Обзор алгоритмов кластеризации данных

Data Mining*

Приветствую!

В своей дипломной работе я проводил обзор и сравнительный анализ алгоритмов кластеризации данных. Подумал, что уже собранный и проработанный материал может оказаться кому-то интересен и полезен. О том, что такое кластеризация, рассказал sashaeve в статье «Кластеризация: алгоритмы k-means и c-means». Я частично повторю слова Александра, частично дополню. Также в конце этой статьи интересующиеся могут почитать материалы по ссылкам в списке литературы.

Так же я постарался привести сухой «дипломный» стиль изложения к более публицистическому.

Понятие кластеризации

Кластеризация (или кластерный анализ) — это задача разбиения множества объектов на группы, называемые кластерами. Внутри каждой группы должны оказаться «похожие» объекты, а объекты разных группы должны быть как можно более отличны. Главное отличие кластеризации от классификации состоит в том, что перечень групп четко не задан и определяется в процессе работы алгоритма.

Применение кластерного анализа в общем виде сводится к следующим этапам:

1. Отбор выборки объектов для кластеризации.
2. Определение множества переменных, по которым будут оцениваться объекты в выборке. При необходимости – нормализация значений переменных.
3. Вычисление значений меры сходства между объектами.
4. Применение метода кластерного анализа для создания групп сходных объектов (кластеров).
5. Представление результатов анализа.

После получения и анализа результатов возможна корректировка выбранной метрики и метода кластеризации до получения оптимального результата.

Лучшее за 24 часа ↓

Ломаем банк в стиле smash the stack!

Искусство переговоров — это просто бизнес, ничего личного

Перевод официальной документации по Backbone.JS

Facebook объявила о достижении соглашения о поглощении Instagram. Цена вопроса — $1 млрд

Как раскрутить «Социальную сеть Ковчег», если это фантастическая трилогия

Квадрокоптер за 1 день и $120

Шеллкоды, эксплойты… Тулзы под Win

Кластерные и «обычные» индексы MySQL (InnoDB)

Go не рекомендуется использовать для разработки на Windows 32bit (UPD: и на Linux тоже)

Парсим Python код с помощью Flex и Bison

« все лучшие

Похожие посты ↓

02.04.2012 → Нечеткий кластерный анализ на примере социально-экономических показателей крупных городов России

11.03.2012 → Data Mining в футболе: давайте оцифруем матч и всех посчитаем!

07.12.2011 → Data Mining в онлайн играх

Меры расстояний

Итак, как же определять «похожесть» объектов? Для начала нужно составить вектор характеристик для каждого объекта — как правило, это набор числовых значений, например, рост-вес человека. Однако существуют также алгоритмы, работающие с качественными (т.н. категорийными) характеристиками.

После того, как мы определили вектор характеристик, можно провести нормализацию, чтобы все компоненты давали одинаковый вклад при расчете «расстояния». В процессе нормализации все значения приводятся к некоторому диапазону, например, [-1, -1] или [0, 1].

Наконец, для каждой пары объектов измеряется «расстояние» между ними — степень похожести. Существует множество метрик, вот лишь основные из них:

1. Евклидово расстояние
   Наиболее распространенная функция расстояния. Представляет собой геометрическим расстоянием в многомерном пространстве:
   $$\rho(x, x') = \sqrt{\sum_i^n (x_i - x_i')^2}$$
2. Квадрат евклидова расстояния
   Применяется для придания большего веса более отдаленным друг от друга объектам. Это расстояние вычисляется следующим образом:
   $$\rho(x, x') = \sum_i^n (x_i - x_i')^2$$
3. Расстояние городских кварталов (манхэттенское расстояние)
   Это расстояние является средним разностей по координатам. В большинстве случаев эта мера расстояния приводит к таким же результатам, как и для обычного расстояния Евклида. Однако для этой меры влияние отдельных больших разностей (выбросов) уменьшается (т.к. они не возводятся в квадрат). Формула для расчета манхэттенского расстояния:
   $$\rho(x, x') = \sum_i^n |x_i - x_i'|$$
4. Расстояние Чебышева
   Это расстояние может оказаться полезным, когда нужно определить два объекта как «различные», если они различаются по какой-либо одной координате. Расстояние Чебышева вычисляется по формуле:
   $$\rho(x, x') = \max(|x_i - x_i'|)$$
5. Степенное расстояние
   Применяется в случае, когда необходимо увеличить или уменьшить вес, относящийся к размерности, для которой соответствующие объекты сильно отличаются. Степенное расстояние вычисляется по следующей формуле:
   $$\rho(x, x') = \sqrt[r]{\sum_i^n (x_i - x_i')^p},$$
   где r и p – параметры, определяемые пользователем. Параметр p ответственен за постепенное взвешивание разностей по отдельным координатам, параметр r ответственен за прогрессивное взвешивание больших расстояний между объектами. Если оба параметра – r и p — равны двум, то это расстояние совпадает с

29.11.2011 → Какой инструмент вы используете для решения задач data mining?

09.08.2011 → Кластеризация. Алгоритм a-квазиэквивалентности

06.04.2011 → Data Mining Cup 2011

25.06.2010 → Технологии data-mining в расследовании террористических актов

02.06.2010 → Data Mining: что внутри

08.08.2009 → Обзор литературы по Data Mining

29.07.2009 → Бизнес кейсы использования Data Mining. Часть 1

расстоянием Евклида.

Выбор метрики полностью лежит на исследователе, поскольку результаты кластеризации могут существенно отличаться при использовании разных мер.

Классификация алгоритмов

Для себя я выделил две основные классификации алгоритмов кластеризации.

1. Иерархические и плоские.
   Иерархические алгоритмы (также называемые алгоритмами таксономии) строят не одно разбиение выборки на непересекающиеся кластеры, а систему вложенных разбиений. Т.о. на выходе мы получаем дерево кластеров, корнем которого является вся выборка, а листьями — наиболее мелкие кластера.
   Плоские алгоритмы строят одно разбиение объектов на кластеры.
2. Четкие и нечеткие.
   Четкие (или непересекающиеся) алгоритмы каждому объекту выборки ставят в соответствие номер кластера, т.е. каждый объект принадлежит только одному кластеру. Нечеткие (или пересекающиеся) алгоритмы каждому объекту ставят в соответствие набор вещественных значений, показывающих степень отношения объекта к кластерам. Т.е. каждый объект относится к каждому кластеру с некоторой вероятностью.

Объединение кластеров

В случае использования иерархических алгоритмов встает вопрос, как объединять между собой кластера, как вычислять «расстояния» между ними. Существует несколько метрик:

1. Одиночная связь (расстояния ближайшего соседа)
   В этом методе расстояние между двумя кластерами определяется расстоянием между двумя наиболее близкими объектами (ближайшими соседями) в различных кластерах. Результирующие кластеры имеют тенденцию объединяться в цепочки.
2. Полная связь (расстояние наиболее удаленных соседей)
   В этом методе расстояния между кластерами определяются наибольшим расстоянием между любыми двумя объектами в различных кластерах (т.е. наиболее удаленными соседями). Этот метод обычно работает очень хорошо, когда объекты происходят из отдельных групп. Если же кластеры имеют удлиненную форму или их естественный тип является «цепочечным», то этот метод непригоден.
3. Невзвешенное попарное среднее
   В этом методе расстояние между двумя различными кластерами вычисляется как среднее расстояние между всеми парами объектов в них. Метод эффективен, когда объекты формируют различные группы, однако он работает одинаково хорошо и в случаях протяженных («цепочечного» типа) кластеров.
4. Взвешенное попарное среднее
   Метод идентичен методу невзвешенного попарного среднего, за исключением того, что при вычислениях размер соответствующих кластеров (т.е. число объектов, содержащихся в них) используется в качестве весового коэффициента. Поэтому данный метод должен быть использован, когда предполагаются неравные размеры кластеров.
5. Невзвешенный центроидный метод
   В этом методе расстояние между двумя кластерами определяется как

Q&A ↓

Finom → Альтернативный формат многослойных векторно-растровых изображений (аналог PSD) 2

Riateche → Выбор монитора 6

Stdit → Нужно ли оптимизировать скрипт? 3

Akson87 → Имеет ли смысл писать статью о доходах от моего первого приложения в App Store? 1

Akson87 → Надо ли студентов учить делать доклады и искать хорошую работу? 1

super → Чем заинтересовать программиста? 10

Norraxx → Конфигурация ПК 5

taliban → Проблемы с Google Reader 5

CaptainFlint → Как заставить Оперу обновить страницу после нажатия кнопки Back 5

SamDark → Медленные модули в Yii 2

« все вопросы

О, работа! ↓

iOS разработчик

Сооснователь-разработчик в технический веб-стартап

Системный администратор

Менеджер по интернет-рекламе

менеджер интернет-проекта

менеджер проекта (консультант 1С)

веб-разработчик 1С Битрикс

расстояние между их центрами тяжести.

6. Взвешенный центроидный метод (медиана)
   Этот метод идентичен предыдущему, за исключением того, что при вычислениях используются веса для учета разницы между размерами кластеров. Поэтому, если имеются или подозреваются значительные отличия в размерах кластеров, этот метод оказывается предпочтительнее предыдущего.

Обзор алгоритмов

Алгоритмы иерархической кластеризации

Среди алгоритмов иерархической кластеризации выделяются два основных типа: восходящие и нисходящие алгоритмы. Нисходящие алгоритмы работают по принципу «сверху-вниз»: в начале все объекты помещаются в один кластер, который затем разбивается на все более мелкие кластеры. Более распространены восходящие алгоритмы, которые в начале работы помещают каждый объект в отдельный кластер, а затем объединяют кластеры во все более крупные, пока все объекты выборки не будут содержаться в одном кластере. Таким образом строится система вложенных разбиений. Результаты таких алгоритмов обычно представляют в виде дерева – дендрограммы. Классический пример такого дерева – классификация животных и растений.

Для вычисления расстояний между кластерами чаще все пользуются двумя расстояниями: одиночной связью или полной связью (см. обзор мер расстояний между кластерами).

К недостатку иерархических алгоритмов можно отнести систему полных разбиений, которая может являться излишней в контексте решаемой задачи.

Алгоритмы квадратичной ошибки

Задачу кластеризации можно рассматривать как построение оптимального разбиения объектов на группы. При этом оптимальность может быть определена как требование минимизации среднеквадратической ошибки разбиения:

$$e^2(X,L) = \sum_{j=1}^{K} \sum_{i=1}^{n_j} \| x_i^{(j)} - c_j \|^2$$

где c_j — «центр масс» кластера j (точка со средними значениями характеристик для данного кластера).

Алгоритмы квадратичной ошибки относятся к типу плоских алгоритмов. Самым распространенным алгоритмом этой категории является метод k-средних. Этот алгоритм строит заданное число кластеров, расположенных как можно дальше друг от друга. Работа алгоритма делится на несколько этапов:

1. Случайно выбрать *k* точек, являющихся начальными «центрами масс» кластеров.
2. Отнести каждый объект к кластеру с ближайшим «центром масс».
3. Пересчитать «центры масс» кластеров согласно их текущему составу.
4. Если критерий остановки алгоритма не удовлетворен, вернуться к п. 2.

В качестве критерия остановки работы алгоритма обычно выбирают минимальное изменение среднеквадратической ошибки. Так же возможно останавливать работу алгоритма, если на шаге 2 не было объектов, переместившихся из кластера в кластер.

К недостаткам данного алгоритма можно отнести необходимость задавать количество кластеров для разбиения.

Нечеткие алгоритмы

Наиболее популярным алгоритмом нечеткой кластеризации является алгоритм c-средних (c-means). Он представляет собой модификацию метода k-средних. Шаги работы алгоритма:

1. Выбрать начальное нечеткое разбиение *n* объектов на *k* кластеров путем выбора матрицы принадлежности *U* размера *n x k*.
2. Используя матрицу U, найти значение критерия нечеткой ошибки:
   $$E^2(X,U) = \sum_{i=1}^{N} \sum_{k=1}^{K} U_{ik} \| x_i^{(k)} - c_k \|^2$$,
   где c_k — «центр масс» нечеткого кластера *k*:
   $$c_k = \sum_{i=1}^{N} U_k x_i$$.
3. Перегруппировать объекты с целью уменьшения этого значения критерия нечеткой ошибки.
4. Возвращаться в п. 2 до тех пор, пока изменения матрицы *U* не станут незначительными.

Этот алгоритм может не подойти, если заранее неизвестно число кластеров, либо необходимо однозначно отнести каждый объект к одному кластеру.

Алгоритмы, основанные на теории графов

Суть таких алгоритмов заключается в том, что выборка объектов представляется в виде графа *G=(V, E)*, вершинам которого соответствуют объекты, а ребра имеют вес, равный «расстоянию» между объектами. Достоинством графовых алгоритмов кластеризации являются наглядность, относительная простота реализации и возможность вносения различных усовершенствований, основанные на геометрических соображениях. Основными алгоритмам являются алгоритм выделения связных компонент, алгоритм построения минимального покрывающего (остовного) дерева и алгоритм послойной кластеризации.

Алгоритм выделения связных компонент

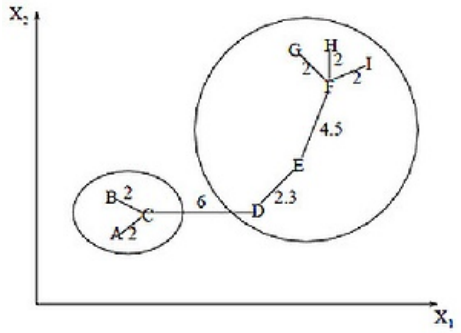
В алгоритме выделения связных компонент задается входной параметр *R* и в графе удаляются все ребра, для которых «расстояния» меньше *R*. Соединенными остаются только наиболее близкие пары объектов. Смысл алгоритма заключается в том, чтобы подобрать такое значение *R*, лежащее в диапазон всех «расстояний», при котором граф «развалится» на несколько связных компонент. Полученные компоненты и есть кластеры.

Для подбора параметра *R* обычно строится гистограмма распределений попарных расстояний. В задачах с хорошо выраженной кластерной структурой данных на гистограмме будет два пика – один соответствует внутрикластерным расстояниям, второй – межкластерным расстояния. Параметр *R* подбирается из зоны минимума между этими пиками. При этом

управлять количеством кластеров при помощи порога расстояния довольно затруднительно.

Алгоритм минимального покрывающего дерева

Алгоритм минимального покрывающего дерева сначала строит на графе минимальное покрывающее дерево, а затем последовательно удаляет ребра с наибольшим весом. На рисунке изображено минимальное покрывающее дерево, полученное для девяти объектов.



Путём удаления связи, помеченной CD, с длиной равной 6 единицам (ребро с максимальным расстоянием), получаем два кластера: {A, B, C} и {D, E, F, G, H, I}. Второй кластер в дальнейшем может быть разделён ещё на два кластера путём удаления ребра EF, которое имеет длину, равную 4,5 единицам.

Послойная кластеризация

Алгоритм послойной кластеризации основан на выделении связных компонент графа на некотором уровне расстояний между объектами (вершинами). Уровень расстояния задается порогом расстояния *c*. Например, если расстояние между объектами $0 \le \rho(x, x') \le 1$, то $0 \le c \le 1$.

Алгоритм послойной кластеризации формирует последовательность подграфов графа *G*, которые отражают иерархические связи между кластерами:

$$G^0 \subseteq G^1 \subseteq \ldots \subseteq G^m,$$

где $G^t = (V, E^t)$ — граф на уровне c^t,
$E^t = \{e_{ij} \in E : \rho_{ij} \le c_t\}$,
c^t – t-ый порог расстояния,
m – количество уровней иерархии,
$G^0 = (V, o)$, o – пустое множество ребер графа, получаемое при $t^0 = 1$,
$G^m = G$, то есть граф объектов без ограничений на расстояние (длину ребер графа), поскольку $t^m = 1$.

Посредством изменения порогов расстояния $\{c^0, \ldots, c^m\}$, где $0 = c^0 < c^1 < \ldots < c^m = 1$, возможно контролировать глубину иерархии получаемых кластеров. Таким образом, алгоритм послойной кластеризации способен создавать как плоское разбиение данных, так и иерархическое.

Сравнение алгоритмов

Вычислительная сложность алгоритмов

Алгоритм кластеризации	**Вычислительная сложность**
Иерархический	$O(n^2)$
k-средних	O(nkl), где k – число кластеров, l – число итераций
c-средних	
Выделение связных компонент	*зависит от алгоритма*
Минимальное покрывающее дерево	$O(n^2 \log n)$
Послойная кластеризация	O(max(n, m)), где m < n(n-1)/2

Сравнительная таблица алгоритмов

Алгоритм кластеризации	**Форма кластеров**	**Входные данные**	**Результаты**
Иерархический	Произвольная	Число кластеров или порог расстояния для усечения иерархии	Бинарное дерево кластеров
k-средних	Гиперсфера	Число кластеров	Центры кластеров
c-средних	Гиперсфера	Число кластеров, степень нечеткости	Центры кластеров, матрица принадлежности
Выделение связных компонент	Произвольная	Порог расстояния R	Древовидная структура кластеров
Минимальное покрывающее дерево	Произвольная	Число кластеров или порог расстояния для удаления ребер	Древовидная структура кластеров
Послойная кластеризация	Произвольная	Последовательность порогов расстояния	Древовидная структура кластеров с разными уровнями иерархии

Немного о применении

В своей работе мне нужно было из иерархических структур (деревьев) выделять отдельные области. Т.е. по сути необходимо было разрезать исходное дерево на несколько более мелких деревьев. Поскольку ориентированное дерево – это частный случай графа, то естественным образом подходят алгоритмы, основанными на теории графов.

В отличие от полносвязного графа, в ориентированном дереве не все вершины соединены ребрами, при этом общее количество ребер равно n–1, где n – число вершин. Т.е. применительно к узлам дерева, работа алгоритма выделения связных компонент упростится, поскольку удаление любого количества ребер «развалит» дерево на связные компоненты (отдельные деревья). Алгоритм минимального покрывающего дерева в

данном случае будет совпадать с алгоритмом выделения связных компонент – путем удаления самых длинных ребер исходное дерево разбивается на несколько деревьев. При этом очевидно, что фаза построения самого минимального покрывающего дерева пропускается.

В случае использования других алгоритмов в них пришлось бы отдельно учитывать наличие связей между объектами, что усложняет алгоритм.

Отдельно хочу сказать, что для достижения наилучшего результата необходимо экспериментировать с выбором мер расстояний, а иногда даже менять алгоритм. Никакого единого решения не существует.

Список литературы

1. Воронцов К.В. Алгоритмы кластеризации и многомерного шкалирования. Курс лекций. МГУ, 2007.
2. Jain A., Murty M., Flynn P. Data Clustering: A Review. // ACM Computing Surveys. 1999. Vol. 31, no. 3.
3. Котов А., Красильников Н. Кластеризация данных. 2006.
3. Мандель И. Д. Кластерный анализ. — М.: Финансы и Статистика, 1988.
4. Прикладная статистика: классификация и снижение размерности. / С.А. Айвазян, В.М. Бухштабер, И.С. Енюков, Л.Д. Мешалкин — М.: Финансы и статистика, 1989.
5. Информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных — www.machinelearning.ru/
6. Чубукова И.А. Курс лекций «Data Mining», Интернет-университет информационных технологий — www.intuit.ru/department/database/datamining/

кластеризация, алгоритмы, data mining, кластерный анализ

комментарии (38)

digreen 11 августа 2010, 10:58 # +1

Молодец, и тут тебе тоже «отл.» в зачетку

hellbee 11 августа 2010, 11:12 # +4

Хоть кто-то в наше время пишет дипломы сам

great_boba 11 августа 2010, 11:14 # 0

Вы еще пропустили популярную меру оценки расстояния как корреляцию между координатами (векторами значений)

andreycha 11 августа 2010, 20:25 # ↑ –1

Уверен, я много что еще пропустил :). Можете рассказать поподробнее про эту меру?

eox425 11 августа 2010, 11:32 # 0

Спасибо большое, отличный обзор для «непосвящённого»

KiriKiri 11 августа 2010, 11:44 # **–2**

Спасибо, очень интересно, а главное я наконец смог понять, для чего используется функциональный анализ.

XenJ 11 августа 2010, 12:37 # ↑ **0**

Все-таки функциональный анализ это немного другая тема.

KiriKiri 11 августа 2010, 13:54 # ↑ **0**

Теорию меры и метрические пространства мы изучали пока что только в курсе функционального анализа.

jerrydevice 11 августа 2010, 11:46 # **0**

Вы большая умничка! У меня как раз был диплом связанный с кластеризацией. Ностальгия охватила.
Вам не приходилось сталкиваться с задачей автоматического определения числа кластеров?

sashaeve 11 августа 2010, 14:33 # ↑ **0**

Мне приходилось. Что конкретно интересует?

jerrydevice 11 августа 2010, 15:03 # ↑ **0**

Интересует, какие подходы и алгоритмы применяются для решения этой задачи на ОЧЕНЬ больших объемах данных?

sashaeve 11 августа 2010, 15:37 # ↑ **0**

Алгоритмы и подходы те же, что и на небольших объемах данных. Разница заключается в том, что а) данные могут обрабатываться параллельно (или могут использоваться кластеры) б) данные могут анализироваться локально (т.е. данные разбиваются на меньшие группы), а потом сравниваться между собой (пост-процессинг) в) если есть какие-то экспертные данные, то делаются допущения на ранних этапах (пре-процессинг), но это может влиять на корректность результата. Где-то так.

andreycha 11 августа 2010, 21:17 # ↑ **0**

А мне кажется, что любой алгоритм, явно не задающий количество кластеров, так или иначе требует задание каких-то косвенных параметров, которые влияют на итоговое количество кластеров? Или я не так понимаю смысл задачи автоматическое определение числа кластеров?

Antelle 11 августа 2010, 19:16 # ↑ **0**

Для этого есть алгоритм X-Means, реализацию его на java можете посмотреть в системе довольно известной datamining-системе weka:
www.java2s.com/Open-Source/Java-Document/Science/weka/weka.clusterers.htm

Основываются они примерно на том, что минимизируется, так сказать, «количество дырок» внутри кластера.

jerrydevice 11 августа 2010, 19:22 # ↑ **0**

О, это интересно. Спасибо.

 **zaartix** 11 августа 2010, 11:47 # **0**

давайте вашу зачетку…

 **mjutu** 11 августа 2010, 12:05 # **0**

у меня тоже похожий диплом… эххх

 **Shens** 11 августа 2010, 12:21 # **+1**

Не встретил в тексте ни слова о самоорганизующейся карте Кохонена. Ну или просто о нейросетевой кластеризации(в том числе и нейронный газ). Вы можете пояснить почему? Просто сам в своих задачах отдавал предпочтение этим алгоритмам.

 **Pilot34** 11 августа 2010, 12:30 # **0**

А как тестировали и результаты напишете? У меня курсач был на эту тему. На новостях самые простые алгоритмы дали лучшие результаты.

 **andreycha** 11 августа 2010, 20:12 # ↑ **0**

С результатами, если честно, туго :). Дальше экспериментов дело не пошло, настало время защиты, и нормальных результатов нет.

Использовал квадрат эвклидова расстояния (чтоб «увеличить» расстояния), а также параметр для «разваливания» дерева (все величины предварительно нормализовывал), соответственно результаты на глаз оценивал.

 **XenJ** 11 августа 2010, 12:49 # **+1**

Советую обратить внимание на такие алгоритмы как LSH и RBVs которые позволяют быстро определять принадлежность произвольного вектора кластеру.

 **shogunkub** 11 августа 2010, 13:28 # **0**

>а объекты разных группы должны быть как можно отличны.
как можно *более* отличны, вероятно.
Из текста непонятно, а темой я, как самоучка, не владею — что понимается под размером кластера? Число элементов в нём, или «протяженность»(расстояние между самыми удаленными элементами). Интуитивно вроде первое, но вдруг…

 **andreycha** 11 августа 2010, 13:51 # ↑ **0**

Да, под размером кластера понимается число объектов в нем.

 **dborovikov** 11 августа 2010, 14:41 # **0**

А не расскажите, какие меры расстояний используют для текстовой информации? Я думаю для DataMining это более актуальная задача.

 **sashaeve** 11 августа 2010, 15:33 # ↑ **0**

Для текстовой информации используются другие методы и подходы. Хотя, в самом простом случае, тексты разбиваются на векторы объектов (слова, фразы и т.д.) и считается расстояние между двумя векторами (меры расстояния специфичны для конкретной задачи).

 **andreycha** 11 августа 2010, 21:30 # ↑ **0**

С Text Mining/IR я не работал, к сожалению. Знаю только, что для качественных

характеристик существуют меры Чекановского-Соренсена и Жаккара.

erley 11 августа 2010, 16:14 # +1

Нужно ещё упомянуть про SVM — Support Vector Machine от профессора Вапника. Те кто плотно работают с задачами кластеризации на практике пользуются этим подходом довольно часто. Есть и коммерческие пакеты, такие как например KXEN.

Melkor 11 августа 2010, 18:03 # +2

Здесь есть слайды к лекции по кластерному анализу. Общий обзор для непрофильной специальности. И пошаговый пример вычисления для иерархического алгоритма.

Melkor 11 августа 2010, 18:07 # ↑ +2

На Slideboom получше выглядит

andreycha 11 августа 2010, 21:19 # ↑ 0

Спасибо. Звук на слайдбуме выносит мозг :)).

Melkor 12 августа 2010, 11:20 # ↑ 0

:) по-моему это стандартные звуки для PP 2010.

Ambrose 11 августа 2010, 18:26 # 0

А можно чуть подробнее про сам диплом? В какой области поставленную задачу решали и какими средствами? Как анализируемые данные были представлены?

Сам в этом году по Data Mining дипломировался, только задачи другие были, ассоциации искал :)

andreycha 11 августа 2010, 20:09 # ↑ 0

В моем случае кластеризация была не в контексте data mining'a. В рамках разработки системы хранилища данных нужно было большие файлы иерархических структур (XML, JSON) разделять на более мелкие, основываясь на статистике обращения к элементам.

webrover 11 августа 2010, 18:45 # 0

писал для диплома алгоритм кластеризации через генетические алгоритмы — модная кстати тема. у меня в городе один профессор этим занимается

serf 11 августа 2010, 22:26 # 0

В свое время использовал (только по учебе) простой алгоритм k-средних (евклидово расстояние) для кластеризации цветовых пятен по сходству.
Запомнился вот этот cgm.computergraphics.ru/ неплохой ресурс, только по графике правда
Вот еще вспомни общее logic.pdmi.ras.ru/~yura/internet/02ia-seminar-note.pdf «Кластеризация данных» (Александр Котов)

hohlandrik 12 августа 2010, 01:31 # 0

Высшая Школа Экономики, факультет Социологии?

andreycha 12 августа 2010, 01:37 # ↑ 0

Политех, факультет технической кибернетики.

gaki 12 августа 2010, 12:09 # **0**

Почему вы примеры не приводите? Очень трудно читать. Про «объяснение нормальным человеческим языком» я, уж ладно, промолчу, ибо понимаю, что в научной среде это нонсенс. Но хоть бы пару примеров типа «вот у нас есть набор из стапицот вислоухих кроликов, и под кластеризацией по критерию вислости ушей мы понимаем то-то, таким-то алгоритмом делаем так-то и получаем бла-бла-бла...»