Муравьев Антон Сергеевич, магистрант кафедры вычислительной техники Института кибернетики ТПУ.
E-mail: an.muravyov@gmail.com
Область научных интересов: машинное обучение, нейронные сети, эволюционные вычисления.

Белоусов Артем Анатольевич, канд. техн. наук, доцент кафедры вычислительной техники Института кибернетики ТПУ.
E-mail: artem.a.belousov@gmail.com
Область научных интересов: нейроэволюционные вычисления, методы улучшения изображений и видеопоследовательностей.

УДК 004.421.5

МОДИФИЦИРОВАННЫЙ АЛГОРИТМ РАСТУЩЕГО НЕЙРОННОГО ГАЗА ПРИМЕНИТЕЛЬНО К ЗАДАЧЕ КЛАССИФИКАЦИИ

А.С. Муравьев, А.А. Белоусов

Томский политехнический университет

Предлагается модель классификатора, основанного на алгоритме растущего нейронного газа. Предложена учитывающая специфику решаемой задачи модификация исходного алгоритма, изменяющая его механизм роста с целью ускорения сходимости. Рассмотрены два подхода к синтезу классификатора: апостериорный и частотный. С использованием наборов данных из репозитория машинного обучения UCI производится сравнение эффективности данных подходов как между собой, так и с другими распространенными методами классификации. Показано, что предложенный алгоритм в ряде случаев превосходит другие алгоритмы аналогичного назначения.

Ключевые слова:
Классификация, кластерный анализ, обучение без учителя, обучение с учителем, растущий нейронный газ, самоорганизующиеся модели.

Введение и постановка задачи

Классификация – одна из задач, относящихся к области машинного обучения и (в более широком смысле) искусственного интеллекта. Целью классификации является определение принадлежности незнакомого объекта (переменной, наблюдения и т. д.) к одной из заранее известных категорий – классов. Данное решение принимается на основе некоторой *обучающей выборки* (*множества*), представляющей собой некий набор примеров, для которых верная принадлежность одному из классов заранее известна. Методы подобного рода, позволяющие воспроизвести корректный отклик системы на сигнал, используя для этого «эталонное» множество, в машинном обучении носят название *методов обучения с учителем*.

Другая область машинного обучения – *обучение без учителя* – не подразумевает наличия обучающего множества. К ней относятся кластерный анализ, векторное квантование и другие задачи, объединенные целью поиска структуры и закономерностей в данных без какой бы то ни было дополнительной информации о них. Именно в связи с этой особенностью в последние годы растет интерес к сочетанию двух названных выше областей, в частности к внедрению элементов обучения без учителя в существующие классификационные алгоритмы [1]. Наиболее распространенные применения алгоритмов обучения без учителя в данном контексте – выделение значащих признаков данных, а также распараллеливание исходной задачи путем разбиения анализируемого пространства на подобласти, в каждой из которых можно настроить отдельный специализированный классификатор.

Одним из наиболее значимых алгоритмов обучения без учителя является «нейронный газ» (neural gas) [2]. Он позволяет распределить фиксированное количество узлов (нейронов) в пространстве данных, описав тем самым его топологию (рис. 1).

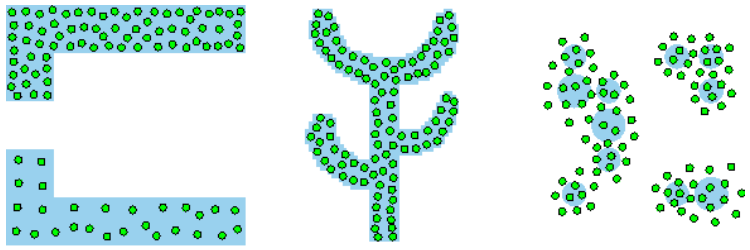

Рис. 1. Пример работы алгоритма нейронного газа

Ограниченность подхода с фиксированным количеством узлов привела к появлению алгоритма растущего нейронного газа (growing neural gas, GNG) [3], в котором применено конкурентное обучение по Хеббу, а также внедрен механизм добавления новых нейронов и удаления невостребованных. Еще одно важное отличие растущего нейронного газа в том, что результатом его работы является граф, ребра которого несут дополнительную топологическую информацию (рис. 2).

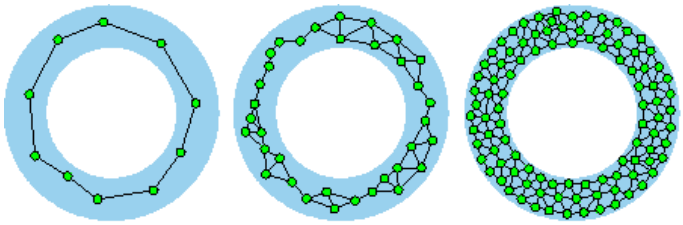

Рис. 2. Воспроизведение топологии данных с помощью растущего нейронного газа

Естественным кажется предположение, что получаемая информация о структуре данных может оказаться очень ценной при решении классификационных задач. Данная работа представляет собой исследование возможности применения растущего нейронного газа в качестве основы для эффективного и высокопроизводительного классификатора.

Растущий нейронный газ и ускорение сходимости

В авторском варианте [3] алгоритм растущего нейронного газа состоит из следующих шагов:

1. Задаются два начальных узла (нейрона) a и b в случайных точках w_a и w_b пространства входных данных $\mathbb{R}^n$. a и b соединяются ребром, возраст которого равен нулю. Ошибка в узлах a и b принимается равной нулю.
2. Выбирается очередной набор данных (сигнал) ξ из входного множества.
3. Определяются ближайший и второй по близости к ξ узел (обозначим их s_1 и s_2 соответственно). Обычно в качестве меры близости используется стандартное евклидово расстояние $\|w(s)-\xi\|$.
4. Возраст всех инцидентных s_1 ребер увеличивается на единицу.
5. Счетчик ошибки нейрона s_1 увеличивается на величину квадрата расстояния до ξ: $\Delta E(s_1)=\|w_{s_1}-\xi\|^2$.
6. s_1 и его топологические соседи (узлы, соединенные с ним ребром) смещаются по направлению к ξ на расстояния $\Delta w_{s1}=\varepsilon_b(\xi-w_{s1})$ и $\Delta w_n=\varepsilon_n(\xi-w_n)$ соответственно, где $0<\varepsilon_b\ll 1$ и $0<\varepsilon_n\ll\varepsilon_b$.

7. Если s_1 и s_2 соединены ребром, его возраст обнуляется; в противном случае между s_1 и s_2 создается новое ребро с возрастом, равным нулю.

8. Все ребра в графе с возрастом более $a_{\max}$ удаляются. В случае если после этого некоторые узлы не имеют инцидентных ребер (оказываются изолированы), эти узлы также удаляются.

9. Если номер текущей итерации кратен величине λ (один из параметров алгоритма), осуществляется вставка нового узла в точке $w_r = 0{,}5(w_q + w_f)$, где q – узел с наибольшей накопленной ошибкой; f – топологический сосед q с наибольшей накопленной ошибкой. Ребро между f и q удаляется, вместо него добавляются ребра между f и r, а также между r и q. Ошибка в узлах f и q уменьшается умножением на константу $\alpha < 1$; значение ошибки нового нейрона r инициализируется равным значению ошибки q.

10. Ошибка во всех узлах уменьшается умножением на константу $d < 1$.

11. Если условие остановки не выполнено, перейти к шагу 2. Стандартным условием остановки является выполнение определенного количества итераций обучения.

Присутствие механизма добавления новых узлов (а также удаления излишних старых) делает данный алгоритм несравнимо более гибким по сравнению с его прямым предшественником, однако эта процедура обладает серьезным недостатком. При малых значениях параметра λ (< 100) чрезмерно частое внедрение новых узлов делает процесс обучения нестабильным, а образующиеся структуры – искаженными. Большие значения λ обеспечивают ожидаемый эффект, но прямым образом приводят к значительному замедлению работы алгоритма (т. к. новые узлы добавляются реже). Данный общепризнанный недостаток привел к появлению ряда работ, видоизменяющих механизм роста с тем, чтобы сделать его более устойчивым и в то же время более быстродействующим. Одно из наиболее распространенных решений – введение «сферы влияния» нейрона. Совокупность таких сфер (в общем случае являющихся гиперсферами) для всех существующих на данный момент узлов покрывает ту часть пространства, топология которой уже описана. Поэтому новый нейрон может быть добавлен в том случае, когда вновь поступившая точка из набора входных данных не находится в «сфере влияния» ни одного из уже существующих нейронов. Сферы могут иметь как фиксированный [4], так и адаптивно подстраиваемый [5, 6] радиус. Вести учет ошибки для отдельно взятых узлов в таком случае нет необходимости.

В данной работе применяется собственный механизм добавления новых узлов, использующий, впрочем, описанную выше идею с введением «радиуса досягаемости» нейронов. Параметры α и d, связанные с учетом ошибки, исключаются, а параметр λ приобретает смысл порогового значения – максимального удаления нейрона от каждой из приписанных ему точек обучающего множества. Каждый нейрон может быть *фиксированным* либо *нефиксированным*. Модифицированный алгоритм назовем *алгоритмом быстрорастущего нейронного газа*. Его полный вариант приводится ниже.

1. Задаются два начальных узла (нейрона) a и b в случайных точках w_a и w_b пространства входных данных $\mathbb{R}^n$. a и b соединяются ребром, возраст которого равен нулю. a и b принимаются нефиксированными.

2. Выбирается очередной набор данных (сигнал) ξ из входного множества.

3. Определяются ближайший и второй по близости к ξ узел (обозначим их s_1 и s_2 соответственно). Обычно в качестве меры близости используется стандартное евклидово расстояние $\|w(s) - \xi\|$.

4. Возраст всех инцидентных s_1 ребер увеличивается на единицу.

5. Если s_1 фиксирован, переходим к шагу 6, иначе к шагу 7.

6. Если $\left\|w_{s_1} - \xi\right\| \le \lambda$, то переходим к шагу 9. Иначе добавляется новый нефиксированный нейрон r в точку, совпадающую с входным набором $w_r = \xi$, а также добавляется новое ребро, соединяющее r и s_1, затем переходим к шагу 10.

7. s_1 и его топологические соседи (узлы, соединенные с ним ребром) смещаются по направлению к ξ на расстояния $\Delta w_{s1} = \varepsilon_b(\xi - w_{s1})$ и $\Delta w_n = \varepsilon_n(\xi - w_n)$ соответственно, где $0 < \varepsilon_b \ll 1$ и $0 < \varepsilon_n \ll \varepsilon_b$.

8. Если $\left\|w_{s_1} - \xi\right\| \le \lambda$, отмечаем нейрон s_1 как фиксированный.

9. Если s_1 и s_2 соединены ребром, его возраст обнуляется; в противном случае между s_1 и s_2 создается новое ребро с возрастом, равным нулю.

10. Все ребра в графе с возрастом более $a_{\max}$ удаляются. В случае если после этого некоторые узлы не имеют инцидентных ребер (оказываются изолированы), эти узлы также удаляются.

11. Если обучающее множество исчерпано, переходим к шагу 12, иначе переходим к шагу 2.

12. Если все нейроны фиксированы, выполнение алгоритма закончено, иначе переходим к шагу 2 и начинаем новую эпоху обучения (повторение обучающего множества).

Приведенный выше алгоритм выполняет построение покрытия обучающего множества с точностью (максимальным отклонением) λ. Проблема со скоростью роста устраняется, поскольку добавление нейронов осуществляется тогда и в те точки пространства, когда и где обнаруживаются еще не описанные существующей структурой наборы из обучающего множества. Помимо этого, данный алгоритм также предоставляет естественным образом вытекающее условие остановки (фиксирование всех нейронов соответствует полному описанию обучающего множества), а также избавляет от необходимости задания верхней границы количества нейронов (поскольку при достижении необходимой точности покрытия добавление узлов прекращается).

Синтез классификатора

Для синтеза классификатора на основе полученного графа необходимо определить две стратегии: стратегию маркирования нейронов и собственно стратегию классификации. Первая из них описывает механизм присвоения меток классов узлам структуры нейронного газа на основе информации, содержащейся в обучающем множестве. Вторая определяет, каким образом будет производиться классификация вновь поступающих данных.

Возможные варианты стратегий рассмотрены в [7]. В данной работе рассматривается единственная и простейшая стратегия классификации (впрочем, являющаяся, согласно [7], наиболее эффективной) – *одиночной связи* (single-linkage). Согласно ей класс вновь поступающей точки x_{new} определяется классом ближайшего к ней узла в графе:

$$class_s(x_{new}) = \arg\min_c(\min_{n \in N(c)} \left|n - x_{new}\right|^2).$$

Для маркировки нейронов по обучающим примерам возможны следующие способы:

1. *По минимальному расстоянию* – нейрону n_i присваивается метка класса наиболее близкого к нему обучающего примера $x \in X_{train}$:

$$l_{\text{mindist}}(n_i) = l(\arg\min_{x \in X_{train}} \left|n_i - x\right|^2).$$

2. *По среднему расстоянию* – нейрон n_i получает метку класса из соображений минимизации среднего расстояния от себя до всех обучающих примеров данного класса $X(c)$:

$$l_{\text{avgdist}}(n_i) = \arg\min_c \sum_{k=1}^{|X(c)|} \frac{|n_i - x_k|^2}{|X(c)|}.$$

3. *По методу большинства* – нейрон n_i получает метку класса, точки которого численно преобладают в области Вороного $v(n_i)$ для n_i (т. е. находятся к n_i ближе, чем к любому из остальных нейронов):

$$l_{major}(n_i) = \arg\max_c |X(c) \cap v(n_i)|.$$

Согласно [7], методы большинства и минимального расстояния демонстрируют достаточно близкие результаты, заметно превосходящие результаты при маркировке по среднему расстоянию. Тем не менее объединяющим фактором для них является тот факт, что они применяются апостериорно – после окончания обучения и формирования структуры нейронного газа. Ввиду вычислительной сложности реализации маркировки по методу большинства (количество обучающих примеров и вершин графа может быть велико, затрудняя поиск) под *апостериорным* вариантом классификатора будем иметь в виду классификатор, полученный с помощью маркирования по минимальному расстоянию.

Альтернативой данному подходу является динамическое маркирование, выполняемое вместе с обучением в основном цикле алгоритма. Введение в систему памяти нейронов позволит еще более снизить вычислительные затраты по сравнению с апостериорным классификатором. Пусть каждый нейрон имеет счетчики, количество которых равно числу классов в решаемой задаче. Счетчик инкрементируется в случае, когда для обучающего примера соответствующего класса данный нейрон оказывается ближайшим (s_1 в приведенной ранее нотации) и выполняется условие $\|w_{s_1} - \xi\| \le \lambda$ (что проверяется на шагах 6 и 8 алгоритма быстрорастущего нейронного газа). После окончания обучения каждый нейрон помечается классом, значение счетчика которого является наибольшим. Данный вариант классификатора назовем *частотным*, поскольку счетчики фактически показывают частоты «попаданий» обучающих примеров в уже покрытые нейронами области пространства.

Экспериментальные результаты

Для проверки работы предложенного алгоритма воспользуемся наборами данных из репозитория машинного обучения UCI [8]. Показатели других распространенных и широко используемых алгоритмов, приведенные для сравнения, получены в [9]. Использованы следующие сокращения:

- FGNG – апостериорный вариант классификатора на основе быстрорастущего нейронного газа;
- FGNG2 – частотный вариант классификатора на основе быстрорастущего нейронного газа;
- MLP – стандартная модель нейронной сети (многослойный персептрон), обучаемая методом обратного распространения ошибки;
- RBF – радиально-базисная нейронная сеть;
- k-NN – метод k ближайших соседей;
- SVM – метод опорных векторов;
- TSEA – нейроэволюционный классификатор на основе сетей высокого порядка; предложен в [9].

Параметры предложенного алгоритма приняты следующими:

- λ – выбирается отдельно для каждого набора данных перебором с сужающимся шагом из интервала 0…100;
- $\varepsilon_b = 0{,}5$;
- $\varepsilon_n = 0{,}05$;
- $a_{\max} = 100$.

В табл. 1 приведены некоторые сведения об использованных наборах данных.

Таблица 1. Задействованные для экспериментов наборы данных

Название	Кол-во примеров	Обуч. примеры	Тест. примеры	Входные параметры	Классы
balance	625	460	165	4	3
cancer	683	500	183	9	2
card	690	500	190	14	2
diabetes	768	575	193	8	2
heart	297	220	77	13	2
hepatitis	155	110	45	19	2
ionosphere	351	260	91	34	2
liver	345	250	95	6	2
newthyroid	215	160	55	5	3
waveform	5000	3750	1250	40	3

В табл. 2 приводятся показатели точности классификации (в процентах) вышеперечисленных методов для различных наборов данных. Для каждого набора полужирным шрифтом выделен лучший метод, курсивом – второй по точности.

Таблица 2. Сравнение точности классификации предложенного алгоритма с другими известными подходами

Метод	*balance*	*cancer*	*card*	*diabetes*
FGNG	83,03	**99,45**	86.32	73.68
FGNG2	79,35	97,27	87.37	76.17
MLP	*93,78*	97,81	84.10	75.94
RBF	88,27	97,20	75.84	77.34
k-NN	91,67	98,85	85.55	75.00
SVM	88,46	98,28	*88.44*	*78.13*
TSEA	**96,20**	*98,98*	**88.68**	**78.63**
Метод	*heart*	*hepatitis*	*ionosphere*	*liver*
FGNG	80,52	**91,11**	91.21	67.37
FGNG2	*85,71*	88,89	91.21	*70.53*
MLP	84,82	84,73	89.12	65.65
RBF	**86,75**	89,30	*92.46*	57.17
k-NN	82,89	86,84	90.91	63.95
SVM	82,89	*89,47*	88.64	58.14
TSEA	83,68	85,79	**93.22**	**74.61**

Метод	*newthyroid*	*waveform*
FGNG	96,36	80,11
FGNG2	*98,18*	80,11
MLP	97,08	84,85
RBF	**98,27**	*87,29*
k-NN	94,44	81,12
SVM	88,89	**88,80**
TSEA	94,88	84,46

По табл. 2 можно отметить, что предложенный алгоритм в ряде случаев демонстрирует очень высокие показатели точности (*cancer*, *hepatitis*, *liver*). На некоторых наборах демонстрируемые результаты оказались заметно хуже по сравнению с другими алгоритмами, что позволяет оценить слабые стороны предложенного подхода. Низкие результаты на наборе *waveform* могут свидетельствовать о плохой устойчивости к шуму, в то время как набор *balance* содержит нелинейности достаточно высокого порядка, приводя к очень сильному падению точности по сравнению с аналогами. Впрочем, необходимо также учитывать, что классификатор на основе растущего нейронного газа не требует большого количества вычислительных ресурсов для обучения (особенно при использовании частотного варианта).

Заключение

Предложенный классификатор на основе быстрорастущего нейронного газа обладает рядом преимуществ. Модифицированный механизм роста позволяет повысить скорость обучения, в то время как основа алгоритма нейронного газа, обеспечивающая малые вычислительные затраты, оставлена без изменений. Эксперименты на десяти различных наборах реальных данных показали, что в среднем предложенный классификатор не уступает существующим аналогам, а в ряде случаев способен их превзойти. Удобство в использовании обусловлено малым количеством задаваемых пользователем параметров, а также «рациональным» условием остановки обучения, не требующим внешнего вмешательства. Необходимость выбора порогового значения λ остается недостатком алгоритма; для его устранения разрабатывается механизм адаптации, подбирающий данное значение без участия человека.

СПИСОК ЛИТЕРАТУРЫ

1. A K-Nearest Classifier Design / Y. Prudent et al. // ELCVIA Electronic Letters on Computer Vision and Image Analysis. – 2005. – V. 5 (2). – P. 58–71.
2. A "neural-gas" network learns topologies / T. Martinetz et al. – University of Illinois at Urbana-Champaign, 1991. – P. 397–402.
3. A growing neural gas network learns topologies / B. Fritzke et al. // Advances in neural information processing systems. – 1995. – V. 7. – P. 625–632.
4. Prudent Y., Ennaji A. An incremental growing neural gas learns topologies //Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on. – IEEE, 2005. – V. 2. – P. 1211–1216.
5. Furao S., Ogura T., Hasegawa O. An enhanced self-organizing incremental neural network for online unsupervised learning // Neural Networks. – 2007. – V. 20 (8). – P. 893–903.
6. An adaptive incremental clustering method based on the growing neural gas algorithm / M.R. Bouguelia et al. // ICPRAM. – 2013.
7. Beyer O., Cimiano P. Online semi-supervised growing neural gas // International journal of neural systems. – 2012. – V. 22 (5).
8. Bache K., Lichman M. UCI Machine Learning Repository. URL: http://archive.ics.uci.edu/ml (дата обращения: 08.04.2014).
9. Tallón-Ballesteros A.J., Hervás-Martínez C. A two-stage algorithm in evolutionary product unit neural networks for classification // Expert Systems with Applications. – 2011. – V. 38 (1). – P. 743–754.

Поступила 06.10.2014.