Все это позволяет повысить конкурентоспособность предприятия, т. к. любое предприятие имеет в своей организационной структуре подразделение складского хозяйства, независимо от того, каким видом деятельности оно занимается – торговлей или производством.

СПИСОК ЛИТЕРАТУРЫ

1. Тюльменков В.Н., Замятина О.М. Эффективное управление складом на базе адресной системы хранения // Молодежь и современные информационные технологии: Сб. трудов 4-ой Всеросс. научно-практ. конф. студентов, аспирантов и молодых ученых. – Томск, 2006. – С. 225–227.
2. Шрайбфедер Д. Эффективное управление запасами. – М.: Альпина Бизнес Букс, 2005. – 304 с.
3. Сергейчев А.Ю. Организация и внедрение адресной системы склада // Складские технологии. – 2005. – № 6. – С. 17–19.
4. Соболев А. Управление складом в Microsoft Axapta. – http://www.sibinfo.ru/warehouse.management.axapta.442.aspx. – 04.08.2006.
5. Еременко А., Шашков Р. Разработка бизнес-приложений в Microsoft Buisness Solution Axapta версии 3.0. – М.: Альпина Бизнес Букс, 2005. – 503 с.

УДК 004.65

ПРИМЕНЕНИЕ ВЕРОЯТНОСТНОГО АЛГОРИТМА СОЕДИНЕНИЯ ЗАПИСЕЙ ДЛЯ ИСКЛЮЧЕНИЯ ДУБЛИРОВАНИЯ ИНФОРМАЦИИ В КОРПОРАТИВНОЙ БАЗЕ ДАННЫХ

А.Е. Пинжин

Томский политехнический университет
E-mail: alex_pinjin@tpu.ru

Рассмотрена возможность применения вероятностного алгоритма соединения записей для устранения дублирования информации в базе данных крупной организации или предприятия. Отражены теоретические основы алгоритма, предложены способы оценки степени сходства по основным типам атрибутов, рассмотрены возможности усовершенствования модели путем учета степени достоверности данных, поступающих из разных источников. Приведены практические результаты работы на примере задачи устранения дубликатов записей о физических лицах в единой базе данных российского вуза.

Введение

Процессы хранения и обработки данных являются неотъемлемой частью любой информационной системы (ИС) крупного предприятия или организации. Рост информационных потребностей, обусловленный как внутренними, так и внешними факторами, может быть удовлетворен только в случае наличия непротиворечивых, актуальных и корректных данных.

При разработке единой информационной среды Томского политехнического университета (ЕИС ТПУ), одним из приоритетных направлений было выбрано построение единой базы данных (БД) согласно принципу безызбыточности. Одной из важнейших частей БД ЕИС ТПУ является единая информационная модель личности, представляющая информацию о физических лицах – студентах, сотрудниках, аспирантах и т. д. [1]. Практика показала, что, несмотря на все усилия по созданию безызбыточной БД, могут иметь место случаи дублирования записей о физических лицах, что является прямым следствием отсутствия общего естественного уникального идентификатора. Приведем результаты следующего исследования. Будем считать, что появление одного дубляжа на 2000 записей является допустимым (0,05 %). К началу 2006 г. в БД ТПУ было зарегистрировано около 48000 студентов (среди них около 20000 выпускников) и 5200 сотрудников. В процессе ручной выверки данных было обнаружено около 500 дубликатов, т. е. примерно 0,9 % от общего числа записей. Известно, что сотрудником является в среднем каждый 50 студент-выпускник и каждый 100 действующий студент, т. е. примерно 1,3 % студентов являются сотрудниками. Таким образом, ручная выверка данных не дала приемлемых результатов. Отметим также высокую трудоемкость процесса выверки – описанные выше результаты были достигнуты усилиями пяти сотрудников в течение нескольких месяцев.

В рамках базы данных корпоративной информационной системы можно выделить следующие типичные ситуации, порождающие ошибки идентификации:

1. Малое пересечение свойств с совпадающими значениями. Например, о двух лицах известны значения свойств «Фамилия Имя Отчество», «Телефон» и «Дата рождения», и значения первых двух свойств совпадают, а значения третьего свойства не совпадают. В таких условиях сложно принять решение об идентичности.
2. Ошибки в значениях атрибутов, т. е. несовпадение фактических и зарегистрированных значений. Можно выделить следующие виды ошибок – опечатки при вводе (например, неверное на-

писание фамилии), ошибки в результате потери или искажения данных, ошибки, связанные с разницей во времени актуализации значений, возникающие, например, при смене фамилии или адреса.

В подобных условиях использование экспертных знаний пользователя не может быть полностью исключено из процесса идентификации. Целесообразным решением является разработка автоматизированной системы поиска потенциальных дубляжей, которая способна оказать содействие пользователю при принятии решения об идентичности записей. Данная статья посвящена описанию формального алгоритма, который мог бы являться основой при создании подобной системы.

1. Основные этапы процесса выявления и устранения дубликатов

Методы поиска и устранения совпадающих записей активно разрабатываются в течение последних десятилетий такими зарубежными исследователями, как Newcombe [2], Fellegi и Sunter [3], Winkler [4], Jaro [5] и др. Особое значение эти вопросы приобрели в последнее время при решении задач интеграции разнородных хранилищ данных, подготовке данных для анализа и «добычи данных» (Data Mining). Следует отметить, что в последние годы возрастает интерес к проблеме и в отечественных источниках [6–8].

Согласно [9], стандартная система слияния записей состоит из блоков, представленных на рис. 1.

Этап стандартизации включает в себя приведение данных к общей структуре и типам. *Этап сегментации* выполняется в тех случаях, когда с практической точки зрения является нерациональным производить сравнение всего множества пар записей, поэтому производится его разделение по какому-либо признаку, и оценка сходства выполняется над выделенным сегментом. Стандартизация и сегментация данных являются важными этапами, однако подробное рассмотрение этих вопросов выходит за рамки статьи.

Сравнение и принятие решения о слиянии записей, является самой сложной частью процесса. Существуют различные методы, применимые к этой задаче, такие как нейронные сети, кластерный анализ и др. Наиболее подходящими для решения поставленной задачи представляются *вероятностные алгоритмы*, обеспечивающие получение интервальной оценки сходства на основе анализа значений атрибутов. Эти алгоритмы широко освещены в зарубежной литературе и применялись при проведении переписи населения, интеграции медицинских, почтовых БД. Основными их достоинствами является наличие формального аппарата, относительно простая реализация, возможность обработки пропущенных и ошибочных значений. Кроме того, в крупных организациях обычно имеется возможность получить необходимые для реализации алгоритма тестовые выборки данных, а также оценки достоверности значений, поступающих из разных источников.

2. Стандартная вероятностная модель слияния записей

Математическая модель вероятностного алгоритма слияния записей была впервые предложена Fellegi и Sunter [3]. Пусть даны два множества объектов реального мира A и B, элементы которых обозначим как a и b соответственно. Некоторые элементы являются общими для A и B. Представим множество $A\times B=\{(a,b);a\in A,b\in B\}$ в виде двух подмножеств M и U. Если a и b, входящие в элемент множества $A\times B$ совпадают (являются одним и тем же объектом), то этот элемент принадлежит M, иначе он принадлежит U.

Представим, что в результате ввода информации об элементах A и B в БД были получены файлы записей L_A и L_B. Обозначим записи, соответствующие элементам множеств A и B, как $\alpha(a)$ и $\beta(b)$. Результатом сравнения двух записей является вектор сравнения, состоящий из таких элементов, как, например «Имя совпадает», «Дата рождения не совпадает» и т. п. Вектор сравнения определяется как векторная функция над $\alpha(a)$ и $\beta(b)$:

$$\gamma[\alpha(a),\beta(b)]=\{\gamma^1[\alpha(a),\beta(b)],...,\gamma^k[\alpha(a),\beta(b)]\}.$$

В дальнейшем будем использовать запись $\gamma(a,b)$, $\gamma[\alpha,\beta]$, или просто γ. Обозначим вероятность $m(\gamma)$ того, что вектор сравнения γ отражает совпадение значений, при условии, что пара записей представляет один и тот же объект

$$m(\gamma)=P(\gamma\mid(a,b)\in M)$$

и вероятность $u(\gamma)$ того, что γ отражает совпадение, если пара записей представляет различные объекты

$$u(\gamma)=P(\gamma\mid(a,b)\in U).$$

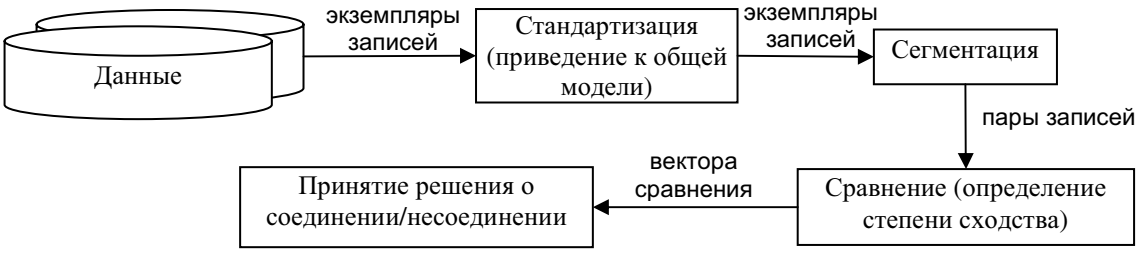

Рис. 1. *Процесс поиска и устранения дубликатов*

Отношение $m(\gamma)/u(\gamma)$ будем называть *степенью сходства записей*.

На практике количество реализаций γ может быть настолько большим, что становится необходимым принятие некоторых допущений. На основе подхода, предложенного в [3], предположим, что компоненты γ могут быть упорядочены и

$$m(\gamma) = m_1(\gamma^1) \cdot m_2(\gamma^2) \cdot \ldots \cdot m_K(\gamma^K),$$
$$u(\gamma) = u_1(\gamma^1) \cdot u_2(\gamma^2) \cdot \ldots \cdot u_K(\gamma^K).$$

Например, γ^1 может включать в себя атрибуты «Фамилия Имя и Отчество», а γ^2 – атрибуты адреса.

Многие авторы считают эффективным использование логарифма полученной оценки для обеспечения аддитивности весов, что позволяет определить

$$w^i(\gamma^i) = \log\left(\frac{m(\gamma^i)}{u(\gamma^i)}\right).$$

Таким образом, можно записать

$$w(\gamma) = w^1 + w^2 + \ldots + w^K$$

и использовать $w(\gamma)$ в качестве *веса соединения*. Вес соединения означает степень сходства записей и является отражением вероятности того, что две записи представляют один и тот же объект.

3. Вычисление весов в условиях корпоративной информационной системы

На практике для вычисления весов может быть использован метод, основанный на частоте вхождения искомого значения в исходную выборку [3, 4]. Интуитивно понятно, что, например, в российском вузе имя «Николай» встречается чаще, чем «Эдуард», поэтому совпадение в первом случае будет иметь меньший вес, чем во втором.

Предположим, что одним из атрибутов записей является фамилия, и мы можем построить список всех безошибочных значений фамилии, а также количество лиц, обладающих каждой из этих фамилий в некоторой тестовой выборке. Пусть пропорция вхождения j-й фамилии в это множество равна p_j. Введем следующие обозначения:

- e_A и e_B – вероятности ошибочного значения фамилии в L_A или L_B.
- e_T – вероятность того, что значения фамилии личности в L_A и L_B различны, однако записаны без ошибок (например, человек сменил фамилию).

Согласно [3, 4] можно сформулировать основные правила для вычисления весов:

- w (фамилия совпадает и является j-й фамилией в списке) $= \log\left(\frac{1}{p_j}\right)$ (1)
- w (фамилия не совпадает) $= \log\left(\frac{e_A + e_B + e_T}{1 - \sum_j p_j^2}\right)$ (2)
- w (фамилия не указана в одной из записей)=0. (3)

Из (1–3) следует, что совпадение по фамилии приведет к появлению положительного веса, и чем реже встречается фамилия, тем больше будет вес; несовпадение приводит к отрицательному весу, который уменьшается с ошибками e_A, e_B, e_T; если фамилия не указана в одной из записей, вес будет равен нулю.

Первичные значения пропорций p_j, в условиях корпоративной информационной системы, могут быть вычислены на этапе ввода первичных данных. Для повышения производительности алгоритма, имеет смысл хранить постоянный список пропорций в БД (рис. 2).

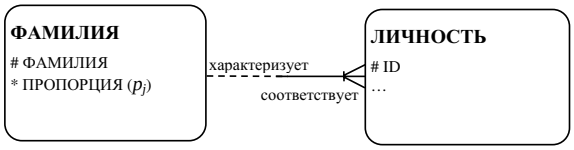

Рис. 2. *Пример концептуальной схемы данных для хранения пропорций*

Первоначальный список пропорций должен вычисляться на основании выборок, обладающих минимальной избыточностью. По мере наполнения БД, через установленные промежутки времени или при накоплении определенного количества новых записей, список пропорций должен обновляться.

Важной практической задачей является установка значений e_A, e_B и e_T. Характерной особенностью многих организаций является наличие множества источников актуализации объектов одного и того же типа. Каждый из источников обладает разной степенью достоверности вводимой информации, следовательно, множества L_A и L_B представляют собой совокупность подмножеств, для каждого из которых можно определить собственные значения e_A, e_B и e_T в отношении того или иного атрибута. Например, представим множество L_B, означающее совокупность зарегистрированных в БД физических лиц, в виде $L_{B_1} \cup L_{B_2} \cup \ldots \cup L_{B_n}$, где L_{B_i} является подмножеством личностей, актуализируемых i-м видом учета. Т. к. каждому виду учета можно сопоставить вероятность ошибки в значении того или иного свойства, то, например, для фамилии $E_B^{фам} = \{e_{B_1}^{фам}, e_{B_2}^{фам}, \ldots, e_{B_n}^{фам}\}$.

В случаях, когда одна и та же личность обладает несколькими ролями, есть возможность избежать множественного соответствия между L_{B_i} и видами учета и, соответственно, несколькими значениями ошибок. В этом случае можно использовать значение ошибки, соответствующее виду учета, обладающему максимальным приоритетом по отношению к свойству личности, включающему в себя атрибут, используемый для сравнения. Таким образом, мы полагаем, что подразделение с наивысшим приоритетом несет наибольшую ответственность за безошибочность значения этого атрибута [10].

Для множества личностей L_A можно применить следующие формулировки при:

1. создании новой личности: $L_A=\{l'\}$, e_A – соответствует виду учета, выполняющему операцию создания;
2. поиске дубликатов среди зарегистрированных личностей: $L_A=L_B$;
3. выполнении импорта данных из автономного источника в БД ЕИС: $L_A=\{l_1,l_2,...,l_m\}$, e_A необходимо определить для этого источника.

Таким образом, алгоритм сравнения пар записей, входящих в $L_A\times L_B$, может быть представлен в виде задачи сравнения пар, входящих в множества $L_A\times L_{B_1}, L_A\times L_{B_2},...,L_A\times L_{B_n}$ на соответствующих уровнях ошибок e_A и $e_{B_1},e_{B_2},...,e_{B_n}$, определенных для каждого атрибута.

Описанный алгоритм вычисления весов применим не только к фамилиям, но и к прочим атрибутам физического лица, имеющим сравнительно небольшой домен значений по отношению к количеству зарегистрированных экземпляров, таких как, Фамилия, Имя, Отчество, Паспортные данные, Название законченного учебного заведения, Страна, Город адреса личности и т. д.

4. Вычисление весов для атрибутов интервального и иерархического типа

Если атрибут принимает не дискретные, а интервальные значения, то вычисление пропорций может принять несколько иной характер. Рассмотрим возможный подход к вычислению пропорций на примере атрибута «Дата рождения физического лица».

Очевидно, что расчет пропорций согласно описанному ранее принципу приведет к низким значениям частот, так как размер домена возможных значений атрибута близок к количеству зарегистрированных личностей, хотя, на интуитивном уровне ясно, что в вузе значительно больше субъектов, обладающих возрастом 20–30 лет, чем обладающих возрастом 10–15 лет. Следовательно, вес совпадения даты рождения в первом случае должен быть меньше, чем во втором.

Рассмотрим практический способ вычисления пропорций. Разобьем область возможных значений атрибута на несколько интервалов и произведем расчет пропорций на этих интервалах. Информация о распределении этих пропорций может быть сохранена в БД в виде структуры, показанной на рис. 3.

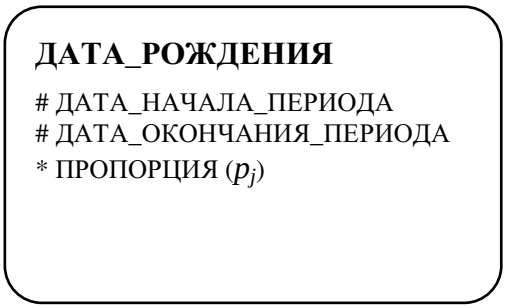

Рис. 3. *Пример концептуальной схемы данных для хранения пропорций атрибутов интервального типа*

При сравнении атрибутов типа Дата, если значение подчиняется определенному формату, например «дд-мм-гггг», может быть применен алгоритм расчета расстояния Хемминга [11].

Алгоритм вычисления весов на основе пропорций может быть адаптирован и для свойств иерархичного характера. В существующих источниках проблема соединения рекурсивных записей обычно не рассматривается, так как предполагается слабая структуризация исходных данных. Однако в БД ЕИС, например, свойство Адрес личности может быть представлено в виде структуры, рис. 4. Модель включает в себя сущность ЭЛЕМЕНТ_АДРЕСА, содержащую рекурсивную ссылку. Безусловно, на практике модель адресов может отличаться, однако мы предполагаем, что другие реализации могут быть приведены к представленной.

Рассмотрим две сравниваемые по адресу записи и обозначим соответствующие наборы атрибутов адреса: $a=\{a_1,a_2,...,a_n\}$ и $a'=\{a'_1,a'_2,...,a'_n\}$. Элементы этих множеств упорядочены по вложенности и представляют собой значения компонентов адреса. Так, a_1 может обозначать страну, a_2 – регион и т. д.

Для элементов адреса может быть произведен расчет пропорций исходя из количества ссылающихся на них адресов личности, при этом наличие ссылки на определенный элемент адреса считаем как наличие ссылки и на старший элемент. В результате каждому элементу конкретной реализации адреса можно поставить в соответствие значения пропорций $p=\{p_1,p_2,...,p_n\}$, причем $p_i>p_j$, при $i<j$.

Далее, суммарный вес совпадения рассчитывается как сумма весов по каждой паре значений (a_i,a'_i), вычисленных по формулам (1–3). Таким об-

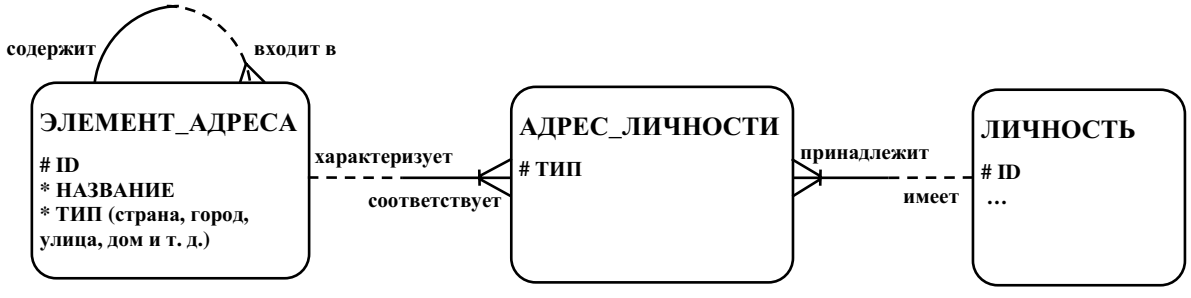

Рис. 4. *Концептуальная схема адреса личности*

разом, совпадение значений более детальных элементов адреса будет давать больший вес по сравнению с совпадением менее детальных элементов. Несовпадение элемента адреса уменьшает суммарный вес.

5. Практические результаты

Алгоритм поиска дублирующихся записей был применен в БД ТПУ для поиска дубликатов среди существующих записей. На момент испытаний, проводившихся в июле 2006 г., в БД было зарегистрировано 48655 записей – 43476 студентов и 5179 сотрудников. Так как было известно, что степень избыточности относительно невелика (1...3 %), пропорции вычислялись непосредственно из БД. Для вычисления степени сходства строк использовались два алгоритма: расстояние Хемминга [11] и расстояние Левенштейна [12]. Исходные данные, условия сравнения и результаты вычисления пропорций приведены в табл. 1.

Таблица 1. *Исходные данные для вычисления пропорций*

Свойство	Количество уникальных знач. свойства	Условие совпадения	Продолжительность вычисления (Pentium 2.4, 512 Мб)
Фамилия	27000	Равенство первой буквы и расст. Левенштейна <2	4 ч
Имя	1500	Расст. Левенштейна <3	25 мин.
Отчество	2000	Расст. Левенштейна <3	30 мин.
Дата рождения	с 1900 по 2007 г., интервал 1 год	Вхождение в интервал, расст. Хемминга <2	2 мин.
Пол	2	Совпадение кода	0,5 мин.
Адрес (с точностью до населенного пункта)	2415	Совпадение кода	7 мин.

Примечание: Продолжительность вычисления зависит от способа реализации и мощности вычислительных ресурсов, поэтому результаты приведены исключительно для сравнительной оценки

Суммарная продолжительность расчета пропорций составила около 5 ч и была признана удовлетворительной, с учетом того, что расчет предполагается производить раз в месяц. В табл. 2 приведен фрагмент таблицы пропорций по именам.

Таблица 2. *Фрагмент таблицы пропорций*

АЛЕКСАНДР	0,0534473
Александр	0,0534473
Аександр	0,0534287
Алекесандр	0,0534287
Александрос	0,0532983
Ольга	0,0489018
Ольга	0,0489018
ольга	0,0489018
Наталия	0,0417668
Наталья	0,041655
Наталья	0,041655
наталья	0,041655

Далее была выполнена сегментация данных, включающая следующие условия:

- Совпадение по признаку Пол.
- Расстояние Хемминга по дате рождения меньше трех.
- Совпадение по первой букве имени и отчества.

Выполнение алгоритма позволило снизить объем выборки с $(48655^2-48655)/2=1183630185$ до 842030 пар. Время работы алгоритма сегментации составило около 15 мин.

Сравнение записей производилось по перечисленным в табл. 1 признакам. Время выполнения составило приблизительно 2 ч. В табл. 3 приведен фрагмент результатов сравнения пар записей с высокими (больше 10) результирующими суммарными весами.

В результате экспертной оценки полученных результатов, значение 20 было установлено в качестве порогового суммарного веса, при котором пара записей считается совпадающей. Такому условию удовлетворяет 532 пары записей, что составляет около 1 % от общего количества записей. Сопоставляя этот результат с данными, приведенными в начале статьи, такие результаты можно считать удовлетворительными. Следует отметить, что при установке порогового значения 15, неверные совпадения (ошибки первого рода) наблюдаются с частотой примерно 1:10, и результаты выполнения алгоритма требуют дополнительного экспертного анализа.

В итоге проделанной работы вероятностный метод был признан достаточно эффективным средством поиска дубликатов информации о физиче-

Таблица 3. *Фрагмент таблицы результатов вычисления весов (w)*

ФИО	w фам.	w имя	w отч.	Дата_рожд.	w д._р.	Пол	w пол	Адрес	w адр.	W_сум.
Сидоров Федор Петрович	6,747	3,147	2,915	01.11.1972	2,708	2	0,693	NULL	0,000	16,210
Сидоров Федор Петровч				01.11.1962		2		NULL		
Иванов Иван Владимирович	-4,42	4,358	2,821	22.02.1988	3,158	1	0,000	Барнаул	4,185	10,100
Петров Иван Владимирович				21.04.1988		NULL		Барнаул		
Сидорова Наталья Александровна	8,118	3,178	2,915	03.03.1981	0,000	2	0,693	NULL	0,000	14,904
сидорова Наталия Александровна				NULL		2		Томск		
Акмедов Сергей Сергеевич	8,588	6,78	4,644	18.03.1958	5,208	2	0,693	Томск	0,000	25,913
Ахмедов Сергей Сергеевч				18.03.1959		2		NULL		

ских лицах. Дальнейшие усилия по решению проблемы избыточности будут направлены на повышение производительности реализованной системы, а также на апробацию альтернативных алгоритмов сравнения атрибутов записей в рамках базового вероятностного метода.

СПИСОК ЛИТЕРАТУРЫ

1. Чудинов И.Л., Пинжин А.Е., Исаев И.В. Об одном подходе к построению информационной модели личности в системах организационного управления // Современные средства и системы автоматизации: Труды IV Научно-практ. конф. – Томск, ТУСУР, 2004. – С. 267–269.
2. Newcombe H.B., Kennedy J.M., Axford S.J., James A.P. Automatic Linkage of Vital Records // Science. – 1959. – V. 130. – № 3381. – P. 954–959.
3. Fellegi L., Sunter A. A Theory for Record Linkage // Journal of the American Statistical Society. – 1969. – V. 64. – № 328. – P. 1183–1210.
4. Winkler W.E. Frequency-Based Matching in Fellegi-Sunter Model of Record Linkage. – Technical Report RR/2000/06, Statistical Research Report Series. – Washington: US Bureau of the Census, DC, 2000. – 14 p.
5. Jaro M.A. Advances in Record Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida // Journal of the American Statistical Society. – 1989. – V. 84. – № 406. – P. 414–420.
6. Карпов В.Э., Карпова И.П. Об одной задаче очистки и синхронизации данных // Информационные технологии. – 2002. – № 9. – С. 25–32.
7. Процедура идентификации личности при отсутствии идентификатора персональных данных: Проект стандарта / Институт архитектуры электронного государства. Разработчик Черенов Ц.В., Бойченко Е.В., Михеев А.В., Одинцова Н.П. – 2006. – 25 с. – [Электронный ресурс] – Режим доступа: – http://www.iaeg.ru/62088.
8. Цыганов Н.Л. Проблемы очистки и избежания дублирования персональных данных с помощью методики нечеткого сопоставления в практике Европейской Организации Ядерных Исследований // Науч. сессия МИФИ-2005: Сб. науч. тр. – М.: МИФИ, 2005. – Т. 12. – С. 192–193.
9. Gu L., Baxter R., Vickers D., Rainsford C. Record linkage: Current practice and future directions. – Technical Report 03/83, CSIRO Mathematical and Information Sciences. – Canberra, ACT 2601, Australia, 2003. – 32 p.
10. Паршин Д.А., Пинжин А.Е. Разграничение приоритетов доступа к свойствам объектов в условиях единой информационной среды вуза // Теоретические и прикладные вопросы современных информационных технологий: Труды VII Научно-техн. конф. – Улан-Удэ, ВСГТУ, 2006. – С. 292–297.
11. Hamming R.W. Error-detecting and error-correcting codes // Bell System Technical Journal. – 1950. – V. 29. – № 2. – P. 147–160.
12. Левенштейн В.И. Двоичные коды с исправлением выпадений, вставок и замещений символов // Доклады АН СССР. – 1965. – Т. 163. – № 4. – С. 845–848.

УДК 519.245:519.688

ПРИМЕНЕНИЕ АДАПТИВНОГО БИНОРМАЛЬНОГО РАСПРЕДЕЛЕНИЯ В МЕТОДЕ ПОИСКА ГЛОБАЛЬНОГО МИНИМУМА SIMULATED ANNEALING

А.А. Хамухин

Томский политехнический университет
E-mail: alex@ad.cctpu.edu.ru

В методе поиска глобального минимума Simulated Annealing предложено использовать бинормальное распределение плотности вероятности следующего шага, моды которого смещены относительно текущего локального минимума, а расстояние между ними и дисперсии функционально связаны с искусственной температурой. Показана эффективность реализаций подхода с помощью численных расчетов.

Метод поиска глобального минимума, известный в литературе под названием Simulated Annealing (SA), или метод «имитации отжига» применяется при построении математических моделей и решения сложных оптимизационных задач в нейрокомпьютерной технике, нефтегазогеологии, микроэлектронике, микробиологии, ядерной физике и др. [1–5]. Метод SA, предложенный Киркпатриком в 1982 г. [6], имеет различные алгоритмические реализации, некоторые из которых включены в такие известные вычислительные пакеты, как Mathematica NMinimize, STATISTICA Neural Networks и ряд других. Главным его достоинством является теоретическое доказательство сходимости к глобальному минимуму при использовании распределения Больцмана [1]. Однако практический сравнительный анализ методов глобальной оптимизации показал, что метод SA является наиболее «хрупким», т. е. заметно зависит от выбранных параметров поиска и нуждается в дополнительной настройке [6]. Так, например, в работе [7] утверждается, что не существует универсального по эффективности алгоритма для разных задач глобальной оптимизации и задача их разработки и модификации остается открытой.

Целью работы автора является создание эффективных инструментальных средств настройки па-