ЛОГИКА И СЕМАНТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ
(Вычислительные системы)
1992 год Выпуск 146

УДК 519.68

ОБ УСТОЙЧИВОСТИ АЛГОРИТМОВ РАСПОЗНАВАНИЯ ОБРАЗОВ

М.В. Сапир

А.С.Нудельман [1] предложил весьма естественный критерий для алгоритмов распознавания образов, который он назвал критерием устойчивости. Представляется интересным исследовать условия, при которых алгоритм распознавания образов является устойчивым - это позволит целенаправленно строить именно такие алгоритмы и упростит проверку устойчивости существующих алгоритмов.

Напомним основные понятия из [1].

Задано признаковое пространство W. Алгоритм распознавания образов R ставит в соответствие обучающей выборке $X = \bigcup_{i=1}^{k} X_i$ (X_i из W, X_i - класс, $X_i \cap X_j = \emptyset$, k - количество классов) кортеж образов $(T_1, \dots, T_k)$ из W. Через $R(X)(x)$ обозначим номер образа, которому принадлежит x, или условное пустое значение 0, которое соответствует отказу от распознавания x. Функция $R(X)(x)$ обычно называется правилом распознавания. По определению будем считать $R_i(X) \stackrel{def}{=} T_i$.

А.С.Нудельман назвал изучаемое свойство алгоритмов распознавания устойчивостью. Поскольку это не единственное свойство такого типа, которое можно ожидать от алгоритмов распознавания, будем называть его детерминированной устойчивостью, или д-устойчивостью.

Алгоритм распознавания образов д-устойчивый, если $\forall y\ R(X')(y) = R(X)(y)$, когда выборки X, X' удовлетворяют условию: $X'_i = X_i \cup \{x\}$, $x \in R_i(X)$, при некотором i и $X'_j = X_j$, когда $j \neq i$.

ТЕОРЕМА 1. *Свойство д-устойчивости правила распознавания образов эквивалентно тому, что каждый образ T_i получается как замыкание подмножества точек соответствующего класса X_i относительно некоторого оператора замыкания.*

ДОКАЗАТЕЛЬСТВО. Напомним свойства оператора замыкания $[\,]$:

1) $[X] \cup [Y] = [X \cup Y]$,

2) $[X] \supseteq X$,

3) $[\varnothing] = \varnothing$,

4) $[[X]] = [X]$,

5) $X \supseteq Y \Rightarrow [X] \supseteq [Y]$.

Заметим, что свойство 1 несущественно в этой ситуации, так как классы в обучающей выборке не могут объединяться - количество классов от обучающей выборки не зависит. Свойство 3 также не имеет смысла устанавливать, так как никакие классы в обучающей выборке не пусты. Требуется доказать, таким образом, что свойства 2,4,5 оператора $R_i(X)$ эквивалентны его д-устойчивости.

Необходимость. Пусть дана некоторая исходная обучающая выборка X^0. Будем изменять обучающую выборку, добавляя точки к множеству X^0_i так, чтобы они попали в образ $R_i(X^0) = T^0_i$. Тогда T_i будет функцией от множества точек $Q_i = T^0_i \cap X_i$: $T_i = Z(T^0_i \cap X_i) = Z(Q_i)$. Покажем, что Z - оператор замыкания для Q. Д-устойчивость означает, что для любого конечного Q_i $\quad T^0_i \cap X_i \subseteq Q_i \subseteq T^0_i$, $Z(Q_i) = T^0_i$. Отсюда сразу получаются свойства 2 и 5 оператора Z. Переходом к пределу при $Q_i \to T^0_i$ получим требуемое свойство 4.

Достаточность. Пусть условия теоремы выполняются: существует оператор замыкания Z: $Z(X_i \cap T_i) = T_i$. Предположим, $y \in T_i$, $X_i' = X_i \cup \{y\}$, $X_j' = X_j$ при $j \neq i$. С одной стороны, по свойству монотонности 5 оператора замыкания, имеем $Z(X_i' \cap T_i) \supseteq Z(X_i \cap T_i) = T_i$, а с другой стороны, по свойству 2, $Z(X_i' \cap T_i) \subseteq Z(T_i) = T_i$. Тем самым д-устойчивость доказана.

Заметим, что оператор замыкания для i-го образа, о котором идет речь в теореме, вообще говоря, зависит от остальных точек обучающей выборки, не попавших в Q_i.

Обозначим через $P_i(X,Y)$ предикат, которому удовлетворяет любая связная область Y из i-го образа, получаемого алгоритмом распознавания на выборке X. Естественно ожидать от алгоритма распознавания выполнения следующих условий:

1. Алгоритм находит все максимальные по вложению области H, для которых выполняется один из предикатов $P_i(X,H)$, $i = 1, \ldots, k$.

В работе [1] предполагается, что правила распознавания правильно распознают все точки из обучающей выборки. Это условие, вообще говоря, не является обязательным. Предположим,

2. $P_i(X,H) = P_{i,1}(X,H) \,\&\, P_{i,2}(X,H)$, где $P_{i,1}$ зависит только от "границы" множества H, $P_{i,2}(X,H)$ зависит только от количества точек каждого класса, попавших в H, а именно, $P_{i,2}(X,H) = P_{i,2}(q_i(X,H), \bar{q}_i(X,H))$, где $q_i(X,H)$ - количество точек i-го образа, попавших в H, $\bar{q}_i(X,H) = \sum\limits_{j \neq i} q_j(X,H)$.

3. $P_{i,2}(a,b) \Rightarrow a \geq b$.

ТЕОРЕМА 2. *Если алгоритм распознавания удовлетворяет условиям 1,2, то он д-устойчив тогда и только тогда, когда* *а) каждое множество* $H: P_{i,1}(X,H)$ *является замыканием множества* $H \cap X_i$ *относительно некоторого оператора замыкания, и*

б) существуют некоторые числа $k_{i,1}, k_{i,2}$ *такие, что*

$$P_{i,2}(X,H) \Leftrightarrow (q_i(X,H) \geq k_{i,1}) \& (\bar{q}_i(X,H) \leq k_{i,2}).$$

ДОКАЗАТЕЛЬСТВО.

Необходимость. Пусть алгоритм д-устойчив и удовлетворяет условиям 1,2. Сначала докажем "а". Предположим, на обучающей выборке X алгоритм нашел связное множество H как подмножество образа T_i. Пусть снова $X'_i = X \cup \{y_1, \ldots, y_m\}$, $y_l \in H$; $X'_j = X_j$ при $j \neq i$. Из д-устойчивости алгоритма следует изотонность предиката $P_{i,2}(x,y)$ по первой переменной: $P_{i,2}(x,y) \Rightarrow P_{i,2}(x+m,y)$. Следовательно, $P_i(X',H)$ тогда и только тогда, когда $P_{i,1}(X',H)$. Доказательство того, что $H: P_i(X,H)$ для д-устойчивого алгоритма распознавания является замыканием множества $Q = H \cap X_i$, в точности повторяет доказательство необходимости из теоремы 1.

Покажем, что выполняется "б". Пусть c'_1 - минимальное число такое, что для некоторого c_2 $P_{i,2}(c'_1, c_2)$, c'_2 - максимальное число такое, что $P_{i,2}(c'_1, c'_2)$. Рассмотрим обучающую выборку X такую,что для некоторого связного Y $P_i(X,Y)$ и $q_i(X,Y) = c'_1$, $\bar{q}_i(X,Y) = c'_2$, и $\{\exists T \quad T \supset Y \& P_{i,1}(X,T) \& \bar{q}_i(X,T) = c'_2 + 1\}$. Обозначим через y элемент, который принадлежит T и не принадлежит Y. Допустим, $y \in R_j(X)$, $j \neq i$. Предположим, существует число z такое, что $P_{i,2}(c'_1+z, c'_2+1)$. Поместив внутрь Y еще $l = z - q_i(X,T)$ точек i-го класса, получим выборку X'. По доказанному свойству "а" для T будет выполняться $P_{i,2}(X',T) \& P_{i,1}(X',T)$. .Это означает, в соответствии со свойством 1, что алгоритм в качестве одной из связных областей i-го образа найдет множество, охватывающее T. При этом $R(X')(y) = i$. Это противоречит д-устойчивости алгоритма R. Следовательно, число $k_{i,2}$ из формулировки теоремы равно $c_2^!$. В силу изотонности по первой переменной предиката $P_{i,2}(X,T)$ число $k_{i,1}$ из формулировки теоремы равно $c_1^!$.

<u>Достаточность</u>. Пусть $X_i' = X_i \cup \{y\}$, $y \in R_i(X)$, $X_j' = X_j$ при $j \neq i$. Очевидно, что множество $H = \underset{\subseteq}{\operatorname{argmax}} (P_i(X,Y))$ будет удовлетворять предикату $P_i(X',H)$. Покажем, что никакое вмещающее H связное множество не будет удовлетворять тому же условию. Допустим, существует множество T, $T \supset H$ & $P_{i,1}(X,T)$. Поскольку H – максимальное подмножество, удовлетворяющее условию $P_i(X,Y)$, подмножество T не удовлетворяет условию $P_{i,2}(X,T)$. Так как этому условию удовлетворяет меньшее множество H, то множество T содержит больше, чем $k_{i,2}$ точек всех классов, кроме i-го. Следовательно, $\neg P_i(X',T)$. Таким образом, на выборке X' алгоритмом будут найдены те же самые связные подмножества. Что и требовалось доказать.

Пользуясь теоремой 2, легко описать ряд разумных д-устойчивых алгоритмов распознавания через условия на связные области i-го образа в признаковом пространстве, которые алгоритм распознавания строит. Зададим эти условия как предикаты $P_i(X,T) = \underset{\subseteq}{\operatorname{argmax}} (P_{i,1}(X,T) \,\&\, P_{i,2}(X,T))$. Предикат $P_{i,2}(X,T)$ определим как в теореме 2.

Тогда $P_{i,1}(X,T)$ может быть таким:

1) T - минимальное полупространство, отделяемое гиперплоскостью, содержащее некоторое множество $Q \subset X_i$;

2) T - минимальная выпуклая оболочка некоторого множества точек $Q \subset X_i$;

3) T - минимальная сфера, содержащая все точки некоторого $Q \subset X_i$;

4) T - одна из "ячеек", на которые заранее разбито признаковое пространство, содержащая точки из X_i ;

5) T - "гиперинтервал", область, ограниченная системой неравенств по каждой переменной вида $\{ a_i \le x_i \le b_i$, $i = 1, \ldots, n-1 \}$, где x_i - переменная i-го признака, а a_i, b_i - некоторые значения i-го признака из обучающей выборки.

В том случае, когда признаки - качественные, номинальные:

6) $P_{i,1}(X,T)$ - конъюнкция атомарных формул вида $(x_i = a_i)$;

7) $P_{i,1}(X,T)$ - любая бескванторная формула исчисления предикатов с атомарными формулами вида $(x_i = a_i)$.

Условие 1 дает нам один из вариантов метода комитетов [4], условие 4 аналогично персептронному подходу к распознаванию образов [5], условие 6 дает разновидность метода "Кора" [6].

Остановимся на алгоритмах с предикатами 5.

Алгоритм гиперинтервалов допускает работу как с количественными, так и с ранговыми признаками, так как в описании искомых областей не участвуют никакие арифметические операции над значениями признаков. Это представляется существенным для задач, где многие признаки измеряются с погрешностью, которая сама зависит от значения, и где признаки могут иметь пороговые величины, по разные стороны от которых значения имеют качественно разный смысл. В этих случаях ввести правильно метрику на признаковом пространстве практически невозможно, и арифметические операции над признаками также теряют смысл. Тем более не имеют смысла операции, в которых участвуют признаки с разными наименованиями. Таким образом, фактически признаки требуется рассматривать как ранговые.

Важное преимущество гиперинтервального подхода состоит в том, что полученные гиперинтервалы можно рассматривать как гипотезы о взаимосвязях сочетаний признаков с целевым признаком. Такие гипотезы легко включаются в конкретно-научный контекст, так как они формулируются близко к тому языку, который привычен исследователю в плохо формализованных областях знания (медицина, биология, геология).

Весьма перспективным свойством этого алгоритма является то, что одновременно с построением правила распознавания сжимается признаковое пространство, т.е. находятся наиболее информативные сочетания признаков, причем информативные именно для данного метода, а не абстрактно. Действительно, если максимальный информативный гиперинтервал содержит весь промежуток значений некоторого признака, значит, из описания этого гиперинтервала этот признак может быть исключен.

Приведем алгоритм поиска максимальных информативных гиперинтервалов. Алгоритм состоит из двух частей. На первом этапе строятся все максимальные информативные гиперинтервалы, на втором этапе формируются наиболее короткие и достаточно "хорошие" правила распознавания из найденных алгоритмом. После первого этапа конкретный специалист, постановщик задачи, может исследовать полученные гиперинтервалы и отобрать для построения правила распознавания те из них, которые он считает наиболее убедительными, осмысленными.

Дадим удобную кодировку гиперинтервала для описания алгоритма. Каждому гиперинтервалу $R = \{r_{i1} \le x_i \le r_{i2}\}_{i=1}^{n-1}$, где r_{ik} принадлежит множеству значений i-го признака из обучающей выборки при каждом $i = 1,\dots, n-1$; $k = 1,2$, поставим в соответствие последовательность длины $(n - 1)$ пар натуральных чисел $q = \langle(a_{1,1},a_{1,2}),\dots, (a_{n-1,1}$,

$a_{n-1,2}\rangle\rangle$, где a_{i1} (a_{i2}) равно числу различных значений i-го признака, которые меньше r_{i1} (больше r_{i2}), $i=1,\ldots,n-1$. Нетрудно видеть, что между множеством непустых гиперинтервалов и множеством последовательностей пар неотрицательных целых чисел $Q=\{a=\langle(a_{1,1},a_{1,2}),\ldots$ $\ldots,(a_{n-1,1},a_{n-1,2})\rangle : \forall i \;\; a_{i1}+a_{i2}<m_i\}$, где m_i - количество различных значений i-го признака в обучающей выборке, устанавливается взаимно-однозначное соответствие, поэтому в дальнейшем в качестве описания гиперинтервала будем использовать соответствующую последовательность из Q; первый индекс всюду означает номер пары в последовательности, второй - номер числа в паре. Через $[a]$ обозначим гиперинтервал, соответствующий последовательности a из Q. Очевидно, $[a]\supseteq[b]$ тогда и только тогда, когда $\forall i\;\forall j\;(1\le i\le$ $\le n-1)\;(1\le j\le 2)\;\; a_{ij}\le b_{ij}$, и $[a]\supset[b]$ тогда и только тогда, когда по крайней мере для одного сочетания i,j достигается строгое неравенство. Перенесем отношение вложенности с гиперинтервалов на соответствующие им последовательности из Q. Будем писать $a\supseteq b$, если $[a]\supseteq[b]$.

Пусть k, m - некоторые наперед заданные числа. Через $K(a)$ обозначим свойство гиперинтервала $[a]$ содержать не больше k объектов всех классов, кроме одного, $M(a)$ будет обозначать, что количество точек одного из классов в гиперинтервале не меньше m.

Теперь задача может быть сформулирована следующим образом. Нужно найти максимальные по отношению к $\subseteq$ элементы среди всех $a\in Q$, удовлетворяющие условию $K(a)\;\&\;M(a)$.

Для описания алгоритма введем на множестве Q отношение лексикографического порядка (л-порядок) $\prec$. Относительно последовательностей a,b из Q будем говорить, что $a\prec b$, если первое слева число, которое в этих последовательностях не совпадает, в последовательности a меньше, чем в b. Перед описанием алгоритма сделаем два очевидных замечания, которые проясняют связь между л-порядком на Q и вложенностью гиперинтервалов.

ЗАМЕЧАНИЕ 1. Если a, b из Q, b - ближайшая л-следующая после a и $[a]$ не является минимальным по вложению, то $b \subsetneq a$ и $\forall c \quad \neg(b \subsetneq c \subsetneq a)$.

ЗАМЕЧАНИЕ 2. Если $b \subseteq a$, то $b \succ a$.

Обозначим через $\oplus$ операцию конкатенации, слияния, последовательностей.

Грубо говоря, алгоритм заключается в том, что в л-порядке перебирают последовательности из Q, вычисляя для каждой из них количество точек каждого класса, попавших в соответствующий гиперинтервал, и пропуская в этом переборе только те последовательности, гиперинтервалы для которых вложены в заведомо "бесперспективный" гиперинтервал (содержащий мало точек каждого класса).

Дадим точное описание алгоритма. Алгоритм представлен как итеративное определение элементов памяти - текущего кортежа y и текущей последовательности памяти R. Условие остановки алгоритма - невозможность выполнить следующее предписание.

Пусть y^i - текущий кортеж из Q; $y^0 = \{(0,0), \ldots, (0,0)\}$; R^i - текущая память алгоритма; $R^0 = \emptyset$. Каждый кортеж y, который записывается в память R, печатается.

Алгоритмом проводятся следующие вычисления:

$$y^{i+1} = \begin{cases} \min\limits_{\prec}(y\colon y \succ y^i), \text{ если } \neg K(y^i) \,\&\, M(y^i), \\ \min\limits_{\prec}(y\colon y \succ y^i \,\&\, \neg(y \subset y^i)) \text{ – иначе;} \end{cases} \qquad (1)$$

$$R^{i+1} = \begin{cases} R^i \oplus \langle y^i \rangle, \text{ если } & K(y^i) \,\&\, M(y^i) \,\& \\ & \&(\forall q \; q \in R^i \Rightarrow \neg(y^i \subseteq q)), \\ R^i - \text{иначе}. \end{cases} \quad (2)$$

Условие (1) означает, что гиперинтервал $[y^i]$, хотя и не является решением задачи, так как предикат $K(y^i)$ не истинен, но может содержать в себе гиперинтервал, удовлетворяющий требованиям задачи. В этом случае следующим текущим кортежем выбирается ближайший лексикографически следующий, как один из ближайших по вложению для $[y^i]$ (см. замечание 1). Если условие (1) не выполняется, следующий текущий кортеж выбирается в л-порядке так, чтобы его гиперинтервал не был вложен в $[y^i]$. Условие (2) означает, что гиперинтервал $[y^i]$ является решением, причем не принадлежит никакому ранее записанному в память решению. В этом и только в этом случае текущий кортеж записывается в память R^{i+1}.

ТЕОРЕМА 3. *Алгоритм поиска максимальных информативных гиперинтервалов решает задачу.*

ДОКАЗАТЕЛЬСТВО. Нетрудно видеть, что алгоритм перебирает в л-порядке все последовательности из Q за исключением л-промежутков $\{y\colon y \succ a \;\&\; y \subset a\}$, где $\neg K(a)$ или $\neg M(a)$, или $P(a)$. В этих промежутках, очевидно, нет решений. Если текущий гиперинтервал удовлетворяет требованиям задачи и среди ранее напечатанных нет вмещающего, то он будет напечатан. Следовательно, если алгоритм не печатает гиперинтервал $[y]$, то $[y]$ не является решением.

Покажем, что если алгоритм печатает гиперинтервал $[y]$, то $[y]$ - решение. Очевидно, что гиперинтервалы, которые печатает алгоритм, удовлетворяют требованиям $T(y) = P(y) \,\&\, \&K(y)$. Нужно показать только, что $[y]$ является максимальным по вложению среди всех гиперинтервалов, удовлетворяющих этим требованиям. Произвольный гиперинтервал $[z]$, вмещаю -

щий $[y]$, л-предшествует $[y]$ (см. замечание 2). Так как просмотр осуществляется в л-порядке, условия $T(z)$ уже проверены на предыдущих шагах, или они заведомо не выполняются. Так как в память алгоритма не записан ни z, ни какой-нибудь вмещающий его гиперинтервал (иначе алгоритм не напечатал бы y), то $T(z)$ ложно. Это и означает максимальность $[y]$.

Построение правила распознавания. При формировании правила распознавания нужно учитывать, что процент правильного распознавания по совокупности гиперинтервалов даже на обучающей выборке может быть существенно ниже, чем по каждому гиперинтервалу в отдельности. Проблема состоит в том, чтобы выбрать из всех "подходящих" гиперинтервалов такой максимальный по объему набор, что его общая информативность достаточно высока.

Предлагается следующий приближенный подход для ее решения. Допустим, в выборке есть всего два класса, или решается задача выделения одного класса среди всех. На первом этапе для каждого из двух классов ищутся такие минимальные совокупности гиперинтервалов данного класса, которые вместе покрывают достаточный процент из всех точек данного класса.

Функция, которая каждому набору гиперинтервалов одного класса ставит в соответствие 0, если процент захвата точек выбранного класса меньше заданного, и 1 - в противном случае, является булевой функцией, монотонно возрастающей по вложенности наборов, и искомые наборы являются ее "нижними единицами". Поэтому для решения задачи первого этапа можно применить методы расшифровки монотонных булевых функций. Наиболее удобными представляются алгоритмы [2,3], в которых первые решения находятся достаточно быстро, и их нахождение не требует завершения работы алгоритма в целом, как в других методах. Так как использовать все решения практически нет возможности, это оказывается существенно. Как показано в [3], время для нахождения первого решения в обоих предложенных алгоритмах линейно зависит от размерности булеана (в данном случае равной количеству гиперинтервалов с преобладанием выделенного класса).

Среди найденных сочетаний гиперинтервалов выделяются те,в которых достаточно высока доля точек соответствующего класса.

На втором этапе перебираются попарно выбранные сочетания гиперинтервалов за первый и второй класс, и выбираются такие пары сочетаний, в которых качество распознавания удовлетворительное.

Алгоритм реализован в пакете программ автора "Empiric". В программе использованы некоторые методы ускорения вычислений , разбор которых далеко выходит за тему данной работы. Программа позволяет ограничить количество признаков, участвующих в описании гиперинтервалов, контролировать гиперинтервалы по контрольной выборке. Алгоритм неоднократно.применялся на медицинских задачах.

Литература

1. НУДЕЛЬМАН А.С. Об одном свойстве методов индукции над стандартными эмпирическими теориями // Машинный анализ сложных структур. - Новосибирск, 1986. - Вычислительные системы: Вып.118. - С.81-99.

2. САПИР М.В. Экстремальные задачи на конечных множествах с монотонной мерой //Автоматика и телемеханика. - 1987. - №5. - С. 149-155.

3. САПИР М.В. Алгоритмы поиска экстремальных подмножеств для монотонной связи //Автоматика и телемеханика. - 1988.-№ 1. - С. 119-125.

4. МАЗУРОВ Вл.Д.,ТЯГУНОВ Л.И. Метод комитетов в распознавании образов //Метод комитетов в распознавании образов.-Свердловск. - 1974. - Вып. 6. - С. 14-40.

5. МИНСКИЙ М., ПЕЙПЕРТ С. Перцептроны. - М.: Мир, 1971.

6. ВАЙНЦВАНГ М.И. Алгоритм обучения распознаванию образов "Кора" //Алгоритмы обучения распознаванию образов. - М.: Сов. радио, 1973. - С.110-115.

Поступила в ред.-изд.отд.
3 августа 1992 года