Комариця Р.В.
студент,
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського

ЗАСТОСУВАННЯ РОЗПОДІЛЕНИХ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ РОЗВ'ЯЗКУ ЗАДАЧІ РОЗПІЗНАВАННЯ КОДУЮЧИХ ДІЛЯНОК ГЕНІВ

Однією з ключових задач біоінформатики є задача розпізнавання білок-кодуючих областей ДНК (екзонів) які в свою чергу розділені сегментами некодуючої ДНК (інтронами). З математичної точки зору поставлена проблема відноситься до задач розпізнавання, алгоритмів вирішення яких є доволі багато, більшість з яких базуються на методах машинного навчання. Та основна проблема полягає в тому, що існуючі об'єми даних неможливо обробити класичними методами машинного навчання, особливо враховуючи обмеження обчислювальних потужностей. Незважаючи на те, що розподілені обчислення відкрили нові шляхи для вирішення задач, які потребують великих обчислювальних потужностей, а стрімкий ріст даних обумовив практично повсюдне використання розподілених обчислень, їх використання в зв'язці з методами машинного навчання досі залишається відносно новою і мало дослідженою задачею. Розпаралелювання обчислень та використання обчислювальних кластерів для вирішення подібного роду задач дозволить значно підвищити швидкість обробки даних.

В даній роботі подається метод вирішення поставленої задачі за допомогою наївного баєсівського класифікатора реалізованого на базі Apache Spark, що забезпечує можливість розгортання на кластері та горизонтального масштабування.

В задачі розпізнавання інтронів та екзонів на заданій ділянці гену задана послідовність символів S окремі елементи якої $S_i, i = 1,.n$ належать скінченному алфавіту R_s [1]. В випадку розпізнавання фрагментів гену, в ролі символів виступать нуклеотиди чотирьох типів: A, C, G, T.

Звідси випливає визначення R_s:

$$R_s = \{A, C, G, T\}. \quad (1)$$

Кожному спостережуваному символу $S_i \in R_s$ ставиться у відповідність один прихований стан H_i з скінченної множини R_h. Для задачі розпізнавання фрагментів генів:

$R_h = \{E, I\}$, (2)

де E – нуклеотиди, що входять в склад екзона, а I– нуклеотиди, що відповідають інтронам. На основі введених позначень можна сформулювати наступну задачу.

Знайти алгоритм $A: R_S^* \to R_h^*$, який довільній послідовності спостережуваних символів $S \in R_s^n$ ставить у відповідність ланцюг прихованих станів такої ж довжини $H \in R_h^n$ згідно з визначеним критерієм якості L

$$A(S) = \arg max_H L(S, H). \quad (3)$$

1. Модель наївного баєсівського класифікатора.

Нехай змінна може приймати значення з деякої множини значень V. Змінна x описується змінними $(a_1, a_2, \dots, a_n)$ – атрибутами. Необхідно знайти найбільш ймовірне значення x, тобто:

$$v_0 = \arg max_{v \in V} P(x = v | a_1, a_2, \dots, a_n). \quad (4)$$

За теоремою Баєса:

$$v_0 = \arg max_{v \in V} \frac{P(a_1, a_2, \dots, a_n | x = v) P(x = v)}{P(a_1, a_2, \dots, a_n)}. \quad (5)$$

Оцінити $P(x = v)$ легко (при наявності навіть невеликого числа тестових даних можна оцінити частоту появи кожного іх них). Але оцінити різні $P(a_1, a_2, \dots, a_n | x = v)$ неможливо, оскільки їх кількість є занадто великою. Для того, щоб отримати кожну із цих ймовірностей, потрібно кожну комбінацію атрибутів спостерігати декілька разів. Це не можливо на практиці. Тому наївний баєсівський класифікатор передбачає умовну незалежність атрибутів при умові даного значеня цільової функції:

$$P(a_1, a_2, \dots, a_n | x = v) = P(a_1 | x = v) \times$$
$$\times P(a_2 | x = v) \times \dots \times P(a_n | x = v) \quad (6)$$

Не дискретні атрибути повинні бути спочатку дискретизовані. Якщо даний клас і значення атрибута ніколи не зустрічались разом в наборі навчальних даних, тоді оцінка, яка базується на ймовірностях буде рівна нулю

Не зважаючи на простоту та наївність баєсівського класифікатора, неодноразово було показано його ефективність навіть при роботі з складними даними [2].

2. Алгоритм розпізнавання екзонів.

Spark API надає розробнику можливість створювати та керувати конвеєром машинного навчання, при цьому немає необхідності зосереджувати свою увагу на інфраструктурі проекту та розподілених обчисленнях. Всю роботу, що пов’язана з розподіленими обчисленнями на себе бере RDD (Resilient Distributed Dataset), що в свою чергу являє собою розподілену таблицю. RDD підтримує широкий набір вбудованих функцій які дозволяють трансформувати дані (наприклад: селекція, фільтрація, об’єднання, перетин, трансформаціяб і т.д.). Крім цього важливою особливістю є можливість об’єднувати виклики функції і будувати ланцюг. Результатом виклику RDD є нова RDD.

Процесс машинного навчання часто представляє собою послідовність певних кроків (наприклад обробка даних, навчання системи, конвертація типів і тд) [3]. Конвеєр машинного навчання, що представлений в Spark API, також являє собою набір кроків. Кожен з кроків може бути представленим однією з двох сущностей – Transformer або Estimator.

Transformer – абстракція що включає в себе трансформатор ознак та моделі для навчання. З технічної точки зору Transformer реалізовує метод transform() який відповідає за певне перетворення поточних даних (поточні дані мають бути представлені у форматі DataFrame – що по суті являє собою динамічну таблицю з колонками). Перетворення даних може включати в себе будь-які операції над даними включаючи агрегацію, фільтрацію, сортування, тощо.

Результат перетворень зазвичай зберігається в вигляді колонок, що додаються до вже існуючих в поточному DataFrame.

Estimator абстрагує концепт алгоритму навчання чи будь-якого іншого алгоритму, що пристовоується або тренується на вхідних даних. З технічної точки зору Estimator реалізовуває метод fit(), який приймає DataFrame, та створює модель, що в свою чергу являється Transformer.

Етапи створення моделі наївного баєсівського класифікатора, необхідного для вирішення задачі розпізнавання, показано на рисунку 1.

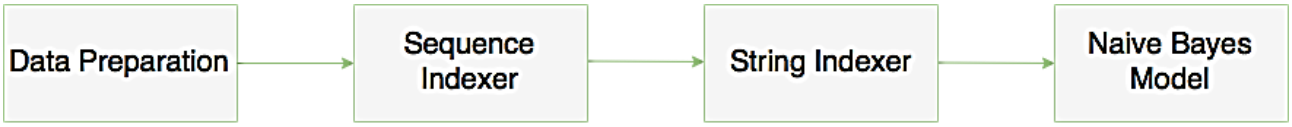

Рис. 1. Модель наївного баєсівськго класифікатора

Джерело: розроблено автором

Data Preparation (Transformer) – зчитування та обробка початкових даних (зчитування файлу з екзонами і генами та їх агрегація). Зазвичай даний крок являється першим при побудові конвеєру.

String Indexer (Transformer) – кодує колонку ярликів (labels), що задані у вигляді текстових рядків в колонку індексів даних ярликів [4]. Дані індекси сортуються за частотою використання ярлика (так наприклад найбільш вживаному ярлику буде присвоєно індекс 0). Якщо ярлики буду представлені у вигляді числових значень, то кожен ярлик буде преведено до текстового типу і після цього проіндексовано.

Sequence Indexer (Transformer) – виконує задачу аналогічну до String Indexer, але з однією відмінністю, а саме: значення вхідної колонки задається у вигляді масиву текстових рядків, а не окремого текстового рядку. Всі строкові значення мають належати одному спільному алфавіту.

Naive Bayes Model (Estimator) – наївний баєсівський класифікатор, який приймає на вхід колонку ярликів та колонку індексованих ознак. Результатом роботи класифікатора є нові колонки.

Розглянутий метод розпізнавання екзонів в ДНК з використанням наївного баєсівського класифікатора розробленого на базі технології Apache Spark, завдяки своїй архітектурі, та концепції конвеєра, забезпечує розподіленість обробки даних та надає можливість надзвичайно простого горизонтального масштабування системи в майбутньому, без необхідності у внесенні будь-яких змін.

Список використаних джерел:

1. Гупал А.М., Сергиенко И.В. Симметрия в ДНК. Методы распознавания дискркетных последовательностей. – Киев: Наукова думка, 2016. – 257 с.

2. Гупал А.М., Сергиенко И.В. Байесовская процедура – оптимальная процедура распознавания и преобразования // Пробл. упр. и информатики. – 2001. – № 3. – С. 5-15.

3. ML Pipelines: A New High-Level API for MLlib [Електронний ресурс] // Режим доступу: https://databricks.com/blog/2015/01/07/ml-pipelines-a-new-high-level-api-for-mllib.html

4. Extracting, transforming and selecting features [Електронний ресурс] // Режим доступу: https://spark.apache.org/docs/latest/ml-features.html