УДК 004.6:004.021

Д.В. Бабин
Институт проблем искусственного интеллекта, г. Донецк, Украина
dmitry.v.babin@gmail.com

Генетический алгоритм решения задачи анализа рыночной корзины

В данной работе рассматривается генетический алгоритм решения задачи анализа рыночной корзины (market basket analysis). Проводится исследование эффективности данного алгоритма, а также возможность его использования на больших объемах входной информации. Определяются оптимальные параметры для алгоритма.

Введение

Задача анализа рыночной корзины представляет собой одну из задач добычи данных (data mining) и, как и прочие задачи извлечения знаний [1], анализ рыночной корзины ориентирован на исследование больших собраний информации в поисках тенденций, шаблонов и взаимосвязей, способных помочь в принятии стратегических решений.

Название этого метода происходит от задачи определения: какие товары, вероятно, покупаются совместно. Однако реальная область его применения значительно шире [2]. В общем виде данная задача призвана выявить ассоциативные направленные правила типа: если произошло событие «A» (например, покупатель купил принтер), то с какой вероятностью произойдет еще и событие «B» (например, покупка картриджа) и еще событие «C» (покупка фотобумаги) и т.д. Можно переформулировать данную задачу иначе – если произошло событие «A», то возникновение каких событий это может за собой повлечь, иными словами, какие наиболее популярные товары у покупателей принтера. Такая информация часто представлена покупателям Интернет-магазинов, однако товары, представленные в категории «Вместе с этим товаром наиболее часто покупают ...», не связаны между собой, а связаны только с покупаемым товаром, т.е. рассматриваются 2-элементные наборы.

Одним из первых алгоритмов, решающих задачу анализа рыночной корзины, был алгоритм Apriori [3], который в дальнейшем был усовершенствован некоторыми исследователями. В настоящий момент существует также множество программных продуктов (PolyAnalyst, Intelligent Miner, Data Miner, SGI Miner и другие [4]), которые решают задачу анализа рыночной корзины. Однако то, что эти продукты являются коммерческими, делает использованные в них алгоритмы недоступными широкой общественности, а сами системы часто не предназначены для работы на персональных компьютерах.

Постановка задачи. Будем рассматривать задачу анализа рыночной корзины в следующем варианте постановки. Исходными данными для задачи являются:

– база данных, содержащая информацию о транзакциях в виде, приводимом к следующей структуре – {tid – код транзакции, iid – код товара (товарной группы)} (на самом деле большинство современных баз данных используют для хранения информации о транзакциях именно такую структуру, дополненную сведениями о количестве и цене приобретаемого товара, а также другой, необходимой для учета и не касающейся данной задачи, информацией);
– размер анализируемой рыночной корзины.

Дополнительно, если это необходимо, задается множество заранее определенных товаров (множество D), такая необходимость может возникнуть, если предстоит выявить, какие товары наиболее часто покупаются совместно с заданными для анализа пользователем товарами.

Результатом работы алгоритма станет набор товаров, которые наиболее часто покупаются совместно, учитывая заданный пользователем набор товаров (если он указан).

Термины и определения. Рыночная корзина – это набор товаров, приобретенных покупателем в рамках одной отдельной транзакции. N-элементный набор – набор, состоящий из N товаров. T – множество товаров, участвующих в транзакциях. Товарная группа – группа товаров со сходными для пользователя характеристиками, которые при анализе считаются одним товаром.

Описание алгоритма

Учитывая формат исходных данных удобно кодировать информацию о товарных наборах в следующем виде. В качестве хромосомы используем вектор чисел V размером N, где N – количество товаров в искомом наборе. Каждый элемент вектора V[i], (i=1..N), представляет собой число, ключ товара, из множества T – множества всех ключей товаров iid, которые представлены в заказах. Таким образом, вектор V=<1,3,5> будет обозначать наличие в заказе товаров с ключами 1, 3 и 5 и то, что производится поиск товарного набора размером N=3. Если пользователь задал определенные товары, то при анализе они входят в каждый из векторов V_i, учитываются при вычислении целевой функции, однако не участвуют в операциях генетического алгоритма.

Целевая функция. Поскольку целевая функция должна отражать как частоту встречаемости i-элементного товарного набора, являющегося подмножеством искомого N-элементного набора, так и должна зависеть от размера данного подмножества, то удобно использовать следующую функцию

$$F = \sum_{i=2}^{N} k[i] * i,$$

где N – длина вектора, k[i] – частота встречаемости товарного набора размером i.

Кроме того, вычисление значения такой функции легко реализовывается средствами SQL, что позволяет проводить вычисления непосредственно на сервере баз данных, не затрачивая времени на передачу данных.

Генетический алгоритм выполняется по классической схеме:

– селекция;
– скрещивание;
– мутация.

Начальная популяция генерируется следующим образом:

– каждая хромосома обязательно содержит в начале элементы множества D (если оно задано);
– остальные позиции заполняются случайным образом ключами товаров из множества T.

Селекция проводится по пропорциональной схеме. Число потомков прямо пропорционально зависит от значения целевой функции для текущего вектора.

Скрещивание проводится по следующей схеме:

– из популяции случайным образом выбираются два вектора;
– для каждой позиции вектора i, i = d..N, с заданной вероятностью вектора обмениваются значениями (d – позиция первого, не принадлежащего множеству D, ключа товара в векторе V, если множество D не задано d = 1).

Для проведения операции мутации в векторе случайным образом выбирается позиция в интервале [d; N], значение в которой заменяется выбранным случайным образом элементом множества T.

Критерием останова алгоритма служит не улучшение целевой функции на протяжении заданного числа шагов. Кроме того, после исследования алгоритма можно выяснить количество итераций алгоритма, за которые получается оптимальное решение, и в дальнейшем останавливать алгоритм, опираясь на это число.

Для оптимизации алгоритма можно провести предварительную сортировку исходного множества ключей товаров T по количеству вхождений в транзакции и при выборе товара в набор отдавать приоритет товарам с большим количеством вхождений. Такой подход показывает свою эффективность при решении задач с небольшим значением длины вектора, при больших же значениях N в итоговый вектор могут попасть редко встречаемые в целом в базе данных ключи товаров, однако, стабильно покупаемые совместно.

Использование в качестве входных данных ключей товара не всегда является эффективным с точки зрения пользователя. Это может быть связано как с тем, что могло произойти товарозамещение (например, вместо клавиатур одной марки стали продавать такие же по характеристикам клавиатуры, но уже другого производителя, либо обновился модельный ряд), так и с тем, что пользователю часто бывает необходимо выяснить, какие товары покупаются совместно, например, с клавиатурами в целом, и ему не важно, какие это были клавиатуры (какая модель, кто производитель, какой цвет). Таким образом необходимо ввести понятие товарных групп, которые будут использованы при анализе как один элемент сформированного множества T_g. В этом случае в качестве ключа товара iid будет использован ключ товарной группы. Кроме того, что это приблизит результаты анализа к потребностям пользователя, это еще и ускорит сам процесс анализа (т.к. размер множества T_g будет меньше либо равен размеру множества T). Процесс определения товарных групп возлагается непосредственно на пользователя, но, учитывая, что современные учетные системы, как правило, хранят информацию о товарах уже с разбивкой на категории, то дополнительных действий со стороны пользователя не потребуется.

Экспериментальные исследования алгоритма

Для проведения экспериментальных исследований генетический алгоритм был реализован программно. Анализ эффективности алгоритма проводился на базе данных, содержащих сведения о 30579 товарах (количество элементов множества T), распределенных по 307 товарным группам (количество элементов множества T_g), и 526016 заказах (транзакциях). Размер таблицы, содержащей информацию о товарах, купленных в рамках одной транзакции, – 2046033 записей. Предметная область базы данных – информация о покупках компьютерных комплектующих, оргтехники, периферийных устройств, компьютерного оборудования и т.п.

Исследования проводились на корзинах различной вместимости: 3, 10, 15 элементов с целью выяснения оптимальных значений для характеристик алгоритма:

– размер популяции;
– вероятность скрещивания;
– вероятность мутации;
– необходимое количество поколений.

Поскольку с увеличением размера популяции количество генерируемых поколений для получения результата уменьшается, то соотношение этих двух характеристик рассчитывалось исходя из минимального времени работы алгоритма.

Многочисленные испытания алгоритма с использованием различных параметров показали следующую зависимость параметров алгоритма от размеров корзин: с увеличением размера анализируемой корзины вероятности скрещивания и мутации

должны уменьшаться. Аналогичное утверждение справедливо и для размеров популяции: с увеличением размеров популяции стоит уменьшить указанные вероятности. Также необходимо учитывать, что количество ключей товаров, используемых при анализе на каждом шаге алгоритма, должно быть больше или равно количеству элементов множества T. То есть должно соблюдаться правило

$$H \cdot N \geq |T|,$$

где H – размер популяции (количество хромосом), N – размер искомого товарного набора (количество элементов вектора V), |T| – мощность множества T (количество элементов множества).

В целом исследования показали:

– 30 – 50 итераций алгоритма позволяют получить искомый набор товаров в случае использования предварительной сортировки и 60 – 90 итераций для несортированных данных;
– оптимальная вероятность скрещивания колеблется от 0,4 до 0,5;
– оптимальная вероятность мутации лежит в пределах от 0,05 до 0,1.

Выводы

Разработанный алгоритм представляет практический интерес в первую очередь для предприятий розничной торговли при проведении анализа покупательского спроса.

Преимуществом данного алгоритма является возможность в большинстве случаев использовать его непосредственно для накопленных о транзакциях данных, не проводя преобразование данных. В разработке алгоритма использован современный, генетический подход, что в свою очередь наделяет данный алгоритм преимуществами, присущими данному классу алгоритмов. В частности, генетические алгоритмы довольно легко распараллеливать. Один из способов описан в [5] и заключается в том, чтобы разбить поколение на несколько групп и работать с каждой из них независимо, обмениваясь время от времени несколькими хромосомами.

Литература

1. Кречетов Н.В. Продукты для интеллектуального анализа данных // Рынок программных средств. – 1997. – № 14. – С. 32-39.
2. Киселев М.В., Соломатин Е.А. Средства добычи знаний в бизнесе и финансах // Открытые системы. – 1997. – № 4. – С. 41-44.
3. Ганти Венкатеш, Герке Йоханнес, Рамакришнан Раджу. Добыча данных в сверхбольших базах данных // Открытые системы. – 1999. – № 9. – С. 36-48.
4. Коржов В.Н. Data Mining по-русски // ComputerWorld. – 2000. – № 34. – С. 54-62.
5. Дюк В.Д. Data Mining – состояние, проблемы, новые решения // Открытые системы. – 1999. – № 3. – С. 12-24.

Д.В. Бабін

Генетичний алгоритм розв'язання задачі аналізу ринкового кошику

У даній роботі розглядається генетичний алгоритм вирішення задачі аналізу ринкової корзини (market basket analysis). Проводиться дослідження ефективності даного алгоритму, а також можливості його використання на великих обсягах вхідної інформації. Визначаються оптимальні параметри алгоритму.

D.V. Babin

The Genetic Algorithm of the Decision of a Market Basket Analysis Task

The genetic algorithm of the decision of a market basket analysis task is considered in this research. To be carried out research of efficiency of the given algorithm, and also an opportunity of its use on great volumes of the entrance information. Optimum parameters for algorithm are defined.

Статья поступила в редакцию 27.06.2006.