УДК 51-76+004.056.53

А.М. Соколов

РАНДОМИЗИРОВАННОЕ ВЛОЖЕНИЕ РАССТОЯНИЯ РЕДАКТИРОВАНИЯ В ЗАДАЧАХ ПОИСКА ГЕНОВ И ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ

Введение

Во многих предметных областях актуальна задача сравнения длинных символьных последовательностей. Примерами являются поиск дубликатов в поисковых машинах [1], генетика [2], информационная безопасность в компьютерных системах [3,4] и др.

Распространенной метрикой, применяемой для сравнения последовательностей, является расстояние Левенштейна (классическое расстояние редактирования) [5]. Оно определяется между двумя символьными строками x, y как минимальное количество операций вставки, замены и удаления символов для преобразования x в y. Эти операции легко интерпретируются в задачах генетики как изменения, происходящие при мутациях и эволюции генов, а также хорошо описывают некоторые способы обхода систем обнаружения вторжений в компьютерные системы [6].

Широко известен классический алгоритм вычисления расстояния Левенштейна, имеющий для строк длиной n сложность $O(n^2)$ [7]. Однако, при больших размерах n и большом количестве строк, что характерно для указанных выше областей, применение этого алгоритма весьма затруднительно.

В [8] был предложен метод вычислительно эффективной оценки расстояния редактирования, использующий вложение расстояния редактирования в векторное пространство. В данной работе исследуется его применение в задачах поиска генов и обнаружения аномалий в компьютерных системах.

Рандомизированное вложение и LSH

Суть предложенного в [8] рандомизированного вложения классического расстояния редактирования в векторное пространство состоит в следующем.

Принято, что подстрока длиной q есть *q-грамма* символьной строки $x\in\Sigma^n$ и $q\in N$. Для вектора вида $v_{n,q}(x)\in(N\cup\{0\})^{|\Sigma|q}$, где каждой q-грамме $\sigma\in\Sigma^q$ соответствует элемент вектора, значение которого – число встречаемости σ в x, введен термин *q-граммный вектор*. Манхетенново (l_1) расстояние между такими векторами, т.е. сумма модулей разностей значений элементов векторов, названо *q-граммным расстоянием*.

Пусть вектор $v^i_{w,q1,q2}(x)$ есть конкатенация q-граммных (от q_1 до q_2) векторов подстроки x[i,i+w–1] длиной w, где i выбрано случайно, независимо и равновероятно из множества возможных позиций i окна шириной w: $i=1,...,n–w+1$. Пусть φ^i – случайный вектор такой же размерности, как и $v^i_{w,q1,q2}(x)$, с элементами, выбранными случайно и независимо из распределения Коши $p(x)=(\pi(1+x^2))^{-1}$. Построим для строки x хеш-вектор размерностью K: $h(x)=(h_1(x),h_2(x),...,h_K(x))$, где

$$h_i(x)=\lfloor((v^i_{w,q1,q2}(x),\varphi_i)+b_i)/r\rfloor, \qquad (1)$$

r и b_i – действительные числа, b_i выбрано случайно, независимо и равновероятно из диапазона [0,r].

Если определить шар $B(q,k)=\{x\in \Sigma^n | ed(q,x)\le k\}$, то семейство $H=\{h:\Sigma^n\to X\}$ (где X – некоторое конечное или счетное множество значений) является [9] (k_1,k_2,p_1,p_2)-чувствительным или просто локально-чувствительным, если для любых $x,y\in\Sigma^n$ и любой независимо и равновероятно выбранной хеш-функции $h\in H$ выполняется:

$$\text{если } x \in B(y,k_1), \text{ то } Prob[h(x) = h(y)] > p_1, \quad (2)$$

$$\text{если } x \notin B(y,k_2), \text{ то } Prob[h(x) = h(y)] < p_2. \quad (3)$$

В [8] доказано, что (1) есть локально-чувствительная функция и на ее основе можно построить процедуру поиска ближайших строк [9], реализуемую с помощью процедуры LSH-лес [10].

Пусть имеется база P строк одинаковой длины n. Необходимо на запрос $q\in\Sigma^n$ возвратить приближенного ближайшего соседа – строку, находящуюся в шаре $B(q,k_2)$. При этом $K = \log_{1/p2}|P|$, количество деревьев $L = |P|^{\rho}$, где

$$\rho = \log(p_1/p_2). \quad (4)$$

При этих условиях описанный ниже алгоритм поиска выдает в выходное множество S с вероятностью большей, чем 1/2, строку y, такую, что $ed(x,y) \le k_2$, где $k_2 = O(zn^{1/3}\ln n)$, z –параметр, которые влияет на значение k_2 и значения вероятностей p_1,p_2. При $p_2<p_1$ выполняется $\rho<1$, что позволяет проводить меньше сравнений, чем при полном переборе базы P.

Реализация поиска ближайшей строки с помощью LSH-леса

Для каждого j=1,...,L все хеш-векторы $h^j(p)$ (размерностью K) всех строк p базы P хранятся в виде отдельного префиксного дерева T_j глубиной до K уровней (корень имеет глубину 0, листья – K). Узлы дерева соответствуют значениям элементов хеш-вектора и содержат ссылки на строки, хеш-векторы которых соответствуют пути от корня дерева к данному узлу. Так, если у хеш-векторов двух строк совпадают первые k их элементов, то и первые k узлов на пути от корня дерева T_j до листьев, соответствующих этим строкам, также совпадают.

При поступлении запроса-строки q:

а. Формируются L его K-мерных хеш-векторов $h^j(q)$;

б. Для каждого хеш-вектора $h^j(q)$ в T_j находится узел, соответствующий $h^j(p)$ с наибольшим числом совпадающих первых элементов этих векторов;

в. Начиная от найденных в 2 узлов, все L деревьев синхронно проходятся по направлению к корню, и соответствующие указанным узлам строки p добавляются в результирующее мультимножество строк S;

г. После того, как все строки, хеш-векторы которых совпадают на данном уровне, добавлены в S, повторяется процедура для следующего (более "высокого") уровня, пока не достигнут корень или |S| не превысит 2L.

В результате получаем мультимножество S строк (кандидатов на приближенного ближайшего соседа к q), упорядоченное в порядке убывания глубины узлов дерева, до которых совпадали первые элементы хеш-векторов соответствующей строки и запроса. Будем далее под уровнем строки из S понимать глубину

последнего узла дерева, на котором еще совпадали первые элементы хеш-векторов запроса и этой строки.

Согласно теореме 5.1 из [10], S будет содержать приближенных ближайших соседей к запросу с ненулевой постоянной вероятностью. Для того, что бы найти действительных ближайших соседей в S, необходимо проверить принадлежность строк из S шару $B(q,k_2)$ с помощью классического алгоритма.

Для всех экспериментов, описываемых ниже, были зафиксированы параметры $k_1=1$, $z=1.01$.

Поиск генов и гиперчувствительных сайтов в ДНК

Поиск генов с помощью рассуждений на основе примеров

Поиск новых генов – это фундаментальная задача вычислительной биологии, состоящая в алгоритмизированном поиске биологически функциональных участков генетических последовательностей. Генами называются определенные подпоследовательности нуклеиновых баз (нуклеотидов) дезоксирибонуклеиновой кислоты (ДНК). Гены высших организмов (эукариотов) состоят из несмежных участков – экзонов. Между экзонами находятся гораздо более длинные некодирующие, "мусорные" последовательности, не относящиеся к генам – интроны [11], у человека суммарно составляющие до 99% длины всего генома [12].

Объем базы генов GenBank удваивается каждые 15 месяцев [13], а длина, например, генома человека составляет около 3.2 миллиардов нуклеотидов. Поэтому поиск генов невозможен без эффективных вычислительных инструментов анализа большого количества длинных последовательностей. Сходство (гомологичность) генетических последовательностей исследуемых организмов с известными позволяет находить новые гены, а также указывает на их эволюционное родство и функцию. Это обусловливает актуальность эффективного поиска гомологичных генетических последовательностей.

Другой важной задачей является поиск гомологичных некодирующих участков в локус-контролирующих областях (locus control region, LCR) бета-глобина, которые управляют транскрипцией в локусе. Известно, что последовательности бета-глобина содержат хорошо сохранившиеся в процессе эволюции области, называемые гиперчувствительными сайтами ДНКазы [14], что позволяет идентифицировать эти области в бета-глобине исследуемых организмов.

Нуклеотиды можно представить как алфавит из четырех символов A, G, T и C, а цепи молекулы ДНК – как последовательности таких символов. Степень отличия двух цепей может быть определена как расстояние редактирования между последовательностями составляющих их нуклеотидов.

В данной работе для поиска экзонов используется подход рассуждений на основе примеров (case-based reasoning, CBR), где роль примеров играют экзоны последовательностей обучающей выборки. По таким примерам ищутся экзоны в тестовых последовательностях, либо участки гиперчувствительных сайтов одного организма, по которым необходимо найти такие же в бета-глобине другого организма. Для повышения эффективности поиска используются описанные выше и в [8] методы вложения.

Результаты экспериментального исследования поиска экзонов

Цель первого эксперимента – оценить эффективность предложенного метода поиска сходных строк в задаче поиска экзонов в генетических последовательностях позвоночных по известным примерам экзонов других организмов.

Были проведены эксперименты по поиску экзонов, где обучающей базой был выбран набор HMR195, использованный в [15] для тестирования программ обнаружения генов (948 последовательностей). Тестовой выборкой служила база последовательностей Burset-Guigo из [16] (570 последовательностей). В обеих базах экзоны размечены.

Поиск экзонов осуществлялся путем распознавания принадлежности экзону отдельных нуклеотидов. Каждому i-му нуклеотиду всех тестовых последовательностей ставился в соответствие счетчик c_i, изначально инициализированный нулем. Для каждого экзона g_i из обучающей последовательности принималось $P=\{g_i\}$ и строился LSH-лес. Далее все тестовые последовательности y нарезались скользящим окном шириной $|g_i|$. Подстроки вида $y[i,i+|g_i|-1]$ поочередно служили запросом к таким образом построенному LSH-лесу. Для всех подстрок, совпавших с g_i на уровне большем или равном K, с помощью классического алгоритма вычислялось расстояние редактирования. Для тех подстрок, расстояние редактирование которых было минимально в пределах данной тестовой последовательности, увеличивалось на единицу значение счетчиков, соответствующих нуклеотидам, принадлежащим данным подстрокам. Далее решение о том, что считать нуклеотидом, принадлежащим экзону, определялось порогом t, отсекающим малые значения c_i.

Чтобы избежать влияния того, что краевые нуклеотиды покрываются меньшим числом окон, предварительно все последовательности из тестовой выборки дополнялись в начале и в конце спецсимволом N (в количестве 2454 символа, что равно длине самого длинного экзона из набора HMR195).

Как интегральная оценка качества поиска экзонов применялась т.н. "приближенная корреляция" (approximate correlation, AC) [16], определяемая как

$$AC = 1/2[\ TP/(TP+FN) + TP/(TP+FP) + TN/(TN+FP) + TN/(TN+FN)] - 1, \quad (5)$$

где TP – true positives, TN – true negatives, FP – false positives, FN – false negatives.

На рис. 1 изображены графики нормализованных значений c_i, а также реальные границы экзонов для последовательности ACU08131 из набора Burset-Guigo для значений K=5 и L=1,4,7,10. Как видно из рисунков, увеличение L приводит к увеличению разницы между значениями c_i для интронов и экзонов, пики становятся более узкими.

Предложенный метод сравнивался с подходом на основе классического алгоритма редактирования [17]. В последнем лучшие значения AC (5), усредненного по всем тестовым последовательностям (AC_{avg}), достигали 0.49. Однако, полученная нами оценка времени работы такого алгоритма на исследованной базе составляет около 6 лет на однопроцессорном компьютере Athlon XP 2600+.

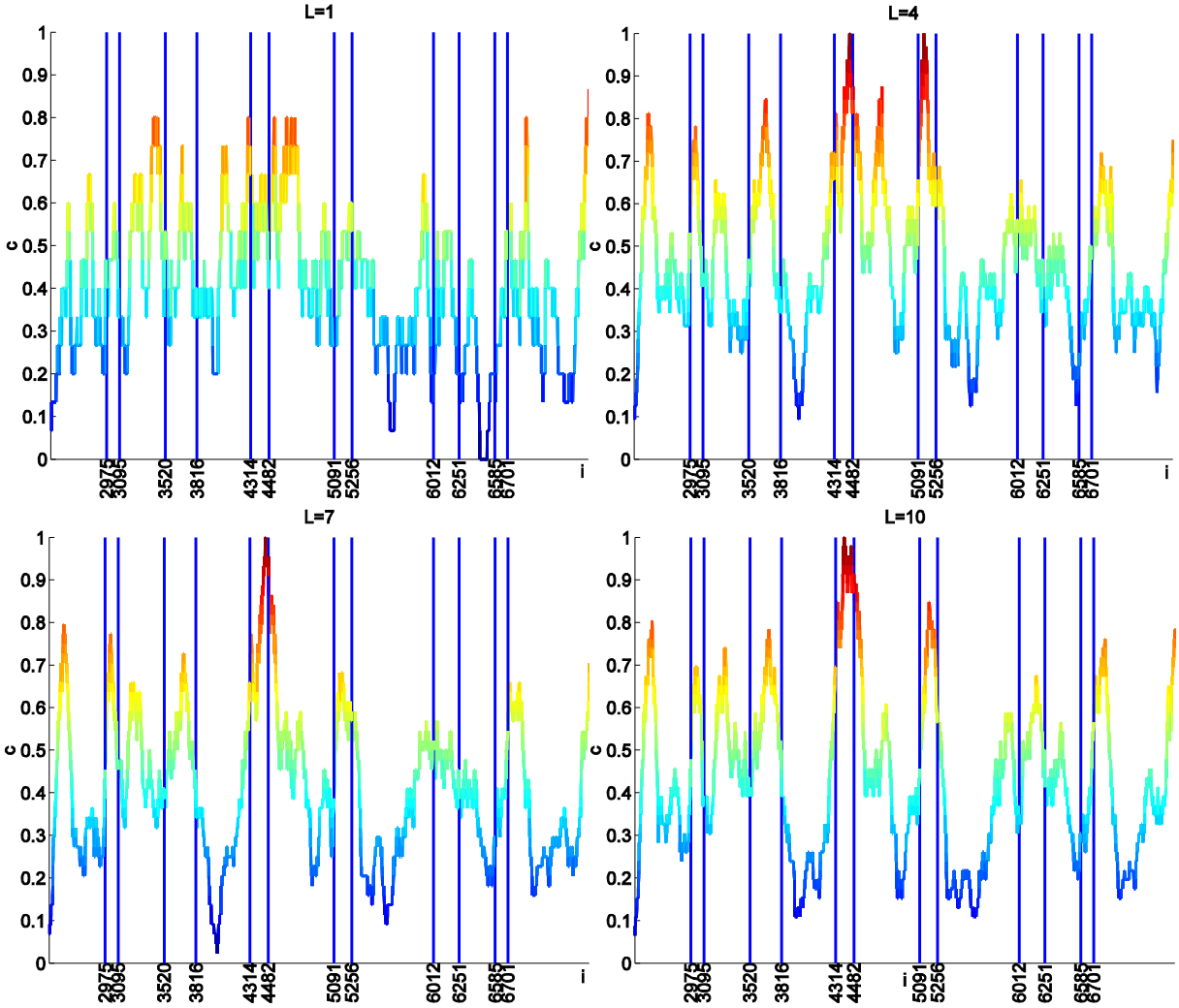

Рисунок 1 – Графики нормализованных значений c_i, а также реальные границы экзонов для последовательности ACU08131 для K=5 и L=1,4,7,10.

В табл. 1 приведены полученные нашим методом значения $AC_{max}=max_t\ AC_{avg}(t)$, усредненные по всем тестовым последовательностям для K=5,7 и L=1,…,10, а также суммарное время, потраченное на поиск кандидатов с помощью предложенного метода, а также на проверку полученных кандидатов с помощью классического алгоритма.

Как видно из табл. 1, метод на основе поиска приближенных ближайших строк с помощью процедуры LSH-лес показывает сравнимые результаты с методом из [17], но за гораздо меньшее время.

Результаты работы предложенного метода поиска генов сравнивались с результатами одной из популярных программ для поиска генов – GeneID 1.3.8. Алгоритм программы основан на использовании ряда разработанных эвристик в комбинации с марковскими цепями со специально подобранными параметрами для разных типов организмов, а также правилами, выведенными экспертами.

Таблица 1

Результаты поиска экзонов в наборе Burset-Guigo, для K=4,5,7

K=4	K=5	K=7

L	время,час	AC_{max}	время,час	AC_{max}	время, час	AC_{max}
1	20	0.440	5	0.408	5	0.308
2	39	0.452	11	0.419	9	0.364
3	56	0.463	23	0.431	24	0.385
4	29	0.468	18	0.438	25	0.391
5	62	0.469	22	0.442	25	0.393
6	42	0.470	34	0.448	29	0.395
7	62	0.470	39	0.449	26	0.397
8	71	0.473	34	0.452	29	0.398
9	80	0.471	38	0.455	32	0.400
10	70	0.474	42	0.458	36	0.402

Для сравнительного анализа использованы следующие наборы параметров, предоставляемые с программой GeneID 1.3.8, – human1iso, human3iso, human.070606 и human.061209.

На рис. 2 сплошными линиями изображены графики изменения точности (отношение количества правильно классифицированных нуклеотидов к общему числу нуклеотидов, классифицированных как кодирующие) от полноты (отношение количества правильно классифицированных нуклеотидов к общему числу кодирующих нуклеотидов) для метода на основе LSH-леса, полученные изменением порога t на значение c_i и усреднением по всем последовательностями тестовой выборки. Маркерами на рисунке изображены результаты GeneID.

Как видно из рис. 2 метод, основанный на процедуре LSH, и не использующий никаких специфических экспертных знаний о природе тестируемых последовательностей, тем не менее, правильно распознает принадлежность 25-50% нуклеотидов, находящихся в экзонах.

В ходе эксперимента было замечено, что короткие экзоны распознаются гораздо хуже, чем длинные, что может объяснять не очень высокий по сравнению с GeneID общий результат. Более высокие результаты работы программы GeneID также объясняются ее высокой специализацией и настройкой на конкретный тип организмов.

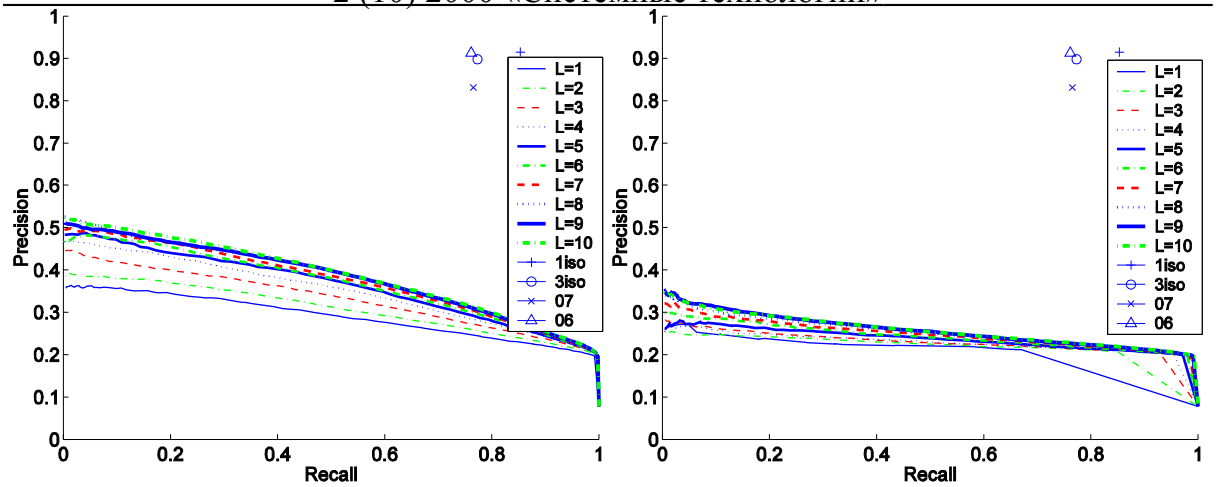

Рисунок 2 – График точность-полнота для K=5,7 и L=1,…,10 с обозначенными результатами работы GeneID для разных наборов параметров (“+” – human1iso, ”o” – human3iso, ”x” – human.070606, ”Δ” – human.061209).

Поиск коротких некодирующих последовательностей

Помимо поиска генов, исследовался поиск коротких гомологичных участков также в некодирующих участках генетических последовательностей для значений n, при которых не выполняется $\rho<1$ (4).

Цель второго эксперимента – исследование эффективности поиска близких коротких последовательностей нуклеотидов, а именно гиперчувствительных сайтов в бета-глобине родственных организмов. Использовалась LCR-последовательность бета-глобина мыши (длина 12Kb, GenBank Z13985) и человека (первые 20Kb, GenBank U01317) из [18].

В качестве обучающей выборки был принят бета-глобин мыши, тестовой – человека. Обозначим обучающую выборку x_m, тестовую – h_m. Все подстроки длиной n обучающей выборки составляли базу P, на которой строился LSH-лес. Создавался также массив счетчиков c, $|c|=|h_m|$, изначально инициализированный нулями. При проходе последовательности h_m скользящим окном шириной n, с помощью LSH-процедуры находились кандидаты на приближенных ближайших соседей к запросам $q_i=h[i,i+n-1]$, $i=1,...,|h_m|$. Если среди строк-кандидатов с глубиной совпадения K, возвращенных LSH- процедурой на запрос q_i, была хотя бы одна подстрока, принадлежащая B(q,e) (это проверялось с помощью классического алгоритма вычисления расстояния редактирования), то значения c(j) для всех j=i,...,i+n–1 увеличивались на единицу. Проводилось усреднение по 100 независимым запускам процедуры поиска с разными инициализациями генератора случайных чисел.

Таблица 2

Пример результатов поиска подстрок длиной n=50, находящихся на расстоянии редактирования не больше 3 в последовательностях бета-глобина мыши и человека для K=7,8

	c	$\sigma^2(c)$	время, мин	память, Мб	c	$\sigma^2(c)$	время, мин	память, Мб
L\K	7				8			
1	0.3	0.5	0.01	2.3	0.3	0.6	0.01	2.7

2	0.6	1.1	0.01	3.4	0.7	1.0	0.01	4.1
3	1.0	1.7	0.01	4.4	0.9	1.4	0.01	5.4
4	1.6	2.5	0.01	5.4	1.4	2.1	0.01	6.8
5	2.0	3.4	0.02	6.4	1.7	2.2	0.02	8.2
6	2.4	3.6	0.02	7.4	2.1	3.1	0.02	9.6
7	2.8	3.8	0.02	8.5	3.0	0.0	0.02	10.9
8	3.2	4.3	0.03	9.5	2.6	3.2	0.03	12.3
9	3.5	5.0	0.03	10.5	2.9	3.4	0.03	13.7
10	3.9	5.7	0.03	11.5	3.2	3.7	0.03	15.1
20	7.2	6.1	0.07	21.7	5.8	7.1	0.07	28.8
30	9.9	7.3	0.11	31.9	8.2	7.4	0.10	42.6
40	12.1	7.6	0.15	42.1	10.2	6.1	0.14	56.3
50	13.6	5.6	0.19	52.2	11.6	5.7	0.18	70.1
100	17.4	1.7	0.40	103.0	16.4	2.8	0.36	138.7
150	18.3	0.7	0.61	153.7	17.9	1.0	0.55	207.3
200	18.6	0.4	0.84	204.3	18.4	0.7	0.74	275.8
300	18.7	0.3	1.23	305.8	18.7	0.3	1.18	412.9

Реальное количество подстрок длиной 50, находящихся на расстоянии редактирования не больше e=3 в исследованных последовательностях, равно 19. В табл. 2 приведены результаты поиска, где в колонке c приведено (усредненное по 100 независимым запускам) количество найденных подстрок. В колонке $\sigma^2(c)$ – дисперсия этого числа. В двух остальных колонках, соответственно, среднее время поиска и затраты памяти.

В работе [19] все 19 подстрок были найдены за 0.4 мин. (на машине Athlon XP 2600+ под оболочкой Cygwin). Как видно из табл. 2 результаты, полученные с помощью метода, основанного на описанной процедуре LSH-лес, в среднем проигрывают результатам [19], что может объясняться слишком короткими строками (n=50), тогда как минимальная длина подстрок, при которой $\rho<1$ (4), равна n=100.

Обнаружение вторжений

Обнаружение хакерских атак является актуальной задачей и, одновременно, сложной, так как поток данных, генерируемых аудит-системами, имеет огромный объем – до гигабайтов в день. Системы обнаружения вторжений (intrusion detection systems, IDS) предназначены для обнаружения попыток несанкционированного доступа в компьютерную систему. Такие системы должны противостоять атакам, даже если злоумышленник с точки зрения соблюдения прав доступа имел необходимые полномочия на свои действия. Одним из подходов к обнаружению атак является создание профиля "нормальной" активности пользователя, а любая активность, не подпадающая под принятое понимание "нормальности", считается опасной. Такие IDS называются системами обнаружения аномалий (anomaly detection systems, ADS). Некоторые существующие подходы к построению ADS рассмотрены в [3].

Одним из направлений обнаружения вторжений есть их обнаружение рассуждениями на основе примеров, где в качестве базы примеров используют

аудит-файлы (логи) пользователей. Используются подходы, оперирующие в каждый момент времени ограниченными участками сессии (окнами). Наиболее востребованы ADS в виде online-систем, которым, очевидно, не доступны будущие команды, а старые логи, как правило, ежедневно или даже ежечасно архивируются, поскольку на их хранение необходимо много ресурсов.

Мы применили приведенный выше подход к поиску генов для обнаружению вторжений в компьютерных системах. Цель эксперимента – проверить, насколько эффективной может быть классификация принадлежности пользовательских сессий (последовательностей системных команд) на основе предложенного метода поиска близких строк.

Роль обучающей выборки, аналогично ситуации с генами, здесь играет множество подпоследовательностей логов сессий. В отличие от задачи поиска генов, где кодирующие экзоны редки и разделены длинными интронами, все команды в логе несут определенную семантическую нагрузку. Поэтому в данной задаче проводилась нарезка и обучающего, и тестового логов на окна одинаковой длины.

Обучающий лог x пользовательской сессии разбивался на пересекающиеся окна вида x[i,i+n–1], i=1,…,|x|–n+1 фиксированной длины n, которые сохраняются для каждого пользователя u в отдельном LSH-лесе F_u. На этапе тестирования создается массив счетчиков C_u для каждого пользователя u. Каждое окно y[j,j+n–1], j=1,…,|y|–n+1 тестового лога y пользователя u* является запросом в F_u для всех присутствовавших в обучающей сессии пользователей u (в эксперименте – 558 пользователей). Обозначим $l_{max}(u,y[j,j+n–1])$ максимальное значение уровня строк, возвращенных в ходе процедуры LSH-лес для леса F_u при запросе y[j,j+n–1]. Если $l_{max}(u,y[j,j+n–1])=K$, то значение счетчика C_u увеличивается на единицу. Пусть U – множество пользователей, таких, что C_u имеет максимальное значение для всех пользователей: $U=\{argmax_u(C_u)\}$. Если $u^* \in U$, то считалось, что пользователь определен правильно. Иначе, пользователь определен неправильно и имеет место аномалия.

Эксперименты проводились на данных, полученных с UNIX сервера физико-технического факультета НТУУ "КПИ", использованных в [4]. Механизмами аудита ОС FreeBSD отслеживались все процессы, которые запускались от имени зарегистрированных в системе пользователей, на протяжении 671 дня (с июня 2001 по декабрь 2003, с перерывами). Всего были получены данные для 717 пользователей, выполнивших 23249986 команд.

В качестве обучающей выборки были приняты все сессии, выполненные за 2001-2002 год (всего 403 дня), в качестве тестовой – сессии 2003 года (268 дней).

Таблица 3

Доля правильно классифицированных сессий для значений ширины окна n=10,20,30,40 и параметров K=5,7, L=1,5,10

	n=10			n=20			n=30			n=40		
L	1	5	10	1	5	10	1	5	10	1	5	10
K=5	0.470	0.984	0.997	0.440	0.942	0.984	0.241	0.787	0.940	0.354	0.848	0.967
K=7	0.431	0.945	0.987	0.403	0.741	0.942	0.327	0.434	-	0.229	0.390	0.836

Результаты (доля правильно классифицированных сессий) для разных значений ширины окна n=10,20,30,40 и параметров K=5,7, L=1,5,10 представлены в табл. 3. Видно, что при увеличении L достигается точность классификации практически 100%. Затраты времени в среднем составляют от 3 до 11 сек. (в зависимости от параметров) на проверку одной сессии. Таким образом, предложенный метод является перспективным в качестве предварительной онлайн-обработки логов.

Выводы

Проведенные эксперименты подтвердили возможность применения рандомизированного метода вложения расстояния редактирования в векторное пространство как метода нахождения приближенно ближайших строк [8]. А именно, показана возможность применения в реальных практических задачах поиска генов и других участков генетических последовательностей, а также в задачах обнаружения вторжений. Учитывая намеренное игнорирование при решении этих задач информации о специфике предметных областей, которые обязательно должны использоваться при решении реальных задач на практике, предложенный метод показал хорошие результаты в обеих задачах и может применяться как составная часть в более специализированных системах.

Перспективным направлением является исследование предложенного подхода на базах компьютерных логов, содержащих следы хакерских атак, для задачи обнаружения вторжений, а также более "осторожных" методов принятия решения относительно тестируемых сессий. В задаче поиска генов представляет интерес исследование объединение предложенного метода с генеративными моделями.

ЛИТЕРАТУРА

1. Соколов А. Исследование ускоренного поиска близких текстовых последовательностей с помощью векторных представлений // Кибернетика и системный анализ. – 2008. – №4.
2. Gusfield D. Algorithms on Strings Trees and Sequences. – Cambridge Univeristy Press, 1997. – 532 p.
3. Соколов А.М. Современные модели обнаружения аномалий в компьютерных системах // Управляющие Системы и Машины.– 2004.– №5.– С. 67-73.
4. Sokolov A.M. An adaptive detection of anomalies in user's behavior // Proc. of the Int. Conference on Neural Networks. – 2003. – 4. – P. 2443-2447.
5. Левенштейн В.И. Двоичные коды с исправлением выпадений, вставок и замещений символов // Докл. АН СССР. – 1965. – Т. 163, – Вып. 4. – С. 845-848.
6. Sokolov A.M, Rachkosvkij D.A. On handling replay attacks in intrusion detection systems // Int. Journal Information Theories & Applications. – 2003. – 10. – №3. – P. 341-347
7. Винцюк Т. К. Распознавание слов устной речи методами динамического программирования // Кибернетика. – 1968.– Вып. 1.– С. 81-88.
8. Соколов А. М. Векторные представления для эффективного сравнения и поиска похожих строк // Кибернетика и системный анализ. – 2007. – №4. – С. 18-38.

9. Indyk P., Motwani R. Approximate nearest neighbors: towards removing the curse of dimensionality // Proc. of STOC. –1998. – P. 604–613.
10. Bawa M., Condie T., Ganesan P. LSH forest: self-tuning indexes for similarity search // Proc. of the Conference on WWW. – 2005. – P. 651-660
11. Zhang MQ. Computational prediction of eukaryotic protein-coding genes // Nat Rev Genet. – 2002. – Vol. 3. – №9. – P. 698-709
12. Sakharkar, M. K., Chow, V. T. and Kangueane, P. Distributions of exons and introns in the human genome. // Silico Biol. – 2004. – 4. – 387-393.
13. Genbank / Benson D.A., Karsch-Mizrachi I., Lipman D.J., Ostell J., Rapp B.A., Wheeler D.L. // Nucleic Acids Research. –2000. – №1. – P. 15-18.
14. Evidence for a locus activation region: the formation of developmentally stable hypersensitive sites in globin-expressing hybrids / Forrester W., Takegawa S., Papayannopoulou T., Stamatoyannopoulos G., Groudine M. // Nucl. Acids Res. – 1987. – 15. – P. 10159–10177.
15. Rogic S., Mackworth A., Ouellette F. B. F. Evaluation of gene-finding programs on mammalian sequences // Genome Research. – 2001. – 11. – P. 817-832.
16. Burset M., Guigó R. Evaluation of Gene Structure Prediction Programs // Genomics. – 1996. – 34. – P. 353-367.
17. Costello E., Wilson D. C. A Case-Based approach to Gene Finding // Workshop on CBR in the Health Sciences. – 2003. – P. 19-28.
18. Buhler J. Efficient large-scale sequence comparison by locality-sensitive hashing // Bioinformatics. – 2001. – 17. – P. 419-428.
19. Narayanan M., Karp R. M. Gapped local similarity search with provable guarantees // Workshop on Algorithms in Bioinformatics. – 2004. – P. 74-86.

1. УДК 51-76+004.056.53

Соколов А.М. Рандомізоване вкладення відстані редагування у задачах пошуку генів і виявлення вторгнень // Системні технології. Регіональний міжвузівський збірник наукових праць. - Випуск 0 (0). - Дніпропетровськ, 2008. - С. 0–0.

У роботі розглядаються застосування рандомізованого методу вкладення класичної відстані редагування до задач пошуку генів та гіперчутливих сайтів, а також до задачі ідентифікації користувачів комп'ютерної системи за логами аудит-системи з метою попередження вторгнень.

Бібл. 19, іл. 2, табл. 4.

UDC 51-76+004.056.53

Sokolov A.M. Randomized edit distance embedding in the tasks of gene finding and intrusion detection // System technologies. – 0 (0). - Dnipropetrovsk, 2008. - P. 0–0.

We apply a randomized classic edit distance embedding method to the tasks of gene and hypersensitive sites finding, and to the task of computer system user identification for preventing intrusions by analyzing system audit logs.

Bib. 19, pic. 2, tabl. 4.

УДК 51-76+004.056.53

Соколов А.М. Рандомизированное вложение расстояния редактирования в задачах поиска генов и обнаружения вторжений // Системные технологии. Региональный межвузовский сборник научных работ. - Выпуск 0 (0). - Днепропетровск, 2008. - С. 0–0.

В работе рассматривается приложение рандомизированного метода вложения классического расстояния редактирования к задаче поиска генов и гиперчувствительных сайтов, а также к задаче идентификации пользователя компьютерной системы по логам аудит-системы з целью предотвращения вторжений.

Библ. 19, ил. 2, табл. 4.

Соколов Артем Михайлович – младший научный сотрудник Международного научно-учебного центра информационных технологий и систем НАН и МОН Украины, г. Киев. Тел. 5026341, sokolov(at)ukr.net