного зрения большинство задач имеют несколько другую формулировку. Например, в задачах обнаружения объектов на изображении цель состоит в минимизации количества ложных срабатываний при достаточной частоте верных обнаружений. При этом число примеров для "объекта" мало по сравнению с числом примером "фона", такое распределение называется несбалансированными классами. При использовании стандартных методов классификации в такой ситуации часто возникает проблема, что уменьшая общую ошибку, классификатор полностью относит интересуемый класс к шуму.

Существует множество работ по проблеме несбалансированных классов в задачах бинарной классификации. Наиболее распространенными являются методы на основе выбора прецедентов обоих классов так, чтобы число прецедентов обоих классов уравнялось. Группа этих методов будет описана в разделах 2.1 и 2.2. Другим примером служат методы на основе изменения порога решения. Данный метод будет рассмотрен в разделе 2.3.

Предлагаемый алгоритм основан на составлении сбалансированной подвыборки данных при условии сильной несбалансированности классов и большого объема входных данных. Описание алгоритма приводится в разделе 3, примеры работы и результаты представлены в разделе 4.

2. МЕТОДЫ НАСТРОЙКИ КЛАССИФИКАТОРОВ ПРИ НЕСБАЛАНСИРОВАННЫХ КЛАССАХ

Введем обозначения, которыми будем пользоваться в дальнейшем. Будем считать, что стоит задача распознавания объектов 1-го класса на фоне шума 2-го класса. За $N+$ обозначим число прецедентов 1-го класса в обучающей базе, а за $N-$ число прецедентов 2-го класса. Пусть классы несбалансированны, т.е. $N+ << N-$.

Рассмотрим наиболее распространенные методы работы с несбалансированными классами:

* Уменьшение большего класса (Under Sampling)
* Увеличение меньшего класса (Over Sampling)

2.1 Уменьшение большего класса (Under Sampling)

Методы этой группы предлагают проводить обучение на всех прецедентах 1-го класса и выбранных некоторым образом прецедентах 2-го класса [5], [6], [7], [8]. Как правило, число прецедентов 2-го класса в новой обучающей выборке полагают равным N+. При уменьшении большего класса происходит значительное сокращение тренировочной базы и, соответственно, времени работы классификатора. Однако это может вызвать потерю важной информации и, как результат, увеличение общей ошибки. Самым простым методом уменьшения большего класса является случайный выбор прецедентов в обучающую выборку. В теории данный метод не учитывает взаиморасположение прецедентов относительно друг друга и разделяющей поверхности, что может привести к ошибкам в нахождении границы классов. Однако было показано, что на практике этот метод является наиболее эффективным [5].

2.2 Увеличение меньшего класса (Over Sampling)

Данные методы предлагают увеличить число прецедентов меньшего класса [9], [10], [11], [12]. Их преимущество заключается в том, что никакая информация не теряется. Однако главным недостатком является значительное увеличение тренировочной базы, что влечет за собой увеличение времени работы алгоритма классификации и требуемые ресурсы компьютера. Самым простым методов является дублирование произвольных прецедентов 1-го класса. При таком подходе никакой информации о границе классов не добавляется.

2.3 Изменение порога решения

Многие алгоритмы классификации выдают степень уверенности в своем предсказании, например, так делает Boosting [12], Bagging [14], SVM [15]. Знак выхода

классификатора говорит, к какому классу относится объект. Таким образом, окончательное решающее правило – это порог по выходу классификатора, обычно равный нулю. Для таких методов, изменяя порог в решающем правиле, можно получать различные разделяющие поверхности.

Среди достоинств этого метода стоит отметить легкость реализации. Однако изменение порога не гарантирует точность формы границы, что зачастую приводит к сильному повышению общей ошибки.

2.4 Взаимосвязь методов

На данный момент методы машинного обучения по несбалансированной выборке имеют довольно скудное теоретическое обоснование. Один из основных результатов получен в работе [3].

Теорема 1.

Чтобы из текущего порога вероятности решения p0 получить желаемый порог p, необходимо в тренировочную выборку включить* $\frac{p^*}{1-p^*}\times\frac{1-p_0}{p_0}$ *прецедентов 2-го класса.*

Этот результат можно проинтерпретировать следующим образом: для задач с несбалансированными классами изменение тренировочной выборки и изменение порога решения приводят к одним и тем же результатам.

В задачах компьютерного зрения тренировочные базы, как правило, имеют большие размеры (порядка $10^5 - 10^7$ объектов для обучения). Это связано с тем, что сбор базы - достаточно дешевая операция и можно собрать базу любого размера. При этом, специфика приложений в компьютерном зрении предполагает не однократную перенастройку классификатора в процессе работы над системой (например, при изменении набора признаков). Очевидно, что для скорости ее разработки тренировочную выборку надо уменьшать, а никак не увеличивать.

3. ОПИСАНИЕ ПРЕДЛОЖЕННОГО МЕТОДА

В данной работе предлагается итеративный метод выбора объектов большего класса, учитывающий положение прецедентов относительно разделяющей поверхности. Алгоритм выбора объектов для обучения описан ниже (см. Алгоритм 1).

В экспериментах использовались следующие значения параметров алгоритма: p_1=20% от верно классифицированных объектов 1-го класса; q_i = 5% от всех неверно классифицированных объектов ($i = \overline{1:NumSteps}$); p_i=10% от верно классифицированных объектов 1-го класса; r_i равно разнице объемов 1-го и 2-го классов; *NumSteps*=20. Подчеркнем важность параметров: q_i позволяет не ошибаться на шумных прецедентах, расположенных вдоль границы; r_i отвечает за сбалансирование классов.

Для наглядности, рассмотрим работу алгоритма по итерациям на модельных данных. Общая база включает в себя 2152 прецедента, из них 232 1-го класса и 1920 2-го. База была разделена на три равные части: обучение, контроль и тест. В качестве классификатора использовался Random Forest из 100 деревьев.

Алгоритм выбора объектов для обучения

Инициализация:

1. *Настраиваем классификатор на TrainData*
2. *Изменяем порог решающего правила Thresh, так чтобы ошибки первого и второго рода на ValidationData сравнялись*
3. *Считаем калиброванные выходы классификатора на всей TrainData*
4. *Выбираем случайным образом* p_1 *прецедентов 1-го класса среди тех объектов, для которых выходы классификатора лежат в интервале* $[Thresh + q_1, 1]$
5. *Выбираем случайным образом* p_1 *прецедентов 2-го класса среди тех объектов, для которых выходы классификатора лежат в интервале* $[0, Thresh - q_1]$
6. *Составляем новую базу NewData из выбранных прецедентов*

Для $i = 2 : NumSteps$

1. *Повторяем 1-3 пункт этапа инициализации, настраивая классификатор на NewData*
4. *Выбираем случайным образом* p_i *прецедентов 1-го класса среди тех объектов, для которых выходы классификатора лежат в интервале* $[Thresh + q_i, 1]$
5. *Выбираем случайным образом* r_i *прецедентов 2-го класса среди тех объектов, для которых выходы классификатора лежат в интервале* $[0, Thresh - q_i]$
6. *Выбираем случайным образом* p_i *неверно классифицированных прецедентов среди тех объектов, для которых выходы классификатора лежат в интервале* $[\min(Thresh, 0.5), \max(Thresh, 0.5)]$
7. *Добавляем выбранные прецеденты в NewData*

Алгоритм 1. *Итеративный выбор объектов для обучения*

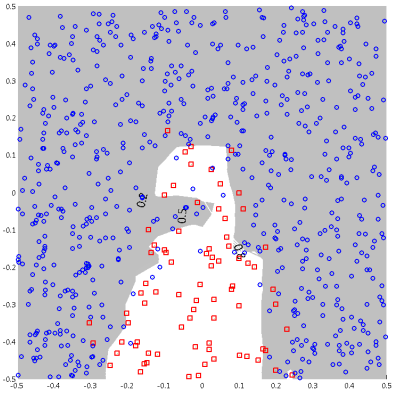
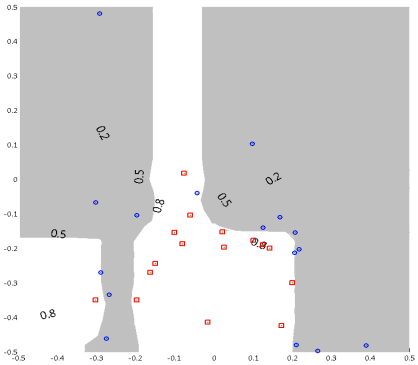


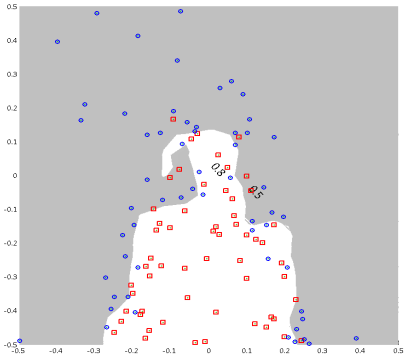
Рисунок 2: *Модельные данные: пример работы классификатора на всей базе*

На рисунке 2 изображена тренировочная база и разделяющая поверхность. Ошибка классификации на тестовой базе оставляет 5%, ошибка после изменения порога 8%.

На рисунке 3 изображены базы, получаемые на разных итерациях алгоритма, и разделяющие поверхности, после обучения на них. Как видно из рисунков, новые разделяющие поверхности сильно отличаются от той, что была на рисунке 2. Очевидно, что из-за нехватки прецедентов 2-го класса, область 1-го класса сильно увеличилась. Обучение на базе из рисунка 3а дает ошибку на контроле 10% и после изменения порога 17%.



(а)



(б)

Рисунок 3: *Пример работы классификатора на модельных данных: (а) - 1 итерация алгоритма, (б) - 10 итерация алгоритма.*

После этапа инициализации выбор неверно классифицированных прецедентов происходит из интервала $[\min(Thresh, 0.5), \max(Thresh, 0.5)]$ (*Thresh* – порог классификатора, при котором ошибка 1-го рода равна ошибке 2-го рода), что позволяет уточнять разделяющую поверхность. Такие прецеденты лежат вдоль искомой границы.

Как видно из рисунка 3б, разделяющая поверхность уже на 10 итерации достаточно сильно уточнилась. Ошибка после обучения на такой базе равна 7%, что приближается к результату, который был получен на всей базе. Однако объем полученной базы значительно меньше исходного, а ошибки на ней сбалансированы. Обычно порядка 20 итераций алгоритма достаточно для получения сбалансированной базы, при обучении на которой тестовая ошибка не хуже, чем на исходной базе. В результате работы алгоритма ошибка в данном примере составляет 6.5%, количество прецедентов равно 72 для обоих классов.

4. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Предложенный алгоритм сравнивался с методом случайного уменьшения большего класса (Random Under Sampling), который дает наилучшие результаты среди существующих методов уменьшения большего класса [5].

Результаты экспериментов показывают, что предложенный алгоритм хорошо справляется с базами большого объема и дает сбалансированные ошибки 1-го и 2-го рода, в то время как метод случайного уменьшения базы теряет важные прецеденты. Помимо этого при очень больших объемах базы даже уменьшения большего класса недостаточно для того, чтобы построить такую базу, чтобы мощностей компьютера хватило для обучения на ней. Поэтому приходится еще раз прореживать базу, а, следовательно, увеличивается вероятность случайного удаления важных прецедентов.

Предложенный метод также сравнивался с методом изменения порога решения, поскольку он лежит в основе предложенного и является самым простым в реализации и очевидным для использования. Предложенный алгоритм в общем случае показывает общую ошибку не хуже чем метод изменения порога. Однако преимущество предложенного метода заключается именно в том, что он строит тренировочную базу много меньше исходной, тогда как в методе изменения порога объем базы не меняется.

Результаты экспериментов на данных из UCI представлены в Таблице 1. В ней же приведены результаты применения алгоритма к данным из области компьютерного зрения (данные LED).

5. ЗАКЛЮЧЕНИЕ

Был предложен итеративный алгоритм выбора прецедентов обучающей базы, устраняющий проблему несбалансированных классов и позволяющий работать на больших базах из области машинного зрения. Был проведен ряд экспериментов по сравнению предложенного алгоритма с распространенными методами выбора прецедентов. Получены результаты, показывающие, что предложенный метод работает стабильно на базах, склонных к классификации, т.е. где ошибка < 40-50%. Также, показано, что алгоритм хорошо справляется с большими объемами информации, что нельзя сказать об остальных методах.

Работа выполнена при поддержке РФФИ 08-01-00883-а

6. СПИСОК ЛИТЕРАТУРЫ

[1] http://archive.ics.uci.edu/ml

[2] P. Viola and M. Jones, ***Fast and robust classification using asymmetric AdaBoost and a detector cascade,*** 2002

[3] Olga Barinova, Anna Kuzmishkina, Alexander Vezhnvets, Vladimir Vezhnevets, ***Learning class specific edges for vanishing point estimation,*** GraphiCon, 2007

[4] Leo Braiman, ***Random Forest,*** 2001

[5] Alexander Yun-chung Liu, B.S., ***The Effect of Oversampling and Undersampling on Classifying Imbalanced Text Datasets,*** 2004

[6] Jason Van Hulse, Taghi M. Khoshgoftaar, Amri Napolitano, ***Experimental Perspectives on Learning from Imbalanced Data,*** ACM International Conference Proceeding Series, Vol. 227

[7] Seyda Ertekin, Jian Huang, Leon Bottou , C. Lee Giles, ***Learning on the Border: Active Learning in Imbalanced Data Classification,*** CIKM, 2007

[8] Miroslav Kubat, Stan Matwin, ***Addressing the Curse of Imbalanced Training Sets: One-Sided Selection,*** In Proceedings of the Fourteenth International Conference on Machine Learning , 1997, pp. 179—186

[9] Nitesh V. Chawla, ***C4.5 and Imbalanced Data sets: Investigating the effect of sampling method, probabilistic estimate, and decision tree structure***, ICML, 2003

[10] Show-Jane Yen, Yue-Shi Lee, Cheng-Han Lin and Jia-Ching Ying, ***Investigating the Effect of Sampling Methods for Imbalanced Data Distributions,*** SMC, 2006, vol. 5, pp. 4163-4168

[11] Bianca Zadrozny, John Langford, ***Cost-Sensitive Learning by Cost-Proportionate Example Weighting***

[12] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, ***SMOTE: Synthetic Minority Over-sampling Technique***

[13] Yoav Freund and Robert E, Schapire***, A decision-theoretic generalization of on-line learning and an application to boosting***

[14] Leo Breiman, ***Bagging Predictors***

[15] B. E. Boser, I. M. Guyon, and V. N. Vapnik, ***A training algorithm for optimal margin classifiers***

[16] Chao-Ton Su and Yu-Hsiang Hsiao, ***An Evaluation of the Robustness of MTS for Imbalanced Data***

Об авторах

Кузьмишкина Анна является студенткой лаборатории Компьютерной графики и мультимедиа Московского Государственного Университета. Ее контактный адрес akuzmishkina@gmail.com .

Баринова Ольга является аспиранткой факультета Вычислительной Математики и Кибернетики Московского Государственного Университета. Ее контактный адрес obarinova@graphics.cs.msu.

Конушин Антон является сотрудником лаборатории Компьютерной графики и мультимедиа Московского Государственного Университета. Его контактный адрес ktosh@graphics.cs.msu .

Данные	*Размер 1-го класса*	*Размер 2-го класса*	*Размер 1-го класса (AS)*	*Размер 2-го класса (AS)*	*Общая ошибка (AS)*	*Ошибка 1-го рода (AS)*	*Ошибка 2-го рода (AS)*	*Общая ошибка (RUS)*	*Ошибка 1-го рода(RUS)*	*Ошибка 2-го рода (RUS)*	*Ошибка с изменением порога решения*
Letter 2vsAll	256	6412	227	261	**4.5%**	4%	4.8%	9%	3.8%	9%	2%
Letter 4vsAll	269	6399	238	273	**2.5%**	4.5%	4%	10.2%	5%	11%	2.5%
Letter 5vsAll	256	6411	227	261	**3%**	5%	4%	9%	4%	9%	5%
Letter 6vsAll	259	6409	228	263	**4%**	2.5%	4%	9%	2.5%	9%	4.5%
Letter 9vsAll	252	6415	220	256	7%	5%	8%	**2%**	5%	1.9%	2.5%
Letter 10vsAll	249	6418	221	257	3.5%	5%	3.5%	**3%**	7.5%	3%	3.5%
Pageblocks 2vsALL	110	1715	102	129	5%	1%	5.5%	7.5%	4.5%	7.5%	**3.5%**
Pageblocks 5vsALL	30	1795	30	46	**6%**	5%	6%	16.5%	~0%	17%	9%
Pendigits 3vsALL	719	6775	632	632	**0.7%**	1.7%	0.7%	1.6%	2.4%	1.5%	1%
Pendigits 4vsALL	780	6714	697	697	**0.5%**	0.5%	0.5%	0.6%	0.5%	0.6%	1.3%
Pendigits 5vsALL	720	6774	645	681	2.5%	2%	2.7%	**1.8%**	4.7%	1.5%	3.3%
Pendigits 6vsALL	720	6774	645	645	**0.4%**	1.5%	0.4%	0.6%	3%	0.4%	0.5%
Pendigits 7vsALL	778	6716	698	753	3%	2%	3%	**1.5%**	8.2%	0.7%	2%
LED	1925	7305	1610	1612	11%	11%	11%	12%	7.5%	14%	**10%**

Таблица 1. *Результаты сравнения предложенного алгоритма (AS) с методом случайного уменьшения большего класса (RUS) на данных из UCI и данных из задач компьютерного зрения. Жирным шрифтом выделены лучшие значения.*