УДК 004.67

Б.А. НОВИКОВ, И.В. СУДОС

ИНДЕКСИРОВАНИЕ ДАННЫХ В ПРОСТРАНСТВЕ БОЛЬШОЙ РАЗМЕРНОСТИ

Новиков Б.А., Судос И.В. **Индексирование данных в пространстве большой размерности.**

Аннотация. Индексирование данных является неотделимой частью задачи поиска. В то время как для данных в пространствах размерностей не более 5 существует хорошо изученный набор эффективных алгоритмов индексации и поиска, для пространств большой размерностей эти алгоритмы оказываются неэффективны или неприменимы. В этом обзоре мы приводим существующие обоснования проблем связанных с индексированием в пространствах большой размерности для задачи поиска ближайшего соседа. Рассматриваются возможные методы решения обозначенных проблем, применимые в областях анализа данных, таких как кластеризация и извлечение скрытых структур. Ставится вопрос о применимости различных методов размерностной редукции к задачи индексирования и поиска ближайшего соседа.

Ключевые слова: индексирование, поиск ближайшего соседа, данные большой размерности, анализ данных, редукция размерности.

Novikov B.A., Sudos I.V. **Indexing of data in high dimensional spaces.**

Abstract. Indexing of the data is an essential part of the search problem. Whilst a well studied set of efficient algorithms of indexing and search exist for a data in the spaces of the dimensionality less then 5, these algorithms are inefficient or not applicable for high dimensional spaces. We review existing studies of the problems related to indexing in a high dimensional space for a nearest neighbour search problem. Possible methods of the designated problems solution that are applicable in the areas of data analysis such as clustering and hidden structure elicitation are regarded. The question on the applicability of the various dimensional reduction methods for an indexing and nearest neighbour search problem.

Keywords: indexing, nearest neighbour search, high dimensional data, data analysis, dimensionality reduction.

1. Введение. Под данными большой размерности в информатике как правило понимаются наборы объектов, характеризуемые большим (более десяти) количеством атрибутов [1].

Рассмотрим задачу поиска в пространствах большой размерности. Задача опирается на многие другие задачи анализа данных в пространстве больших размерностей, в первую очередь: кластеризацию, классификацию и статистический анализ. Решение задачи индексации данных рассматривается как часть задачи построения алгоритма поиска. Задача информационного поиска и индексации в пространствах большой размерности кардинально различается в сложности и основных алгоритмических принципах от задачи информационного поиска в пространствах малой размерности и текстового поиска, подробно описанных в книге [2].

Важно представлять, какие практические задачи используют поиск в пространствах большой размерности, какие ограничения они ставят и с какими теоретическими проблемами сталкиваются.

Каждый рассматриваемый случай применения поиска в пространствах большой размерности может быть рассмотрен со следующих основных позиций:

- свойства данных - элементов пространства поиска (размерность, интенсивность обновления, распределённость).
- вид поискового запроса и требуемая точность поиска.

Мы будем рассматривать в качестве пространства поиска d-мерное векторное пространство. В качестве данных, среди которых осуществляется поиск - множество векторов V, $|V| = N$ в этом пространстве. Множество V формируется в результате процесса обработки исходных данных, например, изображений, видео, сигналов, текстов. Процесс называется трансформацией признаков или выборкой признаков (feature transformation, feature selection), схематически он изображен на рисунке 1. Алгоритм трансформации признаков зависит от происхождения и представления изначальных данных. Каждый элемент $v \in V$ соответствует одному объекту из набора исходных данных, например, одному изображению, сигналу и т.д. Подробнее этот вопрос изучается в работах: [4–6]. В исследовании [16] приведены следующие применяемые на практике запросы: *Запрос-диапазон (range-query)* Такой запрос представляется в виде прямоугольника в некотором подпространстве: $q = [l_{i_1}, r_{i_1}] \times \ldots \times [l_{i_m}, r_{i_m}]$. Результат такого запроса - все существующие данные, соответствующие координаты которых заключены в q. *Запрос ближайших соседей* Пусть q - вектор. Для данного запроса можно выделить 3 подтипа. Первый - нахождение N ближайших относительно некоторых функций расстояния $dist(q, x)$ векторов к вектору-запросу q. Второй подтип - нахождение всех векторов, удаленных не более чем на R от q. Третий - нахождение всех ближайших к q векторов, причем степень близости определяет сам алгоритм поиска. Перейдем к рассмотрению практических примеров.

Примеры применения поиска в пространствах большой размерности. *Мультимедиа коллекции.* Распространенный случай применимости поиска в пространствах большой размерности - это поиск по мультимедиа коллекциям. Под мультимедиа коллекциями подразумеваются наборы изображений, видео, звуков. В этом случае процесс трансформации признаков выделит в качестве векторов данных функции распределения цвета, информацию о форме, спектральные характеристики и др. Способы индексирования для подобных систем рассмотрены Бе-

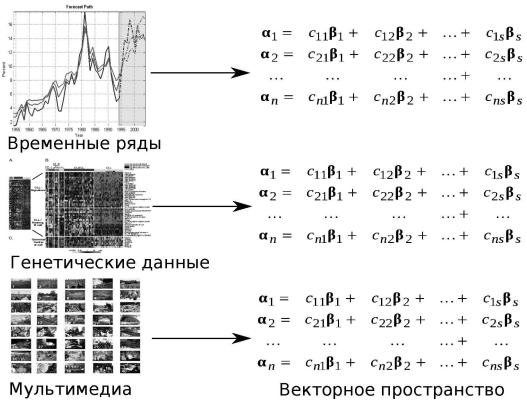


Рис. 1. Преобразование исходных коллекций в вектора

мом и др. в [16]. В этой же работе отмечено одно отличие подобных поисковых систем - разнообразные виды запросов. Наиболее важным считается запрос ближайших соседей, допускающий неточность результата в ряде случаев. Интенсивность обновления может быть весьма высокой, но индексация может осуществляться с определенной задержкой. Релевантность результата запроса поиска может быть оценена с помощью стандартных метрик [2]. Точности:

$$P = \frac{|D_{rel} \cap D_{found}|}{D_{found}},$$

и полноты:

$$R = \frac{|D_{rel} \cap D_{found}|}{D_{rel}},$$

где D_{rel} - множество релевантных данному запросу векторов в базе, а D_{found} - множество найденных векторов. Для подобных систем важна скорость поиска и высокая доступность данных. Система хранения часто содержит дисковую память и может быть распределенной на нескольких узлах, соединенных сетью.

Экспрессия генов. Множество данных, используемых в биоинформатике часто являются высокоразмерными, достигая тысяч координат. Так называемые микромассивы (microarrays) [3] представляют матрицы данных, каждая строка которой соответствует определенному гену, каждый столбец - условию, при котором замерялась экспрессия гена. В качестве элемента такой матрицы выступает численно выраженный “уровень экспрессии” гена, характеризующий его влияние на

конечный белковый продукт при заданных условиях. Особенность этой задачи в том, что пользователь изначально не может утверждать, какие гены являются “ближайшими соседями”. Задача индексирования здесь идет совместно с задачей выявления шаблонов экспрессии. В [7] рассматривается важность восстановления функциональных или эволюционных связей по одному экземпляру генома. В данном случае требуется поиск ближайших соседей. Как правило, подобные задачи подразумевают статичность данных, допустимы задержки как и при индексации, так и при поиске.

Сенсорные сети, системы принятия решений. Интересной задачей по поиску высокоразмерных данных является поиск в сенсорных сетях [8, 9] Предположим, мы имеем набор распределенных в пространстве датчиков. Датчики работают в реальном времени и собирают высокоразмерную информацию, такую как видео, звуки, показатели температур, напряженность поля и т.д. Как показано в [8, 9], наиболее востребован в этих системах поиск ближайшего соседа. Необходимость вести такой поиск в реальном времени, ограниченная пропускная способность канала, а так же ограниченные вычислительные возможности отдельного узла, с которым связан сенсор, делают задачу поиска весьма трудной. Подобные задачи могут быть востребованы в системах принятия решения. На этапе создания модели среды [10] часто требуется определить ближайших соседей к некоторой точке из входных данных и уточнить вероятностную модель, оценки функции полезности.

Приведенные выше примеры показывают, что в реальных поисковых системах может быть различный набор требований на скорость и точность поиска. Данные могут быть как статическими, так и динамическими с высокой скоростью обновления. Для ряда задач на практике возникают серьезные аппаратные ограничения, такие как пропускная способность сетевого канала, необходимость использовать дисковые устройства. Возможно требование на индексирование данных в реальном времени, чтобы обеспечить высокую скорость поиска сразу же после обновления. Реализация каждого из изложенных требований находит трудности, некоторые из которых относятся к фундаментальным теоретическим проблемам, связанным с анализом данных большой размерности. Наличие этих проблем приводит к отсутствию единой модели и устоявшихся подходов решения задач индексации, примененных, например, в реляционной модели. На данный момент, большинство практических задач используют для хранения данных большой размерности неспециализированные базы данных, например NoSql базы, размещая средства индексации и поиска вне базы данных.

Цель этой работы - рассмотреть все возможные, в том числе наиболее общие проблемы индексации в пространствах большой размерности. Так же здесь будут рассмотрены некоторые существующие решения и поставлены вопросы, требующие дальнейших изысканий.

В первом пункте мы рассматриваем общую теоретическую проблему, связанную с анализом данных в пространствах большой размерности. Далее мы более подробно рассмотрим как эта проблема отражается на индексации и кластеризации и делает более трудными определенные практические задачи, такие как индексация распределённых данных и индексация динамических данных.

2. Проблемы индексации и поиска в пространствах большой размерности. Общей и наиболее изученной проблемой для всех приведенных выше задач является проблема именуемая *«Проклятием размерностей»*. Впервые данная проблема была сформулирована Р. Беллманом [11] в книге “Динамическое программирование”. Исследуемая проблема имеет различные интерпретации для задач классификации, регрессионного анализа, кластеризации и поиска. При этом можно выделить два общих аспекта: общее увеличение информации, которую нужно обработать при увеличении числа размерностей, и проблема значимости функции расстояния. Первое вызывает трудности, которые при значительном росте количества размерностей приводят к невозможности численно разрешить задачи анализа данных за приемлемое время. Второй аспект может значительно усложнить исследование функций и параметров распределения данных, скрывая их особенности, что в конечном счете приведет к неверной классификации и нерелевантному поиску. Опишем подробнее проблему расстояния. Для этого обратимся к фундаментальной проблеме ближайшего соседа в пространствах больших размерностей. Бейер и др. [12] ставят вопрос о значимости определения ближайшего соседа в пространстве больших размерностей. Они определяют результат нахождения ближайших соседей как нестабильный (относительно параметра ϵ), если расстояние $dist(q, x)$ от выбранного вектора-запроса q до большинства векторов в наборе данных не превосходит $(1 + \epsilon)dist(q, p)$, где p - ближайший к q сосед, а ϵ - некоторый (малый) параметр. Бейер и др. выводят условие на распределение данных и распределение вектора-запроса, которое позволяет утверждать, что большинство запросов для данных распределений вернет нестабильный результат поиска ближайших соседей. Условием является следующее:

$$\lim_{d\to\infty} \frac{var(dist_d(P_d, Q_d))}{E^2(dist_d(P_d, Q_d))} = 0.$$

Здесь и далее сходимость подразумевается в смысле “по вероятности”. P и Q - произвольный вектор из набора данных и вектор-запрос соответственно. В работе доказано, что выполнение этого условия влечет нестабильность поиска ближайшего соседа, что формально записывается, как:

$$\lim_{d\to\infty} P[dist_{max} \leq (1+\epsilon)dist_{min}] = 1.$$

То есть максимальное возможное расстояние от вектора-запроса до произвольного вектора почти не отличимо от расстояния до ближайшего соседа. Далее, в [12] показано, что на практике приведенное условие выполняется для Евклидова расстояния и ряда наиболее интересных с точки зрения практического применения функций распределения. Таким образом, классическое определение расстояния влечет неразрешимость проблемы ближайшего соседа, которая напрямую связана с задачами поиска, классификации, кластеризации. Гиннебург, Аггарваль и Кеим [13] дополняют результат [12] рядом утверждений для Евклидовой метрики. Основным результатом [13] можно считать рассмотрение проблемы расстояния при уменьшении количества измерений (проекции в подпространство). Здесь выводится качественный критерий таких проекций с точки зрения эффективности функции расстояния в них. Идея основана на применении [30]. Требуется, чтобы среди векторов данных в некоторой проекции были как и близкие к друг другу (вектора внутри одного кластера) так и удаленные друг от друга (принадлежащие разным кластерам или выбросы). На рисунке 2 показаны графики распределения расстояний между двумя произвольными векторами из наборов данных. На рисунке 2a показано распределение расстояний, при котором набор данных располагает хорошо отделимыми кластерами, а на рисунке 2b такое распределение, что большая часть точек оказывается попарно равноудаленными друг от друга. Вариант показанный справа соответствует большему значению качества выбранной проекции. Для численной записи функции качества здесь применяется ядерная оценка плотности. Заметим, что подобный подход является частным случаем задачи поиска наилучшей проекции [31,32]. Г.П. Кригель [14] отмечает, что в рамках задачи кластеризации и поиска ближайших соседей расширяется количество аспектов проклятия размерностей и дополняется следующими двумя: наличие нерелевантных атрибутов у элементов пространства и сложность проведения корреляционного анализа.

Задачи классификации имеют схожие проблемы что и задачи кластеризации. На практике это сводится к тому, что такие классификато-

(a) (b)

Рис. 2. Распределение расстояний между двумя произвольными векторами: (а) с хорошей отделимостью данных; (б) со сложно разделимыми данными

ры, как, например, линейные, оказываются практически бесполезны в пространстве больших размерностей и фактически являются случайными предположениями. Этот факт подробно рассмотрен Фаном Дж. и Фаном И. в [29].

Описанная проблема проклятия размерности приводит к невозможности эффективно применять индексные структуры, пригодные для пространств малой размерности. В следующих пунктах мы более подробнее рассмотрим проблему индексирования, кластеризации и их взаимосвязь в пространствах большой размерности.

3. Индексирование и поиск в пространствах большой размерности. Положим, что мы осуществляем поиск в множестве V векторов-данных в векторном пространстве. В первую очередь нас будет интересовать поиск ближайших соседей. Структуры, которые хорошо применимы для пространств малых размерностей, такие как бинарные деревья, k-d деревья, R-деревья, предложенные Гуттманом [17], всевозможные структуры, основанные на разбиении пространства на практике оказываются неэффективными. Рассмотрим причины неэффективности этих структур. k-d деревья и подобные структуры, основанные на разбиении пространства оказываются неэффективными уже при размерностях $d \geq 10$. Р. Вебер в [18] приводит обоснование неэффективности таких структур. Разбиение пространства по каждой координате производит N^d регионов, где d - размерность, N - количество интервалов разбиения по отдельной координате. При $N = 2$, $d = 100$ мы получим 2^{100} разбиений, что может сильно превзойти количество векторов данных, при этом $N = 2$ может оказаться слишком малым для обеспечения релевантного поиска. Таким образом, при достаточном количестве координат последовательный перебор всех данных может оказаться существенно быстрее поиска с использованием разбиения всего пространства. Данной проблеме будет подвержен любой запрос, кроме может быть точного совпадения. R-деревья и их разновидности стал-

киваются с аналогичной проблемой, которая на практике вырождается в появление большого числа пересекающихся в пространстве ограничивающих прямоугольников [19] Следующие наблюдение в [18] - разреженность данных. Пусть $q = [l_{i_1}, r_{i_1}] \times \ldots \times [l_{i_m}, r_{i_m}]$ - некоторый запрос-диапазон, длинна каждого интервала которого составляет 0.95 от длинны всего диапазона данных для этой координаты. Тогда при $d = 100$ запрос-диапазон накрывает всего лишь $\sim 0.59\%$ объема прямоугольника, ограничивающего все данные во всем пространстве. Отсюда можно утверждать, что выполняя запросы-диапазоны в том числе и как часть поиска ближайших соседей мы с большой вероятностью просмотрим лишь незначительно малую часть всех данных. Если рассматривать запрос на поиск ближайших соседей, то нас будут интересовать вектора, попадающие внутрь сферы с заданным радиусом. Объем сферы равен:

$$\frac{\pi^{n/2}}{\Gamma(\frac{n}{2}+1)} \cdot R^d$$

и его значение окажется существенно меньше, чем объем куба с ребром длинной $2R$, то есть еще меньше чем для запроса-диапазона. Таким образом, рассматривая сферу вокруг заданной точки мы просмотрим незначительную часть всех данных и для того, чтобы организовать покрытие всего пространства некоторыми ограничивающими регионами, количество элементов этого покрытия должно быть достаточно велико. Это общее соображение не позволяет строить существенно меньший по количеству объектов индекс на основе разбиения всего пространства.

Многие из первых работ в области создания индексов для пространств больших размерностей поднимают вопрос адаптации методик индексирования, пригодных для пространств меньших размерностей. К ним относятся X-деревья, основанные на уменьшении количества пересекающихся регионов R-деревьев, предложенные Берчтольдом и Кригелем в [19]. Для большинства практических задач с размерностью более 10 эти подходы оказываются несостоятельны ввиду невосприимчивости особенностей данных и невозможности анализа их распределения во всем пространстве - проблема неотличимости расстояний не позволяет группировать или кластеризовывать данные, что требуется в приведенных подходах. Добкин и Липтон в [33] предложили точный алгоритм поиска ближайшего соседа за сублинейное от количества данных время: $O(2^d logn)$. При этом время создания индексной структуры оценивается как $O(n^{2^d})$. Мейсер [34] предложил алгоритм поиска на основе разбиения пространства гиперплоскостями со сложностью $O(d^5 logn)$ и сложностью индексирования $O(n^d)$. На практике

эти алгоритмы лишь незначительно выигрывают по скорости у полного перебора, а сложность индексации не позволяет вести динамические обновления данных.

Другой подход — использование случайных проекций и *Local Sensitive Hashing (LSH)*, разработанный П. Индиком в [36]. Он позволяет вести приближенный поиск ближайших соседей. Рассмотрим его подробнее. Назовем семейство хэш-функций $\mathcal{H} = \{h : S \to U\}$, где U - некоторое пространство хэшей (p_1, p_2, r_1, r_2) - чувствительными, если для любых q, p выполнено:

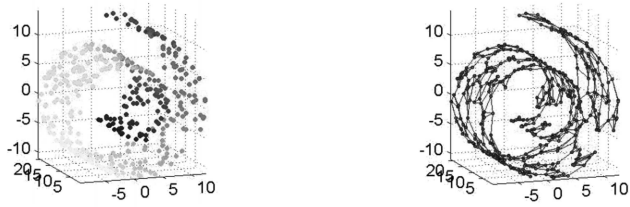
$$p \in B(q, r_1) \Rightarrow P_{\mathcal{H}}(h(p) = h(q)) \geq p_1 \quad (1)$$

$$p \notin B(q, r_2) \Rightarrow P_{\mathcal{H}}(h(p) = h(q)) \leq p_2, \quad (2)$$

где $B(q, r)$ - шар в пространстве поиска с радиусом r и центром q. При этом предполагается, что $p_1 > p_2$ и $r_1 < r_2$. Определим семейство функций $\mathcal{G}\{g : S \to U^k\}$ имеющих вид $g(p) = (h_1(p), \ldots, h_k(p))$, $h_j \in \mathcal{H}$ Далее, для индексации мы выбираем некоторый l и выбираем произвольный набор функций $g_1, \ldots, g_l$ из $\mathcal{G}$. Для каждого вектора данных p мы вычисляем корзину $g_j(p)$ для всех $1 \leq j \leq l$. Чтобы уменьшить количество корзин мы учитываем только непустые. Запрос q осуществляется так: выбираются все существующие в индексе корзины $g_j(q)$, $1 \leq j \leq l$, далее для всех векторов из этих корзин вычисляется расстояние до q до тех пор пока не будут найдены все подходящие или некоторая часть подходящих. Сложность индексирования LSH - $O(nlkt)$, где t - сложность вычисления функции из $\mathcal{H}$. В качестве функций из $\mathcal{H}$ могут быть рассмотрены, например, использующие случайную проекцию: $h(p) = sign(p \cdot w)$, где w - случайно выбранный вектор. Эффективность подхода зависит от выбора $\mathcal{H}$ и метрики в пространстве поиска. В тоже время, для всех известных семейств $\mathcal{H}$ можно наблюдать стремительную деградацию показателей релевантности при $d >> 10$. Также проблеме неразличимости расстояний подвержена часть алгоритма поиска, в которой происходит перебор вычисленных корзин $g_j(q)$, $1 \leq j \leq l$. Сложность алгоритма поиска q с помощью LSH - $O(L(kt + dnp_2^k))$ что делает его на практике одним из наиболее быстрых поисков в пространствах большой размерности. Кроме того, простота обновления хэш-таблиц позволяет вести индексирование динамических данных. Подробное описание и проблемы данного подхода рассматриваются в работах Индика и Андони [35, 37]. Упомянутый подход со случайной проекцией может быть рассмотрен как самостоятельное решение задачи поиска. Его можно рассматривать как грубый и быстрый метод редукции размерностей.

Редукция размерности — общий подход для борьбы с проклятием размерностей. Рассмотрим основные подходы. Метод главных компонент нацелен на аппроксимацию данных линейными многообразиями существенно меньших размерностей чем исходные. Пирсон в [20] рассматривал способы построения линейных многообразий, аппроксимирующих набор векторов. Эквивалентная формулировка этой задачи - поиск подпространства, заданного набором ортогональных векторов (главных компонент), вдоль которых набор данных имеет наибольшую выборочную дисперсию. Главную компоненту можно рассматривать также как вектор, вдоль которого задан сигнал, соответственно ортогональный ей вектор - направление шума. Решение задачи нахождения главных компонент заключается в нахождении ортогонального базиса пространства собственных векторов матрицы ковариации $C = X \cdot X^T$, где X - матрица, составленная из векторов-данных. В качестве базиса отбираются те вектора, которым соответствуют наибольшие значения собственных чисел. Суть метода - спектральное разложение матрицы C может быть рассмотрена как задача о сингулярном разложении (SVD) матрицы X. Вычислительная сложность стандартных алгоритмов оценивается как $O(nd^2)$, где n - количество векторов-данных, а d - размерность. Подробное описание и случаи применения PCA и SVD рассмотрены в книгах [39, 40].

В качестве многообразий меньших размерностей, аппроксимирующих набор данных можно рассматривать и нелинейные многообразия. На рисунке 3а изображено множество (известное среди исследователей как “Швейцарский рулон”), аппроксимируемое гладким многообразием второго порядка. Одним из алгоритмов, позволяющих выделить аппроксимирующее многообразие называется ISOMAP, предложенный Тененбаумом в [41]. Его суть сводится к следующим шагам: 1) по-



(a) (b)

Рис. 3. Множество “Швейцарский Рулон”: (а) исходные данные; (б) граф, связывающий ближайшие соседи

строение графа, связывающего ближайших соседей (на рисунке 3b); 2) определение новой метрики $dist(x_i, x_j)$ между точками на графе, как кратчайший путь на этом графе. 3) применение метода мультиразмерностного шкалирования, описанного в [42,43]. Сложность ISOMAP оценивается как $O(n^2 logn + n^2 d)$.

Большинство работ по редукции размерности не рассматривает ее применение к задачи индексации и поиска. Небольшой обзор применимости PCA к задачи поиска ближайших соседей рассмотрен в [38]. Однако, одна из работ успешно сочетает редукцию размерностей путем проекций на линейные подпространства с традиционными методами сжатия информации для индексации. В работе Р. Дугласа [44] рассматривается файл приближенных векторов (Vector approximation file, VA-файл). Эта работа вводит KVA-файлы (Ядерный VA-файл), которые расширяют VA-Файлы. Результаты экспериментов с использованием больших наборов данных изображений (примерно 100000 изображений с размерностью в 463 координат) доказывают эффективность этого метода.

Главным теоретическим недостатком рассмотренных методов редукции размерностей является его глобальность: анализируется весь набор данных сразу, в то время как для отдельных его частей могут быть выявлены различные аппроксимирующие многообразия. Локальные методы будут рассмотрены в следующем пункте в рамках проблемы кластеризации. Еще одним недостатком является высокая вычислительная сложность. Как было указано, PCA имеет сложность $O(nd^2)$, ISOMAP - $O(n^2 logn + n^2 d)$. На практике такая сложность вынуждает отказываться от подобных алгоритмов, в случае если данные обновляются динамически. Вопросы о разработки инкрементальных алгоритмов редукции и применении редукции к индексированию являются предметом исследований.

4. Кластеризация в пространствах большой размерности. Работа Кригеля [14] является первым наиболее подробным обзором алгоритмов кластеризации в пространствах большой размерности, в которой приведена классификация алгоритмов, их сравнение и анализ применимости. *Подпространственная кластеризация (Subspace clustering)* и *проекционная кластеризация (projection clustering)* рассмотренные в [14, 15] нацелены на нахождение подмножеств элементов близких друг к другу в определенных подпространствах, как, например, показано на рисунке 4. Подразумевается, что подпространства могут быть как и параллельны осям координат так и произвольно ориентированы. Алгоритмы проекционной кластеризации нацелены на нахожде-

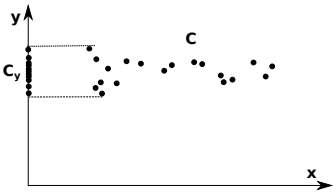


Рис. 4. Набор данных C образует кластер в подпространстве Y

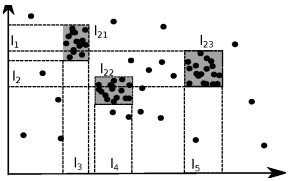


Рис. 5. Пример алгоритма, действующего снизу вверх, обнаруживающего прямоугольники с большой плотностью данных

ние подпространств, в которых элементы образуют кластера. Подпространственная кластеризация [14, 15] напротив, нацелена на нахождение всех кластеров во всех возможных подпространствах. С алгоритмической точки зрения подходы можно разделить на следующие: сверху вниз: [14, 15], когда кластера сначала ищутся в пространствах больших размерностей, а затем осуществляется переход к более низкоразмерным пространствам, и снизу вверх — сначала ищутся кластера в одномерных пространствах, затем в двумерных и так далее. Конкретные реализации алгоритмов предложены в работах [22, 24, 27, 28]

На рисунке 5 изображен пример алгоритма [28] действующего снизу вверх, который с помощью построения сетки с адаптивным размером ячеек выявляет сначала отрезки $l_1, \ldots, l_5$ со скоплениями данных в одномерных пространствах, затем, пересекая их для разных координат, выявляет отрезки в двумерных пространствах: l_{21}, l_{22}, l_{23}. Алгоритм продолжает работу, пока для размерности d' не будет обнаружено достаточно плотных пересечений. Алгоритмы типа [27] начиная с полного пространства делают две выборки C - предполагаемые центройды кластеров и S - предполагаемые данные из кластеров. Далее действуют как алгоритм k-средних (k-means), но кроме того, определяют на каждом шаге релевантные размерности.

У данных алгоритмов существует преимущество перед детальным статистическим анализом данных, основанном на анализе распределения. Во-первых, эти алгоритмы локальны. Для определенного подмножества данных они способны отыскать наиболее подходящее подпространство, в то время как многие алгоритмы размерностной редукции нацелены на поиск одного подпространства для всех данных. Во-вторых, они имеют сложность порядка $O(nd)$.

Следующий класс алгоритмов кластеризации в пространствах большой размерности - *спектральная кластеризация* рассмотренный

в [46, 47]. Спектральная кластеризация нацелена на отыскание кластеров в произвольно ориентированных линейных подпространствах. В ее основе лежит следующий принцип: пусть W - матрица расстояний между векторами из набора данных X размером N. Определим лапласиан матрицы W как : $L = D - W$, где $D_{ij} = 0$ для $i \neq j$ и $D_{ii} = \sum_{1 \leq j \leq N} W_{ij}$. Алгоритм спектральной кластеризации сводится к нахождению L, вычислению собственных k векторов $u_1, \dots u_k$. Вектора выбираются соответствующие наименьшим собственным числам. Далее, применяя известный алгоритм кластеризации, например, k-средних к строкам матрицы U, составленной из $u_1, \dots u_k$ в качестве столбцов. Полученные кластера в пространстве, натянутом на $u_1, \dots u_k$ и есть искомые. В начале работы алгоритма необходимо произвести построение W - матрицы расстояний. Стоит отметить, что можно учитывать только те расстояния между векторами, которые меньше некоторого порога ϵ. Интересный подход к построению W разработан и описан в [45]. Предварительно, для каждого вектора x из изначального набора данных ищется его представление через другие вектора. На языке матриц решается задача:

$$R = ?, \qquad X = XR, \qquad R_{ii} = 0, \quad 1 \leq i \leq N.$$

Причем, для нахождения наиболее компактного такого представления ищется наименьшая по норме матрица: $||R||_{l_1} \to min$. R дает нам представление каждого вектора на некоторой гиперплоскости, все пространство поиска представляется как объединение набора гиперплоскостей. Здесь W может быть выражена как $W = R + R^T$. Подобный подход позволяет перейти к спектральной кластеризации набора дизъюнктных множеств векторов, каждый из которых лежит в отдельной гиперплоскости. Подход получил название *разреженная подпространственная кластеризация (sparse subspace clustering)*.

Интересны также алгоритмы, позволяющие совмещать различные результаты кластеризации для одного и того же набора данных. Данные алгоритмы имеют название Concensus clustering и подробно описаны в исследовании Стрехля и Гоша [26]. Они могут применяться в том случае, если данные были получены из разных источников и сформировано несколько наборов кластеров, что возможно в случае распределенных систем. Основным минусом алгоритмов, рассмотренных в [26] является их сложность: $O(nkr)$ - в лучшем случае, где n - количество всех векторов-данных, r - количество источников данных, k - количество всех известных кластеров. Такая сложность делает затруднительным вычисление совмещенного набора кластеров в реаль-

ном времени.

Также оказываются важны исследования проблемы избыточности подпространственной кластеризации, подробно изученные в [22, 23]. Поднимается еще один аспект проклятия размерностей: наборы кластеров во всех рассмотренных подпространствах могут содержать существенно большее количество элементов чем в одном пространстве малой размерности. Обуславливается это наличием 2^d возможных подпространств, где d - размерность исходного пространства. Часть кластеров в пространствах меньшей размерности может пересекаться с кластерами большей размерности. Возникает избыточность таких наборов. Как правило, кластера обнаруженные в подпространствах большей размерности считаются более интересными, однако для того чтобы назначить кластеру значение его ценности в [22] вводят более общую модель. Пусть определен подпространственный кластер $c = (S, X)$, где S - подпространство, а X - подмножество векторов. В [22] вводят следующую функцию полезности кластера c:

$$value(c) = \frac{|Covered(X)|}{k(S, X)},$$

где $Covered(X)$ - множество элементов из X не покрытых никаким другим известным кластером c', а $k(S, X)$ - функция избыточности кластера, например, размерность S. Введение подобных $value(c)$ функций может быть частично решено при помощи методов поиска наилучшей проекции, предложенных в работах Хьюбера [31] и Фридмана [32].

5. Применение результата кластеризации к построению индексной структуры для поиска ближайших соседей. Задача поиска ближайших соседей (KNN-поиск) в пространстве большой размерности требует построения индексной структуры, проблемы которой описаны выше. Невозможность проводить кластеризацию в пространстве большой размерности также делает невозможность построения индексных деревьев на основе кластеров. Интерес вызывает использование подпростанственных кластеров. В простейшем случае, индексом может являться набор представлений кластеров в подпространстве, а в качестве алгоритма поиска - классификация запроса относительно рассматриваемого множества кластеров. Рассмотрение кластеров помогает существенно сократить поиск, для данных с определенным распределением, в то время как для данных с большим количеством некластеризованных объектов (outliers) этот подход неэффективен. По сравнению с базовыми алгоритмами редукции размерностей, подпространтсвенная кластеризация позволяет выявлять подходящие многообразия меньшой

размерности локально, то есть применительно к ограниченному набору векторов-данных, что делает ее более точным подходом.

Ввиду того, что различные алгоритмы кластеризации для пространств большой размерности могут выявлять наборы кластеров с различными параметрами и, в соответствии с различными моделями, возникает вопрос о нахождении наилучшего подмножества кластеров для использования в индексе.

На текущий момент вопрос применения подпространственной кластеризации к задачи индексирования в пространствах большой размерности остается открытым. В работе Гюннемана [21] предложена иерархическая индексная структура на основе подпространственных кластеров, позволяющая вести как точный поиск так и поиск ближайших соседей. Основной проблемой использования алгоритмов подпространственной кластеризации для индексирования является ее вычислительная сложность и невозможность вести динамическое обновление индекса за приемлемое время. В этом случае требуется разработать адаптации вышеописанных алгоритмов кластеризации к динамическим данным.

6. Заключение. Индексирование данных большой размерности - значительно более сложная задача, чем построение индексов для малого количества атрибутов. Индексация требует применения алгоритмов машинного обучения и анализа данных, таких как редукция размерностей и кластеризации в виду того, что основная ее проблема - проклятие размерностей, концептуально неразрешима для определенных наборов данных без перехода к подпространствам. Индексная структура с точки зрения практики должна поддерживать такие операции как динамическое обновление и иметь возможность создаваться распределённо (на нескольких вычислительных узлах). Однако, большинство известных алгоритмов кластеризации и редукции размерности, необходимых для индексации, крайне трудоемки и не имеют версий для динамических данных и распределенных вычислительных сред. В качестве двух основных вопросов, требующих решения в рамках задачи индексации в пространстве большой размерностей мы выделяем:

1. Разработка алгоритмов и оптимизация, применяющих подпространственную кластеризацию для создания индекса.
2. Оптимизация необходимых алгоритмов анализа данных для ведения инкрементальной (динамической) индексации и выполнения ее в распределенных средах.

Отсутствие единого и эффективного теоретического подхода к индексированию данных большой размерности привело к тому, что на

практике нет общепринятой технологии, позволяющей строить базы данных большой размерности. В то же время, наличие баз данных для распределенных вычислений, например, использующих MapReduce, позволяет несколько упростить решение частных задач индексирования и поиска.

Литература

1. *Donoho D. L.* High-dimensional data analysis: the curses and blessings of dimensionality // American Mathematical Society Conf. Math Challenges of the 21st Century 2000.
2. *Manning C.D., Raghavan P., Schütze H.* Introduction to Information Retrieval // American Mathematical Society Conf. Math Challenges of the 21st Century 2000.
3. *Augenlicht L. H., Kobrin D.* Cloning and Screening of Sequences Expressed in a Mouse Colon Tumor // Cancer Research. 1982. vol.42. pp. 1088-1093.
4. *Dash M., Liu H.* Feature Selection for Classification // Intelligent Data Analysis. 1997. vol.1. pp. 131-156.
5. *Jain A.K., Duin R.P.W., Jianchang M.* Statistical pattern recognition: a review // Pattern Analysis and Machine Intelligence, IEEE Transactions on. 2000. vol.22. pp. 4-37.
6. *Koller D., Sahami M.* Toward optimal feature selection // In 13th International Conference on Machine Learning. 1995. pp. 284–292.
7. *Glazko G., Coleman M., Mushegian A.* Similarity searches in genome-wide numerical data sets // Biology Direct. 2006. vol.1.
8. *Yan T., Ganesan D., Manmatha R.* Distributed Image Search in Sensor Networks // Proceeding SenSys '08 Proceedings of the 6th ACM conference on Embedded network sensor systems. 2008.
9. *Deshpande A., Guestrin C., Madden S.R., Hellerstein J.M., Hongn W.* Model-driven data acquisition in sensor networks // Proceeding VLDB '04 Proceedings of the Thirtieth international conference on Very large data bases. 2004. vol.30. pp. 588-599.
10. *Russell S.J., Norvig P., Canny J.F., Malik J.M. Edwards D.D.* Artificial intelligence: a modern approach // Prentice hall Englewood Cliffs. 1995.
11. *Bellman R. E.* Dynamic programming // Princeton University Press. 1957.
12. *Beyer K., Goldstein J., Ramakrishnan R., Shaft U.* When Is Nearest Neighbor Meaningful // ICDT '99 Proceedings of the 7th International Conference on Database Theory. 1999. pp. 217-235.
13. *Hinneburg A., Aggarwal C., Keim D.A.* What Is the Nearest Neighbor in High Dimensional Spaces // VLDB '00 Proceedings of the 26th International Conference on Very Large Data Bases. 2000. pp. 506-515.
14. *Kriegel H.-P., Kröger P., Zimek A.* Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering // ACM Transactions on Knowledge Discovery from Data. 2009. vol.3 no.3. pp. 506-515.

15. *Parsons L., Haque E., Liu H.* Subspace clustering for high dimensional data: a review // ACM SIGKDD Explorations Newsletter - Special issue on learning from imbalanced dataset. 2004. vol.6.
16. *Böhm C., Berchtold S., Keim D.A.* Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases // ACM Computing Surveys (CSUR). 2001. no.33. vol.3.
17. *Guttman A.* R-Trees: A Dynamic Index Structure for Spatial Searching // Proceedings of 1984 ACM SIGMOD International Conference on Management of Data. 1984. pp. 47-57.
18. *Weber R., Schek H.-J., Blott S.* A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces // VLDB'98, Proceedings of 24rd International Conference on Very Large Data Bases. 1998. pp. 194-205.
19. *Berchtold S., Keim D. A., Kriegel H.-P.* The X-tree: An Index Structure for High-Dimensional Data // VLDB '96 Proceedings of the 22th International Conference on Very Large Data Bases. 1996. pp. 28-39.
20. *Pearson K.* Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions // Philosophical Magazine Bases. 1901. vol.2. pp. 559-572.
21. *Gunnemann S., Kremer H., Lenhard D., Seidl T.* Subspace Clustering for Indexing High Dimensional Data:A Main Memory Index based on Local Reductions and Individual Multi-Representations // Proc. International Conference on Extending Database Technology. 2011. pp. 237-248.
22. *Müller E., Assent I., Günnemann S., Krieger R., Seidl T.* Relevant Subspace Clustering: Mining the Most Interesting Non-Redundant Concepts in High Dimensional Database // Proc. IEEE International Conference on Data Minings. 2009. pp. 377-386.
23. *Assent I., Krieger R., Miller E., Seidl T.* INSCY:Indexing subspace clusters with in-process-removal of redundancy // Data Mining, 2008. ICDM '08. Eighth IEEE International Conference. 2008. pp. 719-724.
24. *Assent I., Krieger R., Miller E., Seidl T.* EDSC: efficient density-based subspace clustering // Proceedings of the 17th ACM conference on Information and knowledge management. 2008. pp. 1093–1102.
25. *Houle M. E., Kriegel H.-P., Kröger P., Schubert E., Zimek A.* Can Shared-Neighbor Distances Defeat the Curse of Dimensionality? // Scientific and Statistical Database Management. 2010. pp. 482-500.
26. *Strehl A., Ghosh J.* Cluster ensembles — a knowledge reuse framework for combining multiple partitions // The Journal of Machine Learning Research. 2003. pp. 583-617.
27. *Aggarwal C.C., Wolf J.L., Phillip S., Yu C.P., Jong S.P.* Fast algorithms for projected clustering // Proceedings of the 1999 ACM SIGMOD international conference on Management of data. 1999. pp. 61–72.
28. *Nagesh H., Goil S.* Adaptive Grids for Clustering Massive Data Sets // SIAM International Conference on Data Mining - SDM. 2002.

29. *Fan J., Fan Y., Wu Y.* High-dimensional classification // World Scientific, New Jersey. 2011. pp. 3-37.
30. *Silverman B.* Density Estimation for Statistics and Data Analysis // Monographs on Statistics and Applied Probability, London: Chapman and Hall. 1986.
31. *Huber P.J.* Projection Pursuit // The Annalas of Statistics. 1985. vol.2. no.13. pp. 435—475.
32. *Friedman J.H.* Exploratory projection pursuit // Journal of American Statistical Association. 1987. no.82. pp. 249—266.
33. *Dobkin D., Lipton R.J.* Multidimensional Searching Problems // SIAM Journal on Computing. 1974. no.5. pp. 181–186.
34. *Meiser S.* Point location in arrangemenm of hyperplanes // Information and Computation. 1993. no.106. pp. 286–303.
35. *Andoni A., Indyk P.* Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions // Communications of the ACM - 50th anniversary issue. 2008. vol.51. pp. 117-122.
36. *Indyk P., Motwani R.* Approximate nearest neighbors: towards removing the curse of dimensionality // Proceedings of the thirtieth annual ACM symposium on Theory of computing. 1998. pp. 604-613.
37. *Datar M., Immorlica N., Indyk P., Mirrokni V.S.* Locality-sensitive hashing scheme based on p-stable distributions // Proceedings of the twentieth annual symposium on Computational geometry. 2004. pp. 253-262.
38. *Deegalla S., Bostrom H.* Reducing High-Dimensional Data by Principal Component Analysis vs. Random Projection for Nearest Neighbor Classification // Proceedings of the 5th International Conference on Machine Learning and Applications. 2006. pp. 245-250.
39. *Ian J.* Principal component analysis // Wiley Online Library. 2005.
40. *Gorban A. N.* Principal manifolds for data visualization and dimension reduction // Springer. 2007. vol.58.
41. *Tenenbaum J.B., De Silva V., Langford J.C.* A global geometric framework for nonlinear dimensionality reduction // Science. 2000. vol.290. no.5500. pp. 2319–2323.
42. *Kruskal J.B.* Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis // Psychometrika. 1964. vol.29. no.1. pp. 1-27.
43. *Cox T.F., Cox M.A.A.* Multidimensional scaling // CRC Press. 2010.
44. *Heisterkamp D.R., Peng J.* Kernel VA-files for relevance feedback retrieval // Proceedings of the 1st ACM international workshop on Multimedia databases. 2003. pp. 48-54.
45. *Heisterkamp D.R., Peng J.* Sparse subspace clustering // IEEE Conference on Computer Vision and Pattern Recognition CVPR 2009. 2009. pp. 2790-2797.
46. *Von Luxburg U.* A tutorial on spectral clustering // Statistics and computing. 2009. vol.17. pp. 395-416.

47. *Shi J., Malik J.* Normalized cuts and image segmentation // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2000. vol.22. pp. 888–905.

References

1. Donoho D. L. High-dimensional data analysis: the curses and blessings of dimensionality. American Mathematical Society Conf. Math Challenges of the 21st Century 2000.
2. Manning C.D., Raghavan P., Schütze H. Introduction to Information Retrieval. American Mathematical Society Conf. Math Challenges of the 21st Century 2000.
3. Augenlicht L. H., Kobrin D. Cloning and Screening of Sequences Expressed in a Mouse Colon Tumor. Cancer Research. 1982. vol.42. pp. 1088-1093.
4. Dash M., Liu H. Feature Selection for Classification. Intelligent Data Analysis. 1997. vol.1. pp. 131-156.
5. Jain A.K., Duin R.P.W., Jianchang M. Statistical pattern recognition: a review. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 2000. vol.22. pp. 4-37.
6. Koller D., Sahami M. Toward optimal feature selection. In 13th International Conference on Machine Learning. 1995. pp. 284–292.
7. Glazko G., Coleman M., Mushegian A. Similarity searches in genome-wide numerical data sets. Biology Direct. 2006. vol.1.
8. Yan T., Ganesan D., Manmatha R. Distributed Image Search in Sensor Networks. Proceeding SenSys ’08 Proceedings of the 6th ACM conference on Embedded network sensor systems. 2008.
9. Deshpande A., Guestrin C., Madden S.R., Hellerstein J.M., Hongn W. Model-driven data acquisition in sensor networks. Proceeding VLDB ’04 Proceedings of the Thirtieth international conference on Very large data bases. 2004. vol.30. pp. 588-599.
10. Russell S.J., Norvig P., Canny J.F., Malik J.M., Edwards D.D. Artificial intelligence: a modern approach. Prentice hall Englewood Cliffs. 1995.
11. Bellman R. E. Dynamic programming. Princeton University Press. 1957.
12. Beyer K., Goldstein J., Ramakrishnan R., Shaft U. When Is Nearest Neighbor Meaningful. ICDT ’99 Proceedings of the 7th International Conference on Database Theory. 1999. pp. 217-235.
13. Hinneburg A., Aggarwal C., Keim D.A. What Is the Nearest Neighbor in High Dimensional Spaces. VLDB ’00 Proceedings of the 26th International Conference on Very Large Data Bases. 2000. pp. 506-515.
14. Kriegel H.-P., Kröger P., Zimek A. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering. ACM Transactions on Knowledge Discovery from Data. 2009. vol.3 no.3. pp. 506-515.
15. Parsons L., Haque E., Liu H. Subspace clustering for high dimensional data: a review. ACM SIGKDD Explorations Newsletter - Special issue on learning from imbalanced dataset. 2004. vol.6.

16. Böhm C., Berchtold S., Keim D.A. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. ACM Computing Surveys (CSUR). 2001. no.33. vol.3.
17. Guttman A. R-Trees: A Dynamic Index Structure for Spatial Searching. Proceedings of 1984 ACM SIGMOD International Conference on Management of Data. 1984. pp. 47-57.
18. Weber R., Schek H.-J., Blott S. A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. VLDB'98, Proceedings of 24rd International Conference on Very Large Data Bases. 1998. pp. 194-205.
19. Berchtold S., Keim D. A., Kriegel H.-P. The X-tree: An Index Structure for High-Dimensional Data. VLDB '96 Proceedings of the 22th International Conference on Very Large Data Bases. 1996. pp. 28-39.
20. Pearson K. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. Philosophical Magazine Bases. 1901. vol.2. pp. 559-572.
21. Gunnemann S., Kremer H., Lenhard D., Seidl T. Subspace Clustering for Indexing High Dimensional Data:A Main Memory Index based on Local Reductions and Individual Multi-Representations. Proc. International Conference on Extending Database Technology. 2011. pp. 237-248.
22. Müller E., Assent I., Günnemann S., Krieger R., Seidl T. Relevant Subspace Clustering: Mining the Most Interesting Non-Redundant Concepts in High Dimensional Database. Proc. IEEE International Conference on Data Minings. 2009. pp. 377-386.
23. Assent I., Krieger R., Miller E., Seidl T. INSCY:Indexing subspace clusters with in-process-removal of redundancy. Data Mining, 2008. ICDM '08. Eighth IEEE International Conference. 2008. pp. 719-724.
24. Assent I., Krieger R., Miller E., Seidl T. EDSC: efficient density-based subspace clustering. Proceedings of the 17th ACM conference on Information and knowledge management.2008. pp. 1093–1102.
25. Houle M. E., Kriegel H.-P., Kröger P., Schubert E., Zimek A. Can Shared-Neighbor Distances Defeat the Curse of Dimensionality?. Scientific and Statistical Database Management. 2010. pp. 482-500.
26. Strehl A., Ghosh J. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. The Journal of Machine Learning Research. 2003. pp. 583-617.
27. Aggarwal C.C., Wolf J.L., Phillip S., Yu C.P., Jong S.P. Fast algorithms for projected clustering. Proceedings of the 1999 ACM SIGMOD international conference on Management of data. 1999. pp. 61–72.
28. Nagesh H., Goil S. Adaptive Grids for Clustering Massive Data Sets. SIAM International Conference on Data Mining - SDM. 2002.
29. Fan J., Fan Y., Wu Y. High-dimensional classification. World Scientific, New Jersey. 2011. pp. 3-37.

30. Silverman B. Density Estimation for Statistics and Data Analysis. Monographs on Statistics and Applied Probability, London: Chapman and Hall. 1986.
31. Huber P.J. Projection Pursuit. The Annalas of Statistics. 1985. vol.2. no.13. pp. 435—475.
32. Friedman J.H. Exploratory projection pursuit. Journal of American Statistical Association. 1987. no.82. pp. 249—266.
33. Dobkin D., Lipton R.J. Multidimensional Searching Problems. SIAM Journal on Computing. 1974. no.5. pp. 181–186.
34. Meiser S. Point location in arrangemenm of hyperplanes. Information and Computation. 1993. no.106. pp. 286–303.
35. Andoni A., Indyk P. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. Communications of the ACM - 50th anniversary issue. 2008. vol.51. pp. 117-122.
36. Indyk P., Motwani R. Approximate nearest neighbors: towards removing the curse of dimensionality. Proceedings of the thirtieth annual ACM symposium on Theory of computing. 1998. pp. 604-613.
37. Datar M., Immorlica N., Indyk P., Mirrokni V.S. Locality-sensitive hashing scheme based on p-stable distributions. Proceedings of the twentieth annual symposium on Computational geometry. 2004. pp. 253-262.
38. Deegalla S., Bostrom H. Reducing High-Dimensional Data by Principal Component Analysis vs. Random Projection for Nearest Neighbor Classification. Proceedings of the 5th International Conference on Machine Learning and Applications. 2006. pp. 245-250.
39. Ian J. Principal component analysis. Wiley Online Library. 2005.
40. Gorban A. N. Principal manifolds for data visualization and dimension reduction. Springer. 2007. vol.58.
41. Tenenbaum J.B., De Silva V., Langford J.C. A global geometric framework for nonlinear dimensionality reduction. Science. 2000. vol.290. no.5500. pp. 2319–2323.
42. Kruskal J.B. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika. 1964. vol.29. no.1. pp. 1-27.
43. Cox T.F., Cox M.A.A. Multidimensional scaling. CRC Press. 2010.
44. Heisterkamp D.R., Peng J. Kernel VA-files for relevance feedback retrieval. Proceedings of the 1st ACM international workshop on Multimedia databases. 2003. pp. 48-54.
45. Heisterkamp D.R., Peng J. Sparse subspace clustering. IEEE Conference on Computer Vision and Pattern Recognition CVPR 2009. 2009. pp. 2790-2797.
46. Von Luxburg U. A tutorial on spectral clustering. Statistics and computing. 2009. vol.17. pp. 395-416.
47. Shi J., Malik J. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2000. vol.22. pp. 888–905.

Новиков Борис Асенович — д-р физ-мат. наук, профессор, заведующий кафедрой информационно-аналитических систем математико-механического факультета Санкт-Петербургского государственного университета. Область научных интересов: методы организации информации, базы данных, индексирование, кластеризация, оптимизация запросов. Число научных публикаций — 46. borisnov@acm.org, http://www.math.spbu.ru/user/boris_novikov; Университетский проспект, дом 28, Санкт-Петербург, 198504, Россия; р.т. +7 921 914 8534.

Novikov Boris — Ph.D., Dr. Sci., professor, head of departemnt of Analytical Information Systems Science, Faculty of Mathematics and Mechanics, Saint-Petersburg State University. Research interests: information managment, databases, indexing, clustering, query optimization. Number of publications — 46. borisnov@acm.org, http://www.math.spbu.ru/user/boris_novikov; 28, Universitetskiy avenue, St. Petersburg, 198504, Russia; office phone +7 921 914 8534.

Судос Иван Викторович — аспирант кафедры информационно-аналитических систем математико-механического факультета Санкт-Петербургского государственного университета. Область научных интересов: методы организации информации, информационный поиск, индексирование, кластеризация. Число научных публикаций — 1. iv.teh.adr@gmail.com, http://blog.sudos.net; Университетский проспект, дом 28, Санкт-Петербург, 198504, Россия; р.т. +7 921 747 81 61.

Sudos Ivan — Ph.D. student of departemnt of Analytical Information Systems Science, Faculty of Mathematics and Mechanics, Saint-Petersburg State University. Research interests: information managment, information retrieval, indexing, clustering. The number of publications — 1. iv.teh.adr@gmail.com, http://blog.sudos.net; 28, Universitetskiy avenue, St. Petersburg, 198504, Russia; office phone +7 921 747 81 61

РЕФЕРАТ

Новиков Б.А., Судос И.В. **Индексирование данных в пространстве большой размерности.**

В статье производится обзор проблем и решений задачи индексации в рамках задачи поиска ближайших соседей в пространствах большой размерности. Данные в пространствах большой размерности характеризуются наличием большого количества атрибутов и могут быть рассмотрены как векторное представление сложных объектов, таких как изображения, сигналы, временные ряды. Авторы статьи ставят задачу оценивания сложности индексирования и поиска ближайших соседей в пространствах большой размерности, которые не возникают в пространствах размерности не более 3. Для этого осуществляется обзор работ, посвященных фундаментальной проблеме ”проклятия размерности” и способам ее разрешения.

Статья состоит из частей: 1) Введение, 2) Проблемы индексации и поиска в пространствах большой размерности, 3) Индексирование и поиск в пространствах большой размерности, 4) Кластеризация в пространствах большой размерности, 5) Применение результата кластеризации к построению индексной структуры для поиска ближайших соседей, 6) Заключение. Во введении формулируется задача поиска в пространстве больших размерностей, обозначаются примеры практического применения, атакже указывается наличие некоторых теоретических проблем и влияния аппаратных ограничений. Во второй части раскрывается суть проблем поиска и индексации в пространствах большой размерности. Описываются сопутствующие аспекты проклятия размерности. В третьей части приводится обзор существующих попыток разрешить обозначенные проблемы индексирования. Рассмотрены алгоритмы, предложенные в работах, дана оценка их применимости. В четвертой части производится обзор работ посвященных задачи кластеризации в пространствах большой размерности. Указывается ряд проблем, успешно преодоленный рассмотренными алгоритмами. Данные проблемы имеют много общего с проблемами в индексировании. Пятая часть ставит вопрос о возможном применении результатов кластеризации в пространствах большой размерности к индексированию. В заключении обобщаются проблемы, обозначенные в обозреваемых работах и выделяется два основных направления исследования.

Авторы подробно дают описание теоретическим аспектам, при рассмотрении существующих работ в области индексирования данных большой размерности, авторы приводят возможные проблемы, не указанные в работах, такие, например, высокая вычислительная сложность. В частности ставится под сомнение эффективность применения алгоритмов размерностной редукции в случае динамически обновляемых данных.

SUMMARY

Novikov B.A., Sudos I.V. **Indexing of data in high dimensional spaces.**

The survey of the problems and solutions of the indexing problem in the scope of nearest neighbour search in high dimensional spaces problem is conducted in the article. High dimensional data is characterized with a large number of attributes and can be considered as a vector representation of complex objects such as images, signals and time series. Authors of the paper pose a problem to determine a difficulties of indexing and search of nearest neighbours in the high dimensional spaces that are not essential for in the spaces with dimensionality 3 or less. For this purpose, the survey of the works devoted to the fundamental problem of the Curse of dimensionality. The works on the curse of dimensionality and the ways of fighting it are regarded.

The article comprise the following sections: 1) Introduction, 2) The problems of indexing and search in the high dimensional space, 3) Indexing an search in high dimensional spaces, 4) Clustering in high dimensional spaces. 5) Application of clustering result to the construction of index structure for a nearest neighbours search 6) Conclusion. The problem of search in high dimensional spaces is posed in the introduction, the applied examples are noted and the existence of some theoretical problems and hardware limitations are given. The second section discovers the core of search and indexing difficulties. The associated aspects of the curse of dimensionality are described. The survey of the existing attempts to resolve the mentioned difficulties is given in the third section. The algorithms proposed in the works are considered and the evaluation of their applicability is given. The survey of the works devoted to the clustering in high dimensional spaces is conducted in the fourth section. The range of difficulties that is successfully overcome by the algorithms considered. The given problems have the same nature with indexing problems. The fifth section rise the question on the possible application of the clustering result in high dimensional spaces to indexing. The conclusion generalizes observed problem and two major directions of research are emphasized.

Authors give a possible difficulties such as computational complexity of high dimensional data indexing that are not specified in the works. In particular, the possibility of dimensional reduction algorithms application is impugned in the case of dynamically updated data.