

СТРУКТУРНЫЙ АНАЛИЗ РУКОПИСНЫХ МАТЕМАТИЧЕСКИХ ВЫРАЖЕНИЙ

В настоящей статье рассматривается разработанный метод структурного анализа для онлайн распознавания рукописных математических выражений. Рассматриваются следующие алгоритмы: размещения, чередования, группировки.

This article discusses the developed algorithm of structural analysis for online recognition of handwritten mathematical formulas. Introduce the following algorithms: placement, interleaving, grouping.

Введение

В настоящее время, в связи с развитием планшетных персональных компьютеров, появляется необходимость ввода данных без использования клавиатуры. Математические выражения составляют основную часть в большинстве научных и технических дисциплин. Ввод математических выражений в компьютеры является более сложным, чем ввод обычного текста, так как математические выражения, как правило, состоят из специальных символов и греческих букв в дополнение к английским символам и цифрам. Задачу распознавания рукописных математических выражений можно разделить на два основных этапа: распознавание символов и структурный анализ. Такое разделение на два этапа имеет ряд преимуществ, в частности подзадачи могут быть решены независимо друг от друга, что позволяет проанализировать результаты и сделать улучшения на каждом из этапов, при этом сохранится целостность всей системы. Целью данной статьи является описание разработанного метода структурного анализа, который позволяет определить пространственные отношения между символами и полностью отобразить структуру, написанного пользователем математического выражения.

Одна из самых ранних работ по распознаванию математических выражений была написана Андерсоном [1]. Для определения структуры математического выражения, Андерсон использовал нисходящий подход. Алгоритм начинается с одной окончательной синтаксической цели и попыток разделить данную цель на подцели до тех пор, пока все подцели не будут выполнены или все попытки потерпят неудачу. Однако эксперименты показали, что данный ал-

горитм структурного анализа не всегда эффективен при работе со сложными математическими выражениями, поскольку разделения до $n-1$ могут быть произведены множеством из n символов, в свою очередь каждое из этих разделений может дополнительно генерировать другие разделения, что ведет к неконтролируемому увеличению подцелей. В [2] в структурно-аналитическом этапе был использован нисходящий парсинг для разделения выражения на подвыражения, затем применялся восходящий парсинг, для того чтобы объединить подструктуры в общую структуру. Однако эксперименты были выполнены только на нескольких простых математических выражениях. В [3] был предложен редактор ввода математических выражений. Система позволяет пользователю вводить цифры и символы в любом порядке. Для того чтобы распознать символы, используется алгоритм соответствия образцу, что накладывает определенные ограничения на работу с программой. В [4] используется подход, основанный на скрытых Марковских моделях. Предполагается, что пользователь всегда пишет выражение в определенном порядке, например, при написании дроби, сначала должен быть написан числитель, затем дробная линия и знаменатель. Такое требование может быть легко нарушено на практике из-за высокой изменчивости стиля написания у различных пользователей.

Предлагаемый в данной статье метод структурного анализа, включающий в себя алгоритмы размещения, чередования и группировки, позволяет анализировать математические выражения любой сложности. Рассматриваемый метод анализирует написанное пользователем математическое выражение после каждого внесенного изменения, что позволяет записывать

составляющие математического выражения в любом порядке, а также вносить изменения в уже написанное выражение.

Обзор процесса распознавания

Дадим определение термина математическое выражение – это символическая запись законченного логического суждения, которая может состоять из цифр, букв и различного вида символов, при этом все цифры, буквы и символы формируют внутреннюю иерархическую структуру. Для того чтобы правильно распознать математическое выражение необходимо точно знать значение каждого из символов, которые в него вошли, и определить расположение этих символов относительно друг друга.

В данной статье будет подробно рассмотрен метод структурного анализа, позволяющий определить расположение символов относительно друг друга. Но сначала кратко рассмотрим процесс сегментации и непосредственно распознавания каждого отдельно сегментированного символа, и после перейдем к описанию самого метода структурного анализа. Такой подход позволит создать полную картину процесса онлайн распознавания рукописных математических выражений.

В предлагаемой системе, используется набор из 77 символов, который позволяет писать тригонометрические, логарифмические функции, интегралы, греческие и латинские буквы.

На первом этапе происходит сегментация символов. В рукописном математическом выражении обычно символы написаны отдельно друг от друга. Будем считать, что каждый математический символ состоит из одного или нескольких отрезков. Отрезок состоит из точки касания пера, точки отрыва пера и всех точек лежащих между ними. При добавлении новых отрезков или удалении уже существующих, может значительно измениться значение символа, поэтому происходит динамическая интерпретация символа после каждого внесенного пользователем изменения.

Для решения задачи распознавания изолированных символов был выбран нечеткий классификатор NEFClass. Система NEFClass прекрасно справляется с необходимой задачей классификации символов, поскольку сочетает в себе нейронные сети и нечеткие системы логического вывода, при этом обладает высокой

скоростью работы алгоритмов обучения. Для задачи распознавания рукописных символов

0	7	e	l	s	z	{	.	Ω	√	Δ
1	8	f	m	t	+	}	,	π	!	∇
2	9	g	n	u	-	[Σ	α	≈	λ
3	a	h	o	v	/]	∞	β	γ	σ
4	b	i	p	w	*	<	≥	μ	θ	∫
5	c	j	q	x	(>	≤	φ	ε	∂
6	d	k	r	y)	%	=	δ	€	Э

Рис. 1. Рассматриваемые символы в системе онлайн распознавания математических выражений

был построен нечеткий классификатор с 12 входами и 77 выходами.

Входам системы соответствуют признаки, которые были подобраны специально для математических символов, цифр, латинских и греческих букв, а выходам соответствует набор из 77 символов, которые пользователь может использовать при написании математических выражений. В качестве алгоритма обучения был выбран алгоритм сопряженных градиентов, который обеспечивает сходимость к оптимальному решению за конечное число шагов. В результате этапа распознавания, на выходе нейронной сети будет получен один из символов.

Структурный анализ

Основной задачей этапа структурного анализа является определение расположения символов в математическом выражении относительно друг друга.

Рассмотрим предлагаемый метод более подробно. Все символы были разделены на группы, в зависимости от допустимых позиций размещения других символов относительно них. На Рисунке 2 схематически изображены все восемь возможных позиций.



Рис. 2. Схематическое изображение допустимых позиций расположения символов относительно текущего рассматриваемого символа

Были выделены следующие группы символов:

1. Цифры. Принимают значения от 0 до 9, π , ∞ . Символы могут быть размещены относительно данной группы в следующих позициях – 1, 2.
2. Математические знаки. К этой группе были отнесены такие символы: $*$, $+$, $-$, $/$, $=$, $<$, \leq , $>$, \geq , $\%$, \approx , ϵ , ε , Δ , ∇ . Допустимо размещение символов во 2 позиции.
3. Скобки открывающие. В эту группу вошли следующие символы: $($, $\{$, $[$. Относительно этой группы символы могут располагаться во 2 позиции.
4. Скобки закрывающие. К этой группе относятся такие символы: $)$, $\}$, $]$. Допустимое размещение символов относительно рассматриваемого символа: 1, 2 позиции.
5. Знаки препинания. К знакам препинания были отнесены: «.», «,», «!». Символы могут располагаться в 1, 2 позиции.
6. Интеграл \int . Допустимое размещение символов во 2, 4, 8 позиции.
7. Извлечение корня $\sqrt{\quad}$. Допустимое размещение: 1, 2, 6 позиции.
8. Сумма \sum . Допустимое расположение символов: 2, 4, 8 позиции.
9. Греческие и латинские буквы, рассмотренные на Рис.1. Допустимое размещение символов: 1, 2, 3 позиции.
10. Тригонометрические функции. Рассматриваются следующие тригонометрические функции \sin , \cos , \tan/tg , $ctg/ctn/cot$. Допустимое размещение символов: 1, 2 позиции.
11. Логарифмические функции \ln , \lg , \log . Допустимое размещение символов: 1, 2, 3 позиции.

Под допустимой позицией расположения символов понимается возможная, но не обязательная к заполнению символом позиция. Под термином «обязательная позиция» имеется в виду такая позиция, которая должна быть обязательно заполнена одним из следующих символов, такая позиция получит приоритет при добавлении нового символа. Например, знак \int

или $\sqrt{\quad}$ предполагает обязательное наличие зависимого символа в позиции 2.

Разработанный метод структурного анализа включает в себя следующие алгоритмы: размещения, чередования, группировки. Ниже в статье рассмотрим каждый из этих алгоритмов подробно.

Алгоритм размещения нового символа относительно других символов работает по следующему принципу.

1. Выражение переписывается слева направо каждый раз при добавлении нового символа.
2. Выбирается позиция для нового символа. А по следующему правилу:
 - Проверяется наличие пустой обязательной позиции, в случае ее наличия данная позиция получает приоритет и в нее помещается символ А.
 - Если пустая обязательная позиция отсутствует, то выбирается ближайшая допустимая позиция, в которую помещается новый символ А.
3. В случае, если позиция, в которую был помещен новый символ А, была занята символом Б, то выбирается новая позиция для символа Б по правилам описанным в пункте 2.

Стоит отдельно рассмотреть знаки «.» и «-». Так точка в выражении может быть десятичной запятой или оператором умножения в зависимости от положения точки и ее соседних символов. Горизонтальная линия может быть дробной чертой или знаком минус в зависимости от длины линии и наличия символов выше и ниже линии. В предлагаемом методе для решения такого рода неопределенности были использованы следующие подходы:

- Знак «-» будет рассматриваться как дробная черта, в том случае, если имеет символы в 4 и/или 8 позициях.
- Знак «.» рассматривается как оператор умножения, если центр масс его сегмента находится ближе к 2 позиции соседнего слева символа.

Рассмотрим работу алгоритма чередования. Несколько букв написанных вместе могут формировать единое значение. Это могут быть тригонометрические функции \sin , \cos , \tan/tg , $ctg/ctn/cot$ или логарифмические функции \ln , \lg ,

log. Рассматривая группу букв написанных подряд алгоритм чередования проверяет, не образуют ли они название функции. В этом случае происходит чередование нескольких букв на одну из известных математических функций. Кроме того, алгоритм проверяет различные варианты написания, так, если на этапе распознавания одна из букв была распознана неправильно, на данном этапе ошибка будет исправлена, как показано на Рис. 3. На Рис.3 видно, что каждый раз при добавлении нового символа математическое выражение начинает анализироваться сначала, в данном случае буква «о» на этапе распознавания была ошибочно распознана как цифра «0». После добавления следующего символа алгоритм чередования просматривает все возможные комбинации, имеющиеся в базе, и, найдя наиболее подходящую, исправит ошибку.

Рассмотрим работу алгоритма группировки. Специальные символы, к ним были отнесены различного вида скобки, дробная черта и знаки препинания: точка, запятая, позволяют сгруппировать несколько отдельных значений в одну группу. В результате работы алгоритма группировки соответствующее оригиналу математическое выражение будет отображено наиболее верно.

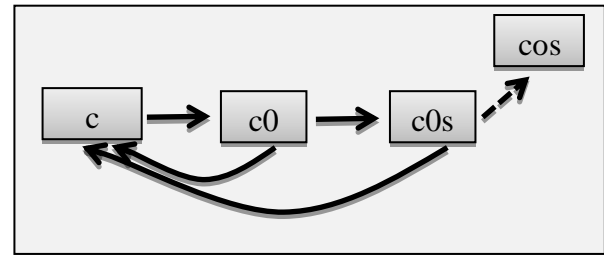


Рис. 3. Пример работы алгоритма чередования

Использование описанного метода позволяет определить пространственные отношения между символами и полностью отобразить структуру, написанного пользователем математического выражения. Кроме того полный анализ структуры математического выражения после добавления каждого нового символа делает систему устойчивой к внесению изменений.

Экспериментальные исследования

Для проверки эффективности работы рассматриваемого метода структурного анализа были проведены тесты в разработанном редакторе ввода рукописных математических выражений Рис. 4.

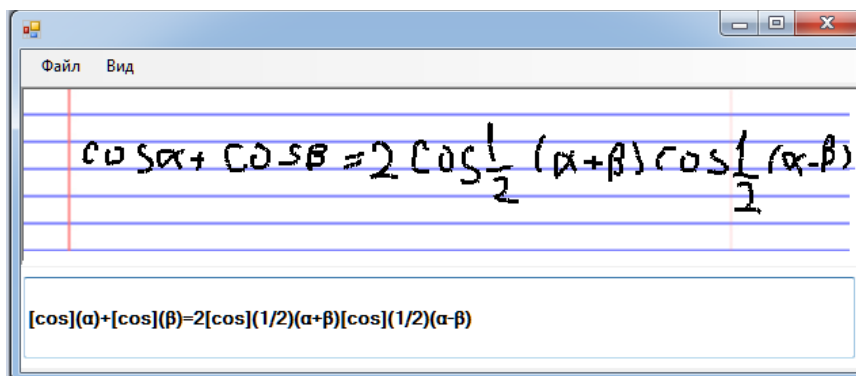


Рис. 4. Пример работы редактора ввода рукописных математических выражений

Для тестирования были отобраны 50 различных математических выражений, включающих в себя тригонометрические, логарифмические функции, операции возведения в степень, извлечения корня, дробного деления, индексы.

На Рис. 4 видно, что редактор хорошо структурирует достаточно сложное математическое выражение, при этом различает последовательность написанных символов в качестве назва-

ния тригонометрической функции, которую показывает в квадратных скобках.

В Таблице 1 приводятся результаты работы метода по количеству неправильно структурированных символов. Можно сказать, что такие символы как буквы и знаки операций имеют меньший процент ошибки, в отличие от логарифмических функций и знака суммы.

Табл. 1. Результаты работы метода структурного анализа

	Кол-во символов, содержащихся в протестированных математических выражениях	Кол-во неправильно структурированных символов	% неправильно структурированных символов
Цифры	371	14	6,46
Знаки операций	194	10	5,15
Скобки	92	8	8,69
Знаки препинания	17	3	17,6
Интеграл	25	3	12
Извлечение корня	23	2	8,69
Сумма	15	2	13,3
Греческие и/или латинские буквы	160	7	4,37
Тригонометрические функции	58	4	6,89
Логарифмические функции	15	2	13,3

На Рис. 5 приводится пример неправильно определенной структуры математического выражения. Здесь метод структурного анализа правильно определил пространственные отно-

шения между символами. Это ошибка могла произойти по причине слишком близкого расположения значения «2» к обязательному полю тригонометрической функции sin.

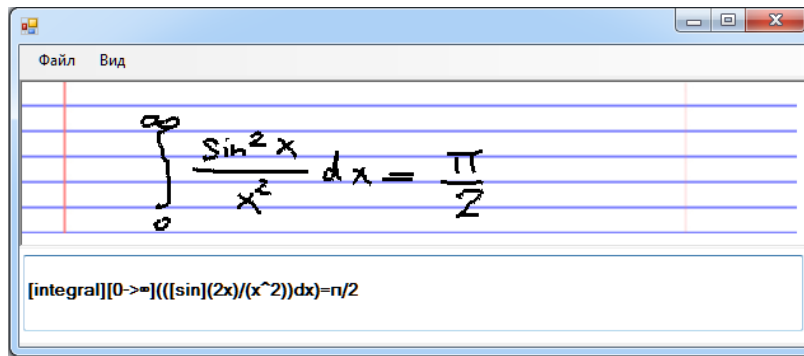


Рис. 5. Пример работы редактора ввода рукописных математических выражений

Табл. 2. Результаты тестирования

Общее кол-во символов, содержащихся в протестированных математических выражениях	Общее кол-во неправильно структурированных символов	% ошибочно структурированных символов
970	55	5,67

Как показано в Таблице 2, в результате тестирования было получено общее значение ошибочно структурированных символов равно 5,67% из проверочного набора данных. В других подходах значение ошибочно структурированных символов лежало в пределах 12%-20% [7]. Полученное значение свидетельствует о достаточно высокой эффективности использования предложенного в работе метода структурного анализа к работе с математическими выражениями любой сложности.

Заключение

В результате проведенного исследования был разработан метод структурного анализа,

который позволяет достаточно точно определить пространственные отношения между символами и как результат полностью отобразить структуру, написанного пользователем математического выражения. Общее значение ошибочно классифицированных символов равно 5,67% свидетельствует об эффективности применения разработанного метода к поставленной задаче. Как показали исследования такие символы как логарифмические функции, знак суммы, знаки препинания имеют самый большой процент неправильно структурированных символов. Это можно объяснить наличием нескольких обязательных позиций у данных символов. Необходимо провести дальнейшие исследования, которые позволят улучшить рас-

смотренный метод структурного анализа. Улучшения будут достигнуты путем доработки алгоритма размещения. На данном этапе модель метода структурного анализа разработана с учетом правила, что пользователь может ввести только одно математическое выражение в редакторе. Для того чтобы пользователь мог ввести несколько математических выражений одновременно, метод должен быть усовершен-

ствован введением дополнительных символов, которые будут разделять математические выражения. Кроме того необходимо разработать систему подсказок, которая позволит пользователю выбирать одну из предлагаемых в списке структур. Предлагаемые улучшения позволят снизить процент ошибочно классифицированных символов.

Список литературы

1. Anderson R. Syntax-directed recognition of hand-printed two-dimensional mathematics. - New York: Academic Press, 1968. – pp. 436-442.
2. Belaid A. Haton J.-P. A syntactic approach for handwritten mathematical formula recognition. - IEEE Transactions on Pattern Analysis and Machine Intelligence, 1984. - pp. 105-107.
3. Nakayama Y. A prototype pen-input mathematical formula editor. – EDMEDIA, 1993. - pp. 400-407.
4. Kosmala A. Rigoll G. On-line handwritten formula recognition using statistical methods. - International Conference on Pattern Recognition 1998 pp. 1306-1308.
5. Зайченко Ю.П. Нечёткие модели и методы в интеллектуальных системах. Учебное пособие для студентов высших учебных заведений. – К.: «Издательский Дом «Слово»», 2008. – с. 344
6. Зайченко Ю.П. Основы проектирования интеллектуальных систем. Учебное пособие. – К.: «Издательский Дом «Слово»», 2004. – с. 352
7. Mouchère H., Viard-Gaudin C., Kim D. H., Kim J. H., Garain U. CROHME2011: Competition on Recognition of Online Handwritten Mathematical Expressions. – IEEE, 2011. – pp. 1497-1500