ИНТЕРНЕТ-СЕРВИС ИДЕНТИФИКАЦИИ ОБЪЕКТОВ ПО МИНИМАЛЬНО ВОЗМОЖНОМУ ЧИСЛУ ПРИЗНАКОВ: АЛГОРИТМЫ И РЕАЛИЗАЦИЯ

Н.В. Афанасенко, А.Т. Вахитов, А.Г. Кирейчук

Введение

Задача идентификации объектов по некоторому набору признаков встречается в жизни достаточно часто; например, при выявлении причин поломки автомобиля, при постановке предварительного диагноза по ряду симптомов, при определении вида животного или растения.

В общем виде эту задачу можно сформулировать так: имеется ряд признаков, каждый из которых может принимать конечное число состояний, имеется набор классов, каждый из которых описывает некоторое количество схожих между собой объектов с помощью их характеристик (то есть класс представляется как набор состояний, которые могут принимать признаки описываемых объектов), и по запросу, содержащему определенные исследователем состояния признаков (не обязательно всех), выявить, каким классам соответствует наблюдаемый объект.

Нередко при решении задачи идентификации мы имеем дело с большим количеством классов и еще большим (до нескольких сотен) количеством признаков. Определение состояний всех признаков при таком их количестве является для исследователя крайне трудоемкой задачей, поэтому важно предоставлять некоторый ранжированный по значимости список признаков, в порядке которого можно наиболее быстро определить класс объекта.

В последнее время к определителям биологических объектов проявляется все больший интерес как к способу развлечения. Подобная программа является одним из популярных приложений для Apple iPhone [1].

Также нельзя не отметить, что в наше время идентификация биологических объектов по ДНК становится все более доступной, но, тем не менее, тем же любителям птиц вряд ли будет доступен подобный подход, плюс, как было отмечено ранее, имеются родственные задачи из других областей, что позволяет нам говорить об актуальности решения данной задачи и в будущем.

Общие проблемы существующего ПО для идентификации в биологии

Определители таксономической принадлежности биологических объектов, ориентированные на персональные компьютеры, впервые появились в 1990-х годах. Обзор наиболее популярных, созданных с тех пор приложений, представлен в работе [2]. Как правило, каждая из этих систем ориентирована на определение объектов только конкретных семейств (например, орхидей или грибов-зигомицетов).

В целом представленные программные продукты обладают схожими функциональными возможностями, однако к их безусловным недостаткам можно в первую очередь отнести:

- Невозможность построения собственного определителя на базе предоставленной системы
- Жесткая привязка к интерфейсу пользователя
- Платформенные ограничения
- Отсутствие API для доступа к функциям определителя

Основные сложности процесса идентификации и пути их оптимизации

Типичный сценарий идентификации обычно он состоит из следующих этапов:

1. Вывод пользователю ранжированного списка признаков
2. Определение пользователем текущих состояний одного или нескольких признаков
3. Анализ ввода, ранжирование классов по мере соответствия их запросу, ранжирование набора еще не определенных признаков
4. Вывод пользователю нового списка признаков и подходящих под запрос классов

К основным характеристикам процесса поиска подходящих наблюдаемому объекту классов, требующим оптимизации в первую очередь, можно отнести следующие.

- **Минимизация средней длины пути определения.** Мы можем предлагать такой порядок признаков, при котором средние длины путей, приводящих к наиболее подходящим классам, минимальны.
- **Повышение надежности определения.** Например, особенно сложные для идентификации признаки (вероятность ошибки в определении которых высока), но сильно влияющие на конечный результат, следует ранжировать ниже более надежных. Данные для подобного улучшения надежности определения могут быть либо предоставлены экспертом, либо получены в результате наблюдения за рядом успешных определений.

Таким образом, помимо общих проблем архитектуры существующих на данный момент определителей, нам также необходимо решить задачу оптимизации процесса идентификации объекта по набору заранее определенных признаков.

Оптимизация процесса идентификации

Если бы мы для каждого класса могли посчитать минимальную по количеству элементов последовательность признаков, определением которых можно отделить данный класс от всех остальных, то в случае, когда наблюдаемый объект вероятно принадлежит данному классу, мы сможем предоставить исследователю оптимальный набор признаков, чтобы в этом убедиться. Для идентификации же произвольного объекта, ранг признаков, входящих в эти минимальные разделяющие последовательности, должен быть, очевидно, выше.

Проблема минимального набора признаков отделяющего класс от всех остальных

Для произвольной пары классов a и b обозначим набор признаков, отделяющих a от b как $c_1^{ab},\ldots,c_{k_{ab}}^{ab}$. Пусть всего у нас N классов. Рассмотрим булевскую формулу, значениями переменных c_m^{ij} которой являются истина в случае, если этот признак определен исследователем, и ложь в противном:

$$\left(c_1^{a1}\vee c_2^{a1}\vee\ldots\vee c_{k_{a1}}^{a1}\right)\wedge\left(c_1^{a2}\vee c_2^{a2}\vee\ldots\vee c_{k_{a2}}^{a2}\right)\wedge\ldots\wedge\left(c_1^{aN}\vee c_2^{aN}\vee\ldots\vee c_{k_{aN}}^{aN}\right) \quad (1)$$

Эта формула принимает значения истина, если при подстановке в переменные значений, соответствующих выбранным признакам, класс a отделим этими признаками от всех остальных. Отметим, что формула находится в конъюнктивной нормальной форме и не имеет отрицаний.

Задача заключается в поиске одного или нескольких минимальных по числу элементов наборов переменных, при присваивании которым значения истина, формула также принимает значение истина. Решив эту задачу для каждого класса, мы получим необходимые для ранжирующей метрики данные.

Необходимый набор переменных может быть найден преобразованием данной формулы в дизъюнктивную нормальную форму, и ответом будут являться минимальные по числу переменных конъюнкты. Эта задача NP-полна, решение ее для больших формул является крайне трудоемким, и вряд ли может быть использовано в нашем случае, поскольку подобные вычисления придется проводить довольно часто и на больших объемах данных.

В работе [3] предлагается использовать нейронную сеть для ранжирования признаков. На вход сети подается набор вероятностей того, что наблюдаемый объект принадлежит одному из классов, на выход сеть выдает ранги признаков, на основании которых строится предлагаемый исследователю набор. Обучать сеть предлагается на малых наборах таксонов, для которых можно решить задачу оптимальным способом.

В ходе работы было отмечено важное свойство, которым обладают формулы вида (1), построенные на основе тестовых баз данных жесткокрылых насекомых, – количество присваиваний, обращающих формулу в истину, составляет в среднем около 73% от всех возможных присваиваний (коих 2^n, где n – число переменных). Для подсчета был использован рандомизированный алгоритм, проверяющий не все 2^n вариантов, а только 10% из них. Мы можем предположить, что подобные значения будут получены и для других определителей, в которых у каждых двух классов достаточно велико количество разделяющих их признаков.

Это наблюдение позволяет нам использовать для решения задачи подход, наиболее соответствующий ее специфике, а именно алгоритм для формул с большим количеством удовлетворяющих присваиваний [4]. Алгоритм возвращает наименьшее по числу элементов присваивание (включая и присваивание переменным значений ложь). Нам же необходима минимизация по числу присваиваний значений истина, а также возможность получения нескольких наименьших присваиваний, поэтому мы будем использовать данный алгоритм с некоторой модификацией.

Пусть дана КНФ-формула f с переменными $a_{1,}a_{2,}\ldots,a_{n.}$ и K – число наименьших присваиваний, которое нужно получить.

1. $i:=0;\Theta:=\varnothing;\Phi_0:=\{f\};\ for\ m\in\{1,\ldots,n\}\ \Phi_m:=\varnothing;$
2. В каждой $f_{i,j}\in\Phi_i=\{f_{i,1},f_{i,2},\ldots,f_{i,q_i}\}$ выбираем наименьший по числу элементов дизъюнкт $l_1\vee l_2\vee\ldots\vee l_{s(i,j)}$
3. Для каждой $f_{i,j}$ получаем на ее основе новые формулы $g_{i,j,1},g_{i,j,2},\ldots,g_{i,j,s(i,j)}$ с помощью следующих присваиваний: $\{l_1\},\{\bar{l}_{1,}l_2\},\ldots,\{\bar{l}_{1,}\bar{l}_{2,}\bar{l}_{3,}\ldots,l_{s(i,j)}\}$, где l_n означает присваивание переменной l_n значения истина, а $\bar{l}_n$ означает присваивание переменной l_n значения ложь. Для всех формул, значение которых истина, добавляем $t(g_{i,j,m})$ в Θ, где $t(g_{i,j,m})$ – множество присваиваний значений истина, которыми из f была получена эта формула. Также проверяем, получили ли мы K присваиваний и возвращаем Θ, если это так.
4. Для всех формул $g_{i,j,m}$, значение которых не определено (то есть не истина и не ложь) добавляем их в $\Phi_{|t(g_{i,j,m})|}$.
5. $i:=i+1;$ Если $i>n$, возвращаем Θ, в противном случае переходим на шаг 1.

Как было отмечено ранее, признаки, входящие в такие наборы, возвращаемые представленным алгоритмом должны ранжироваться выше остальных. Используем эту идею для построения ранжирующей метрики.

Для случая, когда статистические данные еще не накоплены, и мы не имеем никаких экспертных оценок признаков, используем самый простой вариант метрики. В будущем, по мере появления новых факторов ранжирования, метрика может быть модифицирована.

Поскольку признаки в нашем определителе могут иметь произвольное количество состояний, мы в некоторых случаях не сможем говорить о том, разделяет ли данный признак два класса (это случаи, когда множества состояний этого признака, присущих классам, пересекаются и не содержатся один в другом), поэтому будем рассматривать минимальные множества состояний, разделяющих классы.

Рассмотрим определитель, каждый признак в котором имеет только два состояния: признак присутствует у наблюдаемого объекта и признак отсутствует. Мы можем найти K минимальных по числу элементов наборов $\{s_k\}^{i,j}_{j\in[1,n]}$ признаков (состояний в изначальной постановке), отделяющих класс i от остальных. Тогда ранг произвольного признака (в изначальной постановке имеющего несколько состояний) может быть посчитан следующим образом:

$$rate(c)=\sum_{s\in states(c)}\ \sum_{i\in[1,N],\,j\in[1,K]}\left\{P_{i,j}(s)\middle|s\in\{s_k\}^{i,j}_{j\in[1,K]}\right\}$$

$$P_{i,j}(s)=\frac{1}{\left|\{s_k\}^{i,j}_{j\in[1,K]}\right|}$$

В качестве функции $P_{i,j}(s)$ можно использовать отличную от предложенной (например, содержащую экспертные оценки состояния или признака).

Чем выше ранг признака, тем ценнее он в диагностическом плане для исследователя.

Сравнение с энтропийным методом

Традиционно в биологических определителях используется метод на основании расчета энтропии распределения возможных состояний для каждого признака [5].

Признаки сортируются в порядке убывания энтропии Шеннона:

$H(c)=-\sum_{i=1}^{n_i}\left|\chi_c(s_i)\right|\ln\left|\chi_c(s_i)\right|$, где $\chi_c(s_i)$ – множество классов, удовлетворяющих состоянию s_i признака c .

Для сравнения предложенного нами метода с энтропийным методом, мы считали среднее число шагов для определения объекта каждого класса (то есть до момента, когда остается только один класс, подходящий под введенные данные) в порядке ранжированного списка признаков. На трех различных определителях мы получили следующие результаты (где K – количество минимальных путей при подсчете для одного класса):

- **35 классов, 38 признаков, 114 состояний**
  Энтропийный метод: 7.71 шагов
  Метод минимальных разделяющих множеств:

• K = 1	• K = 7	• K = 15
• 10.5	• 9.54	• 6.98

- **130 классов, 24 признака, 103 состояния**
  Энтропийный метод: 14.52 шага
  Метод минимальных разделяющих множеств:

• K = 1	• K = 7	• K = 15
• 11.63	• 11.28	• 11.24

- **1039 классов, 331 признак, 909 состояний**
  К сожалению, у нас не было достаточно вычислительных мощностей для того, чтобы за разумное время провести предлагаемые тесты. Был получен только один результат для K = 1, среднее считалось не по всем 1039 классам, а по случайным 100.
  Энтропийный метод: 53.17 шагов
  Метод минимальных разделяющих множеств: 24.3 шага

По полученным результатам мы можем предположить, что предложенный нами метод с ростом числа классов и возможных признаков и состояний дает ощутимо лучшие результаты нежели энтропийный.

Следует отметить, что при тестировании определителя с ранжированием признаков по методу минимальных разделяющих множеств, ранги признаков считаются только один раз и не пересчитываются после каждого шага определения, в отличие от энтропии, подсчитываемой на каждом шаге. Это, безусловно, поможет сэкономить часть вычислительных ресурсов без уменьшения качества определителя.

Рекомендации по применению метода

При начальном ранжировании признаков рекомендуется проводить ряд тестов для выявления оптимального значения параметра K, при котором среднее число шагов идентификации будет минимальным. Особенно важной эта рекомендация является для небольших определителей, число классов в которых не превышает 100.

По мере накопления статистики по результатам определения, рекомендуется скорректировать значение параметра K, а также модифицировать ранжрующую метрику таким образом, чтобы при описанном тестировании предпочтение отдавалось сокращению пути поиска для наиболее часто встречающихся объектов.

Использование модели MapReduce для параллельного вычисление рангов признаков

Модель MapReduce [6] хорошо подойдет для параллельного вычисления рангов признаков.

Минимальные разделяющие множества вычисляются на этапе Map. Каждый Map-узел возвращает пары из признака и частоты его использования в минимальных путях.

На этапе Reduce вычисляется значение метрики в зависимости от частоты встречаемости признака в минимальных разделяющих множествах, полученной на этапе Map, а также ряда других факторов, предложенных в работе.

Реализация универсального определителя

Как было отмечено ранее, у существующих до настоящего времени определителей есть ряд общих проблем: невозможность построения собственного определителя на базе предоставленной системы, жесткая привязка к интерфейсу пользователя, платформенные ограничения, отсутствие API для доступа к функциям определителя. Клиент-серверная архитектура позволяет решить сразу несколько этих проблем следующим образом:

- **Сервер**
  Хранит данные по зарегистрированным определителям, отвечает за ранжирование признаков и классов в процессе определения, собирает статистику. В API веб-сервиса могут быть добавлены функции регистрации нового определителя и изменения существующих данных.
- **Клиент**
  Наличие общего API для доступа к функциям сервера делает возможным построение клиентского приложения на любой платформе и с любым пользовательским интерфейсом.

Также немаловажным принципом построения демонстрационного приложения является возможность запуска разработанного веб-сервиса на любой популярной в настоящее время операционной системе, будь то *nix, Windows или MacOS.

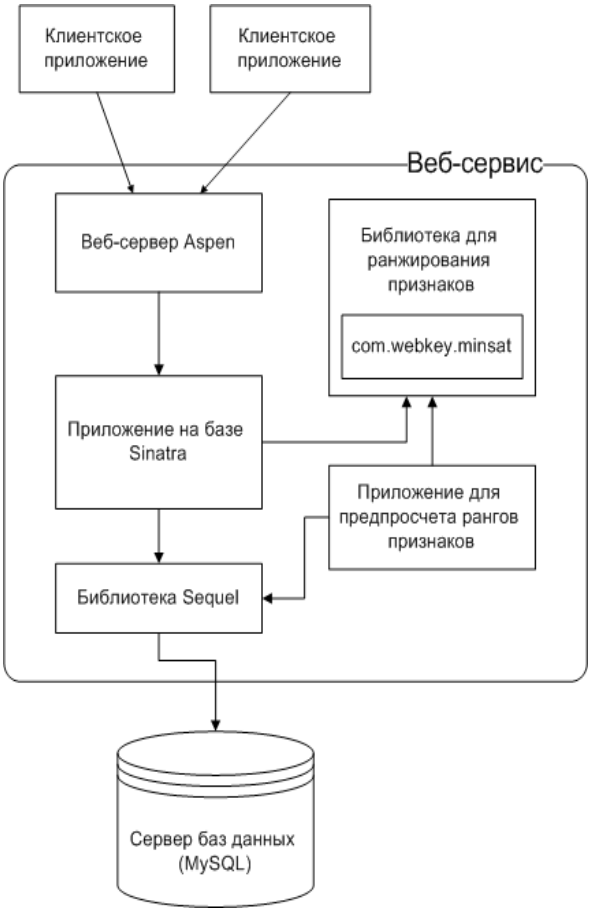

Рис. 1. Архитектура приложения

Общая архитектура приложения

На рисунке 1 представлены основные компоненты системы:

- Ряд клиентских приложений, отвечающих за передачу пользователю информации, полученной от сервера. Демонстрационный клиент представляет собой сайт, работающий на отдельном сервере. Реализован на популярном MVC фреймворке – Zend Framework [8].
- Веб-сервис, постоенный на базе сервера приложений Aspen [9], написанного на JRuby [10], непосредственно самого приложения разработанного на фреймворке Sinatra [11].
- База данных под управлением сервера MySQL [12]. Для доступа к данным была использована ORM библиотека Sequel [13].
- Библиотека для ранжирования признаков Интерфейс написан на JRuby, а чувствительные ко времени исполнения компоненты – на Java.
- Демон предпросчета рангов признаков для вновь добавленных определителей.

Процесс регистрации нового определителя потенциально может занимать заметное для пользователя количество времени даже в случае небольших по объему данных. Поэтому подсчет изначальной ранжировки признаков происходит не сразу, а добавляется в очередь на обработку. Раз в час запускается скрипт, подсчитывающий ранги признаков вновь добавленных определителей. В будущем подобный скрипт может также пересчитывать ранги существующих признаков с учетом собранной статистики.

Достигнутые результаты

- Проведен анализ проблем существующих биологических определителей, а также выявлены их основные недостатки.
- Найден метод оптимизации процесса идентификации объекта по произвольному числу признаков, куда входит:
  - метод решения задачи о минимальном по числу элементов разделяющем два произвольных класса множестве признаков;
  - метод ранжирования признаков для минимизации числа определяемых признаков.

- Проведено имитационное моделирование, которое показало, что разработанный метод ускоряет определение объекта по среднему числу используемых признаков на 20-50% по сравнению с традиционно используемым энтропийным методом.
- Даны рекомендации по применению нового метода в биологических определителях.
- Реализован веб-сервис, лишенный недостатков существующих решений, а также демонстрационное клиентское приложение.

Направления развития

В качестве направлений развития системы можно выделить:

- **Распараллеливание процесса подсчета рангов признаков**
- **Дальнейшую оптимизацию процесса идентификации объектов.** Сюда относится использование данных накопленной по результатам определений статистики, что позволит задать более точную, с точки зрения минимизации определяемых признаков, метрику. Также возможен пересчет рангов признаков в зависимости от определенных пользователем состояний. Это позволит системе оптимизировать поиск даже редких классов. Немаловажным фактором улучшения метрики ранжирования признаков является добавление в ее параметры экспертной оценки надежности признака и вероятности его ошибочного определения.

ЛИТЕРАТУРА:

1. Приложение iBird Explorer PRO // http://itunes.apple.com/app/ibird-explorer-pro/id308018823?mt=8
2. А.Л. Лобанов, А.Г. Кирейчук, И.С. Смирнов. Биологическая диагностика: история, современное состояние, проблемы, 2009. // http://www.zin.ru/projects/WebKey-X/basis.htm
3. А.Т. Вахитов, О.Н. Граничин, А.Г. Кирейчук, А.Л. Лобанов. Параллельный алгоритм обучения для интерактивного политомического определителя биологических видов, 2009.
4. E.A. Hirsch. A Fast Deterministic Algorithm for Formulas That Have Many Satisfying Assignments. L. J. of the IGPL, Vol. 6 No. 1, pp. 59-71, 1998.
5. А.В. Свиридов. Ключи в биологической систематике: теория и практика. М.:Издательство Московского Университета. 1994. 224 с.
6. Jeffrey Dean and Sanjay Ghemawat . MapReduce: Simplified Data Processing on Large Clusters , OSDI 2004. // http://labs.google.com/papers/mapreduce.html
7. Zend Framework Programmer's Reference Guide. // http://zendframework.com/manual/en
8. Документация по компоненту Zend_Http. // http://zendframework.com/manual/en/zend.http.html
9. Git-репозиторий проекта веб-сервера Aspen. // http://github.com/kevwil/aspen
10. Язык JRuby. // http://jruby.org
11. Sinatra framework Documentation. // http://www.sinatrarb.com/documentation
12. MySQL Documentation. // http://dev.mysql.com/doc/index.html
13. Documentation for Sequel. (v3.11.0) // http://sequel.rubyforge.org/documentation.html