ПРОГРАММНАЯ СИСТЕМА ДЛЯ РЕШЕНИЯ ЗАДАЧИ СИНТЕЗА SYNTHESIS 1.0

Александр Докукин

***Abstract**: В статье представлено описание программной системы, предназначенной для решения введенной ранее задачи синтеза в рамках алгебраической теории распознавания. В основе алгоритмов системы лежат теоретические результаты, также описанные ранее. Приводятся оценки скорости работы и качества решения задачи для реальных данных больших объемов.*

***Keywords**: алгебраическая теория распознавания, задача синтеза, программная система.*

***ACM Classification Keywords**: I.5 Pattern Recognition — I.5.0 General.*

Введение

Термин задача синтеза используется в данной статье для обозначения определенного нового типа задач в рамках теории распознавания. А именно, задачи поиска в генеральной совокупности объекта, обладающего в наибольшей степени выраженным свойством, причем свойство это описано только набором прецедентов, т.е. объектов, обладающих или не обладающих этим свойством. При этом, предполагается, что набор прецедентов неполон, т.е. в генеральной совокупности есть неизвестные объекты, и, может быть, противоречив.

Такая общая постановка возникла при решении частной прикладной задачи — синтеза белков с заданным профилем структуроформирующих свойств. Задача решается в предположении, что элементарной единицей, отвечающей за обладание данным отрезком белка структуроформирующими свойствами, является пятибуквенная аминокислотная последовательность [Nekrasov, 2004]. Эта гипотеза получила, в том числе, косвенное подтверждение, поскольку позволила разделить представителей двух классов (имеющих и не имеющих структуроформирующие свойства) методами распознавания с точностью 89% [Senko et al, 2011].

Строгая формальная постановка задачи, а также методы её решения и их обоснование опубликованы ранее [Dokukin, 2013a; Dokukin, 2013b]. В данной статье мы кратко опишем только основную идею метода, понимание которой необходимо для прикладного применения соответствующих алгоритмов. Основной же целью данной статьи является описание программной системы, реализованной на основе полученных ранее результатов.

Метод синтеза

Общая идея решения задачи синтеза заключается в следующем. Если нет возможности оценить выраженность некоторого свойства объекта явно, но имеются прецеденты его наличия или отсутствия, можно обучить на этом наборе прецедентов алгоритм распознавания, и использовать для оценки выраженности свойства оценку принадлежности объекта одному из классов, вычисляемую алгоритмом.

Для увеличения скорости работы, а также для упрощения интерпретации получаемых результатов был разработан особый метод логических закономерностей [Dokukin, 2013a]. Поскольку мы говорим о распознавании пятибуквенных слов в определённом алфавите, логические закономерности приобрели

форму шаблонов слов, т.е. слов с фиксированными буквами в некоторых позициях и пропусками в остальных. В местах пропусков допустимы любые буквы.

Набор закономерностей строится по набору прецедентов, этот этап называется этапом обучения. Качество набора закономерностей можно оценить, распознав на его основе независимую выборку. Этот этап решения задачи, очевидно, будем называть распознаванием.

Последний, третий этап заключается собственно в синтезе, т.е. поиску по заданному шаблону слова с заданными свойствами. Здесь можно выделить два принципиальных случая, которые были названы элементарной и общей задачей синтеза. Элементарная задача заключается в поиске слова по шаблону той же длины, что и прецеденты. При этом, задан целевой класс, соответствующий или наличию, или отсутствию некоторого свойства. Общая задача заключается в поиске слова по шаблону, длина которого существенно превышает длину прецедентов. При этом, на шаблоне задан профиль целевого свойства, т. е. размечены участки его наличия или отсутствия.

Универсальный подход к решению задачи синтеза заключается в независимом восстановлении каждой пропущенной буквы по всему набору шаблонов содержащих её. Например, если имеется шаблон AB-C*D--E и нас интересует восстановление центральной буквы, необходимо выбрать наилучшего кандидата, удовлетворяющего набору шаблонов AB-C*, B-C*D, -C*D-, C*D--, *D--E. Влияние букв в других пропусках не учитывается. Показано, что при определенных, легко проверяемых, условиях на набор закономерностей такой подход даёт оптимальное решение.

Программная система

В описываемой программной системе реализованы алгоритмы для решения всех трёх описанных этапов, а именно: обучения, распознавания и синтеза. При этом, система предоставляет визуальные средства управления, представления промежуточных результатов и т.д.

Для работы с программой необходимо либо создать новый проект, либо загрузить уже существующий. Ключевой характеристикой проекта является набор прецедентов, по которым производится обучение алгоритмов синтеза. То есть, проект предназначен для работы с одним типом задачи. При этом, система позволяет одновременно работать с несколькими проектами.

Основное окно системы разделено на три части. Самая левая содержит дерево проектов и позволяет переключаться между ними, а также между отдельными их свойствами.

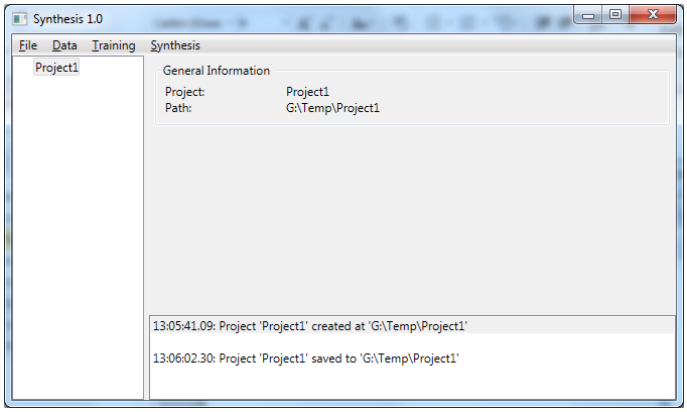

Рисунок 1. Общий вид проекта

Справа вверху расположено окно свойств. Это окно содержит описание текущего элемента дерева проектов. Если выбран проект, то в окне кратко перечисляются основные свойства проекта. Иначе, отображается подробная информация, соответствующая текущему этапу работы.

Наконец, справа внизу расположен журнал проекта, в котором отражаются основные этапы работы с ним. Журнал содержит точные временные отметки, что позволяет оценить продолжительность отдельных этапов.

Исходные данные предоставляются в виде набора из двух текстовых файлов, каждый из которых содержит все прецеденты одного из классов. Формат данных таков: первые пять символов каждой строки соответствуют пятибуквенному прецеденту, остальные символы — комментарий.

Таким образом, принципиальными ограничениями на тип исходной информации является фиксированная длина прецедентов и бинарность целевого профиля. Кроме того, мощность алфавита ограничена допустимым набором аминокислот. Если данные соответствуют этим требованиям, но не имеют отношения к белкам или к их структуроформирующим свойствам, применение всех алгоритмов по-прежнему допустимо. Например, можно работать не со структурообразующими свойствами самими по себе, а с определенной пространственной структурой. Бинарный признак при этом может быть задан в виде “Определенная структура” и “Все остальные”. Обучающая выборка должна быть приведена в соответствие данному признаку.

Исходные данные представлены в свойстве “Training Data” проекта.

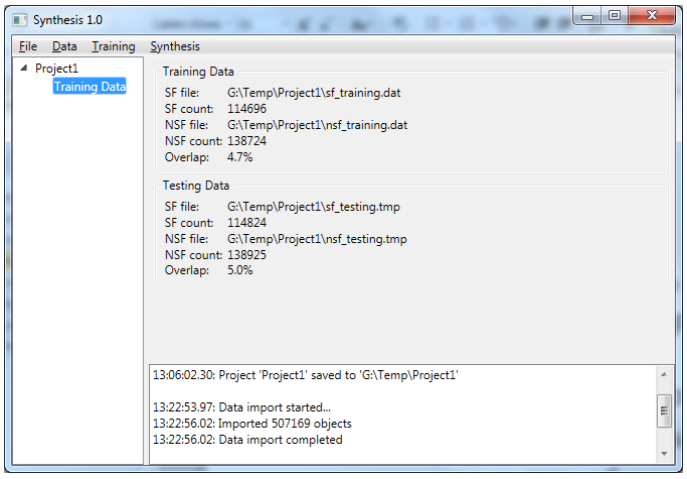

Рисунок 2. Отчёт об исходных данных

В этой версии описание исходных данных состоит только из отчёта. Отчёт содержит две одинаковые секции, посвящённые обучающим (Training Data) и тестовым (Testing Data) данным. Они содержат данные о расположении файлов для обоих классов (SF и NSF file), количество объектов в каждом классе (SF и NSF count) и процент совпадений прецедентов в двух классах (Overlap).

Обучение алгоритма для текущего проекта производится из меню “Training → Train”. Ключевой особенностью разработанного метода является высокая скорость работы. Обработка выборки из пятисот тысяч объектов занимает около двадцати секунд на обычном персональном компьютере. Результаты обучения будут представлены в свойствах “Training Results” проекта.

Отчёт включает статистику числа найденных закономерностей по рангам (Rank N Count), а также статистику нерегулярных закономерностей (Irregular N Count), т.е. закономерностей, которые не являются уточнениями закономерностей меньшего ранга.

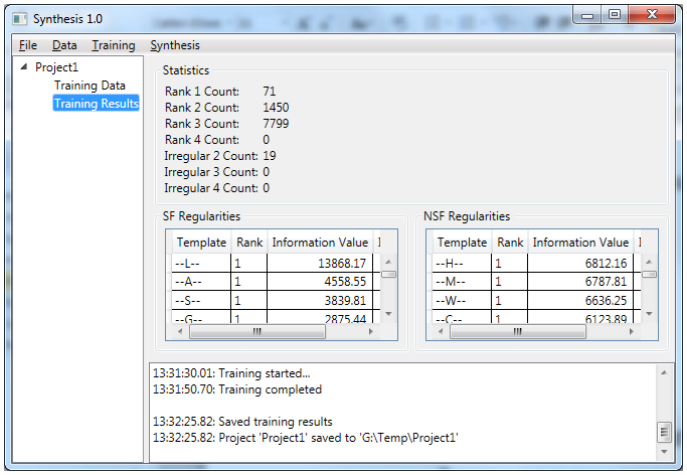

Рисунок 3. Результаты обучения

Кроме того, на этом этапе в удобном виде представлен полный набор найденных закономерностей для каждого из классов (SF и NSF Regularities). Таблицы содержат шаблон закономерности (Template), её ранг (Rank), значимость (Information Value) и регулярность (Is Regular). Таблицы могут быть отсортированы по любому из этих полей.

Проверка результатов обучения является важной частью оценки качества полученного решения. Здесь под проверкой понимается проверка качества распознавания независимой выборки, т.е. обученный алгоритм распознавания приписывает некоторый класс неизвестным ему объектам, и проверяется доля верных ответов алгоритма. Как уже упоминалось выше, наилучший достигнутый результат с небольшими (до нескольких тысяч объектов) обучающими выборками составил 89% [Senko et al, 2011]. Разработанный метод с большими “грязными” выборками (допускающими пересечение классов), дал результат на уровне 78% [Dokukin, 2013a].

Проверка также занимает незначительное время. Для полумиллиона объектов это менее десяти секунд. Отчёт об обучение появляется в свойствах “Testing Results”.

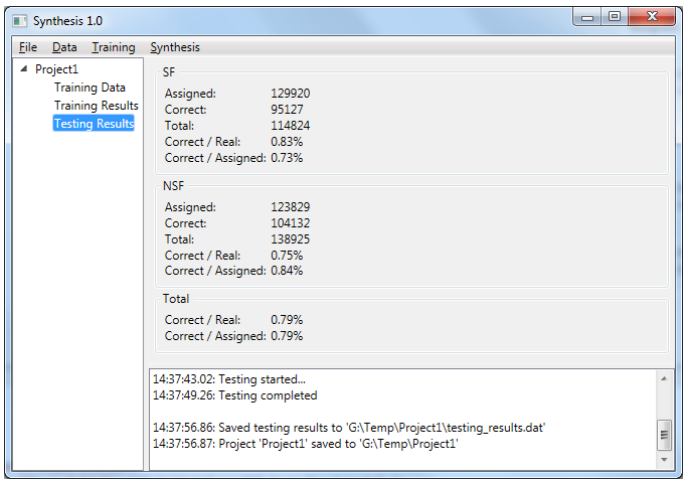

Рисунок 4. Отчёт о распознавании независимой выборки

Здесь тоже присутствуют два одинаковых блока для классов SF и NSF. В них отображается количество объектов, отнесённых алгоритмом к классу (Assigned); количество объектов, отнесённых корректно (Correct); общее количество объектов (Total) и две оценки качества: доля правильно угаданных среди объектов класса, и доля правильно отнесённых. Кроме того, присутствует информация, усреднённая по классам.

Обученный алгоритм можно использовать для решения нескольких задач синтеза одновременно. Для этого нужно импортировать соответствующие шаблоны слов и целевые признаки. Шаблоны описываются текстовыми файлами специального формата:

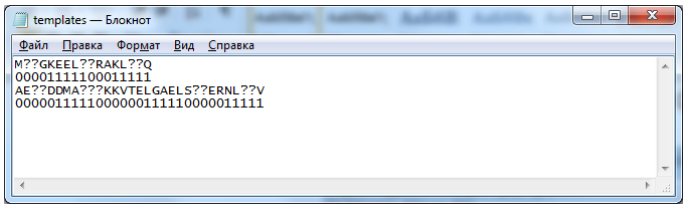

Рисунок 5. Пример файла шаблонов

Здесь, первая строчка задает собственно шаблон, а вторая — профиль целевого признака. Для добавления новых шаблонов в проект надо воспользоваться меню “Synthesis → Import Profiles…”.

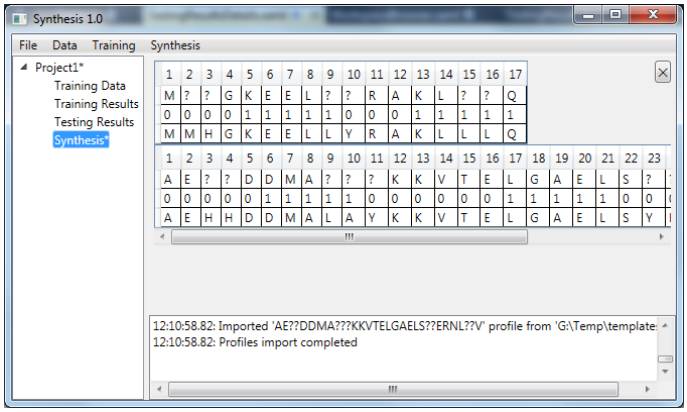

Рисунок 6. Представление шаблонов задач синтеза в программе

Загруженные шаблоны представляются в программе в виде таблицы. Первые две строчки соответствуют загруженным данным, третья — решению задачи, которая решается практически мгновенно. В программе используются обобщённые алгоритмы на случай длинных последовательностей букв синтеза [Dokukin, 2013b]. Таким образом, длина искомых шаблонов может многократно превышать длину обучающих прецедентов.

Шаблоны можно удалять, нажав на соответствующий крестик. В ближайших версиях программы запланировано добавление и редактирование шаблонов вручную, без посредничества текстовых файлов.

Выводы

Нетрудно видеть, что использованные при обучении данные породили SF-шаблон “--L—“ с огромным отрывом по значимости, поэтому большинство случаев недостатка информации разрешается в пользу

этой буквы. Однако, в некоторых случаях (см. шаблон 2, позиция 10) окружение существенным образом влияет на решение. Класс NSF дает более разнообразные решения за счёт наличия нескольких лидирующих закономерностей.

Как уже говорилось выше, на данную ситуацию можно влиять, сужая выборку до интересующего типа последовательностей. Мы надеемся с помощью представленной программы обнаружить такие содержательные постановки.

Кроме того, уже сейчас запланировано развитие алгоритмов для случая нескольких классов. Их реализация будет сразу же внедрена в программу, что, как мы надеемся, ещё более повысит её практическую ценность.

Благодарности

Статья опубликована при поддержке ITHEA ISS (www.ithea.org) и ADUIS (www.aduis.com.ua), проект ITHEA XXI.

Литература

[Dokukin, 2013a] Dokukin A. “On a Logical Regularities Based Method of Definite Quality Object Synthesis”, International Journal “Information Models & Analyses”, 2013, Vol. 2, No. 3, pp. 212–216.

[Dokukin, 2013b] Dokukin A. “On the Formalization of Synthesis Task in Pattern Recognition”, Proceedings of the 11th International Conference “Pattern Recognition and Image Analysis: New Information Technologies” (PRIA-11-2013), 2013, Vol. 1, pp. 84–86.

[Nekrasov, 2004] Nekrasov A. N. “Analysis of the Information Structure of Protein Sequences: A New Method for Analyzing the Domain Organization of Proteins”, Journal of Biomolecular Structure and Dynamics, 2004, Vol. 21, Iss. 5, pp. 615-623.

[Senko et al., 2011] Senko O. V., Nekrasov A. N., Ryazanov V. V., Dokukin A. A. “Prediction of Structure forming Properties of Protein with Help of Pattern Recognition Methods”, Proceedings of 8-th Open German-Russian Workshop “Pattern Recognition and Image Understanding” OGRW-8-2011, pp. 38–40.

Информация об авторе

Alexander Dokukin *– Computing Centre of Russian Academy of Sciences, researcher,*

40 Vavilova St., Moscow, Russia, 119333; e-mail: dalex@ccas.ru.

Major Fields of Scientific Research: Algebraic Approach to Pattern Recognition.

Software System for Solving Synthesis Problem "Synthesis 1.0"

Alexander Dokukin

Abstract: *The software system for solving previously introduced synthesis task in pattern recognition theory is described. The algorithms used in the system are based on theoretical results described earlier. Estimates of its speed and quality in real-world tasks of large scale are presented.*

Keywords: *algebraic recognition theory, synthesis task, software system*