АЛГОРИТЪМ ЗА УЧЕНЕ ОТ МНОГО ИЗТОЧНИЦИ НА ДАННИ

Гергана Лазарова, Милен Чечев, Иван Койчев

Софийски Университет "Св. Климент Охридски"
{gerganal, milen.chechev, koychev}@fmi.uni-sofia.bg

***Резюме**: В съвременния свят все по-често имаме данни за едни и същи обекти, получени от различни източници. Всеки източник на данни изобразява обекта в своето пространство от атрибути (изглед). Процесът на обучение и класификация представлява комбинация от два или повече класификатора, всеки един от които е обучен върху отделен изглед на обектите. В настоящата публикация е представен такъв алгоритъм за смесено индуктивно машинно самообучение, който използва два източника от данни. Предложена е модификация на стандатния алгоритъм за съвместно обучение, която обучава само по-лошия от двата класификатора. Целта на публикацията е да изследва поведението на алгоритъма и сравни неговата класификационна точност с тази на Наивен Бейсов Класификатор.*

***Ключови думи**: смесено индуктивно машинно самообучение, съвместно обучение, класификация*

1. Увод

Първоначално, алгоритъмът за съвместно обучение е използван за клъстеризация на университетски уеб-страници [1]. Учител предварително е класифицирал ръчно малка извадка от примерите. За обучаващо множество са използвани вече класифицираните примери, както и останалите уеб страници, които не са били класифицирани. Използвани са два изгледа. Първият съдържа думите в уеб-страницата, а вторият - хипервръзките, които водят до съответната уеб-страница.

Друг пример за приложение на алгоритъма е разпознаване на хора въз основа на два източника на данни (два изгледа - гласово разпознаване и разпознаване на образи). Дори, така популярните след успеха на Амазон препоръчващи системи могат да се възползват от този алгоритъм, особено в случаите, когато историята от покупки на даден потребител е ограничена или изобщо не съществува. Бихме могли да разделим потребителския профил на два изгледа: $X = (X_1, X_2)$, където X1 = "артикули, за които потребителят е задал рейтинг", X2 = "коментари на потребителя за даден продукт".

Едно по-ново и перспективно направление е учене на емоционалното състояние на човек въз основа на неговите мимики, тембър на гласа, жестикулация и поведение, които се явяват отделни източници на информация за обучение.

След 1998 година, когато Blum и Mitchell [1] публикуват базовия алгоритъм за съвместно обучение, той придоби голяма популярност. Те описват и доказват, че алгоритмите за съвместно обучение се представят по-добре от единичен алгоритъм за обучение, когато съществува естествено разбиване на атрибутите в повече от един изглед. Следните критерии трябва да бъдат изпълнени:

(1) Всеки изглед (множество от атрибути) е достатъчно за самостоятелна класификация

(2) Двата изгледа са условно независими спрямо класа

$$P(X_1|Y, X_2) = P(X_1| Y), P(X_2|Y, X_1) = P(X_2| Y)$$

На практика, почти невъзможно е да срещнем подобен модел от напълно независими изгледи спрямо класа. Не е възможно винаги да намерим таково естествено разделение на атрибутите в отделни изгледи и то зависи от самите данни. Въпреки това се оказва, че алгоритмите за съвместно обучение се представят много добре и подобряват класификационната точност, дори когато условията описани по-горе не са изпълнени.

Съвместното обучение се оказа изключително полезно не само за класифициране на уеб-страници, но и за разпознаване на сцени [9]. Jafar Tanha, Maarten van Someren and Hamideh Afsarmanesh [2] публикуват ансамблово учене въз основа на алгоритъма. Те използват този подход за избор кои примери да бъдат добавени, без да бъде намалена класификационната точност на текущия класификатор. Нашият подход се различава по това, че той подобрява само един от двата класификатора, който е по-слаб. Nigam Kamal and Rayid Ghani [3] анализират ефективността на базовото съвместно обучение и го сравняват с Наивен Бейсов Класификатор и ко-ЕМ алгоритъм. В текущата публикация представеният алгоритъм е сравнен с Наивен Бейсов Класификатор.

2. Индуктивно машинно самообучение с учител - Наивен Бейсов Класификатор

Наивният Бейсов Класификатор (НБК) представлява алгоритъм за учене с учител, който прави предположението, че атрибутите са условно независими помежду си при даден клас.

$$P(y_j)P(x_i \mid y_j) = P(y_j)\prod_{k=1}^{m} P(a_k \mid y_j)$$

3. Смесено индуктивно машинно самообучение

Смесеното индуктивно машинно самообучение представлява комбиниран модел за учене от два вида данни – примери, които са предварително

класифицирани от учител и примери, които нямат известни класификации. В тази област търсенето е към алгоритми, които да постигат добра класификационна точност, имайки малка извадка от класифицирани примери.

Определение 1: Некласифициран пример - без предварително зададена класификация: D-измерен вектор $X = (X_1,\ldots,X_d)$. Няма учител, който да дефинира стойността на класификационната функция. Разполагаме само със стойностите на атрибутите на примера.

Определение 2: Класифициран от учител пример: (*D+1*)-измерен вектор $X=(X_1,\ldots,X_d,Y)$, където Y е стойността на търсената класификационна функция.

3.1. Съвместно индуктивно обучение - базов алгоритъм

Алгоритъмът за съвместно индуктивно обучение решава класификационна задача, в която примерите могат да принадлежат към един от два или повече класа. Обучава два класификатора - L1 и L2, всеки един от които ползва отделно множество от атрибути (изглед). Тези класификатори могат да бъдат и от различен тип (Наивен Бейсов класификатор, невронни мрежи и др.).

Нека D е множеството от всички примери. D1 съдържа само примерите с предварително известни класификации, а D2 съдържа примерите, които не са класифицирани.

$$D_1 = \{(x_i, y_i)\}_{i=1}^{nl} \ , \ D_2 = \{x_j\}_{j=1}^{nu}$$

Обикновено, броят на примерите, които не са класифицирани *u* е много по-голям от тези които са класифицирани *l (u >> l).*

Нека всеки пример X се състои от 2 изгледа, така че X = (X1, X2). X1 и X2 представляват множества от атрибути, характеристики на обекта от двата източника на данни и нека означим класифицираните примери, които отговарят съответно на тези изгледи с U1 и U2.

Алгоритъм за съвместно обучение:

Повтори к на брой пъти {

1. Научи класификатор L1, ползвайки U1.
   Научи класификатор L2, ползвайки U2.
2. Намери класификациите на примерите от D_2:
   - Използвай класификатор L1 върху изглед X1, за класификация на всички примери $(x_{i1}, x_{i2}) \in D_2$
   - Използвай класификатор L2 върху изглед X2, за класификация на всички примери $(x_{i1}, x_{i2}) \in D_2$

3. Добави най-убедителните примери на L1 към U2 и най-убедителните примери на L2 към U1. Премахни тези примери от D_2

}

Броят стъпки *к* зависи от броя примери, които се добавят на стъпка 3 като класифицирани и може да бъде най-много броя на некласифицираните примери.

3.2. Учене от два източника на данни, при което единият класификатор е много по-добър от другия

В тази секция ще представим нов обучаващ алгоритъм за смесено индуктивно машинно самообучение, който използва два изгледа - два източника на данни (Multi-View Semi-supervised Machine Learning Algorithm - MVSSMLA). Представлява модификация на базовия алгоритъм за съвместно индуктивно обучение, при която единия класификатор е по-слаб и не е достатъчен за самостоятелна класификация. Обикновено, когато разполагаме с малко налични класифицирани примери, всеки допълнителен източник на данни може да е от ползва за намаляване на грешката върху нови примери. Тези допълнителни източници могат да имат зашумени данни, както и непълни или липсващи данни, което води до тяхното по-лошо представяне.

Едно от условията за оригиналния алгоритъм е всеки изглед (множество от атрибути) да е достатъчно за самостоятелна класификация. Ако един от тези два класификатора е ненадежден, той би влошил и поведението на другия. Нашият подход се явява решение на този проблем. При примера с класификацията на уеб-страниците, водещ изглед би бил този, който е изграден от думите в самите уеб страници.

Върху всеки един от двата източника се обучава един и същ тип класификатори (L1 и L2), базирани на Наивен Бейсов Класификатор. Нека означим с θ_1 и θ_2 параметрите на тези класификатори. За Наивен Бейсов Класификатор, при който атрибутите са нормално разпределени параметърът $\theta = (\mu, \sigma, \pi)$ представлява вектор от средното, стандартното отклонение на атрибутите и априорната вероятност за класа. За краен резултат се взима комбинация от двата модела.

Нека L2 е по-лошият от двата класификатора, с по-лоша класификационна точност. Нека U2 е множеството от класифицирани примери за класификатор L2. В началото U2 включва всички примери в обучаващото множество с класификации. Нека W съдържа всички останали, некласифицирани примери. В описания по-долу алгоритъм на стъпка 3 се избират кои некласифицирани примери да се добавят към U2. Добавят се примери, които са нямали класификации, но L1 е класифицирал с най-голяма сигурност към един от възможните класове. Важен момент в алгоритъма е, че той се опитва да

намери най-достоверните примери на L1, но не всички от тях се добавят към класифицираните примери. Ако функцията за пригодност на някои примери не надхвърли някаква граница, то те не се добавят и използват като класифицирани.

Алгоритъм:

1. **Инициализация:**
   - $U_1=view_1(D_1)$ – съдържа само атрибутите за изглед 1 (само класифицирани примери);
   - $U_2=view_2(D_1)$ – съдържа само атрибутите за изглед 2 (само класифицирани примери);
   - $W = D_2$: некласифицираните примери в обучаващото множество;
   - K - Брой класове;
   - S – Параметър на паралелизацията. W се разделя на S на брой множества за паралелна обработка на некласифицираните примери
   - Array [K] – съдържа най-убедителните примери $x_i \in W$ за всеки клас
   - Prob [S][K] – най-доброто $P(y_i \mid x_i, \theta_1)$ за всеки клас и множество S
2. **Паралелно Учене:** научи L1 и L2: $L_1(U_1)$ и $L_2(U_2)$, намери θ_1 и θ_2
3. **ПАРАЛЕЛНО ТЪРСЕНЕ:** Раздели W на S множества и едновременно, паралено за всяко едно множество Ws намери за примерите в него:
   - Намери $P(y_i \mid x_i, \theta_1)$ за всеки клас y_i
   - $y^* = \text{argmax}_{y_i} P(y_i \mid x_i, \theta_1)$
   - $p^* = P(y^* \mid x_i, \theta_1)$
   - if(Prob[s][y*] < p*) актуализирай Arrays[y*] с новата стойност xi
4. **КОМБИНИРАНЕ НА РЕЗУЛТАТИ:** За всеки клас y_i сортирай всяко множество от примери Ws спрямо Prob[S][y_i] и слей сортираните Ws, така, че да се запази наредбата. За всеки клас y_i добави най-убедителните примери ($Array_s[y_i],y_i$), ако надхвърлят предварително дефинирана граница (threashold). Изтрий тези примери от W и ги добави в U2.
5. **Край**, Върни θ_1 и θ_2 и класифицирай нови примери въз основа на умножението на $P(y_i \mid x_i, \theta_1)$ и $P(y_i \mid x_i, \theta_2)$.

Представена е и стандартна оптимизационна техника за паралелна обработка на некласифицираните примери. Тя е особено важна в случаите, когато разполагаме с огромно количество некласифицирани примери, тъй като

успява значително да намали времето за избор кои примери да се добавят като класифицирани.

4. Експерименти и резултати

4.1. Постановка на експеримента

Всички тестове са направени по следната схема: Данните се разделят на две множества – обучаващо и тестово. Използвана е Монте Карло кросвалидация [11] за оценка на класификационната точност на алгоритмите.

1. *Обучаващо множество* - съдържа част от оригиналното множество с класифицирани примери. Премахнати са стойностите на класификационната функция на останалите примери. Тези некласифицирани примери са включени към обучаващото множество.

2. За *тестовото множество* използваме всички примери без тези, които сме използвали на стъпка едно като класифицирани.

4.2. Данни

За оценка на алгоритъма са използвани тестови данни от UCI Machine Learning Repository [8]. Всички данни имат реални атрибути, нямат липсващи стойности. Diabetes съдържа 768 примера, разпределени между 2 класа. Вероятността на правилна случайна класификация е 0,5 (2 класа, 50% точност). Примерите в Iris са разделени в 3 класа, всеки един от класовете съдържа по 50 примера. Всеки клас представлява тип растение от вида ирис. Red Wine Quality [7] моделира предпочитанията за вина на база техни свойства. Съдържа 1600 примера, 11 атрибута и 6 класа. Yest предсказва местоположението на протеини в клетките на база 8 атрибута. 1484 примера са разпределение в 10 класа.

4.3. Резултати

Таблица 1: Информация за тестваните данни.

Данни	Класове	Атрибути	Брой примери	Случайна класификация
Iris	3	4	150	33.33%
Diabetes	2	8	768	50.00%
Red Wine Quality (2009)	6	11	1600	16.67%
Yest	10	8	1484	10.00%

Таблица 2 съдържа информация за всички тествани данни, които бяха изследвани (Iris, Diabetes, Red Wine Quality (2009)). Класификационната

точност на двата алгоритъма е сравнена за извадки от обучаващи множества, съдържащи различен процент от класифицирани примери (5%, 10%, 15%, 20%, 50%, 90%). От таблица 2 може да се види, че MVSSMLA се представя сравнително по-добре от алгоритъма за учене с учител (НБК). Дори, когато условието за независимост на двата изгледа не е изпълнено изцяло (напълно независими изгледи се срещат на практика рядко) алгоритъмът се представя значително добре (2.49% подобрение при iris за само 5% класифицирани примери, при Red Wine Quality стига и до 5.07%, за yest – 3.56%). Това подобрение се основа на комбинирането на двата класификатора и предположението за тяхното съгласие относно примери от разпределението.

Таблица 2: Сравнение на класификационната точност на MVSSMLA и НБК

Данни	Алгоритъм	5%	10%	15%	20%	50%	90%
Iris	НБК	80.07	89.81	93.37	94.03	95.08	94.82
Iris	MVSSMLA	**82.56**	**90.77**	**93.53**	**94.10**	**95.08**	**94.82**
Red Wine Quality	НБК	32.67	36.31	39.32	41.97	47.09	49.39
Red Wine Quality	MVSSMLA	**32.40**	**36.63**	**40.33**	**43.34**	**52.16**	**52.16**
Diabetes	НБК	69.74	71.97	72.96	73.53	75.55	80.04
Diabetes	MVSSMLA	**70.00**	**72.19**	**73.16**	**73.71**	**75.70**	**82.76**
yest	НБК	35.53	38.59	44.44	46.70	48.94	50.30
yest	MVSSMLA	**36.16**	**42.15**	**45.55**	**47.78**	**48.98**	**50.68**

Използваният параметър на стъпка 4 за iris е threashold = 0.0 за Red Wine Quality: threashold = 0.0, за Diabetes: threashold = -14 (отрицателна заради логаритмичната вероятност), а за yest: threashold = 3. Различни параметри бяха изследвани за ОА. Когато границата е твърде голяма, малко примери успяват да я преминат и да окажат влияние на модела за учене. Когато параметърът е твърде голям, прекалено много некласифицирани примери оказват влияние на алгоритъма и повлияват представянето на по-слабия алгоритъм. Стойността на този параметър се оказа от голямо значения за процеса на обучение.

Може да се забележи, че при Red Wine Quality, когато използваме малко класифицирани примери – 5% и когато и двата класификатора не са достатъчно убедителни и добри, подобрението на класификационната точност не е гарантирано. Затова условието на стандартното съвместно обучение се запазва и тук – да може да научим достатъчно добри класификатори върху двата изгледа.

Друг интересен момент в изследваните данни е преспециализацията към обучаващото множество. Може да се забележи класификационната точност на

двата алгоритъма върху iris за 50% и 90% класифицирани примери. Вижда се, че с увеличаване на примерите, използвани за обучение, класификационната точност и на двата алгоритъма намалява. Съвмостното обучение също не успява да се справи с този проблем и за неговото решение ще трябва да се подходи с промяна на модела на използваните класификатори (добавяне на регуляризация, дървете с подкастряне и др.).

Заключение

Когато наличните класифицирани примери са малко, всяко подобряване на научения модел е от значение. Резултатите показаха, че можем да се възползваме от априорни знания за поведението на класификаторите върху двата изгледа. Обучаването на само по-лошия от двата класификатора показа значителни подобрения на класификационната точност на алгоритъма спрямо тази на Наивния Бейсов Модел. Подобряването в точността на алгоритмите е изключително важно в модерния свят, където търсенето е към все по-добри и по-добре предсказващи алгоритми.

Благодарности

Работата е частично финансирана от Европейския социален фонд чрез Оперативна програма „Развитие на човешките ресурси”, договор № BG051PO001-3.3.06 - 0052 (2012-2014).

Литература

1. Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In Proceedings of the eleventh annual conference on Computational learning theory, COLT' 98, pages 92–100, New York, NY, USA, 1998. ACM.
2. J. Tanha, M. van Someren & H. Afsarmanesh. Ensemble based co-training. In P. De Causmaecker, J. Maervoet, T. Messelis, K. Verbeeck & T. Vermeulen (Eds.), Proceedings of the 23rd Benelux Conference on Artificial Intelligence: 3-4 November 2011, Ghent, Belgium Vol. 23. BNAIC (pp. 223-231)
3. Nigam, Kamal; Rayid Ghani. "Analyzing the Effectiveness and Applicability of Co-training". Proceedings of the ninth international Conference on Information and Knowledge Management (NY, USA: ACM): 86–93. CiteSeerX: 10.1.1.37.4669
4. Xiaojin Zhu and Andrew B. Goldberg. Introduction to Semi-Supervised Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2009.
5. Olivier Chapelle, Bernhard Scholkopf, Alexander Zien "Semi-supervised Learning", 2006, MIT Press
6. Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In The 20th International Conference on Machine Learning (ICML), 2003.

7. P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009
8. Bache, K. & Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science
9. Xian-Hua Han, Yen-Wei Chen, Xiang Ruan: Multi-class Co-training Learning for Object and Scene Recognition. MVA 2011: 67-70
10. Anoop Sarkar "Applying Co-Training Methods to Statistical Parsing" In Proceedings of the 2nd Meeting of the North American Association for Computational Linguistics: NAACL 2001. pp. 175-182. Pittsburgh, PA, June 2-7, 2001.
11. http://en.wikipedia.org/wiki/Cross-validation_%28statistics%29

A MULTI-VIEW LEARNING ALGORITHM

Gergana Lazarova, Milen Chechev, Ivan Koychev

Sofia University "Kliment Ohridski", Bulgaria
{gerganal, milen.chechev, koychev}@fmi.uni-sofia.bg

Abstract: *Recently, there has been significant interest in multi-dimensional representation of the objects, searching for new features in new sources of information. Each view of the object represents a separate data source for learning a separate classifier. A multi-view teaching algorithm is compared to a single learner. Both models use Naïve Bayes Classifier for the underlying classifiers. The multi-view algorithm is especially applicable in areas where it is difficult to obtain the classifications of the examples.*

Keywords: *Semi-supervised learning, Co-Training, Classification*