

ВЫБОР СПОСОБА ДЕКОМПОЗИЦИИ ПРОСТРАНСТВА ПРИЗНАКОВ ДЛЯ НЕЛИНЕЙНОГО СНИЖЕНИЯ РАЗМЕРНОСТИ

Мясников Е.В.

Самарский государственный аэрокосмический университет имени академика С.П. Королёва
(национальный исследовательский университет) (СГАУ),
Институт систем обработки изображений РАН

Аннотация

В работе рассматриваются два подхода к иерархической декомпозиции пространства признаков для повышения эффективности нелинейного метода снижения размерности. Первый подход, предложенный автором настоящей работы, основывается на декомпозиции исходного пространства признаков с использованием иерархической кластеризации. Второй оригинальный подход основывается на иерархической декомпозиции целевого пространства с использованием KD-деревьев. Анализируемые подходы оцениваются с точки зрения эффективности работы нелинейного метода снижения размерности.

Ключевые слова: снижение размерности, декомпозиция пространства признаков, иерархическая кластеризация, KD-деревья.

Введение

Методы снижения размерности данных, действующие по принципу сохранения попарных расстояний между объектами, могут применяться как средства отображения многомерных данных в научных исследованиях и производственной деятельности в целом ряде областей: биологии, генетике, социологии, экономике и т.д. В современных информационных системах такие методы могут применяться при создании систем навигации по базам данных мультимедийной информации [1, 2, 3, 4, 5], а также при организации интерфейсов доступа к виртуальным каталогам. В области анализа и обработки изображений нелинейные методы снижения размерности нашли своё применение не только при проведении научных исследований, но и при решении ряда прикладных задач: создании автоматизированных систем сегментации изображений, составлении тематических карт по спутниковым снимкам и др.

Существуют различные критерии, применяемые для оценки качества отображения многомерных данных в пространство малой размерности. В частности, ошибка отображения может быть записана в виде:

$$\varepsilon_k = \lambda \sum_{\substack{o_i, o_j \in O \\ i < j}} \omega_{ij} \left(d(o_i, o_j) - d^*(o_i, o_j) \right)^2. \quad (1)$$

Здесь λ, ω_{ij} – некоторые константы, $d(o_i, o_j)$ и $d^*(o_i, o_j)$ – расстояние между объектами o_i и o_j соответственно в многомерном пространстве и пространстве малой размерности, O – всё множество объектов.

При некоторых значениях констант выражение (1) сводится к известным ошибкам Крускала [6] и Сэммона [7] (см. выражение (3) ниже). При этом координаты точек y_i в пространстве малой размерности могут быть легко определены, например, методом наискорейшего спуска:

$$y_i(t+1) = y_i(t) + 2\alpha\lambda \cdot \sigma_i(t), \quad (2)$$

$$\sigma_i(t) = \sum_{\substack{o_j \in O \\ j \neq i}} \omega_{ij} \frac{d(o_i, o_j) - d^*(o_i, o_j)}{d^*(o_i, o_j)} \cdot (y_i(t) - y_j(t)).$$

Здесь настраиваемый параметр α влияет на скорость работы и сходимость алгоритма. Существуют и другие алгоритмы построения такого отображения, например, алгоритм случайного поиска [8], алгоритм SMACOF [9] и др.

Безусловно, методы снижения размерности, действующие по принципу сохранения попарных расстояний между объектами, составляют лишь подкласс нелинейных методов снижения размерности. Информацию о других методах можно почерпнуть, например, в [10].

Кроме того, существует класс методов, основанных на алгоритмах силовой укладки графов. В таких методах точки представляются системой частиц, взаимодействующих друг с другом на плоскости (или в трёхмерном пространстве) в соответствии с некоторой математической моделью физического процесса. Наиболее известными являются модели Идса [11], Фрухтермана–Рейнгольда [12] и Камада–Кавайи [13].

В настоящей работе для оценки качества работы методов снижения размерности многомерных данных взята мера, предложенная Сэммоном в 1969 году [7] и часто называемая ошибкой Сэммона:

$$\varepsilon = \frac{1}{\sum_{\substack{o_i, o_j \in O \\ i < j}} d(o_i, o_j)} \cdot \sum_{\substack{o_i, o_j \in O \\ i < j}} \frac{\left(d(o_i, o_j) - d^*(o_i, o_j) \right)^2}{d(o_i, o_j)}. \quad (3)$$

Это выражение представляет собой частный случай (1) при

$$\lambda = \frac{1}{\sum_{\substack{o_i, o_j \in O \\ i < j}} d(o_i, o_j)}, \quad \omega_{ij} = \frac{1}{d(o_i, o_j)}.$$

Выражение (2) в случае использования ошибки (3) может быть переписано в виде следующего рекуррентного соотношения для координат y_i в пространстве малой размерности:

$$y_i(t+1) = y_i(t) + m \cdot \sigma_i(t), \quad (4)$$

$$m = \frac{2 \cdot \alpha}{\sum_{\substack{o_i, o_j \in O \\ i < j}} d(o_i, o_j)},$$

$$\sigma_i(t) = \sum_{\substack{o_j \in O \\ j \neq i}} \frac{d(o_i, o_j) - d^*(o_i, o_j)}{d(o_i, o_j) \cdot d^*(o_i, o_j)} \cdot (y_i(t) - y_j(t)). \quad (5)$$

Рассматриваемые в настоящей работе методы описываются ниже применительно именно к ошибке (3), однако могут быть обобщены и на случай выбора ряда других ошибок представления многомерных данных.

Работа построена следующим образом. В первом разделе дан краткий обзор методов с пониженной вычислительной сложностью. Второй и третий разделы посвящены описанию используемых в настоящей работе подходов к декомпозиции пространства признаков. В четвертом разделе приводятся результаты численных экспериментов. В конце работы приводится заключение.

1. Методы с пониженной вычислительной сложностью

К сожалению, существенным недостатком методов нелинейного снижения размерности, работающих по итерационной схеме (похожей на представленную выше), является высокая вычислительная сложность. В связи с этим был предложен ряд методов, имеющих пониженную вычислительную сложность.

Одним из способов снижения вычислительной сложности является использование триангуляции [14], при котором сначала с использованием базового алгоритма ищется решение для некоторого количества объектов $M < N$. Затем производится последовательное добавление $(N - M)$ объектов с использованием триангуляции.

В методе снижения размерности с использованием линейного преобразования [15] ищется линейное преобразование, позволяющее, зная матрицу расстояний между объектами в многомерном пространстве, получать положение объектов на плоскости. Указанное линейное преобразование матрицы расстояний определяется по части объектов, к которым уже был применён базовый метод, а затем применяется к матрице расстояний между новыми и уже спроецированными объектами.

Решением проблемы высокой вычислительной сложности может быть алгоритм, использующий аппроксимации приращений координат точек $2\alpha\lambda \cdot \sigma_i(t)$ в (2) на каждой итерации. Для выполнения такой аппроксимации может быть использован подход, подобный предложенному в работе [16], где на каждой итерации для каждого корректируемого элемента формируется два множества. В первом из множеств содержатся элементы, наиболее близкие к корректируемому в многомерном пространстве. Во втором множестве содержатся элементы, отбираемые на каждой итерации случайным образом. Такой подход был использован применительно к модели Идса [11], однако он может быть применён и при минимизации ошибки Сэммона [20].

Позднее вычислительная сложность метода [16] была снижена по сравнению с исходным методом путём применения алгоритма [16] к подмножеству объектов и последующего добавления оставшихся объектов в полученное решение [17].

Эффективным способом снижения вычислительной сложности является иерархическая декомпозиция пространства. После выполнения такой декомпозиции объекты могут учитываться не по отдельности, а группами, что позволяет значительно ускорить итеративный процесс оптимизации. Очевидно, что такой подход позволяет аппроксимировать (2) лишь с некоторой точностью, которая зависит от того, насколько удачно подобрано распределение объектов по группам.

При этом такая декомпозиция может быть выполнена как в исходном, так и в целевом пространстве. Подход, основанный на иерархической декомпозиции исходного пространства, реализован в предложенных автором настоящей работы методах: в нелинейном методе снижения размерности данных с использованием опорных узлов [18], а также ранее в комбинированном методе снижения размерности [2].

Подход, основанный на декомпозиции пространства малой размерности, был реализован, например, в работах [12] (регулярная декомпозиция) и [19] (иерархическая декомпозиция с использованием квадродерева) при решении задачи укладки графов на плоскости для аппроксимации действующих на вершины сил. Следует отметить, что идеи декомпозиции пространства малой размерности широко применяются при моделировании в физике и, в частности, в астрофизике при расчёте взаимодействий большого количества источников. Однако о применении иерархической декомпозиции целевого пространства для решения задач снижения размерности данных, в частности, с использованием методов, действующих по принципу сохранения попарных расстояний, автору неизвестно.

В настоящей работе мы сравниваем оба подхода как с точки зрения качества получаемого отображения, так и с точки зрения времени функционирования.

Исследования ряда других методов снижения размерности, описанных в обзоре настоящей статьи, могут быть найдены в работе автора [20].

2. Декомпозиция в исходном пространстве

Подход к декомпозиции исходного пространства реализован в настоящей работе в виде нелинейного метода снижения размерности данных с использованием опорных узлов [18].

Под опорным узлом в работе понимается либо собственно объект исходного множества, либо некоторый создаваемый в процессе работы метода объект, характеризующий подмножество объектов исходного множества.

Собственно метод снижения размерности состоит из следующих этапов:

1. Построение иерархии кластеров на основе координат объектов в исходном многомерном пространстве.

2. Инициализация координат объектов в целевом пространстве малой размерности.
3. Построение списков опорных узлов для каждого из объектов для расчёта аппроксимации приращений.
4. Выполнение итеративной процедуры оптимизации для всего множества объектов.

Далее указанные шаги описываются более подробно.

Этап 1. Построение иерархии кластеров

Под иерархией кластеров будем понимать древовидную структуру данных, корнем которой является кластер верхнего уровня C^0 , а каждая вершина-кластер C_l^i на уровне $l > 0$ содержит либо подкластеры $\{C_{i_1}^{l+1}, C_{i_2}^{l+1}, \dots, C_{i_{|C_l^i|}}^{l+1}\}$, либо непосредственно объекты $\{o_{i_1}, o_{i_2}, \dots, o_{i_{|C_l^i|}}\}$ исходного множества.

Для построения иерархии кластеров могут быть применены как алгоритмы иерархической агломеративной кластеризации, так и неиерархические алгоритмы кластеризации, запускаемые в рекурсивном порядке (например, алгоритм k-внутригрупповых средних, нейросетевой алгоритм WTA и т. д.).

Этап 2. Инициализация координат объектов в целевом пространстве

Нелинейный метод снижения размерности Сэммона, принятый в настоящей работе в качестве базового, позволяет выполнять инициализацию координат объектов в целевом пространстве (строить начальное приближение) различными способами. В частности, в литературе встречаются способы инициализации случайными значениями, результатами работы метода РСА. При этом отмечено [2], что инициализация с использованием метода РСА позволяет обеспечить в среднем снижение размерности с более высоким качеством.

Этап 3. Построение списка опорных узлов

Под опорным узлом объекта o в настоящей работе будем понимать некоторый объект $o_i \neq o$ или группу объектов $\{o_i\}_{i=1..N}$, обладающих в исходном многомерном пространстве близкими характеристиками и рассматриваемых как единое целое. Под списком опорных узлов S объекта o будем понимать упорядоченное множество опорных узлов объекта o .

При построении списка опорных узлов для каждого из объектов примем во внимание тот факт, что при построении аппроксимации приращений координат объекты, расположенные вблизи рассматриваемого, следует учитывать по отдельности, а группы объектов, расположенные в удалении от рассматриваемого, возможно учитывать совместно.

Таким образом, если при построении аппроксимации какой-либо кластер иерархии находится в удалении от объекта, он может рассматриваться как единое целое. В противном случае необходимо произвести декомпозицию кластера на кластеры более низкого

уровня. В том же случае, когда кластер состоит непосредственно из объектов и не содержит подкластеров, декомпозиция кластера приводит к добавлению в список опорных узлов всех объектов кластера, что может приводить к значительному увеличению объёма списка опорных узлов для расчёта приращений. Чтобы избежать этого, предлагается вместо декомпозиции кластера на отдельные объекты выделять из кластера те объекты, которые близки к рассматриваемому, а остальные объекты кластера объединять во множество, называемое далее неполным кластером, которое учитывать при расчёте приращений как единое целое.

Условие декомпозиции, использованное в настоящей работе, несколько отличается от условия, использованного в [18], и ограничивается значением δ оценки угла α , под которым кластер C наблюдается из объекта o в исходном пространстве (см. рис. 1): $\alpha(C, o) \geq \delta$.

Формирование списка опорных узлов выполняется для каждого из объектов один раз до запуска процесса оптимизации. Ниже приводится более формальное описание предлагаемого алгоритма формирования опорных узлов для объекта.

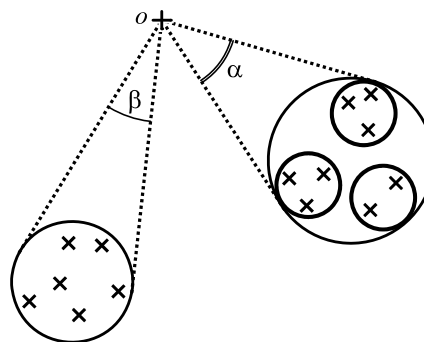


Рис. 1. Иллюстрация к алгоритму построения списка опорных узлов. Кластер, наблюдаемый из объекта o под углом α , будет разбит на три кластера, если $\alpha \geq \delta$. Кластер, наблюдаемый из o под углом β , будет рассматриваться как единое целое при $\beta < \delta$

Пусть o – некоторый объект, для которого требуется сформировать список S опорных узлов, C – произвольный кластер иерархии. Тогда рассмотренные выше принципы могут быть реализованы в виде следующего рекурсивного алгоритма.

1. Если кластер C удовлетворяет условию декомпозиции по отношению к рассматриваемому объекту ($\alpha(C, o) \geq \delta$) и в C содержатся подкластеры $C_i \in C, i = 1..N$, то применить этот рекурсивный алгоритм для каждого из подкластеров C_1, C_2, \dots, C_N .

2. Если кластер C удовлетворяет условию декомпозиции по отношению к рассматриваемому объекту ($\alpha(C, o) \geq \delta$) и в C содержатся объекты $o_i \in C, i = 1..N$, то а) выделить множество V объектов, не удовлетворяющих условию декомпозиции по отношению к рассматриваемому объекту, в неполный кластер: $V = \{o_i | d(o_i, o) < T\}, \alpha(V, o) < \delta$;

б) в том случае, если множество V непустое, добавить его в список опорных узлов: $S = S \cup V$;

в) добавить в список опорных узлов S все оставшиеся объекты кластера: $S = S \cup (C \setminus V)$.

3. Если кластер C не удовлетворяет условию декомпозиции по отношению к рассматриваемому объекту ($\alpha(C, o) < \delta$), то добавить его в список опорных узлов: $S = S \cup C$.

Рассмотренный алгоритм задан с точностью до угла декомпозиции δ и порога T , определяющего, находятся ли два объекта рядом.

Одним из возможных способов выбора порога является задание его равным радиусу кластера, содержащего рассматриваемый объект. То есть

$$T = R(C_k), o \in C_k,$$

где $R(C) = \max\{d(c, o_i)\}$, $o_i \in C$, c – центр кластера C в исходном пространстве. Этот способ выбора порога использовался в настоящей работе.

Функция $\alpha(C, o)$, определяющая значение угла, под которым кластер C наблюдается из объекта o в исходном пространстве, с целью снижения вычислительной сложности рассчитывалась как оценка угла, под которым описывающая кластер C гиперсфера наблюдается из объекта o в исходном пространстве:

$$\alpha(C, o) = 2 \cdot \arcsin(R(C) / d(c, o)).$$

Этап 4. Итеративная процедура оптимизации

Процедура оптимизации в настоящей работе строится на основе аппроксимации вычисления рекуррентного соотношения (4). Наибольший интерес представляет приближённое вычисление для произвольного объекта $o \in O$ суммы $\sigma_i(t)$ в (5), так как расчёт множителя t в выражении (4) может быть сделан заранее и не представляет затруднений. С использованием сформированного на предыдущем шаге работы метода списка опорных узлов S для объекта $o \in O$ приближённое значение суммы (5) на шаге t может быть записано в виде

$$\tilde{\sigma}_i = \sum_{s_j \in S} \zeta(o_i, s_j),$$

где функция $\zeta(o_i, s_j)$ принимает различный вид в зависимости от того, является ли s_j объектом, кластером или неполным кластером.

1. Если s_j является объектом o_j , то

$$\zeta(o_i, s_j) = \frac{d(o_i, o_j) - d^*(o_i, o_j)}{d(o_i, o_j) \cdot d^*(o_i, o_j)} \cdot (\mathbf{y}_i - \mathbf{y}_j).$$

2. Если s_j является кластером C_j , то

$$\zeta(o_i, s_j) = |C_j| \cdot \frac{d(o_i, c_j) - d^*(o_i, c_j)}{d(o_i, c_j) \cdot d^*(o_i, c_j)} \cdot (\mathbf{y}_i - \mathbf{y}_{c_j}),$$

где $d(o_i, c_j)$ – расстояние от объекта до центра кластера в исходном пространстве, $d^*(o_i, c_j)$ – расстояние от объекта до центра кластера в результирующем пространстве, \mathbf{y}_{c_j} – координаты центра кластера в результирующем пространстве.

3. Если s_j является неполным кластером V_j , то аналогично

$$\zeta(o_i, s_j) = |V_j| \cdot \frac{d(o_i, v_j) - d^*(o_i, v_j)}{d(o_i, v_j) \cdot d^*(o_i, v_j)} \cdot (\mathbf{y}_i - \mathbf{y}_{v_j}),$$

где $d(o_i, v_j)$ – расстояние от объекта до центра неполного кластера в исходном пространстве, $d^*(o_i, v_j)$ – расстояние от объекта до центра неполного кластера в результирующем пространстве, \mathbf{y}_{v_j} – координаты центра неполного кластера в результирующем пространстве.

Следует отметить, что центры кластеров и неполных кластеров в многомерном пространстве могут быть рассчитаны заранее. Центры кластеров \mathbf{y}_{c_j} в результирующем пространстве должны уточняться в процессе оптимизации, причём центры неполных кластеров \mathbf{y}_{v_j} могут быть рассчитаны через центры соответствующих полных кластеров \mathbf{y}_{c_j} и позиции исключённых объектов:

$$\mathbf{y}_{c_j} = \frac{1}{|C_j|} \cdot \sum_{o_k \in C_j} \mathbf{y}_k,$$

$$\mathbf{y}_{v_j} = \frac{1}{|V_j|} \cdot \left(|C_j| \cdot \mathbf{y}_{c_j} - \sum_{\substack{o_k \in C_j \\ o_k \notin V_j}} \mathbf{y}_k \right).$$

3. Декомпозиция в целевом пространстве

В отличие от рассмотренного выше подхода выполнение декомпозиции в целевом пространстве неизбежно ведёт к необходимости периодического обновления или перестройки той структуры, с использованием которой выполняется декомпозиция. Причина этого заключается в изменении координат объектов в целевом пространстве по мере выполнения оптимизационного процесса.

В настоящей работе для декомпозиции в целевом пространстве использованы KD-деревья. Построение дерева выполняется перед каждой итерацией оптимизационного процесса с использованием следующей рекурсивной процедуры:

1. Вычисление характеристик текущего узла дерева (средние координаты в исходном и целевом пространстве, границы узла в целевом пространстве).
2. В том случае, если в узле содержится лишь один объект, то выход.
3. Разбиение узла на два дочерних перпендикулярно наиболее вытянутой границе так, чтобы количество объектов в дочерних узлах было приблизительно равным.
4. Выполнение для вновь образованных дочерних узлов этой процедуры.

Построенное дерево используется для выполнения очередной итерации оптимизационного процесса. Пусть построено KD-дерево $Q = \{q_k\}$, $k = 1 \dots K$ с узлами q_1, q_2, \dots, q_K . Тогда вычисление для произвольного

объекта $o \in O$ суммы σ_i в (5) на шаге t может быть сделано приближенно:

$$\tilde{\sigma}_i = \sum_j \zeta(o_i, q_j)$$

путём обхода построенного дерева, начиная с корня дерева q_1 , следующим образом:

1. Если условие декомпозиции для текущего рассматриваемого узла q_j не выполняется и объекты, содержащиеся в терминальных вершинах текущего узла дерева, могут быть учтены группой, то очередное слагаемое рассчитывается следующим образом:

$$\zeta(o_i, q_j) = |q_j| \cdot \frac{d(o_i, q_j) - d^*(o_i, q_j)}{d(o_i, q_j) \cdot d^*(o_i, q_j)} \cdot (y_i - y_{q_j}),$$

где $|q_j|$ – количество объектов (терминальных узлов), содержащихся в соответствующем узле KD-дерева, $d(o_i, q_j)$ и $d^*(o_i, q_j)$ – расстояние между объектом o_i и центром узла q_j соответственно в многомерном пространстве и пространстве малой размерности, y_{q_j} – координаты центра узла в результирующем пространстве.

2. В противном случае критерий декомпозиции выполняется, и аналогичным образом обходятся оба поддерева рассматриваемого узла.

Условие декомпозиции, использованное при обходе дерева, аналогично рассмотренному выше и ограничивает (значением δ) оценку угла, под которым минимальный ограничивающий объект узла прямоугольник наблюдается из объекта o в целевом пространстве (рис. 2).

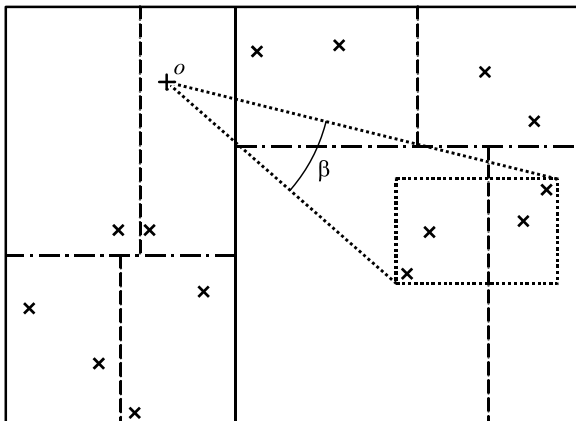


Рис. 2. Иллюстрация декомпозиции в целевом пространстве с использованием KD-дерева. Различные уровни декомпозиции показаны различными линиями: первый уровень показан сплошной линией, второй уровень показан штрихпунктирной линией, третий – штриховой линией. Группа из четырёх объектов (минимальный ограничивающий прямоугольник показан точками) будет разбита на две, если угол β , под которым минимальный ограничивающий прямоугольник наблюдается из объекта o , больше угла декомпозиции δ

Следует отметить, что указанный алгоритм выполняется на каждой итерации для каждого объекта, поэтому для повышения эффективности построенное дерево обходится не рекурсивно, а итеративно с использованием специальных указателей.

4. Экспериментальные исследования

В рамках работы все исследуемые методы были программно реализованы на языке C++ в интегрированной среде разработки Borland Turbo C++ 2006 Explorer. При проведении экспериментов использовался ПК на базе Intel Core i5-3470 CPU 3.2 GHz.

В качестве исходных данных при проведении экспериментов использовался набор данных Corel Image Features Data Set (<http://archive.ics.uci.edu/ml/databases/CorelFeatures/CorelFeatures.data.html>), содержащий признаки, рассчитанные по цифровым изображениям из коллекции Corel (<http://corel.digitalriver.com/>). В частности, в работе использовались следующие наборы признаков.

Набор 1 – цветовые моментные характеристики [21]. Рассчитывалось по 3 характеристики (среднее, СКО и коэффициент асимметрии) для каждой из цветовых компонент. Размерность пространства признаков – 9.

Набор 2 – цветовые гистограммы [22], построенные в цветовом пространстве HSV. Цветовое пространство разбивалось на 8 частей по H и на 4 части по S компонентам. Размерность пространства признаков – 32.

Набор 3 – текстурные признаки на основе матриц совместной встречаемости [23]. Рассчитывались 4 характеристики (второй угловой момент, контраст, обратный момент разности, энтропия) по 4 направлениям (горизонтальному, вертикальному и двум диагональным). Размерность пространства признаков – 16.

Из указанных наборов признаков вырезались фрагменты, содержащие признаковую информацию для необходимого количества изображений, после чего для наборов 1 и 3 к признаковой информации добавлялась единичная компонента и производилась нормировка.

Для оценки эффективности работы методов рассчитывалось значение ошибки представления многомерных данных (3), а также измерялось среднее время выполнения одной итерации процедуры оптимизации.

Работа методов останавливалась при замедлении скорости снижения ошибки (если относительное снижение ошибки за 10 итераций не превышало 0,05). Во всех случаях размерность целевого пространства была выбрана равной двум (решалась задача двумерного отображения). Инициализация выполнялась с использованием метода главных компонент.

Результаты экспериментов приведены на рис. 3–7.

На рис. 3 и 4 приводятся зависимости качественных и временных характеристик от угла декомпозиции δ , при котором осуществляется переход к дочерним узлам соответствующей иерархической структуры (дерева кластеров или KD-дерева).

Представленные на рис. 3–5 результаты приведены для набора признаков 1. Отметим, что эксперименты, проведённые на других описанных выше наборах данных, показывают схожие результаты.

Как видно из приведённых результатов, большие значения угла δ приводят к ожидаемому ухудшению качества отображения по причине более грубой аппроксимации, что выражается в больших значениях

ошибки ϵ представления многомерных данных (рис. 3). Время, затрачиваемое на выполнение одной итерации, уменьшается с увеличением угла δ (рис. 4) за счёт меньшего количества обрабатываемых узлов соответствующей иерархической структуры при построении аппроксимаций.

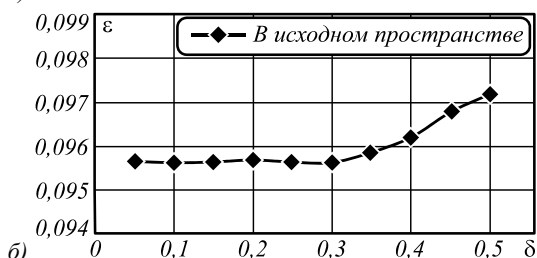
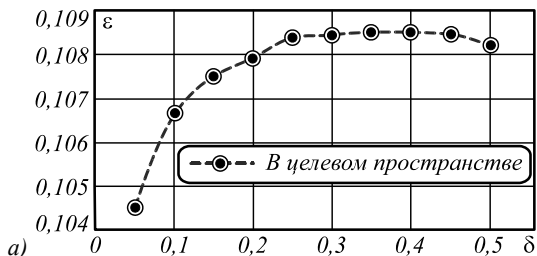


Рис. 3. Зависимость ошибки ϵ представления многомерных данных от угла декомпозиции δ (в долях от π радиан)

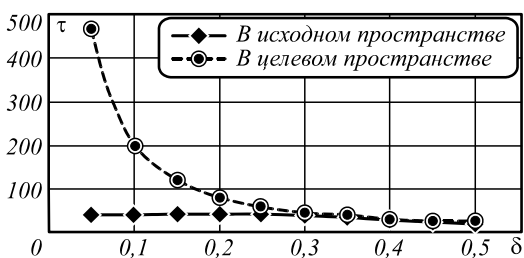


Рис. 4. Зависимость среднего времени выполнения одной итерации (в мс) от угла декомпозиции δ (в долях от π радиан)

Следует отметить, что подход с использованием декомпозиции в исходном пространстве обеспечивает меньшие значения ошибки представления многомерных данных, чем подход с использованием декомпозиции в целевом пространстве. При этом среднее время выполнения итерации остаётся существенно меньшим при использовании подхода с декомпозицией в исходном пространстве на участке $\delta \in [0,05\pi, 0,2\pi]$.

Среднее времени выполнения одной итерации в методе опорных узлов показано более детально на рис. 5.

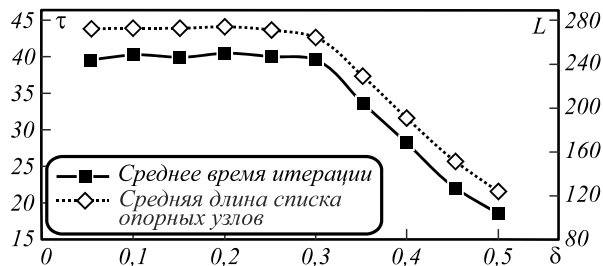


Рис. 5. Зависимость среднего времени выполнения одной итерации τ (в мс) и средней длины списка опорных узлов L от угла разбиения δ (в долях от π радиан) при декомпозиции в исходном пространстве

Там же представлена средняя длина сформированного списка опорных узлов L . Как видно из графика, время выполнения итерации согласуется с длиной сформированного списка опорных узлов, которая зависит от угла декомпозиции δ .

На рис. 6 и рис. 7 приводятся зависимости качественных и временных характеристик от количества объектов для всех трёх описанных наборов признаков. Полученные результаты подтверждают сказанное выше. Качество отображения, измеряемое ошибкой ϵ представления многомерных данных, формируемое с использованием декомпозиции в исходном пространстве признаков, во всех случаях оказалось выше качества отображения, формируемого с использованием декомпозиции в целевом пространстве. При этом подход с декомпозицией в исходном пространстве с использованием опорных узлов позволяет строить отображение быстрее, чем при использовании декомпозиции в целевом пространстве с использованием KD-дерева.

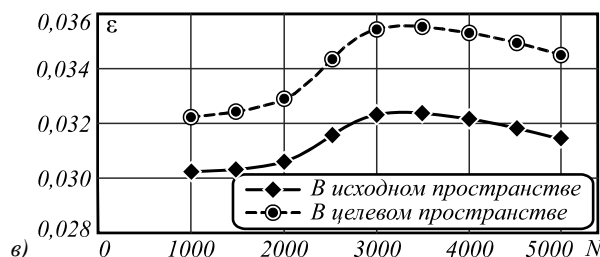
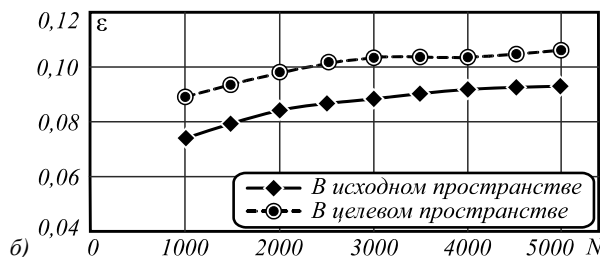
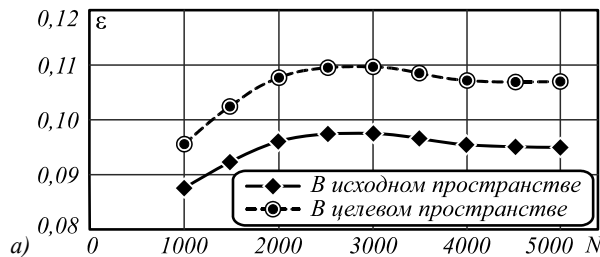


Рис. 6. Зависимость ошибки ϵ представления многомерных данных от количества объектов: для набора признаков 1 (а); для набора признаков 2 (б); для набора признаков 3 (в)

Заключение

В рамках работы проведено исследование двух подходов к декомпозиции пространства с точки зрения качества и скорости работы нелинейного метода снижения размерности данных. Исследование показало, что на представленных наборах данных декомпозиция исходного пространства более предпочтительна, чем декомпозиция целевого пространства и с точки зрения качественных, и с точки зрения временных показателей.

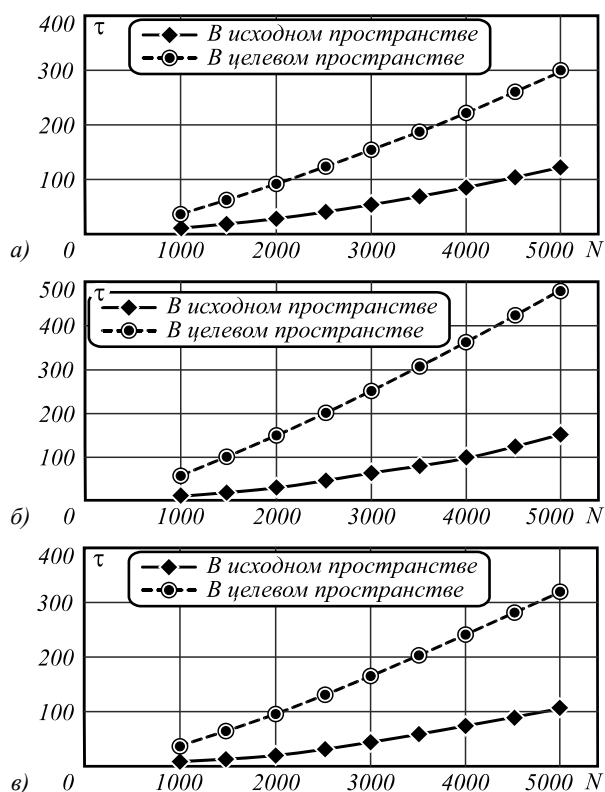


Рис. 7. Зависимость среднего времени выполнения одной итерации (в мс) от количества объектов: для набора признаков 1 (а); для набора признаков 2 (б); для набора признаков 3 (в)

Очевидно, что ни один из рассмотренных подходов не позволяет полностью контролировать ошибку аппроксимации приращений координат точек в (2), так как в первом случае при декомпозиции учитываются лишь характеристики узлов в исходном пространстве, а во втором случае – лишь в целевом. Выходом из этой ситуации и дальнейшим направлением исследований может стать разработка метода, совмещающего в себе оба подхода. Такой метод может быть разработан, например, на основе нелинейного метода снижения размерности с использованием опорных узлов с дополнительным контролем конфигурации объектов узла в целевом пространстве.

Благодарности

Работа выполнена за счёт Российского научного фонда (РНФ), грант № 14-31-00014 «Создание лаборатории прорывных технологий дистанционного зондирования Земли».

Литература (References)

1. **Hiroike, A.** Visualization of information spaces to retrieve and browse image data / A. Hiroike, Y. Musha, A. Sugimoto, Y. Mori // Proceedings of the Third International Conference on Visual Information and Information Systems (VISUAL'99). Lecture Notes in Computer Science. – Springer, 1999. – Vol. 1614. – P. 155-162.
2. **Мясников, Е.В.** Навигация по коллекциям цифровых изображений на основе методов автоматической классификации / Е.В. Мясников // Интернет-Математика 2007: сб. работ. – Екатеринбург: Изд-во Урал. ун-та,

2007. – С. 144-152. (Myasnikov, E.V. Digital image collection navigation based on automatic classification methods / E.V. Myasnikov // Internet-Mathematics 2007: Collected papers. – Ekaterinburg, Russia, 2007. – P. 144-152. – (In Russian).)
3. **Rodden, K.** Evaluating Similarity-Based Visualisations As Interfaces For Image Browsing / K. Rodden. – Technical Report. – University of Cambridge, Computer Laboratory, 2002. – 248 p.
4. **Rose, T.** ANVIL: a system for the retrieval of captioned images using NLP techniques / T. Rose, D. Elworthy, A. Kotcheff, A. Clare, P. Tsonis // The Challenge of Image Retrieval. Electronic Workshops in Computing, 2000.
5. **Rubner, Y.** Adaptive color-image embeddings for database navigation / Y. Rubner, C. Tomasi, L.J. Guibas // Proceedings of the IEEE Asian Conference on Computer Vision, 1998. – P. 104-111.
6. **Kruskal, J.B.** Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis / J.B. Kruskal // Psychometrika. – 1964. – Vol. 29. – P. 1-27.
7. **Sammon, J.W., Jr.** A nonlinear mapping for data structure analysis / J.W. Sammon, Jr. // IEEE Transactions on Computers. – 1969. – Vol. C-18, Issue 5. – P. 401-409.
8. **Calvert, T.W.** Randomly Generated Nonlinear Transformations for Pattern Recognition / T.W. Calvert, T.Y. Young // IEEE Transactions on Systems Science and Cybernetics. – 1969. – Vol. 5. – P. 266-273.
9. **De Leeuw, J.** Applications of convex analysis to multidimensional scaling / J. De Leeuw, J.R. Barra, F. Brodeau, G. Romie [et al.] // Recent developments in statistics. – 1977. – P. 133-145.
10. **Lee, J.A.** Nonlinear Dimensionality Reduction / J.A. Lee, M. Verleysen. – Springer, 2007.
11. **Eades, P.** A Heuristic for Graph Drawing / P. Eades // Congressus Numerantium. – 1984. – Vol. 42. – P. 149-160.
12. **Fruchterman, T.** Graph Drawing by Force-directed Placement / T. Fruchterman, E. Reingold // Software – Practice and Experience. – 1991. – Vol. 21, Issue 11. – P. 1129-1164.
13. **Kamada, T.** An Algorithm for Drawing General Undirected Graphs / T. Kamada, S. Kawai // Information Processing Letters. – 1989. – Vol. 31. – P. 7-15.
14. **Lee, R.C.T.** A Triangulation Method for the Sequential Mapping of Points from N-Space to Two-Space / R.C.T. Lee, J.R. Slagle, H. Blum // IEEE Transactions on Computers. – 1977. – Vol. 26, Issue 3. – P. 288-292.
15. **Pełkalska, E.** A new method of generalizing Sammon mapping with application to algorithm speed-up / E. Pełkalska, D. de Ridder, R.P.W. Duin, M.A. Kraaijeveld // Proc. ASCI'99, 5th Annual Conf. of the Advanced School for Computing and Imaging. – Heijden, The Netherlands: 1999, June 15-17. – P. 221-228.
16. **Chalmers, M.** A Linear Iteration Time Layout Algorithm for Visualizing High-Dimensional Data / M. Chalmers // Proc. IEEE Visualization '96. – San Francisco, 1996. – P. 127-132.
17. **Morrison, A.** Fast Multidimensional Scaling Through Sampling, Springs and Interpolation / A. Morrison, G. Ross, M. Chalmers // Information Visualization. – 2003. – Vol. 2. – P. 68-77.
18. **Myasnikov, E.V.** A Nonlinear Method for Dimensionality Reduction of Data Using Reference Nodes / E.V. Myasnikov // Pattern Recognition and Image Analysis. – 2012. – Vol. 22, Issue 2. – P. 337-345.
19. **Quigley, A.** FADE: Graph Drawing, Clustering, and Visual Abstraction / A. Quigley, P. Eades // Proceedings of the 8-th International Symposium on Graph Drawing. – 2001. – P. 197-210.

20. **Мясников, Е.В.** Анализ методов снижения размерности в задаче представления коллекций цифровых изображений / Е.В. Мясников // Компьютерная оптика. – 2008. – Т. 32, № 3. – С. 296-301. (**Myasnikov, E.V.** The study of dimensionality reduction methods in the task of browsing of digital image collections / E.V. Myasnikov // Computer Optics. – 2008. – Vol. 32(3). – P. 296-301. – (In Russian).)
21. **Stricker, M.** Similarity of color images / M. Stricker, M. Orengo // In Proc. SPIE Conf. on Vis. Commun. and Image Proc. - 1995.
22. **Swain, M.** Color indexing / M. Swain, D. Ballard // International Journal of Computer Vision. – 1991. – Vol. 7(1).
23. **Haralick, R.M.** Texture features for image classification / R.M. Haralick, K. Shanmugam, I. Dinstein // IEEE Transactions on Systems, Man and Cybernetics. – 1973. – SMC-3(6).

THE CHOICE OF A METHOD FOR FEATURE SPACE DECOMPOSITION FOR NON-LINEAR DIMENSIONALITY REDUCTION

E.V. Myasnikov

*Samara State Aerospace University,
Image Processing Systems Institute, Russian Academy of Science*

Abstract

This paper considers two approaches to the hierarchical decomposition of the feature space to improve the efficiency of the non-linear dimensionality reduction method. The first approach suggested by the author of the paper is based on the decomposition of the original feature space using hierarchical clustering. The second original approach is based on a hierarchical decomposition of the target space by using a KD-Tree. The approaches analyzed are evaluated in terms of the efficiency of the non-linear dimensionality reduction method.

Key words: dimensionality reduction, decomposition of the feature space, hierarchical clustering, KD-trees.

Сведения об авторе



Мясников Евгений Валерьевич, 1981 года рождения. В 2004 году окончил с отличием Самарский государственный аэрокосмический университет (СГАУ) по специальности «Автоматизированные системы обработки информации и управления». Кандидат технических наук (2007), доцент кафедры геоинформатики и информационной безопасности СГАУ, старший научный сотрудник научно-исследовательской лаборатории прорывных технологий дистанционного зондирования Земли СГАУ, научный сотрудник лаборатории математических методов обработки изображений Института систем обработки изображений РАН (ИСОИ РАН). Результаты научных исследований отражены более чем в 50 научных работах, соавтор монографии. Область научных интересов: распознавание образов, обработка изображений, геоинформатика, проектирование и разработка программного обеспечения.

E-mail: mevg@geosamara.ru.

Evgeny Valerievich Myasnikov (b. 1981) graduated with honors from Samara State Aerospace University (SSAU) in 2004, majoring in "Automated Systems for Information Processing and Control". PhD in Technical Sciences (2007). Currently he works as Associate Professor at Geoinformatics and Information Security department, senior researcher at the Breakthrough Technologies for Earth's Remote Sensing laboratory in S.P. Korolyov Samara State Aerospace University (National Research University), SSAU and as the Researcher at Mathematical Methods of Image Processing laboratory, Image Processing Systems Institute of RAS (IPSI RAS). The research results are reflected in more than 50 scientific papers, co-author of the monograph. Research interests are pattern recognition, image processing, geoinformatics, software development.

Поступила в редакцию 5 ноября 2014 г.