



Math-Net.Ru

All Russian mathematical portal

M. A. Ryndin, D. Y. Turdakov, Domain adaptation by proactive labeling,
Proceedings of ISP RAS, 2019, Volume 31, Issue 5, 145–152

<https://www.mathnet.ru/eng/tisp460>

Use of the all-Russian mathematical portal Math-Net.Ru implies that you have read and agreed to these terms of use

<https://www.mathnet.ru/eng/agreement>

Download details:

IP: 18.97.14.80

May 18, 2025, 06:29:55



DOI: 10.15514/ISPRAS-2019-31(5)-11



Проактивная разметка примеров для адаптации к домену

¹ М. А. Рындин, ORCID: 0000-0002-7504-3975 <mxrynd@ispras.ru>

^{1,2} Д. Ю. Турдаков, ORCID: 0000-0001-8745-0984 <turdakov@ispras.ru>

¹ Институт системного программирования им. В.П. Иванникова РАН, 109004, Россия, г. Москва, ул. А. Солженицына, д. 25

² Московский государственный университет имени М.В. Ломоносова, 119991, Россия, Москва, Ленинские горы, д. 1

Аннотация. В статье приводятся исследование возможности переноса знаний в целевой домен из другого, но близкого домена-источника с помощью проактивного обучения. Исследуется применимость использования модели машинного обучения, обученной на домене-источнике, как бесплатного ненадежного оракула для определения сложности примера из целевого домена и принятии решения о необходимости его разметки надежным экспертом. Представлен алгоритм такой разметки, одной из особенностей этого алгоритма является его возможность работы с любым классификатором, имеющим вероятностную интерпретацию выхода. Экспериментальное тестирование на наборе данных из отзывов на продукты Амазон подтверждает эффективность предложенного метода.

Ключевые слова: адаптация к домену; проактивное обучение

Для цитирования: Рындин М.А., Турдаков Д.Ю. Проактивная разметка примеров для адаптации к домену. Труды ИСП РАН, том 31, вып. 5, 2019 г., стр. 145-152. DOI: 10.15514/ISPRAS-2019-31(5)-11

Domain adaptation by proactive labeling

¹ M.A. Ryndin, ORCID: 0000-0002-7504-3975 <mxrynd@ispras.ru>

^{1,2} D.Y. Turdakov, ORCID: 0000-0001-8745-0984 <turdakov@ispras.ru>

¹ Ivannikov Institute for System Programming of RAS, 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.

² Lomonosov Moscow State University, GSP-1, Leninskie Gory, Moscow, 119991, Russia

Abstract. Getting tagged data is an expensive and time-consuming process. There are several approaches to how to reduce the number of examples needed for training. For example, the methods used in active learning are aimed at choosing only the most difficult examples for marking. Using active learning allows to achieve results similar to supervised learning, using much less labeled data. However, such methods are often dispersive and highly dependent on the choice of the initial approximation, and the optimal strategies for choosing examples for marking up either depend on the type of classifier or are computationally complex. Another approach is domain adaptation. Most of the approaches in this area are unsupervised and are based on approximating the distribution of data in domains by solving the problem of optimal transfer

or extraction of domain-independent features. Supervised learning approaches are not resistant to changes in the distribution of the target variable. This is one of the reasons why the task of semi-supervised domain adaptation is posed: there are labeled data in the source domain, a lot of unlabeled data in the target domain and the ability to get labels for some of the data from the target domain. In this work, we show how proactive labeling can help transfer knowledge from one source domain to a different but relative target domain. We propose to use a machine learning model trained on source domain as a free fallible oracle. This oracle can determine complexity of a training example to make several decisions. First, this example should be added to training dataset. Second, do we have enough knowledge learnt from source to label this example ourselves or we need to call a trusted expert? We present an algorithm that utilizes these ideas and one of its features is ability to work with any classifier that has probabilistic interpretation of its outputs. Experimental evaluation on Amazon review dataset establishes the effectiveness of proposed method.

Keywords: domain adaptation; proactive learning

For citation: Ryndin M.A., Turdakov D.Y. Domain adaptation by proactive labeling. Trudy ISP RAN/Proc. ISP RAS, vol.31, issue 5, 2019, pp. 145-152 (in Russian). DOI: 10.15514/ISPRAS-2019-31(5)-11

1. Введение

Получение размеченных данных – дорогостоящий и трудозатратный процесс. Существует несколько подходов к тому, как снизить количество примеров, необходимых для обучения.

Например, приёмы, используемые в активном обучении, направлены на выбор для разметки только наиболее трудных примеров. Использование активного обучения позволяет добиться результатов, аналогичных обучению с учителем, используя намного меньше размеченных данных. Однако, часто такие методы дисперсионны и сильно зависят от выбора начального приближения, а оптимальные стратегии выбора примеров для разметки либо зависят от вида классификатора [1], либо вычислительно сложны [2]. Еще одной открытой проблемой является активная разметка примеров несколькими слабоквалифицированными аннотаторами – объединение идей активного обучения и краудсорсинга [3].

Другим подходом является адаптация к домену (предметной области). Задача обычно формулируется следующим образом: даны размеченные данные для задачи в одной предметной области, которая похожа на целевую и требуется построить решение для целевой предметной области, используя эти данные. Большая часть подходов в этой области являются алгоритмами обучения без учителя и основаны на сближении распределения данных в доменах с помощью решения задачи оптимального переноса [4] или извлечения доменно-независимых признаков с помощью, например, понижения размерности [5] или состязательного обучения [6].

Стоит отметить, что природа изменений в распределениях источника и целевого домена может варьироваться: может меняться только распределение примеров, может меняться распределение целевой переменной, эти типы изменений могут происходить одновременно. Подходы, использующие обучение без учителя, плохо справляются с

изменениями, затрагивающими распределение целевой переменной. Это является одной из причин, по которой ставится задача адаптации с учителем: имеются размеченные данные в домене-источнике, много неразмеченных данных в целевом домене и возможность получить метки для части данных из целевого домена. Для решения этой задачи естественным выглядит объединение методов активного обучения и адаптации к домену [7]. Одной из проблем приведённых исследований является зависимость алгоритма от типа модели машинного обучения. Авторы исследуют линейные модели, в то время как использование нелинейных моделей обычно позволяет добиться лучших результатов.

Обычно активное обучение предполагает идеализированную модель среды: имеется один всегда доступный и никогда не ошибающийся оракул, всё исследование сфокусировано на выборе примеров, которые дать ему на разметку. Проактивное обучение [8] снимает эти ограничения, исследователи рассматривают проблему выбора примеров для разметки при разных моделях ошибки оракула, при наличии нескольких оракулов с, возможно, разной стоимостью разметки. В работе [9] исследуется идея проактивной разметки примеров несколькими различными оракулами, авторы показывают, как работать с признаками, отсутствующими в домене-источнике. Однако их решение существенно использует линейность модели машинного обучения, что сказывается на итоговом качестве.

Также большинство алгоритмов активного обучения предполагают многочисленное обучение модели на вновь выбранных для разметки данных. Такой подход может привести к большим временным затратам при обучении сложных нелинейных моделей, поэтому для реального использования необходим алгоритм, минимизирующий количество циклов выбора примеров.

В данном исследовании предлагается использовать модель, построенную на домене-источнике, как дополнительный оракул. Предложенный алгоритм способен работать с произвольными (не только линейными) моделями, имеющими вероятностную интерпретацию выходов. Предложенный алгоритм позволяет добиться показателей качества, близких к качеству внутри домена, выбрав на разметку лишь небольшую часть примеров.

В следующем разделе будет приведена общая схема решения. Затем будет зафиксирована модель оракулов и ограничен класс моделей, с которыми способен работать алгоритм. В четвертом разделе приведено полное описание шагов алгоритма. Пятый раздел посвящен результатам экспериментального тестирования алгоритма.

2. Общая схема предлагаемого решения

На рис. 1 представлена общая схема предложенного подхода, который состоит из 3 шагов.

1. Обучение модели на домене-источнике. Полученную модель обозначим M_s .
2. Проактивная разметка с помощью обученной модели M_s и надежного оракула (эксперта или группа экспертов в предметной области) части примеров из целевого домена.
3. Обучение целевой модели на полученном наборе данных.



Рис. 1. Диаграмма предложенного метода
Fig. 1. Block diagram of proposed algorithm

Табл. 1. Обозначения
Table 1. Definitions

| | |
|----------------------------|---|
| X_s | Примеры из домена-источника |
| y_s | Целевая переменная для примеров из домена-источника |
| X_t | Примеры из целевого домена |
| M_s | Модель, построенная на данных из домена-источника |
| M_t | Целевая модель |
| O | Платный оракул |
| $p^j = M_s(x^j)$ | Вероятность принадлежности к классу «1» для данного примера $x^j \in X_t$, предсказанная бесплатным оракулом |
| $\hat{y}^j = 1(p^j > 0.5)$ | Класс для данного примера $x^j \in X_t$, предсказанный бесплатным оракулом |
| $C = 0.5 - p^j \vee$ | Уверенность бесплатного оракула в данном примере $x^j \in X_t$ |
| θ | Гиперпараметр, граница уверенности бесплатного оракула, до которой мы ему верим |

3. Модели оракулов

Используется два оракула: надежный, который можно рассматривать как группу экспертов (назовем его O) и ненадежный, который является моделью машинного обучения, обученной на домене-источнике (назовем его M_s). Опишем их модель.

- O – платный, считаем, что стоимость разметки одного примера фиксирована и не зависит от сложности примера. M_s по своей природе бесплатный.
- O – надежный и не ошибается. M_s может ошибаться.

Для работы с M_s надо ввести модель ошибки, которая зависит от примера. Выход многих моделей машинного обучения имеет вероятностную интерпретацию (к примеру, это верно для логистической регрессии, нейросетей с активацией в виде софтмакс или сигмоиды в последнем слое). Например, выход бинарного классификатора с активацией-сигмоидой – число в диапазоне от 0 до 1, равное вероятности принадлежности классу «1».

В этом исследовании зафиксируем использование моделей с таким выходом. Для них модель ошибки вводится естественным образом. Например, для бинарной классификации обычно считают, что пример принадлежит классу «1», если выход классификатора больше 0.5, иначе классу «0». Соответственно, чем ближе к 0.5 выход классификатора, тем менее он уверен в примере. Следовательно, величину $0.5 - |0.5 - p^j \vee|$, где p^j – предсказанная вероятность, логично использовать как вероятность ошибки.

4. Алгоритм проактивного выбора примеров и разметки

В табл. 1 представлены обозначения, которые будут использованы далее. Алгоритм 1 описывает шаги при проактивной разметке. В начале принимается решение о необходимости добавить данный пример в обучающую выборку. Для этого смотрим на уверенность бесплатного оракула в данном примере. Чем она меньше, тем более непохож данный пример на примеры из источника, значит этот пример характерен для целевого домена и должен попасть в обучающую выборку. Поэтому пример попадает на разметку (бесплатную или платную решится позже), если реализация случайной величины из распределения Бернулли с параметром C оказалась равна 0. Это почти всегда происходит для сложных примеров, а около половины легких – отсеивается.

Этот шаг вводится сразу для нескольких целей.

- Данный подход похож на выбор примеров по уверенности классификатора из активного обучения. Однако здесь используется не уверенность самого классификатора, а её аппроксимация уверенностью классификатора над доменом-источником.
- Такой способ выбора примера не предполагает многочисленного обучения целевой модели, которое происходит в алгоритмах активного обучения.

- Отсевание части похожих примеров является механизмом, позволяющим забывать особенности одного домена и лучше обучаться к особенностям другого.

Input: M_s

Initialization: $\hat{X} = \{\}, \hat{y} = \{\}$

```
1: for all  $x^j \in X_t$  do
2:   считаем  $p^j, C$ 
3:   if  $Bernoulli(C) = 0$  then
4:     считаем  $\tilde{y}^j$ 
5:      $\hat{X} = \hat{X} \cup x^j$ 
6:     if  $C \leq \theta$  then
7:        $\hat{y} = \hat{y} \cup O(x^j)$ 
8:     else
9:        $\hat{y} = \hat{y} \cup \tilde{y}^j$ 
10:    end if
11:  end if
12: end for
```

Алгоритм 1. Предлагаемый алгоритм проактивной разметки

Algorithm 1. Proposed algorithm of proactive labeling

После этого принимается решение о способе разметки – если уверенность в данном примере выше порога, то пример размечается с помощью M_s , иначе отдается O .

Выходом алгоритма является множество $\hat{X} \subset X_t$ и меток к примерам из этого множества \hat{y} . С помощью этих данных строится целевая модель M_t .

5. Эксперименты

5.1 Набор данных

Эксперименты проводились на наборе данных «Amazon review dataset»¹, содержащем отзывы на различные товары на английском языке. Изначальная целевая переменная – оценка товара по пятибалльной шкале, была заменена на бинарную: оценки до 3 включительно – класс «0» (отрицательный отзыв), более 3 – класс «1» (положительный отзыв).

Сравнение производилось со статьей [9], значения достоверности взяты из неё.

5.2 Используемая модель машинного обучения

При построении классификатора использовалась идея переноса знаний [10] с помощью неглубокой рекуррентной нейронной сети (2 скрытых LSTM слоя и выходной слой с активацией-сигмоидой) над предобученной² на Wikipedia языковой моделью fasttext [11].

¹ <http://jmcauley.ucsd.edu/data/amazon/>

² <https://fasttext.cc/docs/en/english-vectors.html>

Выделим источники случайности всего решения:

- Модели M_s и M_t как нейросети зависят от начальной инициализации весов.
- Выбор примеров в обучающую выборку случаен – серия испытаний Бернулли.
- X_t случайным образом перемешивается и разбивается на выборку для обучения (кандидаты на разметку) и теста в соотношении 9 к 1.

При каждом запуске алгоритма начальные значения генератора псевдослучайных чисел для каждого из источников изменялись. Представленные далее результаты являются осреднением по 15 запускам и округлены до первого значимого знака.

Табл. 2. Результаты на «Amazon review»

Table 2. Results on «Amazon review»

| Источник | Целевой домен | Достоверность, предложенный метод | Достоверность, без адаптации | Достоверность, достигаемая внутри домена | Достоверность, [9] |
|----------|---------------|-----------------------------------|------------------------------|--|--------------------|
| B | E | 90.7 ± 0.2 | 88.6 ± 0.1 | 91.2 ± 0.2 | 78.4 |
| | K | 89.9 ± 0.4 | 86.5 ± 0.2 | 91.3 ± 0.2 | 78.6 |
| E | E | 90.1 ± 0.2 | 87.0 ± 0.1 | 90.9 ± 0.1 | 77.8 |
| | K | 90.2 ± 0.3 | 89.1 ± 0.1 | 91.3 ± 0. | 86.0 |
| K | E | 91.1 ± 0.4 | 88.9 ± 0.3 | 91.2 ± 0.2 | 70.1 |
| | B | 89.3 ± 0.2 | 84.8 ± 0.1 | 90.9 ± 0.1 | 73.2 |

5.3 Результаты

В качестве метрики использовалась достоверность (accuracy). Гиперпараметр θ равнялся 0.25.

Обозначения доменов: B–books, E–electronics, K–kitchen. Результаты представлены в табл. 2. В среднем после первичного отбора на разметку попадало около $55 \pm 5\%$ всех данных, при этом среднее число бесплатно размеченных примеров примерно в 4 раза больше числа обращений к оракулу. При этом средняя доля ошибочно размеченных примеров не превышает $5 \pm 2\%$ от числа размеченных примеров.

Из табл. 2 видно, что предложенный алгоритм позволяет добиться результатов, близких к качеству, которого можно добиться, имея все метки для примеров из целевого домена.

6. Заключение

В данной работе представлен алгоритм адаптации к домену с учителем, использующий проактивную разметку. Этот алгоритм способен работать с любыми моделями машинного обучения, имеющими вероятностную интерпретацию выхода. Экспериментально показано, что алгоритм способен строить модели, близкие к качеству внутри домена, выбирая лишь часть примеров для разметки.

Список литературы / References

- [1] Cai Wenbin, Zhang Yexun, Zhang Ya, Zhou Siyuan, Wang Wenquan, Chen Zhuoxiang, Ding Chris. Active Learning for Classification with Maximum Model Change, ACM Transactions on Information

- Systems, vol. 36, issue 2, 2017, pp. 15:1–15:28.
- [2] Ozan Sener, Silvio Savarese. Active Learning for Convolutional Neural Networks: A Core-Set Approach. arXiv:1708.00489, 2017.
- [3] Гилязов Р.А., Турдаков Д.Ю. Активное обучение и краудсорсинг: обзор методов оптимизации разметки данных. Труды ИСП РАН, том 30, вып. 2, 2018 г, стр. 215-250 / Gilyazev R.A., Turdakov D.Y. Active learning and crowdsourcing: a survey of annotation optimization methods. Trudy ISP RAN/Proc. ISP RAS, vol. 30, issue 2, 2018, pp. 215-250 (in Russian). DOI: 10.15514/ISPRAS-2018-30(2)-11.
- [4] Nicolas Courty, Rémi Flamary, Devis Tuia, Alain Rakotomamonjy. Optimal Transport for Domain Adaptation. arXiv:1507.00504, 2015.
- [5] Minmin Chen, Zhixiang Eddie Xu, Kilian Q. Weinberger, Fei Sha. Marginalized Denoising Autoencoders for Domain Adaptation. arXiv:1206.4683, 2012
- [6] Yaroslav Ganin, Victor Lempitsky. Unsupervised Domain Adaptation by Backpropagation. Proceedings of the 32nd International Conference on Machine Learning, 1180–1189, 2015.
- [7] Rai Piyush, Saha Avishek, Hal Daumé III, Venkatasubramanian Suresh. Domain Adaptation Meets Active Learning. Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing, 2010, 27–32.
- [8] Pinar Donmez and Jaime G. Carbonell, From Active to Proactive Learning Methods. Advances in Machine Learning I. Springer, Berlin, Heidelberg, 2010. 97-120.
- [9] Krishnapuram Raghu, Rajkumar Arun, Acharya Adithya, Dhara Nikhil, Goudar Manjunath, Sarashetti Akshay P. Online Domain Adaptation by Exploiting Labeled Features and Pro-active Learning. Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, 2018.
- [10] Howard Jeremy, Ruder Sebastian. Universal Language Model Fine-tuning for Text Classification. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, 328–339.
- [11] Armand Joulin, Edouard Grave, Piotr Bojanowski, Tomas Mikolov. Bag of Tricks for Efficient Text Classification. CoRR, abs/1607.01759, 2016.

Информация об авторах / Information about authors

Денис Юрьевич ТУРДАКОВ, кандидат физико-математических наук, заведующий отделом информационных систем ИСП РАН, доцент кафедры системного программирования МГУ. Научные интересы: обработка естественного языка, машинное обучение, интеллектуальный анализ данных, анализ социальных сетей, распределенная обработка данных.

Denis Yuryevich TURDAKOV, Ph.D. in Physics and Mathematics, Head of the Information Systems Department at ISP RAS, Associate Professor of the System Programming Department of Moscow State University. Research interests: natural language processing, machine learning, data mining, social network analysis, distributed data processing.

Максим Алексеевич РЫНДИН, аспирант ИСП РАН. Научные интересы: методы адаптации к домену и переноса знаний, онлайн обучение, обработка текстов на естественном языке, генеративные модели, активное и проактивное обучение, анализ социальных сетей.

Maxim Alekseevich RYNDIN, PhD student of ISP RAS. Research interests: methods for adapting to the domain and transferring knowledge, online learning, natural language processing, generative models, active and proactive learning, analysis of social networks.