PRACTICE OF WEB DATA MINING METHODS APPLICATION

WEB DATA MINING METOŽU LIETOŠANAS PRAKSĒ

P.Osipovs and A.Borisov

Keywords: web data mining, duplicate document detection, internet users behavior paterns

Современные темпы роста количества информации в Internet предъявляют высокие требования к эффективности алгоритмов её обработки.
В данной статье рассмотрены некоторые алгоритмы из области Web Data Mining, доказавшие свою эффективность во многих существующих приложениях.
Статья разбита на две логические части, в первой рассматривается теоретическое описание алгоритмов, а во второй примеры их успешного использования в для решения реальных задач.
Алгоритмы поиска нечётких дубликатов документов активно используются всеми ведущими поисковыми системами в мире, приведены описания следующих алгоритмов: шинглы, сигнатурные методы, алгоритмы базирующиеся на изображениях.
Такие методы классификации как кластеризация методом нечётких к-средних (Fuzzy c-means/FCM clustering) и кластеризация методом муравьиной колонии (Standard Ant Clustering Algorithm SACA).
В заключении описан опыт успешного применения методов нечёткой кластеризации совместно с программным инструментальным пакетом DataEngine для повышения эффективности работы банка «BCI Bank» а также совместное использование кластеризации методом муравьиной колонии совместно с линейным генетическим программированием для решения увеличения эффективности прогнозирования нагрузки на серверы высоконагруженного Internet портала института Монаш.

Введение

С развитием Internet значимость методов Web Data Mining постоянно увеличивается. Разрабатываются и внедряются всё новые, зачастую неожиданные варианты применения методов Data Mining для Web. В данной статье рассмотрен опыт применения различных алгоритмов для решения некоторых задач Web Data Mining. Статья написана на основе анализа публикаций в рассматриваемой области.

Рассмотренные алгоритмы успешно применяются в сферах банковских услуг, бизнес исследований, медицинских исследованиях, используются крупнейшими Internet поисковыми системами.

Алгоритмы поиска нечётких дубликатов документов:

- шинглы;
- сигнатурные методы;
- алгоритмы базирующиеся на изображениях.

В рамках анализа статьи [1] о практическом применении методов поиска поведенческих шаблонов у пользователей банковского Internet портала приведены описания

- метода нечёткой кластеризации Fuzzy c-means;
- инструментального пакета Data mining: «*DataEngine*».

В работе рассмотрен пример успешного использования кластеризации методом муравьиной

колонии совместно с линейным генетическим программированием в реальных проектах[2], также были описаны основы методов муравьиной колонии и рассмотрен пакет Discipulus™ [3].

Поиск нечётких дубликатов документов

Задачи

- Анализ уникальности документов (duplicate document detection DDD [4]) поисковыми машинами в Internet (Google, Bing, Yandex) для предотвращения лишней индексации одинаковых документов;
- Выявление случаев плагиата;
- Архивирование документов (уменьшение используемого пространства за счёт отказа от хранения схожих документов);
- Кластеризация документов по мере их схожести (используется для выдачи более корректных документов при поиске, когда из каждого кластера выбираются наиболее релевантные документы);
- Поиск и фильтрация спама (документов не несущих смысловой нагрузки).

Алгоритм шинглов

Применяется для определения меры схожести документов. В простейшем случае для определения идентичности документов используют сравнение их контрольных сумм. Но главный недостаток такого подхода заключается в большой чувствительности к минимальным изменениям в документе, можно изменить буквально один символ, и контрольная сумма изменится кардинально.
Поэтому был предложен алгоритм шинглов [5,6] (от англ. Shingles – чешуйки), позволяющий определять меру схожести документов в численном виде.

Описание алгоритма

Основная идея заключается в разбиении всего текста на некие равные части, к примеру, по десять слов. Причём перед разбиением происходит очистка (канонизация) текста от предлогов, союзов, дополнительной текстовой разметки (HTML теги), также есть рекомендации удалять прилагательные, так как они не несут смысловой нагрузки и зачастую используются автоматизированными средствами для придания тексту видимости уникальности.
Важной особенностью также является то, что выделенные части (шинглы) идут не встык, а внахлест, для исключения потери информации.
Далее для каждого шингла высчитывается контрольная сумма (CRC32, MD5, и пр.), и происходит сравнение случайных выбранных последовательностей хешей (*сигнатур*) для сравниваемых документов. Даже одно совпадение является указанием на большую вероятность схожести анализируемых документов.
Ещё одним важнейшим преимуществом относительно других алгоритмов является временная сложность. Если обычно нужно сравнить все документы со всеми, что даёт оценку сложности $O(N^2)$ (где N – количество сравниваемых документов), то при сравнивании только сигнатур оценка сложности приближается к $O(N * \log(N))$ [5]; однако при большом N информация о всех анализируемых документах не может быть обработана одним компьютером, возникает задача распределения вычислений, что повышает сложность практически до $O(N^2)$.

В настоящее время алгоритмы на основе шинглов активно используются и развиваются. Перспективными направлениями считаются *алгоритмы супершинглов* [6], когда значения полученных шиглов распределяются по кластерам, и уже на основе значений их хешей производится анализ схожести документов. Этот алгоритм значительно эффективнее по скорости, однако сам автор указал на плохие результаты при наличии небольших документов.

Идея объединения шинглов получила дальнейшее развитие в алгоритме мегашинглов, которые являются попарным сочетанием 6 супершинглов, в этом случае документы считаются совпадающими, если у них совпадает хотя бы один мегашингл.

Алгоритмы, базирующиеся на терминах, сигнатурные методы (Term Based Algorithms)

Впервые предложены в [7]. В отличие от алгоритма шинглов, в качестве основных единиц измерения использует отдельные слова. Данный класс алгоритмов более фокусируется на семантической, нежели синтаксической схожести документов, отказываясь от анализа структуры, оформления документа. Используется построение словаря наиболее часто встречающихся во всех анализируемых документах слов, и в качестве меры близости принимают значение пересечения $\cup$ множества ключевых слов документа и общего словаря.
Числовое значение сходства можно получить с помощью метрики семантического подобия [7]:

$$r(A,B) = \frac{|S(A) \cap S(b)|}{|S(A) \cup S(B)|} \qquad (1)$$

где
$r(A,B)$ - мера близости документов A и B;
$S(A); S(B)$ - множества ключевых слов (k-грамм) документов A и B.

При проведении анализа каждый документ должен быть сравнен со всеми остальными, что даёт сложность алгоритма равную $O(N^2)$. Считается, что данный алгоритм сложнее и более требователен к ресурсам, нежели метод шинглов, и поэтому используется редко.

Как вариант, вместо построения словаря предлагают использовать метод *Tf-IDF* [8] (*TF* — term frequency, *IDF* — inverse document frequency), когда каждый документ представляется в виде числового вектора, отражающего важность использования каждого слова в документе. Размерность вектора зависит от количества слов в общем наборе. Такой подход называется векторной моделью (*VSM*) и позволяет производить сравнение документов с использованием кластерного анализа.
Подробнее о методе *Tf-IDF*.

Метод используется для статистической оценки важности слова в контексте отдельно взятого, но являющегося частью коллекции документа. Вес каждого слова пропорционален количеству употреблений его в рассматриваемом документе и обратно пропорционален частоте его употребления в остальных документах.
Данная мера является произведением двух отношений:
Частота Слова (TF) – являющаяся отношением числа вхождения некоего слова к общему количеству слов в документе

$$TF = \frac{n_i}{\sum_k n_k} \qquad (2)$$

где

n_i есть число вхождений слова в документ;

$\sum_k n_k$ общее число слов в документе.

И *Обратная Частота Документа (IDF)* - инверсия частоты, с которой некоторое слово встречается в документах коллекции (особенность *IDF* в том, что данная мера уменьшает вес часто употребляемых слов)

$$IDF = \log \frac{|D|}{|d_i \supset t_i|} \qquad (3)$$

где

$|D|$ - количество документов в коллекции;

$|d_i \supset t_i|$ - количество документов, в которых встречается t_i.

В результате применения данной меры наибольший вес получают слова с высокой частотой употребления в одном документе и низкой в других.

Алгоритмы, базирующиеся на изображениях (Image Based Algorithms)

В этой группе алгоритмов [9,10] документы представляются и сравниваются в виде изображений.
Одним из простейших алгоритмов этой области является *GQView* [11].
Его идея проста: изображение разбивается попиксельно на квадраты размером 32 на 32 пикселя, и далее берётся среднее значение цвета пикселей в каждом блоке. В таком случае разница представляется в виде суммы значений отличий для одинаковых блоков двух изображений, причём значение отличия для каждого блока нормализовано к диапазону $0 \div 1$.

$$dif(img1, img2) = \sum_{i=1}^{n} norm(img1_i - img2_i) \qquad (4)$$

где

$dif(img1, img2)$ - значение отличия изображения 1 от изображения 2;

n – количество пиксельных блоков в каждом изображении;

$img1_i$ - среднее значение цвета в i-ом блоке;

$norm$ – процедура нормализации.

Системы классификации пользователей на основе поведенческих шаблонов

Одной из трёх основных составляющих Web Mining является Web Usage Mining. Основной целью этого направления является анализ и моделирование шаблонов поведения пользователей на Internet ресурсе в целях адаптивной подстройки выдаваемой информации для получения наиболее релевантных данных каждым пользователем индивидуально, или прогнозирования поведения пользователей в будущем.
Основные направления, в которых используется Web Usage Mining это:

- Internet порталы банков;
- Бизнес исследования (Business Intelligence);
- Прогнозирование посещаемости на высоконагруженных проектах;
- Системы персонификации Internet магазинов.

Web Usage Mining для банковских Internet порталов

Прежде чем непосредственно приступать к описанию проведённого исследования, следует описать использованные в нём методы.

Кластеризация методом нечётких с - средних (Fuzzy c-means/FCM clustering)

При использовании алгоритмов нечёткой кластеризации каждый объект может принадлежать более чем одному кластеру.
В результате отработки метода для каждого рассматриваемого объекта *x* возвращается не номер кластера, к которому он принадлежит, а значения вероятности принадлежности к *к*-ому кластеру

$u_i(k)$. Также обычно сумма этих вероятностей равна 1.

$$\forall x(\sum_{k=1}^{n} u_k(x) = 1) \qquad (5)$$

где
n – количество кластеров вероятность принадлежности к которому больше 0.

В таком случае, при использовании данного алгоритма центр каждого кластера находится как среднее значение всех его точек, с учётом весов имеющих значение вероятности принадлежности точки к анализируемому кластеру:

$$center_k = \frac{\sum_x u_k(x)^m x}{\sum_x u_k(x)^m} \qquad (6)$$

где
m – экспоненциальный вес ($m > 1$).

Обычно m выбирают равным 2, но этот выбор теоретически не обоснован, и некоторые методы предлагают другие значения. Есть исследования [12], рекомендующие выбирать значения m в диапазоне $3.2 \div 3.7$.

Мера принадлежности точки кластеру обратно пропорциональна близости её к центру кластера:

$$u_k(x) = \frac{1}{d(center_k, x)}. \qquad (7)$$

Алгоритм Fuzzy c-means очень похож на классический k-means, за исключением того, что точка может принадлежать нескольким кластерам одновременно.

Возможности пакета DataEngine

Многофункциональный программный пакет инструментов **DataEngine**[13] представляет собой комплексное решение для работы с различными Data Mining алгоритмами.
Пакет использует множество подходов для решения задач Data Mining.

- Клиент – серверная архитектура для обеспечения расширяемости;
- Алгоритмы нечётких вычислений;
- Автоматизированное создание нейронных сетей различных топологий;
- Сети карты Кохонена;
- Алгоритмы нормализации по различным шкалам;
- Различные стратегии обработки недостающих значений и выбросов;
- Богатые инструменты визуализации;
- Стандартные интерфейсы создания дополнений и API для задач автоматизации.

Описание проведённого исследования

В банке «*BCI Bank*» Чили Сантьяго было проведено исследование, наглядно показавшее преимущества, получаемые от внедрения методов Web Mining в банковской сфере.
Так как для этого банка стоимость обработки операций клиентов, производимых посредством Internet, примерно на 90% ниже, чем в обычных представительствах, то есть реальная выгода от привлечения как можно большего количества клиентов к использованию именно этого вида платежей.
Несмотря на более выгодные тарифы, большинство клиентов предпочитало использовать классические методы использования банковских услуг, и возникла задача на основе имеющихся профилей клиентов классифицировать клиентов по степени привлекательности для них Internet обслуживания. И в дальнейшем проводить с ними разъяснительную работу с целью показать все преимущества от использования Internet обслуживания.
В начале была проведена сегментация профилей пользователей по различным атрибутам. Также была проведена процедура анонимизации профилей, в основном в целях приватности, но также и ввиду того, что атрибут «*имя*» не несёт в себе важной информации.
Была получена итоговая таблица:

Table 1
Пример таблицы с выделенными параметрами

Атр. 1	Атр. 2	Атр. 3	Атр. 4	Атр. 5
38	1	1702	62	234
26	0	833	21	34
41	0	500	58	123
40	1	1240	46	314

Далее с использованием алгоритма кластеризации нечётких с - средних (fuzzy c-means) была произведена сегментация клиентов по 5 классам.

- **L1** – «Молодые», редко пользуются Internet услугами банка, оперируют преимущественно небольшими суммами.
- **L2** – «Очень молодые», ещё меньшее использование Internet транзакций.
- **M1** – «Старые», используют Internet сравнительно часто, оперируют бóльшими суммами чем «Молодые».
- **M2** - «Средний возраст», используют Internet сравнительно часто, оперируют бóльшими суммами чем «Молодые».
- **H** - «Современные», активно используют все возможности работы в Internet.

Далее всё множество профилей было разделено на две части, обучающая выборка 20% и основная 80%, соответственно. С использованием пакета *DataEngine* была создана нейронная сеть MLP (многослойный персептрон), которая была обучена распределять профили по классам на 20% выборке.
После обучения на вход сети была подана вся база профилей клиентов банка, и проведена классификация.
В результате были получены данные о том, что 21% профилей являются потенциально активными пользователями Internet портала банка, и им были отправлены буклеты с подробным описанием преимуществ от использования такого вида платежей.
По результатам проведённой компании было выявлено увеличение количества пользователей Internet транзакций на **1,4%**! Что для банка означает неплохое увеличение прибыли.

Прогнозирование посещаемости на высоконагруженных проектах

Другим важным направлением Web Mining является прогнозирование посещаемости Internet ресурсов.
Важнейшим и основным источником данных для этого служат серверные журналы запросов и средства их обработки и анализа. Также их преимущество в распространённости, так как серверы генерируют их автоматически, а специализированные средства сбора статистики хоть и позволяют получать только требуемую информацию, но нуждаются в установке и настройке.
Из журналов серверных запросов можно извлечь следующие данные:

- Пути навигации пользователей;
- Время просмотра страниц;
- Структура гиперссылок и содержания страниц.

Всё это может увеличивать эффективность в таких сферах как e-business, e-services, e-learning, получать новых пользователей, удерживать существующих, увеличивать эффективность и полезность маркетинга, рекламных компаний.

Сначала нужно рассмотреть две важные технологии, применяемые при решении задач такого типа.

Кластеризация методом муравьиной колонии (Standard Ant Clustering Algorithm SACA)

Методы кластеризации с использованием муравьиных алгоритмов, наравне с другими "*биологическими*" методами, начали активно развиваться в начале 21 века. Именно тогда было высказано и подтверждено предположение о том, что синергетический эффект от применения алгоритмов, сходных с природными, совместно с использованием постоянно возрастающих вычислительных мощностей компьютеров может дать превосходные результаты в различных областях науки.

Рассмотрим классическую кластеризацию методом муравьиной колонии.
В основу данного метода положена модель поведения при поиске пищи, когда сначала муравьи движутся в случайных направлениях, но по мере нахождения пищи наиболее выгодные пути следования помечаются всё большим количеством феромона, который, однако, выветривается со временем, если путём перестают пользоваться.
Движение на каждой итерации определяется по следующей формуле:

$$P_i = \frac{\left(l_i^q * f_i^p\right)}{\sum_{n=0}^{N}\left(l_n^q * f_n^p\right)} \qquad (8)$$

где

P_i - вероятность перехода по *i*–му пути;

l_i - длина *i*-ого перехода;

f_i - количество феромона на *i*-ом переходе;

q – коэффициент "жадности" алгоритма;
p – коэффициент "стадности" алгоритма.

Несмотря на то, что алгоритмы данного типа выдают приближённый результат, но ввиду своих статистических свойств увеличение количества итераций увеличивает точность результата.

В продолжение развития данной идеологии было предложено много различных вариантов увеличения классического алгоритма, *ACLUSTER* [14], адаптивный метод муравьиной кластеризации [15], метод *ATTA* [16].

Пакет Discipulus™ и его возможности

Пакет основан на генетических алгоритмах и предназначен для решения разнообразных регрессионных и классификационных задач. Дополнительно позволяет использовать алгоритмы нейронных сетей, классификационных деревьев, метод опорных векторов (*Support Vector Machines*) и некоторые другие.
Утверждается, что основным преимуществом данного программного комплекса является его беспрецедентно высокая скорость работы (благодаря фирменной технологии *AIMLearning™),* которая позволяет добиться огромного выигрыша в скорости по сравнению с другими инструментами моделирования.

Линейное генетическое программирование (ЛГП)

Понятие ЛГП является развитием идеологии генетического программирования (ГП), однако главным отличием является то, что если в простом ГП хромосомами (узлами строящегося дерева программы) могут быть только элементарные операторы или переменные языка программирования, то в более сложном – уже используются целые логические блоки, такие как функции или их последовательности (подпрограммы). Также важнейшей особенностью является то, что используется императивные языки программирования (такие как C), в отличие от функциональных (LISP) в ГП.
Обычно используется некий массив регистров *r*, который может содержать в себе переменные и константы. Также одна из переменных (обычно *r*[0]) используется для вывода значения хромосомы. Ещё такой классический вариант называют Single Solution Genetic Programming (*SS-LGP*).
В процессе работы алгоритма используются 2 типа операций: кроссовер и мутация.
Кроссовер обменивает инструкции у родителей.
Мутация возможна в двух вариантах:

- Макромутация – может удалить или добавить случайную инструкцию;
- Микромутация – удаляет или изменяет оператор или константу инструкции.

Других важных отличий от генетических алгоритмов нет, происходят итерации, на каждой из которых вычисляется значение фитнес функции, на основе результатов которой осуществляется естественный отбор и проверка на останов.

В качестве иллюстрации использования методов Web Mining в этой области рассмотрим исследование, проведённое в университете Монаш [17]

Средняя загруженность серверов университета оценивается примерно в 7 миллионов хитов в сутки, однако в зависимости от сезона, месяца, дня недели и даже часа нагрузки могут меняться очень значительно, и эффективное их прогнозирование позволит эффективнее настроить политики балансировки нагрузок на серверы.
Главной целью данного исследования было сравнение эффективности использования кластеризации методом муравьиной колонии и последующим построением программы классификатора с использованием методов линейного генетического программирования по сравнению с другими возможными методами.
Предметом исследования было выявление шаблонов в использовании Internet портала университета Монаш для прогнозирования распределения нагрузок на серверы.

Вначале было выделено обучающее множество, данные серверного журнала за некий период времени были очищены, и произведена кластеризация методом ACLUSTER по таким параметрам, как количество байт, запрошенных у домена, количество запросов в час и в день.
В итоге была произведена кластеризация по количеству запросов ко всем доменам относительно времени их совершения. Конечно, в зависимости от параметров алгоритма результаты несколько различались, но это свойство подобного рода приближённых алгоритмов, к тому же, благодаря его вероятностному характеру, результаты улучшаются с увеличением количества итераций.

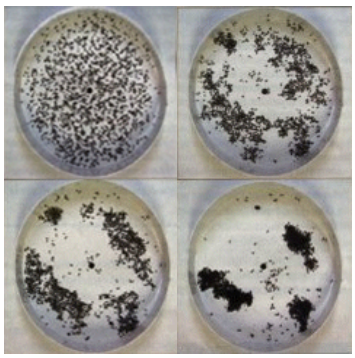

Рисунок 1 Повышение качества кластеризации при использовании алгоритма ACLUSTER при увеличении количества итераций (с сайта авторов алгоритма [18])

Далее, используя методы генетического линейного программирования, нужно было построить с классификатор для прогнозирования нагрузок.
Для создания программы-классификатора был использован пакет Discipulus™, который использует идеологию генетического линейного программирования.
В результате применения различных настроек пакета было создано несколько программ - классификаторов, из которых была выбрана имеющая лучшие результаты.
Коэффициент корреляции между реальными результатами и предсказанными программой составил 0,9921 для почасового предсказания и 0,9963 для предсказания по дням.

Основной целью исследования было сравнить эффективность использования данной связки алгоритмов по сравнению другими, что и было произведено в заключении.
Использовались следующие сокращения:
ANT-LGP рассмотренный алгоритм, муравьиная колония + линейное генетическое программирование;
i-Miner – гибридная система нечёткой кластеризации + нечёткий вывод (тех же авторов);
SOM-LGP – самоорганизующиеся карты + линейное генетическое программирование;
SOM-ANN – самоорганизующиеся карты + нейронная сеть.

Таблица 2

Сравнение результатов прогнозирования различными методами

Метод	Значения		Коэффициент корреляции
	Реальное	Тестовое	
ANT – LGP	0,2561	0,035	0,9921
i-Miner	0,0012	0,0051	0,9981
SOM-LGP	0,0546	0,0639	0,9493
SOM-ANN	0,0654	0,0516	0,9446

Из таблицы 2 видно, что данная связка алгоритмов при решении задачи показала результат, сравнимый с лучшим.

Заключение

Использование современных методов Data Mining позволяет получить реальный экономический эффект. Также, на основе анализа реального опыта внедрения, можно утверждать, что совмещение нескольких методов для решения подзадач является обоснованным и даёт ожидаемый эффект.

References

1. Araya S., Silva M., Weber R. Identifying web usage behavior of bank customers // Berlin: Springer, October 2003. – P 951-958.
2. Ajith A., Vitorino R. Web Usage Mining Artificial Ant Colony Clustering and Linear Genetic Programming // CEC´03 - Congress on Evolutionary Computation / IEEE Press, ISBN 078-0378-04-0, 8-12 December 2003. - Canberra, Australia, P. 1384-1391.
3. Software Discipulus™ Web Site URL: http://www.rmltech.com/ - Visit date September 2009.
4. Ye S., Wen J., Ma W. A systematic study on parameter correlations in large scale duplicate document detection // Knowledge and Information Systems. 14(2), (2008), - University of California Postprints P. 217-232.
5. Manber U. Finding similar files in a large file system // Usenix Winter 1994 Technical Conference, January 1994.
6. Broder A., Glassman S., Manasse M. Syntactic Clustering of the Web // Sixth World Wide Web Conference, September 1997.
7. Chowdhury A., Frieder O., Grossman D., McCabe M.C. Collection statistics for fast duplicate document detection // ACM Transactions on Information System 20(2), (2002) P. 171-191.
8. Salton, G. and McGill, M. J. Introduction to modern information retrieval // New York McGraw-Hill, ISBN 0-07-054484-0 Chapter 3, (1983) P. 52-117.

9. Daniel P. L. Models and Algorithms for Duplicate Document Detection // Information Retrieval, Kluwer Academic Publishers Hingham, MA, USA Volume 4 , Issue 2, (2001) P. 153-173.
10. Bharat K., Broder A. A systematic study on parameter correlations in large-scale duplicate document detection // London: Springer ISSN 0219-1377 , March 2007. - P. 217-232.
11. Description of GQView algorithm URL: http://www.elliotglaysher.org/2006/03/19/duplicate-image-algorithms/ - Visit date September 2009.
12. Киселева Е.М., Блюсс О.Б. особенности некоторых алгоритмов многокритериальной нечеткой кластеризации // Questions of Applied Mathematics and Mathematical modeling, Dnepropetrovsk National University of Oles Gonchar КВ № 5713 (2008).
13. Software DataEngine Web Site URL: http://www.dataengine.de/ - Visit date September 2009.
14. Ramos V., Muge F., Pina P. Self-Organized Data and Image Retrieval as a Consequence of Inter- Dynamic Synergistic Relationships in Artificial Ant Colonies // Soft Computing Systems Design, Management and Applications, 2nd Int. Conf. on Hybrid Intelligent Systems, IOS Press, , 2002. - P. 500-509.
15. Andre L. Vizine, Leandro N. de Castro [etc] Towards Improving Clustering Ants: An Adaptive Ant Clustering Algorithm // Informatica 29 (2005) ISSN 0350-5596, P. 143-154.
16. Handl J., Knowles J., Dorigo M. Ant-based clustering and topographic mapping // Artificial Life Volume 12 , Issue 1, Cambridge, MA, USA: MIT Press ISSN:1064-5462, January 2006. – P. 35-61.
17. Monash university Web site URL: http://www.monash.edu.au/ - Visit date September 2009.
18. Artificial Ant Colonies alghoritm description URL: http://www.chemoton.org/ref39.html - Visit date September 2009.

Об авторах

Pavel Osipov Mg.Sc.Eng. Ph.d. student, Institute of Information Technology, Riga Technical University. He received his masters diploma in Transport and Telecommunications Institute, Riga. His research interests include web data mining, machine learning and knowledge extraction.

Arkady Borisov, Dr.habil.sc.comp.,Professor, Institute of Information Technology, Riga Technical University, 1 Kalku Street, Riga LV-1658 Latvija, e-mail: arkadijs.borisovs@cs.rtu.lv.

Pāvels Osipovs, Arkadijs Borisovs. Practice of web data mining methods application

Recent growth of information on the Internet have high demands for efficiency of processing algorithms.
In this paper some algorithms from the field of Web Data Mining, have proved effective in many existing applications.
Paper divided into two logical parts, the first is considered a theoretical description of algorithms, and second examples of their successful use to solve real problems.
Search algorithms of fuzzy duplicates of documents actively used by all the leading search engines in the world, are descriptions of the following algorithms: shingles, signature methods, algorithms based on the images.
Such methods of classification as a method of fuzzy clustering to-medium (Fuzzy c-means/FCM clustering) and clustering by ant colony (Standard Ant Clustering Algorithm SACA).
In conclusion, described the experience of the successful application of fuzzy clustering in conjunction with the software toolkit DataEngine to improve the efficiency of the bank «BCI Bank» as well as the sharing of the ant colony clustering method in conjunction with linear genetic programming to meet the increasing efficiency of predicting the load on the servers of highly Internet portal Monash Institut .

Pāvels Osipovs, Arkadijs Borisovs. Web Data Mining metožu lietošanas praksē

Nesenās izaugsmes informāciju par interneta ir augstas prasības pēc efektivitātes apstrādes algoritmiem.
Šajā rakstā daži no lauka Web Data Mining algoritmiem, ir izrādījušies efektīvi, daudzās esošās programmas.
Pants ir sadalīts divās loģiskās daļās, pirmā ir uzskatāma teorētisks apraksts par algoritmiem, un otrā to veiksmīgi izmantot piemērus, lai risinātu reālas problēmas.
Meklēt algoritmi izplūdušo dublikātu dokumentu aktīvi izmanto visas vadošo meklēšanas dzinēju pasaulē, ir apraksti par šādu algoritmi: jostas roze, parakstu metodes, algoritmi balstās uz attēliem.
Šādu metožu klasifikāciju kā izplūdušās klasterizācijas metodes vidējā (Fuzzy c-means/FCM apvienību veidošana) un apvienības ar skudru kolonijas (Standard Ant Clustering algoritms SACA).
Nobeigumā aprakstītās pieredzi veiksmīga pielietojuma izplūdušās klasterizācijas saistībā ar programmatūras rīku komplekts DataEngine, lai uzlabotu efektivitāti, bankas «BCI Bank», kā arī daloties skudru kolonija klasterizācijas metodes saistībā ar lineāro ģenētisko programmu, lai apmierinātu pieaugošo efektivitāti prognozēšanā slodzes uz ļoti interneta portāla Monash institūts serveri.