MeLIF+: МНОГОПОТОЧНЫЙ ФИЛЬТРУЮЩИЙ АЛГОРИТМ ОТБОРА ПРИЗНАКОВ

Исаев И. П., студент 4 курса кафедры КТ Университета ИТМО,
isaev@rain.ifmo.ru
Сметанников И. Б., аспирант кафедры КТ Университета ИТМО,
smeivan@mail.ru

Аннотация

Поиск ансамбля нескольких алгоритмов, то есть наилучшей их комбинации — подход, широко используемый в машинном обучении. Алгоритм MeLiF использует эту технику для алгоритмов фильтрующего отбора признаков. В этой статье мы предложим многопоточную реализацию этого алгоритма, и покажем, что это не только значительно ускорит производительность, но и позволит улучшить качество отбора признаков.

Введение

В современном мире машинное обучение является одной из наиболее перспективных и исследуемых научных и технологических областей, во многом, благодаря тому, что его можно применить для решения задач из практически любой предметной области, порождающей данные. Одним из примеров таких областей является биоинформатика [1,2], в которой порождаются огромные объемы данных об экспрессии генов различных организмов, которые позволяют определить, какие участки ДНК отвечают за некоторый видимый признак, или какие гены реагируют на определенные изменения внешней среды. Основная проблема таких данных — большое количество атрибутов и сравнительно малое количество образцов. Из-за слишком большой размерности пространства крайне сложно построить модель, которая хорошо обобщает данные,. Мешает также и то, что большинство признаков не имеют никакого отношения к предсказываемому результату, а следовательно, являются шумом.

Разумное решение в таком случае — каким-то образом отобрать наиболее релевантные признаки и обучать классификатор только на них, а остальные отбросить. Эта задача решается в рамках раздела машинного обучения, называемого отбор признаков (*feature selection*). Существуют три основных метода отбора признаков: фильтры (*filter*) [3] — отбор на

основе статистических свойств каждого конкретного признака, обертки (*wrapper*) [4] — в процессе которых происходит поиск зависимостей между признаками и встроенные методы (*embedded*) [5], являющиеся комбинацией двух предыдущих.

Особенность фильтров в том, что они являются наиболее быстрыми из вышеперечисленных методов отбора, поэтому их часто используют для предобработки, чтобы передать получившееся подмножество атрибутов другому wrapper или embedded — методу. Это особенно важно для решения задач биоинформатики, так как количество признаков в наборах данных из этой области составляет десятки и сотни тысяч.

В последнее время многие алгоритмы машинного обучения так или иначе используют ансамбли[6]. Алгоритм MeLiF [7] пытается применить этот подход к фильтрации признаков — он строит линейную комбинацию фильтров, которая отбирает наиболее релевантные признаки. Структура алгоритма такова, что его достаточно легко распараллелить. В данной работе описывается многопоточная версия алгоритма, которая позволила значительно сократить время работы, не ухудшив качество отбора признаков.

Структура статьи построена следующим образом: раздел 2 более подробно описывает однопоточный вариант MeLiF, раздел 3 — предложенную схему распараллеливания, раздел 4 — организацию экспериментов, раздел 5 — полученные результаты.

MeLiF

Псевдокод алгоритма выглядит следующим образом:

```
MeLiF(points, delta, evaluate)
q* = 0
p*
for p: points
q = evaluate(p)
if (q > q*)
p* = p
q* = q
smthChanged = true
while smthChanged
for dim: p.size
p+ = p{p[dim] + delta}
q+ = evaluate(p+)
if(q+ > q*)
q* = q+
p* = p
```

```
smthChanged = true
break
p- = p{p[dim] − delta}
q- = evaluate(p-)
if(q- > q*)
q* = q-
p* = p-
smthChanged = true
break
return (p*, q*)
```

Стартовыми точками алгоритма являются некоторые линейные комбинации базовых фильтров. Опытным путем было установлено, что наилучший результат достигается при следующем выборе стартовых точек: $(1,0,\dots,0),(0,1,\dots,0),\dots,(0,0,\dots,1)$ – играет роль только один из базовых фильтров и $(1,1,\dots,1)$ – все базовые фильтры равноправны. Алгоритм проходит по всем переданным стартовым точкам и для каждой точки пытается сместить значение каждой из координат на $+\delta$ и $-\delta$. Если хотя бы одно из таких значений лучше максимального найденного — алгоритм выбирает его и начинает поиск с первой координаты точки получившейся точки. Если улучшить результат не удалось – алгоритм останавливается.

Для каждой выбранной точки алгоритм измеряет значение заданной линейной комбинации фильтров для каждого признака в наборе данных. Полученные результаты сортируются и отбираются N признаков с наилучшим показателем. Далее, производится запуск некоторого классификатора только на этом подмножестве признаков и полученный результат сохраняется для сравнения с последующими точками.

MeLiF+

На данном этапе разработки алгоритма предлагается следующее: мы можем запускать алгоритм параллельно из каждой стартовой точки с поддержкой глобального максимума через точку синхронизации потоков. Кроме того, предлагается запускать функцию $evaluate$ для точек $+\delta$ и $-\delta$ одновременно, далее выбирая точку с наилучшим результатом. Мы покажем, что это не только ускоряет работу алгоритма, но и зачастую улучшает полученные результаты. Последнее может быть объяснено следующим образом: так как алгоритм является жадным, т.е. принимает локально-оптимальные решения, то, позволив ему просматривать одновременно две точки, мы выбираем лучшее из локально-оптимальных решений. Это может привести алгоритм к другому локальному оптимуму в худшем случае (такие случаи показаны в эксперименте), но на практике —

в среднем, улучшает полученный результат.

Схема эксперимента

В качестве классификатора использовался SVM из библиотеки WEKA[1], с полиномиальным ядром и 1. Для улучшения устойчивости работы алгоритма использовался метод скользящего окна по пяти стратам (5-fold cross-validation). Количество отбираемых признаков $N = 100$.

Эксперимент производился на сервере со следующими характеристиками: 32-ядерный процессор AMD Opteron 6272 @ 2.1 GHz, 128 GB RAM. Распараллеливание производилось в $N = 50$ потоков, где

$N = 2pf$, где p — количество стартовых точек, f — количество страт.

В качестве базовых метрик алгоритма были выбраны следующие: Spearman Rank Correlation, Symmetric Uncertainty, Fit Criterion, VDM [8]. Алгоритм запускался в двух режимах — параллельном и последовательном, и производилось измерение времени работы и наилучшего качества классификации.

Для эксперимента были взяты 36 наборов данных различных размеров из архивов GEO[2], Broad institute. Cancer Program Data Sets[3], Kent Ridge Bio-Medical Dataset[4], Feature Selection Datasets at Arizona State University[5], RSCTC'2010 Discovery Challenge[6]

Результаты

В Таблице 1 представлены результаты эксперимента. Колонки T и F_1 обозначают время и найденное значение F_1-меры для последовательного алгоритма, $T+$ и F_1+ — соответственно, для параллельного. Строки отсортированы по возрастанию времени работы параллельного алгоритма.

Dataset	**Shape**	T	F_1	$T+$	F_1+
CNS	7129 x 60	30	0,769	5	0,769

[1] Weka 3: Data Mining Software in Java. http://www.cs.waikato.ac.nz/ml/weka/
[2] http://www.ncbi.nlm.nih.gov/geo
[3] http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi
[4] http://datam.i2r.a-star.edu.sg/datasets/krbd
[5] http://featureselection.asu.edu/datasets.php
[6] http://tunedit.org/repo/RSCTC/2010/B/public

GDS2960	4132 x 101	30	0,816	10	0,806
GDS2901	15923 x 84	81	1,0	**10**	1,0
GDS3145	22690 x 64	87	1,0	12	1,0
GDS2961	8448 x 67	51	0,887	13	0,887
DLBCL	7129 x 135	60	0,800	15	0,800
GDS2962	8448 x 67	47	0,741	15	0,741
GDS4261	22769 x 104	128	1,0	**17**	1,0
Prostate	12600 x 136	90	0,774	18	**0,779**
GDS3553	26496 x 96	139	1,0	18	1,0
GDS3116	22283 x 116	140	0,723	20	0,723
GDS3929	24526 x 183	284	0,754	**29**	0,754
GDS2947	54675 x 64	212	1,0	29	1,0
GDS3622	45101 x 110	260	1,0	34	1,0
Breast	24188 x 97	145	0,853	35	0,853
GDS3995	35557 x 90	179	1,0	36	1,0
GDS2819	54675 x 99	306	1,0	**37**	1,0
GDS4109	22283 x 79	160	0,955	38	0,955
GDS2819	54675 x 99	293	1,0	38	1,0
GDS3257	22283 x 107	167	0,991	39	0,991
GDS4130	54675 x 104	307	1,0	40	1,0

GDS4129	54675 x 120	347	1,0	45	1,0
GDS4336	28869 x 90	179	0,934	45	0,934
arizona5	19993 x 187	211	0,788	58	**0,794**
GDS4837	54675 x 88	313	0,982	61	0,982
GDS3244	61170 x 160	489	1,0	62	1,0
GDS2771	22215 x 192	267	0,850	65	0,850
data6	59004 x 92	462	0,7002	74	0,7003
GDS4222	54675 x 130	427	0,972	81	0,972
GDS4103	54675 x 78	301	0,971	106	0,971
GDS2819	54675 x 99	411	0,812	106	**0,822**
GDS4318	33252 x 108	246	0,945	108	0,921
GDS4600	54675 x 170	527	0,970	110	**0,977**
GDS4431	54613 x 146	529	0,672	131	0,672
data4	54675 x 113	453	0,810	143	**0,855**

Таблица 1: Результаты эксперимента

Параллелизм ускорил работу алгоритма в среднем в 5,5 раз, при этом, незначительно улучшив результат в среднем (0,900 против 0,899).

Заключение

Предложенная схема распараллеливания алгоритма позволила ускорить его работу в среднем, в 5,5 раз, не ухудшив выбор точек. К сожалению, в текущей схеме невозможно получить линейный прирост скорости, поскольку в каждый момент обрабатывается фиксированное число точек. В дальнейшем я планирую разработать метод, который будет использовать компромисс между обходом всего линейного пространства и обработкой наиболее значимых точек, что позволит добиться линейного

прироста производительности за счет произвольного количества одновременно обрабатываемых точек.

Литература

1. Bolón-Canedo V., Sánchez-Maroño N., Alonso-Betanzos A., Benítez J.M., Herrera F. A review of microarray datasets and applied feature selection methods // Information Sciences. 2014. Vol. 282. P. 111–135.
2. Saeys Y., Inza I., Larranaga P. A review of feature selection techniques in bioinformatics // Bioinformatics. 2007. Vol. 23, no. 19. P. 2507–2517.
3. Sánchez-Maroño N., Alonso-Betanzos A., Tombilla-Sanromán M. Filter Methods for Feature Selection — A Comparative Study // Intelligent Data Engineering and Automated Learning (IDEAL 2007). Lecture Notes in Computer Science. Volume 4881. P. 178–187.
4. Kohavi R., John G.H. Wrappers for feature subset selection // Artificial Intelligence. 1997. Vol. 97, no. 1–2. P. 273–324.
5. Lal, T.N., Chapelle O., Weston J., Elisseeff A. Embedded methods // Feature extraction, pp. 137–165. Springer Berlin Heidelberg, 2006.
6. Bolón-Canedo V., Sánchez-Maroño N., Alonso-Betanzos A. An ensemble of filters and classifiers for microarray data classification // Pattern Recognition. 2012. Vol. 45 (1) P. 531–539.
7. Smetannikov I., Filchenkov A. MeLiF: Filter Ensemble Learning Algorithm for Gene Selection. // Advanced Science Letters. American Scientific Publisher. — 2016 (in press).
8. Auffarth B., Lopez M., Cerquide J. Comparison of Redundancy and Relevance Measures for Feature Selection in Tissue Classification of CT images // Advances in Data Mining. Applications and Theoretical Aspects. Lecture Notes in Computer Science. Vol. 6171. P. 248–262.