Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование
информационных систем

Системное программирование

Лунина Полина Сергеевна

Комбинирование нейронных сетей и синтаксического анализа для обработки вторичной структуры последовательностей

Дипломная работа

Научный руководитель:
к. ф.-м. н., доцент Григорьев С. В.

Рецензент:

Санкт-Петербург
2019

SAINT-PETERSBURG STATE UNIVERSITY

Software and Administration of Information Systems

Software Engineering

Polina Lunina

The composition of neural networks and parsing for secondary structure processing

Graduation Thesis

Scientific supervisor:
Assistant Professor Semyon Grigorev

Reviewer:

Saint-Petersburg
2019

Оглавление

Введение

В совершенно разных предметных областях встречаются концептуально схожие задачи, связанные с анализом различных символьных цепочек. Например, распознавание и классификация геномных последовательностей в биоинформатике или поиск аномалий в цепочках системных вызовов в компьютерной безопасности. Часто оказывается, что исследуемые последовательности обладают достаточно специфической синтаксической структурой и, учитывая каким-либо способом ее особенности при разработке алгоритмов для решения различных задач, можно значительно повысить их точность и эффективность. В биологических терминах синтаксическая структура геномных последовательностей называется вторичной, и далее мы будем использовать этот термин применительно также и к другим областям.

Одним из классических способов описания вторичной структуры являются формальные грамматики, в частности, контекстно-свободные. Они обладают широкими выразительными возможностями и позволяют описать связь между символами, находящимися на большом расстоянии. Например, как показали исследования в области биоинформатики, с помощью вероятностных грамматик можно смоделировать синтаксическую структуру всей цепочки. Тем не менее, в общем случае создание такой грамматики — достаточно сложная, а иногда и невозможная задача. Поэтому имеет смысл использовать более простую грамматику для описания только ключевых особенностей вторичной структуры, а для ее полноценного анализа применять другие методы.

Существенной проблемой при работе с реальными данными является возможное присутствие различного рода шумов, мутаций и случайных всплесков, что делает точные методы неприменимыми. Распространенный способ учесть эту проблему — использование методов машинного обучения, в особенности, искусственных нейронных сетей. Кроме того, нейронные сети предоставляют возможность эффективно находить сложные и не поддающиеся формализации структурные закономерности во входных данных.

В данной работе мы предлагаем новый подход для класса проблем, связанных с обработкой символьных данных, обладающих некоторой синтаксической структурой. Основная идея подхода — комбинация методов синтаксического анализа и машинного обучения. Мы используем грамматику для описания основных особенностей синтаксической структуры, извлекаем эти особенности с помощью алгоритмов синтаксического анализа, преобразуем полученные данные в удобный формат и используем в качестве входных данных для нейронной сети, сконструированной и обученной для решения конкретной задачи. Мы предоставляем экспериментальные исследования предложенного метода на некоторых биологических задачах: распознавание 16s рРНК и классификация тРНК. Полученные результаты показывают применимость предложенного подхода к реальным исследовательским областям. Исходный код и докуметация доступны по ссылке https://github.com/LuninaPolina/diplo

1. Постановка задачи

Целью данной работы является разработка подхода для анализа вторичной структуры последовательностей с использованием комбинации синтаксического анализа и нейроннх сетей.

Для достижения данной цели в рамках работы были поставлены следующие задачи.

- Разработать архитектуру решения.
- Провести экспериментальные исследования.
- Создать документацию.

2. Обзор областей применения

2.1. Биоинформатика

Одной из областей, где необходим анализ большого количества символьных данных, является биоинформатика. Точные и эффективные методы для решения таких задач, как распознавание и классификация организмов по их генетическим данным, предсказание функций и вторичной структуры белков, аннотация геномов и т.п. стали ключевыми направлениями в современной вычислительной геномике (протеомике).

Материалом для изучения являются нуклеотидные (или, в случае белков, аминокислотные) последовательности, определенные участки которых соединяются между собой по определенным закономерностям, образуя сложную и стабильную вторичную структуру (рис. 1).

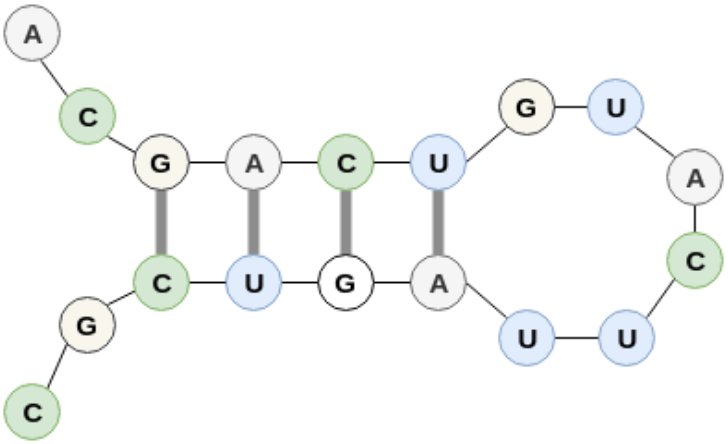

Рис 1: Образоваие вторичной структуры рнк

Идея о том, что именно особенности вторичной структуры генетических цепочек существенны для решения задач распознавания и классификации, описана в различных научных работах и широко используется на практике [11, 14]. Существующие способы описания и моделирования вторичной структуры используют в основе различные подходы и алгоритмы, такие как скрытые марковские модели, ковариационные модели [2] и формальные грамматики [12, 10, 3]. Распространенные проблемы при реализации данных подходов заключаются в больших вре-

менных затратах и сложности создания точной грамматики или модели. Тем не менее, они успешно применяются на практике для создания различных инструментов [5, 8].

Концептуально другим подходом к решению задач биоинформатики является использование методов машинного обучения [13, 6]. Они позволяют находить сложные закономерности в больших объемах данных и учитывать характерную для биологических данных зашумленность.

Предложенный в данной работе подход позволяет, во-первых, совместить некоторые преимущества подходов, основанных на задании вторичной структуры, и на машинном обучении, а во-вторых, повысить производительность этапа синтаксического анализа относительно классических способов, так как описание только харатерных особенностей вторичной структуры вместо моделирования ее для всей цепочки позволит существенно сократить размер грамматики и,как следствие,время затраченное на работу парсера.

2.2. Компьютерная безопасность

Еще одной потенциальной областью применения предложенного подхода является компьютерная безопасность. Одна из самых острых проблем в данной области — борьба с вредоносными программами. Для обнаружения их воздействия на систему применяют различные подходы, заключающиеся в поиске аномалий в последовательностях системных вызовов (трассах), совершенных другими программами [7, 15, 4]. Для этого нужно реализовать способ идентификации процессов, т.е. по некоторому набору особенностей научиться различать трассы различных программ и выявлять вирусы и отклонения.

Трассы представляют из себя некоторые последовательности символов, в которых присутствуют закономерности, характерные для определенных видов программ, следовательно, формальное описание этих закономерностей может оказаться полезным при исследовании системных аномалий.

В работах [17, 18] описан алгоритм обнаружения процессов вредо-

носных программ, основанный на поиске в трассах некоторых характерных шаблонов, по которым для набора процессов строятся описывающие их модели. Для каждого нового процесса проводится специальная оценка и подбирается наиболее близкая модель из существующих. Затем составляется вектор характеристик, оценивающих поведение процесса в рамках модели и подобные вектора используются для обучения нейронной сети, осуществляющей бинарную классификацию: процессы легитимных и вредоносных программ.

Предложенный в данной работе подход может оказаться применимым в данной области, так как TODO

3. Разработка архитектуры решения

В данном разделе сначала представлена общая схема архитектуры решения, а затем детальное описаны все части предложенного подхода.

3.1. Описание предложенного подхода

Предложенный в данной работе подход может использоваться для решения различных задач во многих исследовательских областях. Ограничения, накладываемые на потенциальную область для апробации подхода следующие. Во-первых, исследуемые данные — некоторый набор символьных последовательностей с метаданными, для которых нужно решить задачу классификации по каким-либо признакам. Во-вторых, на основе анализа специфики области исследования и визуального изучения некоторого подмножества последовательностей можно выделить некоторые характерные шаблоны и закономерности в их образовании, т.е. синтаксическую структуру.

Процесс проведения эксперимента выглядит следующим образом. Сначала создается грамматика, описывающая характерные особенности вторичной структуры рассматриваемых последовательностей. Затем эти особенности извлекаются путем применения некоторого алгоритма синтаксического анализа ко входным данным по заданной грамматике. Полученные в результате работы синтаксического анализатора матрицы разбора приводятся к удобному для дальнейшей обработки виду и подаются на вход нейронной сети, осуществляющей классификацию в соответствии с условиями поставленной задачи. Кроме того, в процессе работы могут понадобиться некоторые дополнительные действия по обработке данных, уникальные для конкретного эксперимента, например, составление выборки, фильтрация и т.п.

Для удобства использования предложенного подхода на практике было необходимо унифицировать и задокументировать все шаги от начальной обработки данных до фиксации результатов. Архитектура решения представлена на рис. 2 и состоит из следующих частей.

- Грамматика в формате yard или аналогичном
- База данных, хранящая сами входные последовательности, метаданные, результат парсинга, различные промежуточные данные и т.д.
- Parsing Tool — утилита для обработки данных синтаксическим анализатором с возможностью сохранения результата в различные форматы
- Neural Networks — модуль для обучения и тестирования нейронных сетей
- Data Processing — модуль для промежуточной обработки данных

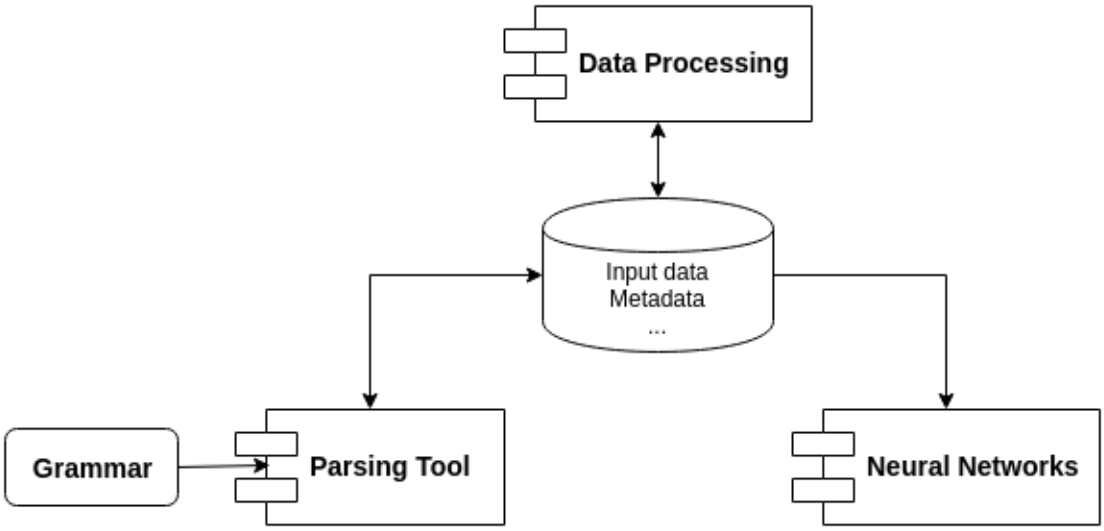

Рис 2: Архитектура решения

В следующих разделах будут детально описаны все шаги по использованию предложенного подхода.

3.2. Подготовка входных данных

В данном разделе описаны необходимые для использования шредложенного подхода данные, а также детали их хранения и представления.

В процессе апробации подхода были выведены некоторые общие требования, позволяющие наиболее эффективно и удобно проводить экспериментальные исследования.

Все необходимые данные помещаются в некоторое облачное хранилище, откуда могут извлекаться по мере необходимости. Входные данные — исследуемые цепочки — последовательно записаны в файлы формата fasta (для, например, биологических данных) или любого аналогичного текстового формата, причем каждой цепочке соответствует уникальный числовой идентификатор. Вся остальная информация о цепочках (метаданные, принадлежность к тестовой или обучающей выборке, класс и др.) хранится в специальной ссылочной таблице с доступом по идентификатору.

Кроме того, в процессе исследования могут понадобиться специфические скрипты для промежуточной обработки данных, например, выборка для обучения нейронной сети, изменение длин цепочек, фильтрация входных данных и др. Эти скрипты реализуются в модуле Data Processing и могут быть переиспользованы в похожих экспериментах.

3.3. Генерация данных с помощью синтаксического анализатора

В рамках предложенной архитектуры синтаксический анализатор — консольная утилита, принимающая на вход грамматику, файл с цепочками и список желаемых выходных форматов. Опишем основные принципы его работы.

Синтаксический анализ – процесс проверки выводимости некоторой подстроки в заданной грамматике. В контексте предложенного решения терминальными символами грамматики являются символы исследуемых последовательностей, правила грамматики описывают характерные особенности их вторичной структуры, а алгоритм синтаксического анализа используется для извлечения этих особенностей путем поиска всех выводимых подстрок для данной строки для всех нетерминалов. Наш подход не зависит от выбора конкретного алгоритма синтаксиче-

ского анализа, однако в описанных ниже экспериментах мы предлагаем использовать разработанный в рамках проекта YaccConstructor [16] в лаборатории JetBrains [9] алгоритм, основанный на матричных операциях [1], который демонстрирует высокую производительность на практике в связи с использование параллельных вычислений.

Результатом работы матричного синтаксического анализатора для строки и фиксированного нетерминала является верхнетреугольная битовая матрица разбора. Мы предлагаем использовать такие матрицы как входные данные нейронной сети, поэтому необходимо привести их к удобному для обработки формату, учитывая специфику конкретной задачи. На данный момент в рамках расширения используемого алгоритма реализованы два формата преобразования матриц.

- Вектора характеристик, сохраняемые в виде строк csv-файла. Генерация векторов осуществляется следующим образом: отбрасывается пустая часть матрицы ниже главной диагонали, оставшиеся строки последовательно преобразовываются в битовый вектор, а затем сжимаются в байтовый вектор.

- Черно-белые изображения в формате bmp, получаемые путем замены нулевых битов матрицы на белые пиксели, а единичных — на черные.

Кроме того, в исходный код инструмента можно легко добавить другие выходные форматы.

Классификация как векторов характеристик, так и изображений относится к классическим сценариям использования нейронных сетей, однако стоит отметить, что ввиду специфики конкретной задачи и особенностей входных цепочек, выбор формата данных может повлиять на эффективность и скорость обучения. Например, в процессе экспериментальных исследований было обнаружено, что несмотря на то, что векторные данные занимают меньше памяти и ускоряют процесс обучения, их использование предполагает выравнивание всех цепочек до одинаковой длины, что может оказаться не самым эффективным решением для задач, где входные последовательности имеют принципиально

разные длины, так как большая часть вектора для коротких цепочек в данном случае будет заполнена незначащими нулями. Изображения же можно сгенерировать для цепочек разных длин и затем привести к одному разрешению, что позволяет аккуратно сохранить особенности вторичной структуры даже для коротких цепочек, однако это приводит к ухудшению скорости обучения и вызывает необходимость хранения больших объемов данных.

3.4. Обучение нейронных сетей

Искуственные нейронные сети – широко применямый метод решения задач классификации в областях, где входные данные обладают сложно формализуемыми закономерностями и могут содержать шумы и неточности. Мы предлагаем использовать нейронные сети для обработки сгенерированных синтаксическим анализатором данных, предполагая, что в них закодированы существенные для классификации особенности синтаксической структуры.

Архитектура нейронной сети уникальна для каждой конкретной задачи и предметной области, однако экспериментальные исследования выявили некоторые общие закономерности и интуиции. Для векторизованных данных высокую эффективность показало чередование полносвязных (dense) слоев ввиду утери информации о взаимном расположении элементов изначальной битовой матрицы и дропаут (dropout) слоев с нормализацией для разжатия данных. Для изображений мы предлагаем использовать небольшое количество сверточных слоев (так как они применяются в основном для извлечения каких-либо особенностей из входных данных, а в нашем случае это уже сделано на этапе синтаксического анализа), затем линеаризацию, а далее перейти к чередующимся dense и dropout слоям, аналогично архитектуре для веторизованных данных.

Выше был описан стандартный в рамках нашего подхода способ использования нейронных сетей, однако возможны различные его модификации, основанные на констуировании более сложных моделей с

загрузкой весов уже обученных, что позволяет упростить задачу или повысить точность результата. Например, расширение нижней части нейронной сети слоями с большим количеством нейронов позволяет провести классификацию на большее количество классов, а расширение верхней части предоставляет возможность подачи на вход данных в другом формате, например, изначальной символьной последовательности вместо результата парсинга.

Остановимся подробнее на последней модификации. Большинство алгоритмов синтаксического анализа работают за полином от длины входа, поэтому генерация большого количества данных на длинных цепочках потребует существенных временных затрат. Поэтому мы предлагаем следующую идею.

- Сгенерировать некоторый набор данных с помощью синтаксического анализатора и обучить на них нейронную сеть (NN1).
- Создать новую нейронную сеть (NN2), которая расширяет вход NN1 несколькими слоями, верхний из которых принимает символьные цепочки.
- Подгрузить веса NN1 на нижнюю часть NN2 и дообучить NN2.

Эта идея может быть применена как к векторизованным данным, так и к изображениям. В случае изображений необходимо использовать веса NN1, начиная с линеаризованного слоя. Таким образом можно, во-первых, уменьшить размер выборки для генерации парсером, а во-вторых, улучшить точность уже обученных нейронных сетей без временных затрат на дополнительную генерацию данных. Высокая точность и скорость обучения такой нейронной сети была подтверждена экспериментальным путем.

Заключение

В ходе данной работы были получены следующие результаты.

- Раработана архитектура решения для использования предложенного подхода.
- Проведены экспериментальные исследования предложенного подхода на задачах распознавания 16s рРНК и классификации тРНК.
- Задокументировано описание схемы работы подхода и результаты проведенных экспериментов.
- Опубликована статья "The Composition of Dense Neural Networks and Formal Grammars for Secondary Structure Analysis" на конференции BIOINFORMATICS 2019
- Представлен постер "16s rRNA Detection by Using Neural Networks" на конференции Biata 2018

Существует несколько направлений дальнейшего развития полученных результатов.

- Эксперименты в области кибербезопасности — поиск анмалий в последовательностях системных вызовов.
- Предсказание функций белков.
- Распознавание химер.
- Моделирование вторичной структуры геномных пследовательностей
- Рибозимы???

Список литературы

[1] Azimov Rustam, Grigorev Semyon. Context-free Path Querying by Matrix Multiplication // Proceedings of the 1st ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA). — GRADES-NDA '18. — New York, NY, USA : ACM, 2018. — P. 5:1–5:10. — Access mode: http://doi.acm.org/10.1145/3210259.3210264.

[2] Biological sequence analysis: probabilistic models of proteins and nucleic acids / Richard Durbin, Sean R Eddy, Anders Krogh, Graeme Mitchison. — Cambridge university press, 1998.

[3] Dowell Robin D, Eddy Sean R. Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction // BMC bioinformatics. — 2004. — Vol. 5, no. 1. — P. 71.

[4] Ghosh Anup K, Schwartzbard Aaron. A Study in Using Neural Networks for Anomaly and Misuse Detection. // USENIX security symposium. — Vol. 99. — 1999. — P. 12.

[5] HMMER [Электронный ресурс]. — Access mode: http://hmmer.org/ (online; accessed: 05.05.2019).

[6] Higashi Susan, Hungria Mariangela, Brunetto MADC. Bacteria classification based on 16S ribosomal gene using artificial neural networks // Proceedings of the 8th WSEAS International Conference on Computational intelligence, man-machine systems and cybernetics. — 2009. — P. 86–91.

[7] Hofmeyr Steven A, Forrest Stephanie, Somayaji Anil. Intrusion detection using sequences of system calls // Journal of computer security. — 1998. — Vol. 6, no. 3. — P. 151–180.

[8] Infernal [Электронный ресурс]. — Access mode: http://eddylab.org/infernal/ (online; accessed: 05.05.2019).

[9] JetBrains Programming Languages and Tools Lab [Электронный ресурс]. — Access mode: `https://research.jetbrains.org/groups/plt_lab` (online; accessed: 05.05.2019).

[10] Knudsen Bjarne, Hein Jotun. RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. // Bioinformatics (Oxford, England). — 1999. — Vol. 15, no. 6. — P. 446–454.

[11] RNAscClust: clustering RNA sequences using structure conservation and graph based motifs / Milad Miladi, Alexander Junge, Fabrizio Costa et al. // Bioinformatics. — 2017. — Vol. 33, no. 14. — P. 2089–2096.

[12] Rivas Elena, Eddy Sean R. The language of RNA: a formal grammar that includes pseudoknots // Bioinformatics. — 2000. — Vol. 16, no. 4. — P. 334–340.

[13] Sherman Douglas. Humidor: Microbial Community Classification of the 16S Gene by Training CIGAR Strings with Convolutional Neural Networks. — 2017.

[14] Variation in secondary structure of the 16S rRNA molecule in cyanobacteria with implications for phylogenetic analysis / Klára Řeháková, Jeffrey R Johansen, Mary B Bowen et al. // Fottea. — 2014. — Vol. 14. — P. 161–178.

[15] Wespi Andreas, Dacier Marc, Debar Hervé. Intrusion detection using variable-length audit trail patterns // International Workshop on Recent Advances in Intrusion Detection / Springer. — 2000. — P. 110–129.

[16] YaccConstructor [Электронный ресурс]. — Access mode: `https://github.com/YaccConstructor` (online; accessed: 05.05.2019).

[17] Баклановский Максим Викторович, Ханов Артур Рафаэльевич.

Поведенческая идентификация программ // Моделирование и анализ информационных систем. — 2015. — Vol. 21, no. 6. — P. 120–130.

[18] Оценка точности алгоритма распознавания вредоносных программ на основе поиска аномалий в работе процессов / МВ Баклановский, АР Ханов, КМ Комаров, ПА Лозов // Научно-технический вестник информационных технологий, механики и оптики. — 2016. — Vol. 16, no. 5.