

УДК 004.932.75'1

Распознавание рукописных математических выражений, вводимых в режиме реального времени

*Луховцев И. А., студент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Программное обеспечение ЭВМ и информационные технологии»*

*Научный руководитель: Корниенко В. В., ассистент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Программное обеспечение ЭВМ и информационные технологии»
vasiliykornienko@mail.ru*

Введение

В математических выражениях символы и буквы представлены в виде сложной двухмерной структуры. Все символы и буквы, будучи правильно сгруппированными, формируют строгую иерархическую структуру. В то же время правильная группировка символов в математических выражениях является очень сложной задачей. Во-первых, существует два типа символов. Первый тип включает в себя все базовые символы, а второй включает в себя специальные математические символы, такие как « $\sqrt{\quad}$ » и « \sum », скобки и символы-операторы. Каждый из этих типов имеет свой критерий группировки. Во-вторых, существует два типа операторов: явные и неявные операторы. Явные операторы — это символьные операторы, такие как «+», «-» и так далее, а неявные операторы определяют отношения между символами по их относительным позициям, например, « a^2 ». В-третьих, некоторые символы могут иметь различные значения в зависимости от контекста.

Каждый символ имеет своё собственное значение. В то же время в математических выражениях различные символы, сгруппированные вместе, могут иметь другое значение, нежели по отдельности. Например, цифры вместе обычно формируют число, когда все они одного размера, находятся рядом друг с другом и лежат на одной горизонтальной прямой. В то же время эти же символы, но с различными размерами и позициями могут нести совершенно другой смысл, например, число 310 и 3^{10} . Также некоторые буквы вместе могут формировать некий блок, например, названия тригонометрических функций, таких как \cos , \sin и так далее. Прежде чем распознать эти буквы как произведение

переменных, необходимо проверить, не формируют ли они вместе название какой-либо функции.

Все эти свойства делают процесс распознавания математических выражений очень сложным даже тогда, когда все отдельные символы и буквы могут быть распознаны правильно.

Распознавание выражений

Процесс распознавания математических выражений делится на три основных этапа: сегментация математического выражения, распознавание отдельных символов и структурный анализ выражения. На этапе сегментации в исходном выражении выделяются все отдельные символы. На этапе распознавания каждый символ, прошедший этап сегментации, проходит классификацию. На последнем этапе проводится анализ результатов предыдущих этапов и формируется структура математического выражения. Входные данные делятся на два типа: вводимые в режиме реального времени и подготовленные заранее (изображения). Схема этапов процесса распознавания показана на рис. 1.

Сегментация символов

Сегментация символов подразумевает составление групп из всех физически разделённых компонентов математического выражения. В то же время некоторые буквы и символы, такие как «i», «j» и «=», состоят из нескольких компонентов. В результате необходимо объединять соответствующие компоненты, прежде чем распознавать отдельные буквы и символы. Также некоторые символы, такие как «√» и «Σ», обычно содержат другие символы внутри себя как подвыражения. Необходимо принимать особые меры при сегментации таких символов.

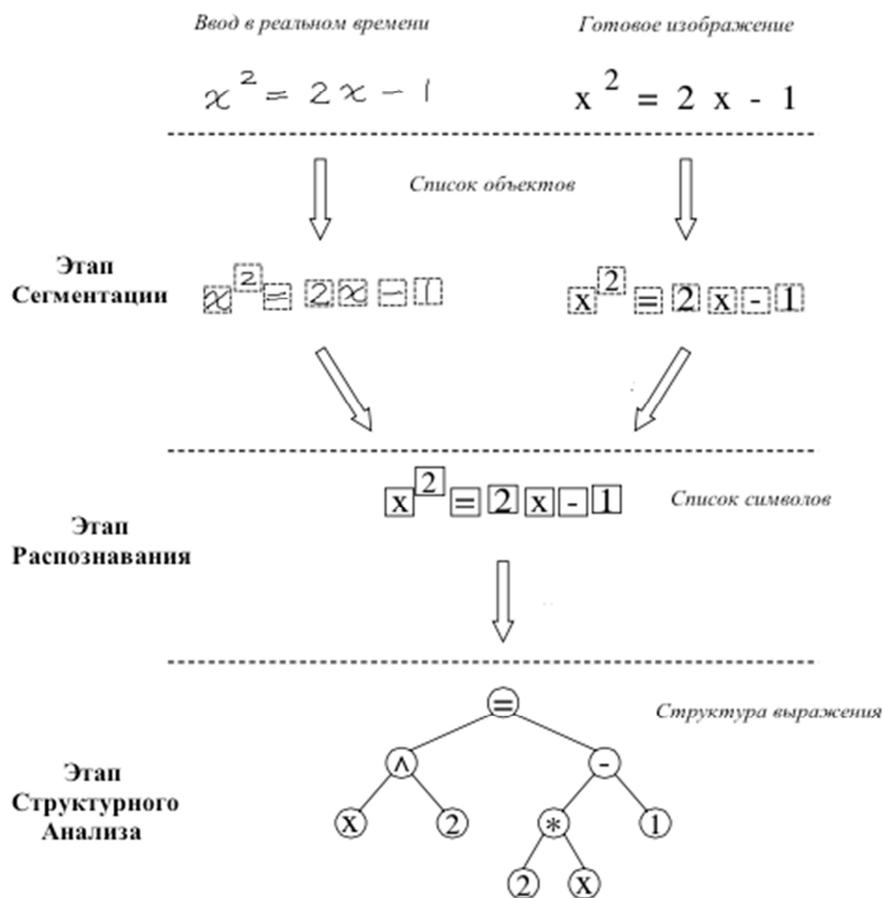


Рис. 1. Этапы распознавания математического выражения

Анализ существующих методов сегментации

Существует метод модульной сегментации рукописных математических выражений, описанный в [1]. Этот метод состоит из двух модулей сегментации. Модуль перемещения данных сначала строит дерево отношений по исходному выражению. Проекция на оси X и Y используются для того, чтобы решить, как сегментировать данные из исходного выражения. Но такой подход применительно к таким символам, как « $\sqrt{\quad}$ » и обыкновенная дробь, зачастую даёт неправильный результат. Для снижения вероятности такой ошибки применяются дополнительные операции предварительной сегментации таких символов и их подвыражений, а также множество дополнительных проверок. Далее второй модуль преобразует дерево отношений, пытаясь соединить различные части вместе, например, в поисках таких символов, как « i », « j » и « $=$ ». Всё вышперечисленное говорит о сложности и малой эффективности данного метода.

В [8] описывается метод сегментации математического выражения путём рекурсивной нарезки выражения вертикальными и горизонтальными линиями. Этот метод также нуждается в большом количестве дополнительных проверок для символов,

состоящих из нескольких компонентов, а также для символов, содержащих подвыражения. Этот метод в основном используется для сегментации не рукописных математических выражений на заранее подготовленных изображениях. Применимо к рукописным математическим выражениям, вводимым в реальном времени, этот метод имеет очень низкую точность.

Метод «Рекурсивной X-Y нарезки» [4] основан на помещении каждого символа в ограничительные рамки. Он похож на предыдущий метод, но использует ограничивающие рамки для каждого символа вместо полного пиксельного выделения. Этот метод быстрее предыдущего, но крайне плохо работает с рукописными математическими выражениями, так как при таком вводе символы могут не иметь своей чёткой границы, а, например, касаться или частично закрывать другие символы.

Наилучшим методом в рамках данной задачи можно назвать метод «Прогрессивной группировки компонентов» [6, 11]. В данном методе этапы сегментации и распознавания работают вместе. Сначала формируются все возможные упорядоченные множества для небольшого набора росчерков. Затем все эти множества проходят через классификатор и предпочтение отдаётся тому множеству, у которого наименьший уровень ошибки по результатам классификации. Данный метод может выдавать неверный результат в том случае, когда компоненты математического выражения вводятся не последовательно, например, если ставить точку над символом «i» после написания символов, которые следуют за «i». Такая ошибка требует ручной доработки. Этот метод является наилучшим в рамках данной задачи, так как полностью использует преимущество того факта, что исходные данные вводятся в режиме реального времени и не требует большого числа дополнительных проверок в тех случаях, в которых остальные алгоритмы требуют.

Распознавание символов

По завершении стадии сегментации математического выражения мы имеем список объектов с некоторыми известными атрибутами, такими как размер и местоположение. Задача этапа распознавания — идентифицировать каждый выделенный на предыдущем этапе символ. Если же система подразумевает совместную работу этапа сегментации и распознавания, как описано в последнем проанализированном методе сегментации, то этап распознавания начинается после каждой итерации этапа сегментации.

Анализ существующих методов распознавания символов

Сегодня наиболее распространенными подходами к классификации рукописных символов являются алгоритмы сопоставления с образцом [7], скрытые Марковские модели [3, 5], алгоритмы на основе искусственных нейронных сетей [2]. В последнее время начинает активно использоваться аппарат искусственных нейронных сетей с целью решения различных задач классификации. Преимуществом искусственных нейронных сетей является их способность к обобщению полученной информации, таким образом обученная на ограниченном множестве выборочных данных нейронная сеть может вернуть верный результат применительно к данным, которые не участвовали в процессе обучения. Поскольку решение об идентификации рукописных символов принимается системой распознавания в условиях неполной и неточной информации, можно считать подходящим выбор метода распознавания на основе нечеткого нейронного классификатора [2], сочетающего в себе преимущества нейронных сетей и нечетких систем логического вывода.

Структурный анализ математических выражений

На входе этапа структурного анализа имеется список всех символов математического выражения, а также атрибуты, такие как размер и относительное местоположение, для каждого из них. Следующим шагом станет построение иерархической структуры из этих символов.

Анализ существующих методов структурного анализа математических выражений

В [9] описывается метод, основанный на двух синтаксических анализаторах, которые работают по принципу преобразования исходного выражения в более краткие подвыражения, путём сканирования его сверху вниз и снизу вверх. Синтаксический анализатор, работающий по принципу сверху вниз, делит исходное выражение на подвыражения, а второй анализатор после этого соединяет все подвыражения в одну большую структуру. Этот метод хорошо работает только с простыми арифметическими выражениями и тригонометрическими функциями, но в сложных математических выражениях данный метод часто неправильно определяет отношения между функциональными операторами и их подвыражениями.

Одним из наиболее точных методов структурного анализа можно назвать метод, основанный на базовых линиях символов [9]. Базовая линия является списком, который

содержит все атрибуты символа и таким образом представляет горизонтальное расположение символов относительно друг друга в исходном выражении. Каждый символ имеет ссылки на другие базовые линии. Основная идея метода заключается в использовании семантической информации символов для их группировки и построения дерева базовых линий, которое наилучшим образом отображает структуру математического выражения. Этот метод требует большого количества памяти для хранения всей информации, но при этом даёт высокую точность, что и делает его наилучшим в рамках данной задачи.

Описание полученного метода

В результате анализа и соединения компонентов был получен метод, описанный далее. Вначале необходимо выполнить предварительную обработку данных. Символ является упорядоченным набором росчерков, а росчерк является упорядоченным набором пар координат, где первая пара соответствует точке касания, а последняя – точке отрыва. Каждый символ рисуется определенным набором росчерков.

Все точки, из которых состоит росчерк, сдвигаются так, что начальная точка является верхним левым углом ограничивающей рамки символа. Далее все точки масштабируются с сохранением отношения ширины к высоте так, что символ ровно помещается внутри квадрата определённого размера. Далее, набор точек каждого росчерка изменяется так, чтобы точки располагались равноудаленно, количество точек должно быть фиксировано и равно $36/N$, где N – количество росчерков в символе, а 36 делится на 1, 2, 3 и 4, где 4 — максимальное количество росчерков в одном символе [10].

По завершении данного этапа имеется набор векторов одной размерности, где каждому классу символов соответствует определенное количество векторов. Эту информацию необходимо сначала использовать для обучения нейронной сети, а затем и для подготовки росчерков к распознаванию.

После того, как пройден этап предварительной обработки данных, можно начинать проводить этапы сегментации и распознавания. Классификатор распознает символы по отдельности. Однако, когда необходимо распознавать потоковый ввод, он не может заранее знать, какие росчерки составляют символ, и сколько символов всего введено. Предположим, что введенные росчерки упорядочены по времени, то есть при вводе символа, состоящего из трех росчерков, последовательно вводятся росчерки данного символа. Например, не допускается ставить точку над символом j после написания символов, которые следуют за j . С учетом данного предположения, для определения

символов, состоящих из нескольких росчерков, необходимо последовательно рассматривать группы росчерков размером от 1 до N. Если символ из нескольких росчерков распознается классификатором с ошибкой, меньшей определенного порога, то отдается предпочтение данному символу. В случае, если ни один из результатов классификации не превосходит порогового значения, пороговое значение увеличивается и шаг повторяется, либо возникает ошибка распознавания.

Также необходимо определить допустимые символы для каждого определенного количества росчерков. Так, согласно этому методу, два росчерка не могут быть распознаны как символ, состоящий из одного или трех росчерков. Диаграмма работы процесса сегментации представлена на рис. 2.



Рис. 2. Диаграмма работы процесса сегментации

На этом этапе могут возникать неоднозначности. Так, например, классификатор не может знать, ввели ли мы символ «минус» или символ «дробь». Для этого осуществляется дополнительная проверка. Их различие состоит в том, что на определенном расстоянии снизу и сверху от черты «минус» не будет расположено никаких символов, а у символа «дробь» будут. На основании этого производится переименование символа «черта» в символ «минус» или «дробь» соответственно.

По завершении этапов сегментации и распознавания отдельных символов наступает этап структурного анализа математического выражения.

Отношения символов и операторов определяются по позиции и относительному размеру символа в исходном выражении. Для определения этих отношений используются пространственные области «верхний левый индекс», «сверху», «верхний индекс», «нижний индекс», «снизу», «снизу слева» и «подвыражение». Например, ожидается, что подвыражения числитель и знаменатель горизонтальной линии оператора дроби будут лежать в областях «сверху» и «снизу» относительно горизонтальной линии дроби. Пространственные области символов представлены на рис. 3.

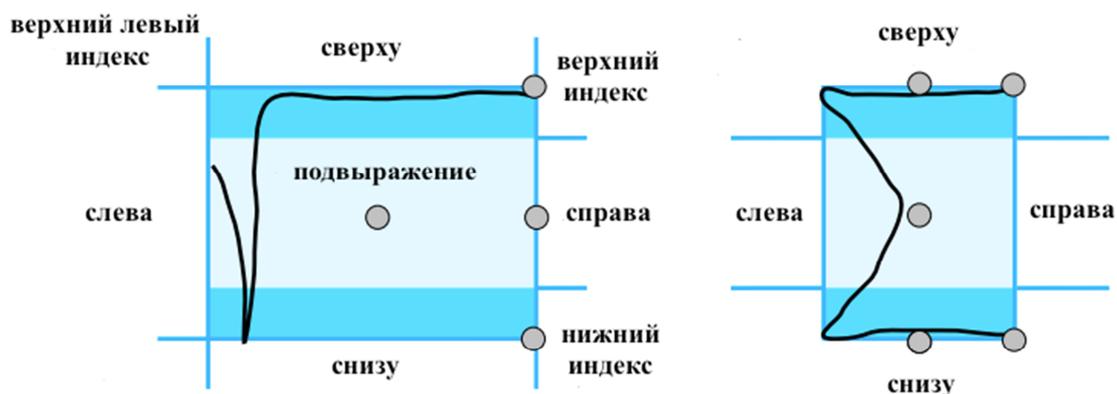


Рис. 3. Пространственные области символов

Пользуясь простыми атрибутами (левая граница, правая граница, и так далее), необходимо определить порог верхнего индекса, порог нижнего индекса и так далее. Данные атрибуты необходимы для того, чтобы определить все зоны вокруг символа. Также, исходя из этих атрибутов, необходимо определить центральную точку. Центральная точка — это точка, которая отвечает за расположение символа в какой-либо из областей. Для определения атрибутов символ классифицируется как с надстрочным элементом, с подстрочным элементом или центральный, как показано в таблице 1.

Таблица 1

Классификация различных классов символов

	Центральный	С надстрочным элементом	С подстрочным элементом
Без индексов	+ - * / (∞ =		
С верхним индексом	e π √ log sin cos tan	0 1 2 3 4 5 6 7 8 9	

С индексами	$a c x z)$	$b d \partial \Delta$	$g y$
Типа «Сумма»	$\Sigma \Pi \int$		
Типа «Предел»	$\lim \max \min$		

Каждый класс символов имеет по-разному определяемые атрибуты, как показано в таблице 2, где x и y – координаты центра символа, а H и W – высота и ширина символа соответственно. Это помогает избегать неоднозначностей при анализе структуры.

Базовая линия — это список, который представляет горизонтальное расположение символов в выражении [10]. Каждый символ имеет ссылки на другие базовые линии, которые удовлетворяют различным пространственным отношениям.

Таблица 2

Атрибуты различных классов символов

	Верхний порог	Нижний порог	Центроида
Символ с надстрочным элементом	$y + 0.8H$	$y + 0.2H$	$x + 0.5W$ $y + 0.33H$
Символ с подстрочным элементом	$y + 0.9H$	$y + 0.6H$	$x + 0.5W$ $y + 0.6H$
Центральный символ	$y + 0.8H$	$y + 0.2H$	$x + 0.5W$ $y + 0.5H$

Доминирование определяется следующим образом: символ A доминирует над символом B , если B лежит в области A , а A не лежит в области B . Однако оба символа могут лежать в областях друг друга. Тогда необходимо вычислить функцию доминирования следующим образом (каждый следующий шаг выполняется, если на предыдущем не удалось определить относительное доминирование символов):

1. Сравнение относительного местоположения символов.
2. Проверка идентификации символов с целью выявить явное доминирование.
3. Сравнение размеров символов относительно друг друга.

Доминирующая базовая линия — это базовая линия, на которую не ссылается никакой символ. Во время анализа математического выражения сначала производится поиск крайнего левого доминирующего символа, далее всех оставшихся доминирующих символов базовой линии.

Структура данных, которая представляет выражение, называется деревом базовых линий. На рисунке 4 представлено математическое выражение и дерево базовых линий для него.

$$x_{ij} \cdot y + \frac{a+b}{c}$$

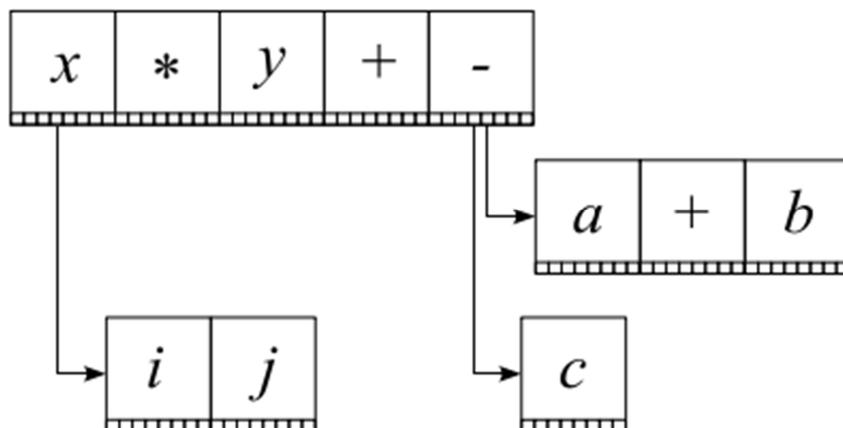


Рис. 4. Дерево базовых линий математического выражения

Полученное дерево представляет собой структуру математического выражения, которую можно преобразовать в любой нужный вид, например, в $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ или MathML.

Заключение

В заключение стоит добавить, что данный метод можно улучшить на этапе сегментации. Суть в том, что в данном методе очень важна последовательность введения росчерков. Если росчерки для какого-либо символа ввести не последовательно, велика вероятность ошибки при распознавании. Эту ошибку можно исправить путём игнорирования последовательности введения росчерков и вместо этого группировки росчерков по относительным пространственным координатам. Также для увеличения точности сегментации можно группировать больше символов, чем максимальное количество росчерков в символах, с учётом пространственных координат. Это немного понизит быстродействие метода, но в разы повысит точность сегментации и распознавания, покрывая тем самым множество нестандартных подходов пользователей к написанию символов и всего математического выражения в целом. К тому же, современные вычислительные системы настолько быстрые и мощные, что повышение

числа обрабатываемых групп самым незначительным образом скажется на быстродействии метода.

Список литературы

1. Belaid A., Haton J.P. A syntactic approach for handwritten mathematical formula recognition. IEEE Trans. Pattern Anal. Mach. Intel, 1984. P. 105–111.
2. Dimitriadis Y. A new interactive mathematical editor, using on-line handwritten symbol recognition, and error detection-correction with an attribute grammar. ICDAR, 1991. P. 885-893.
3. Faure C., Wang Z.X. Automatic perception of the structure of handwritten mathematical expressions. Computer Processing of Handwriting, World Scientific, 1990. P. 337–361.
4. Ha J., Haralick R.M., Phillips I.T. Understanding mathematical expressions from document images. ICDAR, 1995. P. 956–959.
5. Kosmala A., Rigoll G. On-line handwritten formula recognition using statistical methods. International Conference on Pattern Recognition, 1998. P. 1306-1308. DOI: 10.1109/icpr.1998.711941.
6. Matsakis N. Recognition of handwritten mathematical expressions. Cambridge: Massachusetts Institute of Technology, 1999. 59 p.
7. Nakayama Y. A prototype pen-input mathematical formula editor. EDMEDIA, 1993. P. 283-407.
8. Okamoto M., Miao B. Recognition of mathematical expressions by using the layout structures of symbols. ICDAR, 1991, pp. 242–250. DOI: 10.1109/icassp.1995.480046
9. Smithies S., Novins K., Arvo J. A handwriting-based equation editor. Graphics Interface, 1999. P. 84–91.
10. Winkler H.J., Fahrner H., Lang M. A soft-decision approach for structural analysis of handwritten mathematical expressions. ICASSP, 1995. P. 2459-2462.
11. Zanibbi R., Blostein D., Cordy J. Recognizing Mathematical Expressions Using Tree Transformation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002. P. 195-213. DOI: 10.1109/tpami.2002.1046157.