А.А. Россиев

ИТЕРАЦИОННОЕ МОДЕЛИРОВАНИЕ НЕПОЛНЫХ ДАННЫХ С ПОМОЩЬЮ МНОГООБРАЗИЙ МАЛОЙ РАЗМЕРНОСТИ

Красноярск – 2000

Итерационное моделирование неполных данных с помощью многообразий малой размерности. Россиев А.А.

Описывается метод моделирования неполных данных с помощью последовательности кривых, обобщающий метод главных компонент. Обсуждаются три версии метода:

а) линейный – с моделированием данных последовательностью линейных многообразий малой размерности;

б) квазилинейный – с построением "главных кривых" (или "главных поверхностей"), однозначно проектируемых на линейные главные компоненты;

в) существенно нелинейный – основанный на построении "главных кривых" ("главных струн и балок") с использованием вариационного принципа; итерационная реализация этого метода близка методу самоорганизующихся карт Кохонена.

Все полученные зависимости, кроме линейной, нуждаются в экстраполяции, которая производится с помощью формул Карлемана. Метод трактуется как построение нейросетевого конвейера, решающего следующие задачи:

а) заполнение пробелов в данных;

б) ремонт данных – корректировка значений исходных данных так, чтобы наилучшим образом работали построенные модели;

в) построение вычислителя, заполняющего пробелы в поступающей на вход строке данных (в предположении, что данные о новых объектах связаны теми же самыми отношениями, что и в исходной таблице).

Разработанная технология предназначена для решения широкого спектра задач, связанных с обработкой неполных данных. Она реализована в программных продуктах FAMaster и ModelAnalyzer.

СОДЕРЖАНИЕ

Введение

Практически любое научное исследование связано со сбором и обработкой данных, в большинстве случаев структурируемых в виде таблиц. Достаточно часто таблицы содержат пробелы (пустые ячейки), возникающие вследствие невозможности наблюдения, отсутствия необходимых инструментов и т.п. Кроме пробелов, по тем же причинам, а также из-за неточных входных данных, из-за погрешностей, вносимых на отдельных этапах вычислений, погрешностей самих методов вычислений, данные могут содержать искаженные значения (например, случайный шум).

Большинство известных методов анализа данных не могут обрабатывать такую информацию. Иногда на место пропущенных значений можно попытаться поставить среднее значение данного показателя по всей таблице, либо исключить из таблицы показатель (столбец) или объект (строку) с пробелом. Но и то, и другое приводит к потере информации или к ее значительному искажению, а иногда и вообще делает решение задачи невозможным.

Несмотря на перечисленные трудности, данные нуждаются в обработке. Поэтому явно или неявно возникает необходимость в процедуре заполнения пропусков и ремонта (здесь и далее под "ремонтом" подразумевается поиск и исправление неправдоподобных значений) данных – процедуре предобработки.

Следует особо подчеркнуть, что, рассматривая данные проблемы, невозможно говорить ни об истинных значениях данных, ни даже о статистической доказательности, но только о правдоподобии. Особую трудность (и в то же время особый интерес) описанные задачи имеют в тех случаях, когда плотность пробелов высока, расположены они нерегулярно, а данных немного, например, число строк таблицы примерно равно числу столбцов.

Рассматривая множество данных как набор точек (линейных многообразий при наличии пропусков) в пространстве, можно строить их линейные и нелинейные приближения – модели, с помощью которых можно восстанавливать имеющиеся пробелы и ремонтировать данные.

Кроме этого, такие модели позволяют проводить дополнительные эксперименты, которые по тем или иным причинам невозможны с реальным объектом исследования. В частности, представляет большой интерес задача оценки правдоподобности полученных данных.

В настоящем исследовании описывается метод моделирования неполных данных с помощью последовательности кривых, обобщающий метод главных компонент. Он также может быть интерпретирован как процедура сингулярного разложения матриц с пробелами. Обсуждаются три версии метода, которые имеют ясную физическую интерпретацию:

а) линейный – с моделированием данных последовательностью линейных многообразий малой размерности;

б) квазилинейный – с построением "главных кривых" (или "главных поверхностей"), однозначно проектируемых на линейные главные компоненты;

в) существенно нелинейный – основанный на построении "главных кривых" ("главных струн и балок") с использованием вариационного принципа; итерационная реализация этого метода близка методу самоорганизующихся карт Кохонена.

Все полученные зависимости, кроме линейной, нуждаются в экстраполяции, которая производится с помощью формул Карлемана. Метод трактуется как построение нейросетевого конвейера, решающего следующие задачи:

а) заполнение пробелов в данных;

б) ремонт данных – корректировка значений исходных данных так, чтобы наилучшим образом работали построенные модели;

в) построение вычислителя, заполняющего пробелы в поступающей на вход строке данных (в предположении, что данные о новых объектах связаны теми же самыми отношениями, что и в исходной таблице).

Разработанная технология итерационного моделирования неполных данных с помощью многообразий малой размерности предназначена для решения широкого спектра задач, связанных с обработкой неполных данных. Она реализована в программных продуктах FAMaster и ModelAnalyzer.

В **первой главе** работы дан обзор литературы по существующим методам обработки данных с пробелами, а также по теории главных кривых. В частности, рассматриваются методы регрессионного анализа, методы максимального правдоподобия, эмпирические методы и нейросетевые методы заполнения пропусков в таблицах.

Далее приводятся теоретические и методологические основы моделирования данных с пробелами многообразиями малой размерности. При этом вначале рассматриваются эмпирические таблицы и задачи эмпирического предсказания, требования к методам обработки таблиц эмпирических данных, а затем описываются теоретические основы самого метода моделирования данных с пробелами многообразиями малой размерности.

Пусть задана прямоугольная таблица $A=(a_{ij})$ типа "объект-признак", где строки соответствуют объектам, а столбцы – признакам, и пусть часть информации в таблице отсутствует (т.е. некоторые $a_{ij}=@$, где $@$ – значок, означающий отсутствие данных).

Тогда каждая строка таблицы – это вектор данных x с k пробелами, который представляется как k-мерное линейное многообразие L_x, параллельное k координатным осям, которые соответствуют пропущенным данным. При наличии априорных ограничений на пропущенные значения место L_x занимает прямоугольный параллелепипед $P_x \subset L_x$. Ищется многообразие M заданной малой размерности (чаще всего – кривая), наилучшим образом приближающее данные и удовлетворяющее некоторым требованиям регулярности. Из данных вычитаются ближайшие к ним точки многообразия M – получается остаток – и процесс повторяется, пока остатки не приблизятся в достаточной степени к

нулю. Дальнейшая конкретизация метода состоит в указании того, как строится многообразие *M*.

Приводится связь предложенного алгоритма с методами главных кривых и сингулярного разложения таблиц данных, рассматриваются некоторые условия применимости и, наконец, приводится постановка задачи.

Требуется построить модели, которые позволяли бы решать следующие три задачи, связанные с восстановлением пропущенных данных:

1) правдоподобно заполнить имеющиеся пробелы в данных;
2) отремонтировать данные, т.е. исправить их значения таким образом, чтобы наилучшим образом работали построенные модели;
3) построить по имеющейся таблице вычислитель, который бы заполнял пробелы в данных и ремонтировал бы их по мере поступления (в предположении, что данные в поступающей на вход строке связаны теми же соотношениями, что и в исходной таблице).

Вторая глава "Линейные и квазилинейные многообразия малой размерности" посвящена методу моделирования неполных данных с помощью линейных и квазилинейных многообразий малой размерности.

Первоначально дается краткое описание классического метода главных компонент.

Далее рассматривается геометрическая интерпретация метода итерационного моделирования неполных данных с помощью линейных и квазилинейных многообразий малой размерности.

Возьмем за основу прямую $f(x)=xy+b$, которая задается направляющим вектором y и проходит через точку, определяемую вектором b, и расположим ее так, чтобы она наилучшим (в некотором точном смысле) образом приближала исходные данные. Исходные данные однозначно проецируются на полученную прямую, при этом проекция определяется как ближайшая к данному точка прямой (как для полных, так и для неполных данных). Если из данных вычесть их проекции, получим уклонения от первой модели (в предположении, что прямая моделирует данные). Это множество уклонений также можно смоделировать прямой данного вида, спроецировать на нее и вычесть проекции – и так далее. Над полученной прямой можно построить кривую – квазилинейную модель, где в качестве аргумента гладкой вектор-функции выступает значение проекции на исходную прямую. В этом и заключается итерационный процесс моделирования неполных данных линейными и квазилинейными многообразиями малой размерности.

Пусть задана прямоугольная таблица $A=(a_{ij})$, ячейки которой заполнены действительными числами или значком @, означающим отсутствие данных.

Требуется представить исходную матрицу A в виде суммы матриц P_q: $A=\sum_q P_q$, где каждая P_q имеет вид $x_i y_j+b_j$.

Если потребовать, чтобы вектор b всегда был нулевым (то есть, чтобы моделирующая прямая проходила через начало координат), то получим сингулярное разложение матриц с пропусками, иначе имеем метод главных компонент для неполных данных.

Основная процедура – поиск наилучшего приближения таблицы A, содержащей пропуски, матрицей вида $x_i y_j + b_j$ методом наименьших квадратов:

$$\Phi = \sum_{\substack{i,j \\ a_{ij} \neq @}} (a_{ij} - x_i y_j - b_j)^2 \to \min . \qquad (1)$$

Решение дается последовательными итерациями по явным формулам – при фиксированных значениях одних переменных находятся значения других. Этот итерационный процесс является сходящимся – так как на каждом этапе значение функционала уменьшается.

Таким образом, для данной матрицы A находим наилучшее приближение матрицей P_1 вида $x_i y_j + b_j$. Далее, из матрицы A вычитаем полученную матрицу P_1, и для полученной матрицы уклонений $A - P_1$ вновь ищем наилучшее приближение P_2 этого же вида и т.д. Контроль ведется, например, по остаточной дисперсии столбцов.

В результате исходная матрица данных A представляется в виде суммы матриц P_q, т.е. $A = P_1 + P_2 + \ldots + P_q$.

С использованием Q полученных факторов (матриц заданного вида) можно решать задачи заполнения пропусков в таблице и ремонта искаженных значений:

Q-факторное заполнение пропусков: пропущенные значения в исходной матрице A определяются из суммы Q полученных матриц вида $x_i y_j + b_j$;

Q-факторный "ремонт" таблицы: значения в исходной матрице заменяются на сумму Q полученных матриц вида $x_i y_j + b_j$.

В функционале (1) используется матрица данных, но если переписать его для вектора данных, то получается процедура заполнения пропусков и ремонта в отдельно взятом векторе.

Совершенно аналогично строятся линейные многообразия большей размерности – два и три. Однако им свойственны некоторые особенности – например, образующие их векторы не обязательно ортогональны и может потребоваться процедура их ортогонализации, а также может возникнуть необходимость в процедуре вращения – для более наглядного представления.

Метод построения квазилинейных моделей основывается на методе построения соответствующих линейных моделей. Его предлагается проводить в три этапа.

1. *Построение линейной модели*.

2. *Интерполяция (сглаживание)*: строится вектор-функция $f(t)$, минимизирующая функционал:

$$\Phi = \sum_{\substack{i,j \\ a_{ij} \neq @}} (a_{ij} - f_j(\sum_k a_{ik} y_k))^2 + \alpha \int_{-\infty}^{+\infty} (f''(t))^2 dt ,$$

где $\alpha > 0$ – параметр сглаживания.

3. *Экстраполяция.* Решение этой задачи возможно провести несколькими способами, но наиболее интересным представляется использование такой экстраполяции, где сглаженная вектор-функция $f(t)$ экстраполируется с некоторого конечного множества $\{t_k\}$ (которое не обязательно связано с

проекциями на прямую $z_j=ty_j+b_j$ исходных строк данных) на всю вещественную прямую с использованием формул Карлемана (с помощью формул Карлемана экстраполируется отклонение кривой $f(t)$ от прямой $ty+b$):

$$f(t) \approx ty + b + \sum_{k=1}^{m} (f(t_k) - t_k y - b) \frac{2(e^{\lambda t} - e^{\lambda t_k})}{\lambda (e^{\lambda t} + e^{\lambda t_k})(t - t_k)} \prod_{\substack{j=1 \\ j \neq k}}^{m} \frac{(e^{\lambda t_k} + e^{\lambda t_j})(e^{\lambda t} - e^{\lambda t_j})}{(e^{\lambda t_k} - e^{\lambda t_j})(e^{\lambda t} + e^{\lambda t_j})}, \quad (2)$$

где λ – параметр метода, характеризующий, насколько широка полоса на плоскости комплексных чисел, в которой гарантированно голоморфна экстраполируемая функция (эта ширина равна π/λ).

Приводятся различные решения задачи интерполяции: с помощью полиномов небольшой степени и кубических сплайнов.

Далее подробно рассматриваются проблемы экстраполяции, оптимального аналитического продолжения. В качестве решения этих проблем предлагается использовать формулу Карлемана, которую можно применять не только для экстраполяции, но и для интерполяции.

Процедура использования квазилинейных моделей несколько отличается от аналогичной процедуры в линейном случае.

Точка на построенной кривой $f(t)$, соответствующая полному ("комплектному") вектору данных a, строится как $f((a,y))$. В этом и заключается квазилинейность метода: сначала ищется проекция вектора данных на прямую $\Pr(a)=ty+b$, $t=(a,y)$, а затем строится точка на кривой $f(t)$. Также и для неполных векторов данных – сначала ищется на прямой ближайшая к ним точка $t(a)$ (проекция неполного вектора a), а затем – соответствующая точка на кривой $f(t)$ при $t=t(a)$. Следующим шагом из данных вычитаются их проекции на построенную кривую, то есть матрица данных заменяется на матрицу уклонений от модели. Далее процедура продолжается до тех пор, пока не выполнится некоторое условие остановки (например, близость уклонений к началу координат).

В результате исходная таблица предстает в виде Q-факторной модели:

$$a_{ij} \cong \sum_{q} f_j^q(t_i^q). \quad (3)$$

Описанные методы построения многообразий малой размерности, наилучшим образом приближающих данные, имеют ясные механические представления, лежащие в их основе.

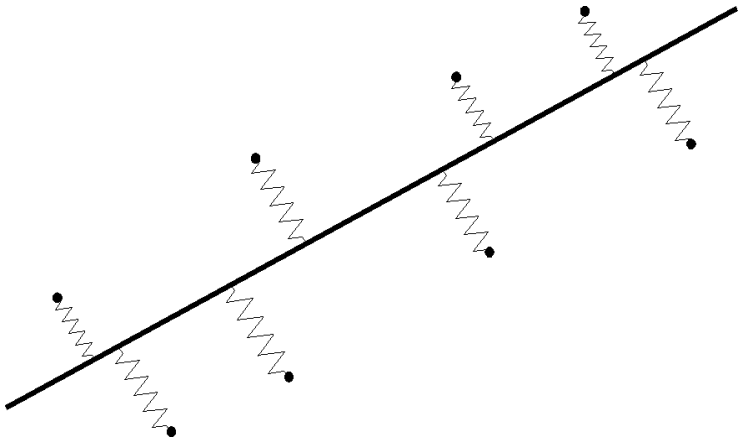

Рис. 1.

В линейном случае, в пространство данных помещена прямая жесткая балка (рис. 1), свободно соединенная пружинками с данными. Минимум функционала (1) соответствует минимуму суммарной энергии растяжения пружинок.

При этом решение задачи минимизации функционала (1) полностью аналогично поиску равновесия балки. Зафиксируем ее начальное положение и найдем такое положение пружинок, которое отвечает минимуму энергии растяжения. После этого зафиксируем положение концов пружинок на балке, освободим балку и дадим ей прийти в механическое равновесие. Далее зафиксируем балку в новом положении и вновь освободим концы пружинок.

Эта итерационная процедура сходится – так как на каждом этапе суммарная энергия растяжения пружинок уменьшается.

Если же предположить, что балка может упруго отклоняться от прямого вида, то получим следующую картину (рис. 2).

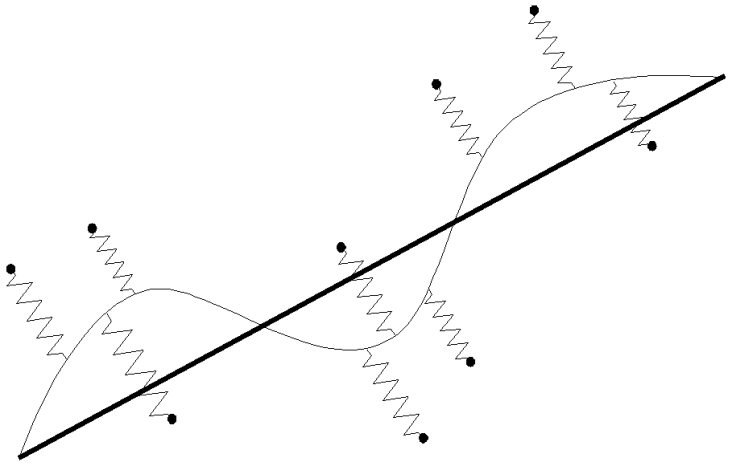

Рис. 2.

Места крепления пружинок в балке определяются проекциями на прямую балку (т.к. модель квазилинейная).

При этом возникает задача определения поведения концов балки за границами области данных – описанная выше задача экстраполяции. Формула Карлемана в этом случае имеет аналогию с закреплением бесконечных концов балки на прямой.

Если учесть, что построенная квазилинейная модель является суммой

квазилинейных вектор-функций (3), то можно сделать вывод, что указанный алгоритм допускает нейросетевую интерпретацию.

Действительно, построим такую нейронную сеть, где с каждой кривой из (2) был бы связан один сумматор (в качестве его весов будем использовать координаты вектора y^q), набор из n свободных слагаемых ("порогов") – координат вектора b^q, и n нелинейных преобразователей, каждый из которых вычисляет одну координату точки на кривой по формуле (2).

Действие такого "нейрона" на вектор a входных сигналов специфично: сначала вычисляется проекция данного $t(a)$ на прямую (работает сумматор), далее нелинейные элементы вычисляют $f^q(t(a))$, а затем разность $f_j^q(t(a))$ ($a_j \neq @$) передается следующему нейрону.

При прохождении a по этому конвейеру одновременно накапливается сумма величин $f_j^q(t(a))$ ($a_j = @$). Они и образуют вектор выходных сигналов – предлагаемые значения пропущенных данных. В случае необходимости провести ремонт данных накапливается сумма величин $f_j^q(t(a))$ для каждой координаты j.

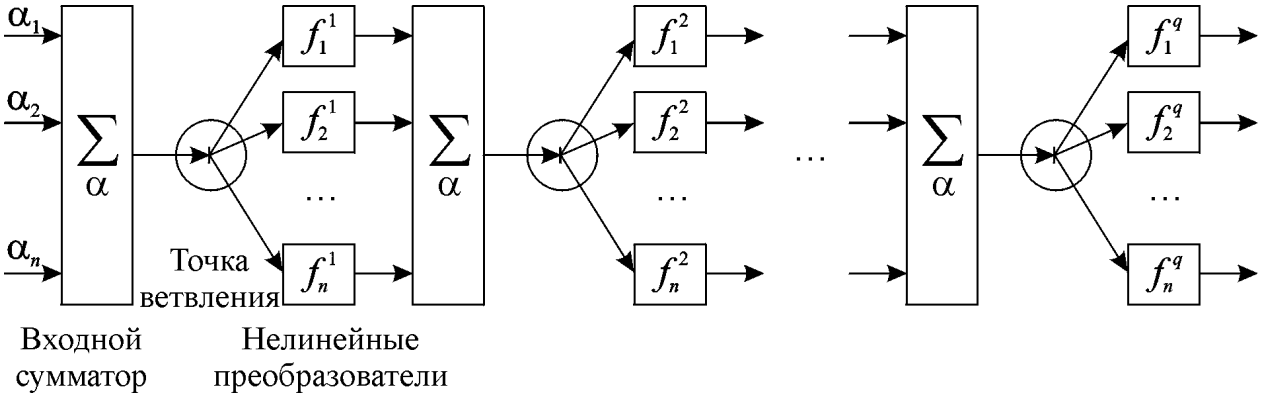

Рис. 3.

В результате получена следующая нейронная сеть (рис. 3). Структура каждого нейрона в ней нестандартна – он имеет один входной сумматор и n нелинейных преобразователей (по размерности вектора данных).

В **третьей главе** *"Самоорганизующиеся многообразия малой размерности"* описываются принципы построения и использования самоорганизующихся многообразий малой размерности (Self-Organizing Curve – SOC).

Первоначально вводится понятие главной кривой, как гладкой линии, проходящей через центры масс точек, проецируемых на нее, и рассматривается идея итерационного алгоритма построения главной кривой. Приводится алгоритм Hastie-Stuetzle, который работает итерационно, начиная с главной компоненты множества данных.

Далее в главе рассматривается идея построения самоорганизующихся кривых на основе самоорганизующихся карт Кохонена и описывается алгоритм расчета одномерных нелинейных многообразий.

Пусть SOC определяется набором точек (ядер) $Y=\{y_j\}$ ($j=1..m$), последовательно расположенных на кривой (в первом приближении пусть SOC

– просто ломаная Y) и требуется отобразить на ней набор точек данных $X=\{x_i\}$. Введем преобразование $П$, которое каждому вектору $x \in X$ сопоставляет ближайшую к нему точку из Y:

$$x \xrightarrow{П} y_j, \left\| y_j - x \right\|^2 \to \min,$$

каждому ядру y_j сопоставляется его таксон.

$$K_j = \left\{ x \in X \middle| x \xrightarrow{П} y_j \right\}.$$

Метод построения SOC напоминает метод динамических ядер, за исключением добавления дополнительных ограничений на связность и нелинейность. Минимизируемая величина строится из следующих слагаемых:

1) мера приближения данных:

$$D_1 = \sum_j \sum_{x \in K_j} \left\| x - y_j \right\|^2,$$

2) мера связности (близкие точки на кривой должны переходить в близкие в пространстве данных):

$$D_2 = \sum_j \left\| y_j - y_{j+1} \right\|^2,$$

3) мера нелинейности (равномерности):

$$D_3 = \sum_j \left\| 2y_j - y_{j-1} - y_{j+1} \right\|^2.$$

Таким образом, для построения SOC требуется минимизировать функционал:

$$D = \frac{D_1}{|X|} + \lambda \frac{D_2}{m} + \mu \frac{D_3}{m} \to \min,$$

где λ, μ – параметры связности и нелинейности – "модули упругости" (деление на число точек $|X|$ и число ядер m означает нормировку "на одно слагаемое" и позволяет для выборок разной мощности использовать одинаковые способы изменения λ и μ.).

Аналогично процедуре построения самоорганизующихся кривых строятся соответствующие самоорганизующиеся карты. Приводятся их две основные вариации – с квадратными и гексагональными ячейками сетки.

В отличие от линейных и квазилинейных моделей, при построении самоорганизующихся кривых возникает ряд проблем, основная из которых – попадание в область локального минимума функционала D. Предлагаются возможные пути ее решения – метод "отжига" и многосеточные методы. В первом случае система первоначально ставится в очень жесткие рамки (за счет увеличения коэффициентов в функционале D), которые затем постепенно ослабляются, а во втором в процессе построения кривой изменяется число составляющих ее узлов.

Еще одна проблема заключается в кусочной гладкости полученных многообразий. Поэтому для сглаживания предлагается использовать кусочно-линейные проекторы на ломаную с последующим сглаживанием с

использованием формулы Карлемана, которая при этом еще решает и задачу экстраполяции.

Аналогично линейным и квазилинейным многообразиям, принцип построения самоорганизующихся многообразий малой размерности также имеет ясную механическую интерпретацию.

Искомая упругая балка (кривая – SOC) представляется в виде ломаной, узлы которой свободно соединены с данными (рис. 4).

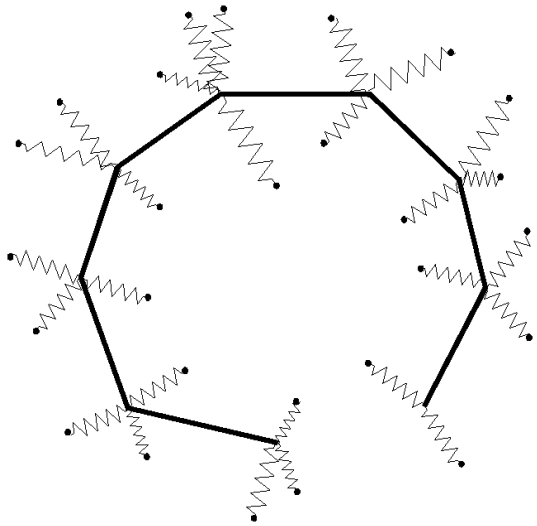

Рис. 4.

Аналогично жесткому случаю, система через несколько итераций придет в равновесие. Причем их число будет конечно, т.к. на каждом шаге суммарная энергия растяжения пружинок уменьшается, а число возможных состояний (способов крепления узлов ломаной пружинками к данным) конечно.

Введенные модули упругости представляют собой соответственно степень притяжения узлов ломаной друг к другу и степень сопротивления изгибу в узлах.

Четвертая глава *"Экспериментальные результаты"* посвящена экспериментальной проверке метода итерационного моделирования неполных данных с помощью многообразий малой размерности.

Совместно с Красноярской государственной медицинской академией был проведен анализ административных территорий Красноярского края по показателям здоровья и здравоохранения – типизация и выделение относительно похожих групп в системе из 49 регионов (административных территорий) Красноярского края с помощью факторного (линейного и нелинейного) и кластерного анализа.

Процесс моделирования данных с пробелами проиллюстрирован также на основе таблицы результатов выборов президентов США, которая содержит 31 предвыборную ситуацию (с 1860 по 1980 гг.), где для каждых выборов в таблице содержатся данные по 12-ти бинарным признакам.

Метод также использовался для верификации связей между двумя динамическими системами – в задачах гелиофизики при анализе различных временных рядов. Из полного ряда годовых значений чисел Вольфа было удалено 50% точек. Для восстановления пропусков использовалась SOC.

Строки исходной таблицы представляли собой m-мерные запаздывающие векторы Такенса, с m=6, вида: $a_{kj} = x_k, x_{k+1}, ..., x_{k+5}$, так что удаление одного отсчета в таблице приводит к удалению соответствующей диагонали. С использованием метода удалось удовлетворительно восстановить даже вершины циклов.

Далее описывается использование метода моделирования неполных данных с помощью многообразий малой размерности при обнаружении факторов, влияющих на течение и прогноз заболевания у больных со сложными нарушениями ритма и проводимости сердца. При этом были замечены различия между разными группами больных и были выявлены факторы, способствующие возникновению ряда осложнений.

В заключении суммированы основные результаты работы и сделаны выводы.

1. Для решения задачи заполнения пропусков и ремонта искаженных данных разработан метод итерационного моделирования неполных данных с помощью многообразий малой размерности. Приведены три вариации метода: начиная с простейших линейных многообразий, продолжая построенными на их основе квазилинейными многообразиями и заканчивая методом главных кривых для данных с пробелами.

2. Для параллельной реализации метода итерационного моделирования данных с пробелами разработан способ построения нейронного конвейера, решающего задачи заполнения пробелов и ремонта данных.

3. Разработаны программные продукты FAMaster и ModelAnalyzer, реализующие предложенные технологии.

4. Численные эксперименты показали высокую эффективность итерационного моделирования неполных данных с помощью многообразий малой размерности. Метод хорошо зарекомендовал себя при решении трудных задач с большим числом пропущенных данных, а в более простых (стандартных) случаях приводит к тем же результатам, что и классические методы статистического анализа.

Глава I. Проблемы обработки и итерационного моделирования неполных данных с помощью многообразий малой размерности

Введение

В первом параграфе дан обзор литературы по существующим методам обработки данных с пропусками. В частности, рассматриваются методы регрессионного анализа, методы максимального правдоподобия, эмпирические методы. Также в рассмотрение включаются и нейросетевые методы заполнения пропусков в таблицах.

Во втором параграфе приводится обзор существующей теории главных кривых.

В третьем параграфе приводятся теоретические и методологические основы моделирования неполных данных с помощью многообразий малой размерности. При этом вначале рассматриваются эмпирические таблицы и задачи эмпирического предсказания, требования к методам обработки таблиц эмпирических данных, а затем, в четвертом параграфе, описываются теоретические основы самого метода моделирования неполных данных с помощью многообразий малой размерности. Рассматриваются различные варианты применимости метода. В итоге ставятся задачи, решению которых посвящен разработанный метод.

I.1. Методы обработки данных с пропусками

В рамках классической теории статистического прогнозирования [41, 42, 49] известно, что оптимальным статистически достоверным прогнозом отсутствующего значения является его условное математическое ожидание – регрессия. Однако весь аппарат математической статистики основан на предположении о том, что функции распределения нормальны или близки к нормальным. Поэтому при восстановлении функции условного математического ожидания требуется проверять гипотезу о распределении эмпирических данных по нормальному закону или использовать аппарат непараметрической статистики, восстанавливающий оценки плотностей распределения вероятностей.

В докомпьютерное время (до 1960 г.), начиная со статьи Уилкса [86], исследования по проблеме заполнения пропусков носили в основном теоретический характер и касались большей частью оценок максимального правдоподобия (МП–оценки) по некомплектным выборкам. На практике же использовались примитивные способы борьбы с пробелами – вычеркивание некомплектных строк или столбцов, замена пробелов средними по столбцу, использование в вычислениях только комплектных пар и т.п. Полный обзор этих и многих других методов до 1966 г. можно найти в [55], некоторые из них приведены в [25].

С распространением ЭВМ были предложены более сложные машинные алгоритмы, основанные на методе наименьших квадратов: регрессионный метод [57, 85], метод главных компонент [64], пошаговая регрессия [62], метод многомерной линейной экстраполяции [50], метод прогностических переменных [36]. Учитывая тот факт, что оценки первых двух моментов полностью определяют оценки регрессии, многие авторы сосредоточились на проблеме оценивания ковариационной матрицы по данным с отсутствующими значениями [61, 63, 71]. Одновременно выяснилась и некоторая ограниченность методов, основанных на методе наименьших квадратов. Так, Уилкинсон [41, стр. 167] указывал, что если пробелы имеются только в отклике, то при совместном оценивании пробелов и коэффициентов регрессии метод наименьших квадратов требует вычеркивать все строки с пробелами. Это приводит к неполному использованию информации, содержащейся в данных.

Со второй половины 70-х годов особых успехов добилось направление, связанное с МП-оценками, особенно в рамках нормальных распределений. Появились практические алгоритмы, вычисляющие МП-оценки пробелов, например [56, 67, 84]. В работе [59] предложена мощная вычислительная процедура – ЕМ-алгоритм для решения общей задачи оценивания параметров в условиях некомплектной выборки. К настоящему времени эти методы интенсивно развиваются, созданы эффективные робастные варианты ЕМ-алгоритма [80]. Возобладала тенденция поиска для всех классических статистических методов аналогов, способных работать с некомплектными данными, не заполняя пробелов [70, 80, 82]. Более полный обзор теории и практики содержится в монографиях [60, 78].

Достоверное статистическое оценивание должно давать для отсутствующих данных их условные математические ожидания (условия – известные значения других признаков) и характеристики разброса – доверительные интервалы. Это, однако, требует либо непомерно большого объема известных данных, либо очень сильных предположений о виде функций распределения. Приходится вместо ***статистически достоверных*** уравнений регрессии использовать ***правдоподобные эмпирические*** методы заполнения пропусков.

Так, один из известных эмпирических методов – алгоритм “ZET” [39, 40] – основан на изучении “похожести” объектов. Из предположения избыточности строится "предсказывающая подтаблица", состоящая из наиболее связанной с интересующим нас неизвестным элементом информации. Далее из принципа локальной линейности прогнозируется оценка неизвестного значения.

В [27, 28, 29] приводится пример прогнозирования эмпирических (измеряемых) данных по теоретическим данным (например, месту объекта в какой-либо естественной классификации) – прогнозируются высшие потенциалы ионизации атомов на основе атомных номеров. Не так давно был создан новый эмпирический метод – метод транспонированной регрессии [30, 31, 32, 65], близкий по идее к полуэмпирическому методу [27–29].

Основной проблемой в задаче транспонированной регрессии является нахождение для каждого объекта наилучшей функции из данного класса

функций и наилучшей опорной группы объектов, по которым строится эта функция.

I.2. ГЛАВНЫЕ КРИВЫЕ

Впервые понятие главной кривой появилось в [68] в 1989 году. В этой работе главные кривые были определены как "self-consistent" гладкие кривые, которые проходят через середину *d*-мерного вероятностного распределения или облака данных. Self-consistency означает, что каждая точка кривой есть среднее всех точек данных, которые проецируются в эту точку кривой.

В качестве связи между главными компонентами и главными кривыми показано, что если прямая линия есть self-consistent, тогда она является главной компонентой. Также показано, что при некоторых условиях главные кривые представляют собой критические точки расстояния от наблюдения. Основываясь на этом свойстве, авторы разработали алгоритм для нахождения главных кривых и для распределений, и для множеств данных.

Определение главной кривой, данное в [73], имеет то преимущество, что главные кривые существуют всегда, если распределение имеет конечные вторые моменты. Новое определение также делает возможным проведение теоретического анализа обучения главных кривых на основе учебных данных. Основываясь на этом, определен алгоритм построения ломаных, который успешно сравнивается с предыдущими методами.

В [83] дано альтернативное определение главной кривой, основанное на смешанной модели. Оценка выполнена с использованием метода максимального правдоподобия.

В [58] на основе обобщения полной дисперсии вводится понятие главной точки, а потом рекурсивно определяются главные кривые и поверхности.

I.3. ТАБЛИЦЫ ЭМПИРИЧЕСКИХ ДАННЫХ

В данной работе исследуется классическая задача обнаружения эмпирических закономерностей [37, 38], рассмотрению которой посвящен широкий спектр работ. Объектом исследования в них является таблица эмпирических данных, в которой систематизированы сведения о результатах измерений некоторых свойств изучаемых объектов. Эта таблица используется для предсказания значения некоторого выделенного исследователем свойства (или множества свойств) объекта при отсутствии информации о структуре объекта и его внутренних взаимосвязях. Для этого используют в основном способ рассуждений по аналогии, при этом в таблице должны содержаться объекты с известными значениями выделенного свойства. Обычно строки эмпирической таблицы соответствуют множеству объектов, а столбцы – множеству свойств или признаков, отсюда другие названия: таблица “объект-свойство”, “объект-признак”. В результате анализа данных в таблице исследователь получает некоторые эмпирические закономерности, которые используются для прогнозирования значений. Методы обнаружения

закономерностей в таблицах эмпирических данных, в свою очередь, являются широким полем для исследований [26, 37, 38, 43]. При этом в каждом отдельном случае понятие закономерности конкретизируется.

Задачи эмпирического предсказания

Итак, под эмпирической таблицей понимается таблица, элементы которой есть результаты измерений ряда признаков у подмножества объектов A, выбранных из некоторого множества Г. Множество Г задается в зависимости от целей исследования. При этом считается, что исследователь имеет правило, по которому он может определить, принадлежит или не принадлежит множеству Г произвольный рассматриваемый объект. Любому объекту $a \in \Gamma$ можно сопоставить вектор $\mathbf{x}=(x_1,\ldots, x_j,\ldots, x_n)$ и значение x_0 в пространстве признаков $X_1,\ldots,X_j,\ldots,X_n$; X_0. Признак X_0 выделен в качестве целевого признака. Для каждого признака X_j определена область его значений D_j (j=1,…,n; 0) и указан тип шкалы, в которой он измерен [48]. Важно различать группы шкал, предназначенных для измерения признаков следующих трех видов: количественных (шкалы интервалов, отношений и абсолютная); порядковых (шкалы порядка, частичного порядка, рангов, баллов); номинальных (шкала наименований). Пусть выбрано некоторое множество объектов $A=\{a^1,\ldots,a^i,\ldots,a^N\}$, $A \subseteq \Gamma$. Множеству A соответствует таблица $\mathbf{X}=\{x_j^i\}$ (i=1,…,N, j=1,…,n; 0). Таблица **X** используется для решения пяти типов задач эмпирического предсказания [43], которые перечислены ниже.

1. ***Распознавание образов*** (предсказание значения целевого признака x_0 для любого объекта $a \in \Gamma$ по его описанию **x**; в этом случае признак X_0 замерен в шкале наименований).

2. ***Предсказание значения целевого признака*** x_0 ***для объекта*** $a \in \Gamma$ ***по его описанию x***. Признак X_0 – порядковый или количественный.

3. ***Упорядочивание объектов по их перспективности с точки зрения некоторого критерия*** (предсказание порядка на объектах некоторого подмножества A' ($A' \neq A$, $A' \subset \Gamma$).

4. ***Автоматическая группировка объектов***. В данном случае значения признака X_0 для подмножества объектов A не заданы. Необходимо эти значения определить, используя свойство “похожести” объектов по их описанию.

5. ***Динамическое прогнозирование значения целевого признака*** x_0 ***объекта*** $a \in \Gamma$, ***использующее временные измерения значений признаков*** $X_1,\ldots,X_n$ (анализ временных рядов). В качестве примера задачи динамического прогнозирования можно привести задачу ранней диагностики заболеваний на основе профилактических осмотров пациентов.

При решении данных задач используется следующая эмпирическая гипотеза: считается, что при выборе объектов подмножества A из множества Г не делается предпочтения одного объекта другому: объекты подмножества A выбираются из Г случайным образом, т. е. рассматривается статистическая постановка задачи. Известные методы обработки эмпирических таблиц строят решающее правило, максимизирующее качество предсказания на объектах

подмножества *A*. Если допустить, что указанная гипотеза не верна, то всегда можно так подобрать объекты подмножества *A* из Г, что это правило будет плохо работать на остальных объектах множества Г. Поэтому большинство методов обработки таблиц в явном или неявном виде используют эту гипотезу.

Очевидно, что все вышеприведенные задачи могут рассматриваться как единая задача заполнения пропусков в таблице. В начале этой главы были рассмотрены существующие методы решения этой задачи. В дальнейшем основное внимание будем уделять задаче восстановления пропущенного количественного или порядкового признака (второго типа).

Требования к методам обработки таблиц эмпирических данных

Рассмотрим основные особенности указанных в предыдущем разделе задач для случая изучения сложных объектов.

1. Задачи приходится решать в условиях высокой априорной неопределенности, когда практически ничего неизвестно о виде функций распределения вероятностей в пространстве признаков. Всякое “сильное” предположение (например, о нормальности распределения, некоррелированности признаков и т.д.) ставит вопрос об адекватности предлагаемого вида действительному. То же самое можно сказать и о предположении об унимодальности функций распределения. Поэтому методы решения задач должны быть универсальными, т.е. ориентированными на достаточно слабые ограничения на вид распределений.
2. При изучении сложных объектов возникают большие трудности при задании исходной системы признаков для их описания. Поэтому в признаковом пространстве может быть много “дублирующих” и “шумящих” признаков. В результате проблема выбора наиболее информативной подсистемы признаков приобретает важное значение, поскольку уменьшение числа признаков часто улучшает качество решения (и сокращает экономические и временные затраты на измерения или сбор информации).
3. Для описания объектов используются признаки, измеренные в разных шкалах и, возможно, разнотипные.
4. В связи со сложностью измерения некоторых параметров, отказом датчиков и т.д. в таблице могут отсутствовать некоторые значения исходных признаков и даже целевых у некоторых объектов.

В связи с этим методы решения задач обработки экспериментальных данных должны удовлетворять следующим требованиям:

1) должны работать при наличии пропусков в таблице;
2) работать даже в случае, если число измеренных признаков превышает число объектов, и число объектов достаточно мало;
3) должны обеспечивать возможность обработки разнотипных экспериментальных данных (без сведения всех признаков к одной шкале) и инвариантность к допустимым преобразованиям шкал признаков;
4) должна обеспечиваться достаточно высокая вычислительная эффективность.

И, дополнительно:

5) должен использоваться класс решающих функций, имеющий малую меру сложности;
6) должны обеспечиваться наглядность и легкая интерпретируемость полученных решающих правил.

I.4. ИТЕРАЦИОННОЕ МОДЕЛИРОВАНИЕ НЕПОЛНЫХ ДАННЫХ С ПОМОЩЬЮ МНОГООБРАЗИЙ МАЛОЙ РАЗМЕРНОСТИ

Всем перечисленным в предыдущем параграфе требованиям к методам решения задач обработки экспериментальных данных удовлетворяет метод моделирования данных с пробелами многообразиями малой размерности.

Итак, пусть задана таблица данных, строки которой соответствуют объектам, а столбцы – признакам. Пусть, далее, часть информации в таблице отсутствует – есть пробелы. Основная возникающая в связи с этим задача – правдоподобно заполнить существующие пропуски. Ей сопутствуют еще одна задача – произвести "ремонт" таблицы: выделить данные, имеющие неправдоподобные значения, и исправить их. Кроме того, по таблице, как правило, полезно построить правило вычислений для заполнения пробелов в данных о новых объектах (по мере их поступления) и ремонта этих новых данные. Построение такого правил вычисления предполагает, что данные о новых объектах связаны между собой теми же соотношениями, что и в исходной таблице.

Следует особенно подчеркнуть, что в этих проблемах невозможно говорить ни об истинных значениях данных, ни даже о статистической доказательности, но только о правдоподобии. Особую трудность (и в то же время – притягательность) описанные задачи имеют в тех случаях, когда плотность пробелов высока, расположены они нерегулярно, а данных немного, например, число строк примерно таково же, как и число столбцов.

Обычные алгоритмы регрессии состоят в построении эмпирических зависимостей одних данных от других. Этот подход здесь неприменим. Если расположение пробелов нерегулярно, то фактически требуется построение зависимостей неизвестных данных от известных для всех возможных их положений в таблице. Это означало бы построение 2^{n-1} зависимостей, где n – число признаков. Только в этом случае можно будет восстанавливать любой неизвестный набор данных, если хоть что-то известно. В связи с этим приходится использовать метод моделирования данных многообразиями малой размерности.

Суть метода моделирования данных такова. Вектор данных x с k пробелами представляется как k-мерное линейное многообразие L_x, параллельное k координатным осям, которые соответствуют пропущенным данным. При наличии априорных ограничений на пропущенные значения место L_x занимает прямоугольный параллелепипед $P_x \subset L_x$. Ищется многообразие M заданной малой размерности (чаще всего – кривая), наилучшим образом приближающее данные и удовлетворяющее некоторым требованиям

регулярности. Для комплектных векторов данных точность приближения определяется как обычное расстояние от точки до множества (нижняя грань расстояний до точек множества). Для неполных данных вместо него используется нижняя грань расстояний между точками M и L_x (или, соответственно, P_x). Из данных вычитаются ближайшие к ним точки многообразия M – получается остаток – и процесс повторяется, пока остатки не приблизятся в достаточной степени к нулю. Близость линейного многообразия L_x или параллелепипеда P_x к нулю означает, что мало расстояния от нуля до ближайшей к нему точки L_x (соответственно, P_x). Дальнейшая конкретизация метода состоит в указании того, как строится многообразие M.

Идея же моделирования данных с помощью многообразий малой размерности возникла давно. Самая известная, давняя и очень практичная ее реализация для данных без пробелов – это классический метод главных компонент. Он состоит в том, что данные моделируются с помощью их ортогональных проекций на "главные компоненты" – собственные векторы корреляционной матрицы, которым соответствуют наибольшие собственные значения. Другая алгебраическая интерпретация метода главных компонент – сингулярное разложение таблицы данных. Как правило, для достаточно точного представления данных требуется сравнительно немного главных компонент и размерность сокращается иногда в десятки раз.

Обобщение первой главной компоненты на нелинейный случай ("главная кривая") было предложено в 1988 г. [68, 75, 76]. Известны также обобщения классического метода главных компонент на данные с пробелами.

В работе описан метод построения системы моделей для некомплектных данных. В простейшем случае эти модели являются обобщением классического (линейного) метода главных компонент на данные с пробелами. Далее следует квазилинейный метод, надстраиваемый над линейным и использующий его результаты. Наконец, с помощью формализма самоорганизующихся кривых строится существенно нелинейный метод.

Для каждого метода приводится соответствующая механическая интерпретация, которая показывает сходства методов и их последовательное развитие.

В результате, построенная технология моделирования данных с пробелами многообразиями (линейными и нелинейными) малой размерности в общем случае представляется более эффективной по сравнению с обычными уравнениями регрессии.

Разрабатываемый алгоритм заполнения пробелов в отличии от многих других алгоритмов, предназначенных для той же цели, не требуют их предварительного априорного заполнения данных. Однако, что вполне естественно, он требует предварительной нормировки данных ("обезразмеривания") данных – перехода в каждом столбце таблицы к "естественной" единице измерения. Следует заметить, что в задаче обработки данных с пробелами невозможно перейти к однородной задаче центрированием

данных.

А что касается расположения пробелов в данных, то приведенный алгоритм применим в том случае, когда матрица данных не может быть приведена перестановкой строк и столбцов к следующему блочно-диагональному виду:

$$A=\begin{bmatrix} A_1 & @ & \dots & @ \\ @ & A_2 & \dots & @ \\ \dots & \dots & \dots & \dots \\ @ & @ & \dots & A_n \end{bmatrix},$$

где @ – прямоугольные матрицы с неизвестными элементами. Для таких таблиц связь между различными блоками A_i установить невозможно, а поэтому и невозможно решать задачу восстановления пропущенных данных по известным.

Постановка задачи

Пусть задана прямоугольная таблица $A=(a_{ij})$, клетки которой заполнены действительными числами или значком @, означающим отсутствие данных.

Требуется построить модели, которые позволяли бы решать следующие три задачи, связанные с восстановлением пропущенных данных:

1) правдоподобно заполнить имеющиеся пробелы в данных;

2) отремонтировать данные, т.е. исправить их значения таким образом, чтобы наилучшим образом работали построенные модели;

3) построить по имеющейся таблице вычислитель, который бы заполнял пробелы в данных и ремонтировал бы их по мере по мере поступления (в предположении, что данные в поступающей на вход строке связаны теми же соотношениями, что и в исходной таблице).

Первый возникающий вопрос: *как (в какой метрике) оценивать ошибку модели?* Выбор меры ошибки необходим и для построения моделей и для их тестирования. С точки зрения простоты вычислений наиболее привлекателен метод наименьших квадратов (МНК). Ошибка в нем вычисляется как сумма квадратов уклонений по всем известным данным (среднеквадратичная ошибка Mean Square Error – MSE). Однако и здесь имеется произвол, связанный с выбором масштабов, то есть с нормировкой данных.

В классическом методе главных компонент обычно производится нормировка исходных данных на единичную дисперсию. После такой нормировки первая главная компонента определяется как такое направление (вектор), что ортогональные проекции данных на него имеют максимальную дисперсию. Она соответствует главной оси эллипсоида рассеяния.

Однако нормировка на единичную дисперсию не всегда соответствует сути дела. Кроме среднего квадратичного уклонения σ данной величины на роль естественного масштаба претендуют также точность ее измерения и, что особенно важно, допуск на ее изменение.

Понятие "допуск" происходит из технических приложений и означает тот произвол в значении величины, который может быть допущен без ущерба для решения значимых задач. Допуск определяется индивидуальным пользователем или особыми соглашениями о стандартных допусках. Именно величина допуска, скорее всего, может быть наилучшим естественным масштабом измерения. Следует, однако, помнить, что эта величина определяется не только таблицей данных, но еще и теми задачами, которые с ее помощью будут решаться.

Исходно по построению главных компонент столько же, сколько исходных признаков – просто совершается переход к новой системе координат. Однако нет необходимости вычислять все главные компоненты и, тем более, сохранять их все в модели. Достаточно оставить несколько из них. Если из p данных отобрано m главных компонент ($m<p$), то приходим к так называемой m-факторной модели.

Всюду далее предполагаем, что данные нормированы приемлемым образом (например, на соответствующие допуски) и оцениваем ошибки по методу наименьших квадратов.

Глава II. Линейные и квазилинейные многообразия малой размерности

Введение

В первом параграфе вводится понятие главных компонент. Если это определение расширить на случай некомплектных данных, а также ввести некоторую нелинейность, то получим метод итерационного моделирования неполных данных с помощью линейных и квазилинейных многообразий малой размерности, геометрическая интерпретация которого дана во втором параграфе.

Третий параграф целиком посвящен линейным многообразиям малой размерности, начиная с сингулярного разложения матриц с пропусками и заканчивая методом главных компонент для таблиц с пробелами.

В четвертом параграфе изложен принцип заполнения пропусков и ремонта в отдельной векторе данных. А в пятом описываются алгоритмы построения многомерных линейных многообразий малой размерности и присущих им особенностей.

С шестого параграфа начинается описание квазилинейных многообразий, а в седьмом и восьмом приводятся основные проблемы при построении квазилинейных многообразий: интерполяция и экстраполяция соответственно.

В девятом параграфе заканчивается изложения принципов работы с квазилинейными многообразиями малой размерности и последние два параграфа, десятый и одиннадцатый, посвящены уже прикладной стороне использования многообразий малой размерности: во-первых, рассмотренные модели допускают ясную механическую интерпретацию, а во-вторых, на их основе может быть построен нейронный конвейер, способный решать задачи заполнения пробелов в данных, а также ремонтировать эти данные.

II.1. Метод главных компонент

Анализ главных компонент [81], возможно, наиболее известная методика многомерного анализа и используется для снижения размерности и извлечения особенностей. Рассмотрим случайный вектор $X=(X^{(1)},\ldots, X^{(d)})$ размерности d и с конечными случайными моментами. Первая линейная главная компонента для X представляет собой прямую линию, ожидаемое Евклидовое расстояние от которой до X минимальное среди всех прямых линий. Это свойство делает первую главную компоненту сжатым одномерным приближением распределения X и проекция X на эту линию дает наилучшее линейное представление данных. Для эллиптических распределений первая главная компонента также *self-consistent*, т.е. любая точка линия есть условное математическое ожидание X над теми точками пространства, которые проецируются в данную точку.

Рассмотрим множество данных X с элементами $v=(v_1,\ldots,v_n)\in \Re^d$. Главная компонента P описывает множество данных X как линейная функция f одного

переменного t, т.е. $x \in X$ представляется как $f(v)=\lambda(v)c+c_0 \in P$. Задавая X, значения векторов c и c_0 определяются как минимум квадратичной ошибки приближения

$$c, c_0: \frac{\partial}{\partial c} E = 0, \frac{\partial}{\partial c_0} E = 0, \tag{2.1}$$

где

$$E = \int_X \|v - \lambda c - c_0\| P(v) dv. \tag{2.2}$$

$P(v)$ – вероятностное распределение данных. Уравнения (2.1) и (2.2) подразумевают, что дистанция $\|v - w\|$ минимальна по изменению w вдоль P. Другими словами, проекция точки данных v определяется ее ближайшей точкой на главной компоненте.

II.2. ГЕОМЕТРИЧЕСКАЯ ИНТЕРПРЕТАЦИЯ

Предлагаемый метод моделирования данных с пропусками данных линейными многообразиями малой размерности имеет ясную геометрическую интерпретацию.

Пусть имеется таблица типа "объект-признак", то есть каждая строка этой таблицы соответствует объектам, а столбцы – признакам. Сопоставим ей матрицу $A=(a_{ij})$, каждый элемент a_{ij} которой соответствует j-му свойству i-го объекта. Каждая строка этой матрицы есть вектор данных a с k пробелами, который представляется как k-мерное линейное многообразие L_a, параллельное k координатным осям, которые соответствуют пропущенным данным. При наличии априорных ограничений на пропущенные значения место L_a занимает прямоугольный параллелепипед $P_a \subset L_a$.

Построим моделирующее эти данные линейное многообразие малой размерности следующим образом:

За основу возьмем прямую $f(x)=xy+b$, которая задается направляющим вектором y и проходит через точку, определяемую вектором b. Причем, задавая ограничения на значения свободного члена b, мы можем требовать, чтобы прямая проходила или не проходила через начало координат. Далее, расположим эту прямую так, чтобы она наилучшим (в некотором точном смысле) образом приближала исходные данные. Если взять в качестве проектора данных на эту прямую ортогональный проектор, то исходный вектор данных a ортогонально проецируется таким образом в вектор $x=\mathrm{Pr}_f(a)$ на полученной прямой.

Для исходных данных можно посчитать их уклонения от линейной модели, которые находятся из разницы между исходными данными и их проекциями на полученную прямую. Для полученных уклонений также можно построить приближающее наилучшая (в определенном точном смысле) прямая, для которой тоже можно рассчитать уклонения.

Для более лучшего приближения исходных данных можно подобрать такую гладкую вектор-функцию одного переменного, определяемого через проекцию данных на уже построенную прямую, что суммарное значение

квадратов уклонений будет минимальным среди всех возможных функций данного класса (естественно, при одинаковых ограничениях на гладкость). Такой тип линий называется квазилинейным.

В результате, получается итерационный процесс моделирования данных, который заключается в том, что для исходных данных строится наилучшая (в определенном точном смысле) модель – линейное или квазилинейное многообразие M малой размерности. Далее из данных a (соответственно L_a или P_a) вычитаются проекции $x=\Pr_M(a)$. Получаем уклонения от первой модели. Для этого множества уклонений снова строится простая модель и т.д., пока все уклонения не станут достаточно близки к нулю.

II.3. Линейные многообразия малой размерности

Сингулярное разложение матриц с пропусками

Идея сингулярного разложения матриц, содержащих пропуски, на сумму одноранговых матриц была предложена старшим научным сотрудником Новосибирского института математики С.В. Макаровым. И хотя материал этого раздела непосредственно в обработке данных не используется, он дает нам простейший пример и прототип для дальнейших построений.

Пусть задана прямоугольная матрица $A=(a_{ij})$, клетки которой заполнены действительными числами или значком @, означающим отсутствие данных.

Требуется представить исходную матрицу A в виде суммы одноранговых матриц P_q: $A=\sum_q P_q$.

Таким образом, ставится задача поиска наилучшего приближения A матрицей вида x_iy_j методом наименьших квадратов:

$$\Phi=\sum_{\substack{i,j\\ a_{ij}\neq @}}(a_{ij}-x_iy_j)^2\to\min. \qquad (2.3)$$

Решение этой задачи дается последовательными итерациями по явным формулам. При фиксированном векторе y_j значения x_i, доставляющие минимум форме (2.3), однозначно и просто определяются из равенств $\partial\Phi/\partial x_i=0$:

$$\frac{\partial\Phi}{\partial x_i}=-2\sum_{\substack{j\\ a_{ij}\neq @}}(a_{ij}-x_iy_j)y_j=0,$$

$$x_i=\left(\sum_{\substack{j\\ a_{ij}\neq @}}a_{ij}y_j\right)\Bigg/\left(\sum_{\substack{j\\ a_{ij}\neq @}}(y_j)^2\right).$$

Введем обобщенное на случай данных с пробелами определение скалярного произведения $(\cdot,\cdot)_a$ и нормы $\|\cdot\|_a$.

Определение 1: Скалярное произведение $(y_1,y_2)_a$ векторов y_1 и y_2 называется скалярным произведением по известным компонентам вектора a и считается следующим образом:

$$(y_1, y_2)_a = \sum_{\substack{i \\ a_i \neq @}} y_{1i} y_{2i} .$$

Определение 2: Норма $\|y\|_a$ вектора y называется нормой по известным компонентам вектора a и считается следующим образом:

$$\|y\|_a = \sqrt{(y,y)_a} = \sqrt{\sum_{\substack{i \\ a_i \neq @}} y_i^2} .$$

С учетом этих определений можно заметить, что значения проекций x_i находятся как нормированное скалярное произведение вектора данных a_i (i-ая строка матрицы A) на вектор y:

$$x_i = \frac{(a_i, y)_{a_i}}{\|y\|_{a_i}^2},$$

где, напомним, скалярное произведение $(\cdot,\cdot)_{a_i}$ и норма $\|\cdot\|_{a_i}$ вычисляются по известным компонентам вектора $a_{i,}$, т.е. мы имеем дело с обобщенным на случай данных с пробелами скалярным произведением и нормой.

Аналогично и при фиксированном векторе x_i значение y_j, доставляющее минимум форме (1.3), определяются явно из равенств $\partial\Phi/\partial y_j=0$:

$$\frac{\partial \Phi}{\partial y_j} = -2 \sum_{\substack{i \\ a_{ij} \neq @}} (a_{ij} - x_i y_j) x_i = 0,$$

$$y_j = \left(\sum_{\substack{i \\ a_{ij} \neq @}} a_{ij} x_i \right) \Bigg/ \left(\sum_{\substack{i \\ a_{ij} \neq @}} (x_i)^2 \right).$$

Аналогично вычислению проекции x_i через скалярное произведение, можно записать соответствующую проекцию j-го столбца a_j матрицы A на вектор x:

$$y_j = \frac{(a_j, x)_{a_j}}{\|x\|_{a_j}^2},$$

где скалярное произведение и норма вычисляются по известным компонентам вектора a_j.

Процесс вычисления итерационный, поэтому в качестве начального приближения вектора y возьмем случайное значение, но потребуем, чтобы y был единичной длины:

y – случайный, нормирован на 1 (т.е. $\|y\|^2 = \sum_j y_j^2 = 1$).

В качестве критерия остановки будем использовать малость относительного улучшения значения минимизируемого функционала на итерации, т.е. критерий остановки – малость относительного улучшения $\Delta\Phi/\Phi$, где $\Delta\Phi$ – полученное за цикл уменьшение значения Φ, а Φ – само текущее значение. Естественно и использование второго критерия остановки – малость самого значения Φ.

Таким образом, итерационная процедура останавливается, если $\Delta\Phi/\Phi<\varepsilon$ или $\Phi<\delta$ для некоторых ε, $\delta<0$.

В результате для матрицы A получили наилучшее приближение матрицей P_1 вида x_iy_j. Далее, из матрицы A вычитаем полученную матрицу P_1, и для полученной матрицы уклонений $A–P_1$ вновь ищем наилучшее приближение P_2 этого же вида и т.д., пока, например, норма A не приблизится в достаточной степени к нолю.

В результате получили опять же итерационную процедуру разложения матрицы A в виде суммы матриц ранга 1, т.е. $A=P_1+P_2+\ldots+P_q$.

Из теории сингулярного разложения матрицы в виде суммы одноранговых матриц известно, что в случае полной (без пробелов) матрицы число полученных одно-ранговых матриц не превышает число столбцов исходной матрицы. В общем же случае при наличии пробелов это не так.

Метод главных компонент для таблиц с пробелами

В предыдущем разделе был описан метод моделирования данных прямыми, проходящими через начало координат. Однако, такие однородные модели нужны далеко не всегда. Обобщим задачу на случай моделирования данных прямыми, не обязательно проходящими через начало координат.

Таким образом, требуется представить исходную матрицу A в виде суммы матриц P_q: $A=\sum_q P_q$, где каждая P_q имеет вид $x_iy_j+b_j$.

Аналогично случаю с одно-ранговыми матрицами, *основная процедура – поиск наилучшего приближения таблицы с пропусками матрицей вида* $x_iy_j+b_j$ [1, 4].

Следовательно, ставится задача поиска наилучшего приближения A матрицей вида $x_iy_j+b_j$ методом наименьших квадратов:

$$\Phi=\sum_{\substack{i,j\\ a_{ij}\neq@}}(a_{ij}-x_iy_j-b_j)^2\to\min. \qquad (2.4)$$

Решая эту задачу опять же последовательными итерациями по явным формулам, мы получим линию, на которую не накладывается ограничение обязательного прохождения через начало координат. При фиксированных векторах y_j и b_j значения x_i, доставляющие минимум форме (2.4), однозначно и просто определяются из равенств $\partial\Phi/\partial x_i=0$:

$$\frac{\partial\Phi}{\partial x_i}=-2\sum_{\substack{j\\ a_{ij}\neq@}}(a_{ij}-x_iy_j-b_j)y_j=0,$$

$$x_i=\left(\sum_{\substack{j\\ a_{ij}\neq@}}(a_{ij}-b_j)y_j\right)\Bigg/\left(\sum_{\substack{j\\ a_{ij}\neq@}}(y_j)^2\right).$$

Как и в предыдущем разделе, значения проекций x_i находятся как нормированное скалярное произведение центрированного вектора данных $a_i–b$ (центрированная i-ая строка матрицы A) на вектор y:

$$x_i = \frac{(a_i - b, y)_{a_i}}{\|y\|^2_{a_i}},$$

где скалярное произведение и норма вычисляются по известным компонентам вектора a_i.

Аналогично и при фиксированном векторе x_i значения y_j и b_j, доставляющие минимум форме (2.4), определяются явно из двух равенств $\partial\Phi/\partial y_j=0$ и $\partial\Phi/\partial b_j=0$:

$$\frac{\partial\Phi}{\partial y_j} = -2\sum_{\substack{i \\ a_{ij}\neq@}}(a_{ij} - x_i y_j - b_j)x_i = 0,\ \frac{\partial\Phi}{\partial b_j} = -2\sum_{\substack{i \\ a_{ij}\neq@}}(a_{ij} - x_i y_j - b_j) = 0.$$

Представляя полученные уравнения в виде системы, получим:

$$\begin{cases} y_j \sum\limits_{\substack{i \\ a_{ij}\neq@}} x_i^2 + b_j \sum\limits_{\substack{i \\ a_{ij}\neq@}} x_i = \sum\limits_{\substack{i \\ a_{ij}\neq@}} a_{ij}x_i, \\ y_j \sum\limits_{\substack{i \\ a_{ij}\neq@}} x_i + b_j \sum\limits_{\substack{i \\ a_{ij}\neq@}} 1 = \sum\limits_{\substack{i \\ a_{ij}\neq@}} a_{ij}. \end{cases}$$

То же самое в векторной форме:

$$\begin{cases} y_j (x,x)_{a_j} + b_j (x,1)_{a_j} = (a_j, x)_{a_j}, \\ y_j (x,1)_{a_j} + b_j (1,1)_{a_j} = (a_j, 1)_{a_j}. \end{cases}$$

Таким образом:

$$\begin{cases} y_j A_{01}^j + b_j A_{00}^j = B_0^j \\ y_j A_{11}^j + b_j A_{10}^j = B_1^j \end{cases}, \text{ где } A_{kl}^j = \sum_{\substack{i \\ a_{ij}\neq@}} x_i^{k+l},\ B_k^j = \sum_{\substack{i \\ a_{ij}\neq@}} a_{ij}x_i^k,\ k=0..1,\ l=0..1.$$

Выражая из первого уравнения b_j и подставляя полученное значение во второе, получим:

$$y_j = \frac{B_1^j - B_0^j \dfrac{A_{10}^j}{A_{00}^j}}{A_{11}^j - A_{01}^j \dfrac{A_{10}^j}{A_{00}^j}},\ b_j = \frac{B_0^j - y_j A_{01}^j}{A_{00}^j}.$$

Решение полученной системы также может быть найдено при помощи метода Крамера или метода квадратного корня [51].

Поскольку, как и в предыдущем разделе, процедура является итерационной, то в качестве начального приближения вектора y возьмем случайное значение, но нормированное на 1, а в качестве b возьмем средние значения в соответствующих столбцах исходной матрицы A:

y – случайный, нормирован на 1 (т.е. $\|y\|^2 = \sum_j y_j^2 = 1$).

$b_j = \dfrac{1}{n_j}\sum\limits_{\substack{i \\ a_{ij}\neq@}} a_{ij}$, где $n_j = \sum\limits_{\substack{i \\ a_{ij}\neq@}} 1$ (число известных данных в j-ом столбце).

Задаваясь практически произвольными начальными приближениями для y_j и b_j, ищем значение x_i, далее объявляем неизвестными y_j и b_j, находим их значения при фиксированном x_i и т.д. – эти простые итерации сходятся, так как на каждой итерации происходит уменьшение функционала (3.4).

Хотя, несмотря на уменьшение функционала на каждой итерации, возможна ситуация, когда процесс идет так медленно, что относительное уменьшение значения на каждой итерации близко к нолю, то вводится критерий остановки – не когда процесс полностью сойдется, а когда относительное уменьшение на каждой итерации станет меньше напер заданного значения.

Как и для задачи (3.3), критерий остановки – малость относительного улучшения $\Delta\Phi/\Phi$, где $\Delta\Phi$ – полученное за цикл уменьшение значения Φ, а Φ – само текущее значение. Второй критерий – малость самого значения Φ. Процедура останавливается, если $\Delta\Phi/\Phi<\varepsilon$ или $\Phi<\delta$ для некоторых ε, $\delta<0$.

Теперь определим процедуру последовательного итерационного исчерпания матрицы A:

Последовательное исчерпание матрицы A. Решая задачу (2.4), для данной матрицы A находим наилучшее приближение матрицей P_1 вида $x_iy_j+b_j$. Далее, из матрицы A вычитаем полученную матрицу P_1, и для полученной матрицы уклонений $A–P_1$ вновь ищем наилучшее приближение P_2 этого же вида и т.д. Контроль ведется, например, по остаточной дисперсии столбцов.

В результате исходная матрица данных A представляется в виде суммы матриц P_q, т.е. $A=P_1+P_2+\ldots+P_q$. Если пробелы отсутствуют, т.е. все значения a_{ij} известны, то описанный метод приводит к обычным главным компонентам – сингулярному разложению центрированной исходной таблицы данных. В этом случае, начиная с $q=2$, $P_q = x_i^q y_j^q$ ($b=0$). В общем случае это не так. *Следует обратить особое внимание на то, что центрирование (переход к нулевым средним) к данным с пробелами неприменимо.*

С использованием Q полученных факторов можно решать задачи заполнения пропусков в таблице и ремонта искаженных значений:

Q-факторное заполнение пропусков: пропущенные значения в исходной матрице A определяются из суммы Q полученных матриц вида $x_iy_j+b_j$;

Q-факторный "ремонт" таблицы: значения в исходной матрице заменяются на сумму Q полученных матриц вида $x_iy_j+b_j$.

Остается заметить, что при отсутствии пробелов полученные прямые будут ортогональны и мы получим ортогональную систему факторов. Для неполных данных это не так, но возможен процесс ортогонализации полученной системы факторов, который, к примеру, заключается в том, что исходная таблица восстанавливается при помощи полученной системы факторов, после чего эта система пересчитывается заново, но уже на дополненных данных.

II.4. ВОССТАНОВЛЕНИЕ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ ЛИНЕЙНЫХ МОДЕЛЕЙ

Когда для исходной матрицы A имеется набор исчерпывающих ее с заданной точностью матриц P_q, то, как это было показано в предыдущем параграфе, операции заполнения пропусков и "ремонта" данных не представляют особого труда. Однако часто возникает необходимость обработать отдельно взятый вектор данных, при условии что данные в нем связаны теми же соотношениями, что и в исходной матрице.

На основании этого опишем операцию восстановления данных в поступающей на обработку строке a_j с пробелами (некоторые a_j=@). Пусть построена последовательность матриц P_q вида $x_i y_j + b_j$ ($P_q = x_i^q y_j^q + b_j^q$), исчерпывающая исходную матрицу A с заданной точностью. Для каждого q по заданной строке определим число $x^q(a)$ и вектор a_j^q:

$$a_j^0 = a_j,\ (a_j \neq @);$$

$$\ldots\ldots\ldots\ldots\ldots \qquad (2.5)$$

$$x^q(a) = \left(\sum_{\substack{j \\ a_j \neq @}} (a_j^{q-1} - b_j^q) y_j^q \right) \Bigg/ \left(\sum_{\substack{j \\ a_j \neq @}} (y_j^q)^2 \right);$$

$$a_j^q = a_j^{q-1} - x^q(a) y_j^q - b_j^q,\ (a_j \neq @);$$

$$\ldots\ldots\ldots\ldots\ldots$$

Здесь многообразие M – прямая, координаты точек на M задаются параметрическим уравнением $z_j = t y_j + b_j$, а проекция $\mathrm{Pr}_M(a)$ определяется согласно (2.5):

$$\mathrm{Pr}(a) = t(a) y_j + b_j,$$

$$t(a) = \left(\sum_{\substack{j \\ a_j \neq @}} (a_j - b_j) y_j \right) \Bigg/ \left(\sum_{\substack{j \\ a_j \neq @}} (y_j)^2 \right). \qquad (2.6)$$

То же самое в векторной форме:

$$t(a) = \frac{(a_i - b, y)_a}{\|y\|_a^2}.$$

Для Q-факторного восстановления данных полагаем:

$$\overline{a}_j = \sum_{q=1}^{Q} x^q(a) y_j^q + b_j^q,\ (a_j \neq @). \qquad (2.7)$$

Геометрическая интерпретация этой процедуры состоит в том, что для заданного входного вектора a находим его проекцию a_1 на прямую $f_1(t) = t y_1 + b_1$. Далее, вычитая из вектора полученную проекцию, находим его уклонение от прямой f_1. Для этого уклонения снова ищем проекцию a_2, но уже на прямую $f_2(t) = t y_2 + b_2$ и т.д. Таким образом, исходный вектор представляется в виде суммы Q векторов:

$$a = \sum_{q=1}^{Q} a^q .$$

В итоге, используя это разложение, восстанавливаются пропущенные значение в поступающем на обработку векторе *a*, а также находятся "исправленные" оценки уже известных его значений.

II.5. МНОГОМЕРНЫЕ ЛИНЕЙНЫЕ МНОГООБРАЗИЯ

Для построения линейных многообразий размерности больше 1 используются совершенно аналогичные формулы. Но так как в составе одного многообразия размерности больше 1 задающие его векторы могут быть не ортогональными, то для облегчения интерпретации полученных факторов рекомендуется проводить соответствующую ортогонализацию.

Остановимся на двумерных и трехмерных линейных многообразиях, для которых опишем процедуры построения и ортогонализации.

Однако, ко всему сказанному хочется заметить, что использование многомерных линейных моделей не приносит ничего существенно нового. Увеличение размерности линейных моделей может дать эффект только при использовании их соответствующих квазилинейных форм.

Ортогонализация базисной системы векторов

Пусть дано подпространство размерности *n* и в нем система линейно независимых векторов $\{y_k\}$, k=1..n. По определению, они образуют базис этого подпространства. Требуется провести процедуру ортогонализации этого базиса, т.е. построить такой новый базис этого подпространства $\{\bar{y}_k\}$ (k=1..n), чтобы $(\bar{y}_i, \bar{y}_j) = 0$ для любых $i \neq j$.

Для начала решим следующую задачу.

Пусть набор векторов $\{y_k\}$ (k=1..n–1) уже ортогонализирован, а y_n – еще нет. Тогда построим новый вектор $\bar{y}_n = \sum_{i=1}^{n-1} \alpha_i y_i + y_n$ и потребуем, чтобы $(\bar{y}_n, y_k) = 0$ для всех k=1..n–1. В результате имеем систему из n–1 уравнения с n–1 неизвестными:

$$\left(\sum_{i=1}^{n} \alpha_i y_i + y_n, y_k \right) = 0 , \; k=1..n-1.$$

Учитывая, что (y_i,y_k)=0 при $i \neq k$, то легко заметить, что полученная система будет диагональной. Отсюда неизвестные легко находятся по формулам:

$$\alpha_k = -\frac{(y_n, y_k)}{(y_k, y_k)} .$$

Следовательно, если задана система из *n* векторов, в которой имеется ортогональная подсистема из n–1 вектора, то оставшийся вектор "ортогонализируется" по формуле:

$$\overline{y}_n = y_n - \sum_{i=1}^{n-1} \frac{(y_n, y_i)}{(y_i, y_i)} y_i \,. \tag{2.8}$$

Тогда процедура ортогонализации системы из n базисных векторов выглядит как последовательная ортогонализация систем из 2, 3,…, n векторов с использованием (2.8).

Двумерные линейные модели

Для построения двумерного линейного многообразия минимизируем квадратичную форму:

$$\Phi = \sum_{\substack{i,j \\ a_{ij} \neq @}} (a_{ij} - x_{1i} y_{1j} - x_{2i} y_{2j} - b_j)^2 \to \min \,. \tag{2.9}$$

Решение дается последовательными итерациями по явным формулам. При фиксированных y_{1j}, y_{2j} и b_j значения x_{1i} и x_{2i} однозначно находятся из системы равенств $\partial\Phi/\partial x_{1i}=0$ и $\partial\Phi/\partial x_{2i}=0$:

$$\begin{cases} \dfrac{\partial \Phi}{\partial x_{1i}} = -2 \sum\limits_{\substack{j \\ a_{ij} \neq @}} (a_{ij} - x_{1i} y_{1j} - x_{2i} y_{2j} - b_j) y_{1j} = 0, \\ \dfrac{\partial \Phi}{\partial x_{2i}} = -2 \sum\limits_{\substack{j \\ a_{ij} \neq @}} (a_{ij} - x_{1i} y_{1j} - x_{2i} y_{2j} - b_j) y_{2j} = 0. \end{cases}$$

То же самое в векторном виде:

$$\begin{cases} x_{1i} (y_1, y_1)_{a_i} + x_{2i} (y_1, y_2)_{a_i} = (a_i - b, y_1)_{a_i}, \\ x_{1i} (y_1, y_2)_{a_i} + x_{2i} (y_2, y_2)_{a_i} = (a_i - b, y_2)_{a_i}. \end{cases}$$

Аналогично при фиксированных x_{1i} и x_{2i} значения, доставляющие минимум квадратичной форме (2.9), однозначно находятся из равенств $\partial\Phi/\partial y_{1j}=0$, $\partial\Phi/\partial y_{2j}=0$ и $\partial\Phi/\partial b_j=0$:

$$\begin{cases} \dfrac{\partial \Phi}{\partial y_{1j}} = -2 \sum\limits_{\substack{i \\ a_{ij} \neq @}} (a_{ij} - x_{1i} y_{1j} - x_{2i} y_{2j} - b_j) x_{1i} = 0, \\ \dfrac{\partial \Phi}{\partial y_{2j}} = -2 \sum\limits_{\substack{i \\ a_{ij} \neq @}} (a_{ij} - x_{1i} y_{1j} - x_{2i} y_{2j} - b_j) x_{2i} = 0, \\ \dfrac{\partial \Phi}{\partial b_j} = -2 \sum\limits_{\substack{i \\ a_{ij} \neq @}} (a_{ij} - x_{1i} y_{1j} - x_{2i} y_{2j} - b_j) = 0. \end{cases}$$

То же самое в векторной форме:

$$\begin{cases} y_{1j} (x_1, x_1)_{a_j} + y_{2j} (x_1, x_2)_{a_j} + b_j (x_1, 1)_{a_j} = (a_j, x_1)_{a_j}, \\ y_{1j} (x_1, x_2)_{a_j} + y_{2j} (x_2, x_2)_{a_j} + b_j (x_2, 1)_{a_j} = (a_j, x_2)_{a_j}, \\ y_{1j} (x_1, 1)_{a_j} + y_{2j} (x_2, 1)_{a_j} + b_j (1, 1)_{a_j} = (a_j, 1)_{a_j}. \end{cases}$$

Полученная система решается любым численным способом, например, методом квадратного корня [51] (т.к. полученная матрица является симметричной).

Учитывая, что полученные векторы y_1 и y_2 могут быть не ортогональны, то в некоторых случаях их необходимо ортогонализировать. А так как они образуют базис подпространства размерности 2, то эта задача легко решается с использованием (2.8).

Трехмерные линейные модели

Для построения двумерного линейного многообразия минимизируем квадратичную форму:

$$\Phi = \sum_{\substack{i,j \\ a_{ij}\neq @}} (a_{ij} - x_{1i}y_{1j} - x_{2i}y_{2j} - x_{3i}y_{3j} - b_j)^2 \to \min. \qquad (2.10)$$

Решение дается последовательными итерациями по явным формулам. При фиксированных y_{1j}, y_{2j}, y_{3j} и b_j значения x_{1i}, x_{2i} и x_{3i} однозначно находятся из системы равенств $\partial\Phi/\partial x_{1i}=0$, $\partial\Phi/\partial x_{2i}=0$ и $\partial\Phi/\partial x_{3i}=0$:

$$\begin{cases} \dfrac{\partial\Phi}{\partial x_{1i}} = -2\sum\limits_{\substack{j \\ a_{ij}\neq @}} (a_{ij} - x_{1i}y_{1j} - x_{2i}y_{2j} - x_{3i}y_{3j} - b_j)y_{1j} = 0, \\ \dfrac{\partial\Phi}{\partial x_{2i}} = -2\sum\limits_{\substack{j \\ a_{ij}\neq @}} (a_{ij} - x_{1i}y_{1j} - x_{2i}y_{2j} - x_{3i}y_{3j} - b_j)y_{2j} = 0, \\ \dfrac{\partial\Phi}{\partial x_{3i}} = -2\sum\limits_{\substack{j \\ a_{ij}\neq @}} (a_{ij} - x_{1i}y_{1j} - x_{2i}y_{2j} - x_{3i}y_{3j} - b_j)y_{3j} = 0. \end{cases}$$

То же самое в векторном виде:

$$\begin{cases} x_{1i}(y_1,y_1)_{a_i} + x_{2i}(y_1,y_2)_{a_i} + x_{3i}(y_1,y_3)_{a_i} = (a_i - b, y_1)_{a_i}, \\ x_{1i}(y_1,y_2)_{a_i} + x_{2i}(y_2,y_2)_{a_i} + x_{3i}(y_2,y_3)_{a_i} = (a_i - b, y_2)_{a_i}, \\ x_{1i}(y_1,y_3)_{a_i} + x_{2i}(y_2,y_3)_{a_i} + x_{3i}(y_3,y_3)_{a_i} = (a_i - b, y_3)_{a_i}. \end{cases}$$

Аналогично при фиксированных x_{1i}, x_{2i} и x_{3i} значения, доставляющие минимум квадратичной форме (2.10), однозначно находятся из равенств $\partial\Phi/\partial y_{1j}=0$, $\partial\Phi/\partial y_{2j}=0$, $\partial\Phi/\partial y_{3j}=0$ и $\partial\Phi/\partial b_j=0$:

$$\begin{cases} \dfrac{\partial \Phi}{\partial y_{1j}} = -2 \sum\limits_{\substack{i \\ a_{ij} \neq @}} (a_{ij} - x_{1i} y_{1j} - x_{2i} y_{2j} - x_{3i} y_j - b_j) x_{1i} = 0, \\ \dfrac{\partial \Phi}{\partial y_{2j}} = -2 \sum\limits_{\substack{i \\ a_{ij} \neq @}} (a_{ij} - x_{1i} y_{1j} - x_{2i} y_{2j} - x_{3i} y_{3j} - b_j) x_{2i} = 0, \\ \dfrac{\partial \Phi}{\partial y_{3j}} = -2 \sum\limits_{\substack{i \\ a_{ij} \neq @}} (a_{ij} - x_{1i} y_{1j} - x_{2i} y_{2j} - x_{3i} y_{3j} - b_j) x_{3i} = 0, \\ \dfrac{\partial \Phi}{\partial b_j} = -2 \sum\limits_{\substack{i \\ a_{ij} \neq @}} (a_{ij} - x_{1i} y_{1j} - x_{2i} y_{2j} - x_{3i} y_{3j} - b_j) = 0. \end{cases}$$

То же самое в векторной форме:

$$\begin{cases} y_{1j}(x_1, x_1)_{a_j} + y_{2j}(x_1, x_2)_{a_j} + y_{3j}(x_1, x_3) + b_j(x_1, 1)_{a_j} = (a_j, x_1)_{a_j}, \\ y_{1j}(x_1, x_2)_{a_j} + y_{2j}(x_2, x_2)_{a_j} + y_{3j}(x_2, x_3) + b_j(x_2, 1)_{a_j} = (a_j, x_2)_{a_j}, \\ y_{1j}(x_1, x_2)_{a_j} + y_{2j}(x_2, x_2)_{a_j} + y_{3j}(x_3, x_3) + b_j(x_3, 1)_{a_j} = (a_j, x_3)_{a_j}, \\ y_{1j}(x_1, 1)_{a_j} + y_{2j}(x_2, 1)_{a_j} + y_{3j}(x_3, 1) + b_j(1, 1)_{a_j} = (a_j, 1)_{a_j}. \end{cases}$$

Полученная система решается любым численным способом, например, методом квадратного корня [51] (т.к. полученная матрица является симметричной).

Учитывая, что полученные векторы y_1, y_2 и y_3 могут быть не ортогональны, то в некоторых случаях их необходимо ортогонализировать. А так как они образуют базис подпространства размерности 3, то эта задача легко решается с использованием (2.8).

II.6. КВАЗИЛИНЕЙНЫЕ МНОГООБРАЗИЯ МАЛОЙ РАЗМЕРНОСТИ

В данном разделе рассматривается простейший вариант нелинейного метода главных компонент, который надстраивается над линейным. Предлагается использовать квазилинейные многообразия малой размерности, допускающие простые явные формулы обработки данных и опирающиеся на описанные алгоритмы построения линейных многообразий.

Метод построения квазилинейных моделей

Пусть, как и в случае линейных моделей, задана таблица с пропусками $A=(a_{ij})$, т.е. некоторые $a_{ij}=@$. Построение квазилинейных моделей, наилучшим (в определенном точном смысле) образом приближающих данные, предлагается проводить в несколько этапов.

1. *Построение линейной модели:* решение задачи (2.4). Для определенности полагаем, что $(y,b)=0$, $(y,y)=1$ – этого всегда можно добиться.

2. *Интерполяция (сглаживание):* строится вектор-функция $f(t)$, минимизирующая функционал:

$$\Phi = \sum_{\substack{i,j \\ a_{ij}\neq @}} (a_{ij} - f_j(\sum_k a_{ik} y_k))^2 + \alpha \int_{-\infty}^{+\infty} (f''(t))^2 dt\,, \qquad (2.11)$$

где $\alpha>0$ – параметр сглаживания.

Решение задачи сглаживания может проводиться различными методами. Далее мы рассмотрим решение этой задачи полиномами небольшой степени и кубическими сплайнами. Также будет рассмотрено применение функции Карлемана [24] для решения задачи интерполяции.

Следует заметить, что использование полиномов небольшой степени хоть и позволяет с малыми вычислительными затратами получить удовлетворительную интерполяцию, но не дает хорошей экстраполяции. Использование кубических сплайнов требует большей вычислительной мощности, но зато дает более качественную интерполяцию, хотя экстраполяция тоже оставляет желать лучшего.

Поэтому возникает необходимость в третьем этапе построения квазилинейной модели, а именно в этапе экстраполяции полученной функции на всю вещественную ось.

3. *Экстраполяция:* самая простая экстраполяция полученной вектор-функции $f(t)$ может быть получена при использовании касательных к полученной функции на концах интервала. Намного более интересным представляется использование для экстраполяции формул Карлемана.

Таким образом, сглаженная вектор-функция $f(t)$ экстраполируется с некоторого конечного множества $\{t_k\}$ (которое не обязательно связано с проекциями на прямую $z_j=ty_j+b_j$ исходных строк данных) на всю вещественную прямую с использованием формул Карлемана (с помощью формул Карлемана экстраполируется отклонение кривой $f(t)$ от прямой $ty+b$).

$$f(t) \approx ty + b + \sum_{k=1}^{m} (f(t_k) - t_k y - b) \frac{2(e^{\lambda t} - e^{\lambda t_k})}{\lambda (e^{\lambda t} + e^{\lambda t_k})(t - t_k)} \prod_{\substack{j=1 \\ j\neq k}}^{m} \frac{(e^{\lambda t_k} + e^{\lambda t_j})(e^{\lambda t} - e^{\lambda t_j})}{(e^{\lambda t_k} - e^{\lambda t_j})(e^{\lambda t} + e^{\lambda t_j})}, \qquad (2.12)$$

где λ – параметр метода, характеризующий, насколько широка полоса на плоскости комплексных чисел, в которой гарантированно голоморфна экстраполируемая функция (эта ширина равна π/λ).

II.7. Интерполяция

Интерполяция полиномом небольшой степени

Для решения задачи интерполяции полиномом степени n ставится задача наилучшего приближения матрицы A полиномами вида

$$f_j(x) = f_n^j x^n + f_{n-1}^j x^{n-1} + \ldots + f_1^j x + f_0^j\,.$$

На этот полином накладывается ограничения гладкости, то есть в результате требуется решить следующую задачу минимизации:

$$\Phi = \sum_{\substack{i,j \\ a_{ij}\neq @}} (a_{ij} - f_j(\sum_k a_{ik} y_k))^2 + \alpha \int_{-\infty}^{+\infty} (f''(t))^2 dt \to \min, \qquad (2.13)$$

где $\alpha>0$ – параметр сглаживания.

Так как значения $x_i=(a_i,y)$, где a_i – i-ая строка матрицы A, фиксированы (вычислены на предыдущем этапе), то значения коэффициентов полинома f_k^j (k=0..n), доставляющие минимум функционалу Φ, определяются из системы равенств $\partial\Phi/\partial f_k^j = 0$ (k=0..n):

$$\frac{\partial\Phi}{\partial f_k^j} = -2 \sum_{\substack{i \\ a_{ij}\neq @}} (a_{ij} - \sum_{l=0}^{n} f_l^j x_i^l) x_i^k + \alpha \frac{\partial I}{\partial f_k^j} = 0, \text{ где } I = \int_{-\infty}^{+\infty} (f''(t))^2 dt.$$

Вычислим значение интеграла при параметре сглаживания:

$$f_j'(x) = \sum_{k=1}^{n} k f_k^j x^{k-1}, \; f_j''(x) = \sum_{k=2}^{n} k(k-1) f_k^j x^{k-2}, \text{ тогда:}$$

$$\left(f_j''(x)\right)^2 = \sum_{k,l=2}^{n} k(k-1)l(l-1) f_k^j f_l^j x^{k+l-4}.$$

Учитывая, что искомая функция определена на отрезке [–1,1], получим значение интеграла:

$$\int_{-\infty}^{+\infty} \left(f_j''(t)\right)^2 dt = \int_{-1}^{+1} \left(f_j''(t)\right)^2 dt = \sum_{k,l=2}^{n} f_k^j f_l^j k(k-1)l(l-1) \frac{t^{k+l-3}}{k+l-3}\Bigg|_{-1}^{+1} =$$

$$= \sum_{\substack{k,l=2 \\ k+l \text{ - четное}}}^{n} f_k^j f_l^j k(k-1)l(l-1) \frac{2}{k+l-3}.$$

Продифференцируем полученное выражение по f_k^j:

$$\frac{d}{df_k^j} \int_{-1}^{+1} \left(f_j''(t)\right) dt = \sum_{\substack{l=2 \\ l\neq k \\ k+l \text{ - четное}}}^{n} f_l^j k(k-1)l(l-1) \frac{2}{k+l-3} + 2 f_k^j k^2 (k-1)^2 \frac{2}{2k-3}.$$

Окончательно, для k=0..n, имеем:

$$\frac{\partial\Phi}{\partial f_k^j} = -2 \sum_{\substack{i \\ a_{ij}\neq @}} (a_{ij} - \sum_{l=0}^{n} f_l^j x_i^l) x_i^k +$$

$$+ \alpha \sum_{\substack{l=2 \\ l\neq k \\ k+l \text{ - четное}}}^{n} f_l^j k(k-1)l(l-1) \frac{2}{k+l-3} + 2 f_k^j k^2 (k-1)^2 \frac{2}{2k-3} = 0.$$

Группируя коэффициенты при f_k^j (k=0..n), получим:

$$\sum_{l=0}^{n} A_{kl}^j f_l^j = B_k^j, \text{ для } k\text{=0..}n, \text{ где:}$$

$$A_{kl}^{j} = \sum_{\substack{i \\ a_{ij} \neq @}} x_i^{k+l} + \alpha \cdot \begin{cases} 0, \text{если } k < 2 \text{ или } l < 2 \text{ или } k + l \text{ - нечетное}, \\ 2k^2 (k-1)^2 \dfrac{2}{2k-3}, \text{если } k = l, \\ k(k-1)l(l-1)\dfrac{2}{k+l-3}, \text{если } k \neq l. \end{cases}$$

$$B_k^j = \sum_{\substack{i \\ a_{ij} \neq @}} a_{ij} x_i^k \ .$$

Полученная система уравнений может быть решена численным методом.

Интерполяция кубическими сплайнами

Опишем решение задачи интерполяции за счет использования кубических сплайнов, коэффициенты которых находятся из равенства нулю частных соответствующих производных Ф (2.11) на некоторой равномерное сетке. Для тех узлов сетки, на которые не попадают данные, коэффициенты находятся из условий согласования (непрерывности самой функции и непрерывности первой и второй производной).

Таким образом, требуется с использованием кубических сплайнов решить следующую задачу минимизации:

$$\Phi = \sum_{\substack{i,j \\ a_{ij} \neq @}} (a_{ij} - f_j(\sum_k a_{ik} y_k))^2 + \alpha \int_{-\infty}^{+\infty} (f''(t))^2 dt \to \min,$$

где $\alpha>0$ – параметр сглаживания.

Пусть задан отрезок [–1,1]. Разобьем его на n частей точками t_s, s=0..n, где:

$-1 = t_0 < t_1 < \ldots < t_{n-1} < t_n = 1$, при этом $h_s = t_s - t_{s-1}$.

Пусть $t(x) = x - t_{s-1}$, где $t_{s-1} < x < t_s$, s=1..n, тогда имеем:

$$f(x) = f_{s3}t^3 + f_{s2}t^2 + f_{s1}t + f_{s0} = \sum_{l=0}^{3} f_{sl}t^l \ . \qquad (2.14)$$

Вычислим значение интеграла при параметре сглаживания. Для этого вычислим вторую производную функции $f(x)$ и возведем ее в квадрат:

$$f'(x) = \sum_{l=1}^{3} l f_{sl} t^{l-1} \ , \ f''(x) = \sum_{l=2}^{3} l(l-1) f_{sl} t^{l-2} \ , \qquad (2.15)$$

$$\left(f_j''(x) \right)^2 = \sum_{k,l=2}^{3} k(k-1)l(l-1) f_{sk}^j f_{sl}^j t^{k+l-4} \ .$$

Учитывая, что искомая функция определена на отрезке [–1,1], получим значение интеграла:

$$I=\int_{-1}^{1}\left(f''(t)\right)^2dt=\sum_{s=1}^{n-1}\int_{t_{s-1}}^{t_s}\sum_{k,l=2}^{3}k(k-1)l(l-1)f_{sk}^{j}f_{sl}^{j}t^{k+l-4}dt=$$

$$=\sum_{s=1}^{n-1}\left(\sum_{k,l=2}^{3}f_{sk}f_{sl}k(k-1)l(l-1)\frac{t^{k+l-3}}{k+l-3}\Bigg|_{t_{s-1}}^{t_s}\right)=$$

$$=\sum_{s=1}^{n-1}\left(\sum_{k,l=2}^{3}f_{sk}f_{sl}k(k-1)l(l-1)\frac{h_s^{k+l-3}}{k+l-3}\right).$$

Таким образом, исходная задача сглаживания запишется в следующем виде:

$$\Phi=H+\alpha I\to\min,\text{ где}$$

$$H=\sum_{\substack{i\\ a_i\neq@}}\left(a_i-\sum_{l=0}^{3}f_{sl}t_i^l\right),\ I=\sum_{s=1}^{n-1}\left(\sum_{k,l=2}^{3}f_{sk}f_{sl}k(k-1)l(l-1)\frac{h_s^{k+l-3}}{k+l-3}\right).$$

Следует учесть, что искомая сплайн-функция должна быть непрерывная в узлах сетки вместе со своей первой и второй производной:

$$\begin{cases}f(t_s-0)=f(t_s+0)\\ f'(t_s-0)=f'(t_s+0)\\ f''(t_s-0)=f''(t_s+0)\end{cases},\text{ где } s=1..n-1.$$

Подставив соответствующие значения из (2.14) и (2.15), получим линейные ограничения на коэффициенты сплайнов:

$$\begin{cases}f_{s+1,0}=f_{s3}h_s^3+f_{s2}h_s^2+f_{s1}h_s+f_{s0},\\ f_{s+1,1}=3f_{s3}h_s^2+2f_{s2}h_s+f_{s1},\\ f_{s+1,2}=3f_{s3}h_s+f_{s2},\end{cases}\qquad\text{где } s=1..n-1.$$

Учитывая все вышеописанное, будем решать исходную задачу сглаживания методом множителей Лагранжа:

$$L=\Phi+\sum_{s=1}^{n-1}\lambda_{s0}\varphi_{s0}+\sum_{s=1}^{n-1}\lambda_{s1}\varphi_{s1}+\sum_{s=1}^{n-1}\lambda_{s2}\varphi_{s2},\qquad(2.16)$$

где:

$$\begin{cases}\varphi_{s0}=f_{s+1,0}-f_{s3}h_s^3-f_{s2}h_s^2-f_{s1}h_s-f_{s0},\\ \varphi_{s1}=f_{s+1,1}-3f_{s3}h_s^2-2f_{s2}h_s-f_{s1},\\ \varphi_{s2}=f_{s+1,2}-3f_{s3}h_s-f_{s2},\end{cases}\qquad(s=1..\ n-1),$$

есть линейные ограничения, полученные из условий непрерывности.

Дифференцируя (2.16) по f_{sl} и λ_{sl}, получим следующую систему:

$$\begin{cases}\dfrac{\partial L}{\partial f_{sl}}=0,\text{ для } l=0..3,\ s=1..n,\\ \dfrac{\partial L}{\partial \lambda_{sl}}=0,\text{ для } l=0..2,\ s=1..n-1,\end{cases}$$

т.е. систему из (7*n*−3) уравнений и с (7*n*−3) неизвестными.

Первые строчки этой системы можно расписать следующим образом:

$$\frac{\partial L}{\partial f_{sl}} = \frac{\partial H}{\partial f_{sl}} + \alpha \frac{\partial I}{\partial f_{sl}} + L_{sl}\text{, где:}$$

$$\frac{\partial H}{\partial f_{sl}} = -2 \sum_{\substack{i \\ a_i \neq @}} \left(a_i - \sum_{p=0}^{3} f_{sp} t_i^p \right) t_i^l ,$$

$$\frac{\partial I}{\partial f_{sl}} = \sum_{\substack{k=2 \\ k \neq l}}^{3} f_{sk} k(k-1) l(l-1) \frac{2}{k+l-3} + 2 f_{sl} l^2 (l-1)^2 \frac{2}{2l-3},$$

$$\begin{cases} L_{s0} = \lambda_{s-1,0} - \lambda_{s0}, \\ L_{s1} = \lambda_{s-1,1} - \lambda_{s0} h_s - \lambda_{s1}, \\ L_{s2} = \lambda_{s-1,2} - \lambda_{s0} h_s^2 - \lambda_{s1} 2h_s - \lambda_{s2}, \\ L_{s3} = -\lambda_{s0} h_s^3 - \lambda_{s1} 3h_s^2 - \lambda_{s2} 3h_s, \end{cases}$$

при этом коэффициенты с индексом *s*−1 берутся для *s*=2..*n*, коэффициенты с индексом *s* берутся для *s*=1..*n*−1.

Подставив все это в исходную систему, можно заметить, что она имеет практически блочно-диагональный вид, где размер отдельных блоков достаточно мал, поэтому, несмотря на большой размер ((7*n*−3)×(7*n*−3)), она с хорошей точностью решается даже простыми численными методами.

II.8. ЭКСТРАПОЛЯЦИЯ

Проблема экстраполяции, оптимальное аналитическое продолжение и формула Карлемана

Проблема экстраполяции за пределы разброса имеющихся данных хорошо известна. Отказаться от ее решения нельзя – нет гарантии, что следующие данные попадут строго в диапазон изменения имеющихся и не всегда возможно заранее ограничить величину выхода за пределы этого диапазона. Необходимость в построения новых формул для всех возможных значений данных вызвана еще двумя обстоятельствами: во-первых, построенная на втором этапе сглаженная зависимость в принципе интерполяционная и не может быть экстраполирована, во-вторых, в ней фактически содержится в явном виде информация о каждой строке матрицы данных. Сглаживание, например, просто многочленом небольшой степени по методу наименьших квадратов свободно от второго недостатка (информация "сворачивается" в несколько коэффициентов), но не дает хорошей экстраполяции.

При популярной (но весьма грубой) экстраполяции прямыми полученная функция *f*(*t*) экстраполируется с отрезка (например, [*a,b*]) на всю вещественную ось за счет использования первого приближения, построенного в концах отрезка: *f*(*t*)= *f*(*a*)+ *f*′(*a*)(*t*–*a*) при *t*<*a* и *f*(*t*)= *f*(*b*)+ *f*′(*b*)(*t*–*b*) при *t*>*b*.

Более интересен вопрос об оптимальной экстраполяции. Для его строгой формальной постановки надо включить задачу аналитического продолжения функции (с конечного множества точек на прямую или пространство) в бесконечную последовательность таких задач. Удобно также перейти от действительной переменной t к полосе на плоскости комплексных переменных.

Итак, рассматривается задача аналитического продолжения функции, заданной на *бесконечной* последовательности точек $\{t_k\}$ (k=1,2,…). Требуется построить формулу продолжения с *конечного* множества, наилучшую в следующем смысле: последовательность функций $f_n(t)$, полученных продолжением с множества $\{t_k\}$ (k=1,2,…,n), сходится быстрее, чем для всех остальных формул этого класса. Требуется, конечно, доопределить следующие понятия: что такое "формулы этого класса", какая сходимость имеется в виду и др. Все это сделано в соответствующей математической литературе [24].

Сглаженная вектор-функция $f(t)$ может быть оптимальным образом экстраполирована с некоторого конечного множества $\{t_k\}$ (которое не обязательно связано с проекциями на прямую $z_j=ty_j+b_j$ исходных строк данных) на всю вещественную прямую с использованием формул Карлемана [24]:

$$f(t) \approx ty + b + \sum_{k=1}^{m} (f(t_k) - t_k y - b) \frac{2(e^{\lambda t} - e^{\lambda t_k})}{\lambda (e^{\lambda t} + e^{\lambda t_k})(t - t_k)} \prod_{\substack{j=1 \\ j \neq k}}^{m} \frac{(e^{\lambda t_k} + e^{\lambda t_j})(e^{\lambda t} - e^{\lambda t_j})}{(e^{\lambda t_k} - e^{\lambda t_j})(e^{\lambda t} + e^{\lambda t_j})}, \quad (2.17)$$

где λ – параметр метода, характеризующий, насколько широка полоса на плоскости комплексных чисел, в которой гарантированно голоморфна экстраполируемая функция (эта ширина равна π/λ).

Как правило, в качестве множества узлов $\{t_k\}$ для экстраполяции по Карлеману мы используем точки, равномерно расположенные на отрезке, а не исходные экспериментальные данные. Значения $f(t)$ в этих точках находим по интерполяционным формулам.

С помощью формул Карлемана экстраполируется отклонение кривой $f(t)$ от прямой $ty+b$. Формулы Карлемана обеспечивают хорошую экстраполяцию аналитических функция на всю прямую (конечно, строго говоря, в каждом конкретном случае нет гарантий, что именно по формуле (2.17) будет получена наилучшая экстраполяция, однако существует ряд теорем о том, что формула (2.17) и родственные ей дают наилучшее приближение в различных классах аналитических функций [24]).

Процессы сглаживания и экстраполяции по формуле Карлемана можно объединить в один, т.е. проводить одновременно и интерполяцию и экстраполяцию. Это становится возможным, если для интерполяции использовать следующую вариацию формулы Карлемана:

$$f(t) \approx ty + b + \sum_{k=1}^{m} f_k \cdot \frac{2(e^{\lambda t} - e^{\lambda t_k})}{\lambda (e^{\lambda t} + e^{\lambda t_k})(t - t_k)} \prod_{\substack{j=1 \\ j \neq k}}^{m} \frac{(e^{\lambda t_k} + e^{\lambda t_j})(e^{\lambda t} - e^{\lambda t_j})}{(e^{\lambda t_k} - e^{\lambda t_j})(e^{\lambda t} + e^{\lambda t_j})},$$

где коэффициенты f_k находятся из формулы (2.11) аналогично задаче сглаживания кубическими сплайнами.

Интерполяция и оптимальное сглаживание по формуле Карлемана

Требуется решить с использованием формулы Карлемана следующую задачу сглаживания:

$$\Phi = H + \alpha I \to \min, \text{ где:}$$

$$H = \sum_{\substack{i,j \\ a_{ij}\neq @}} (a_{ij} - f_j(\sum_k a_{ik} y_k))^2,\ I = \int_{-\infty}^{+\infty} (f''(t))^2 dt,$$

$$f(t) = ty + b + \sum_{k=1}^{m} f_k E_k(t), \text{ – вектор-функция,}$$

$$E_k(t) = \frac{2(e^{\lambda t} - e^{\lambda t_k})}{\lambda(e^{\lambda t} + e^{\lambda t_k})(t - t_k)} \cdot \prod_{\substack{j=1 \\ j\neq k}}^{m} \frac{(e^{\lambda t_k} + e^{\lambda t_j})(e^{\lambda t} - e^{\lambda t_j})}{(e^{\lambda t_k} + e^{\lambda t_j})(e^{\lambda t} + e^{\lambda t_j})},$$

и, наконец, $\alpha>0$ – параметр сглаживания.

Заметим, что вычисления для каждого j производятся одинаково, поэтому в вычислениях их можно не учитывать.

Значения коэффициентов f_k (k=0..n), доставляющие минимум функционалу Φ, определяются из системы равенств $\partial\Phi/\partial f_k = 0$ (k=0..n) следующим образом:

$$\frac{\partial H}{\partial f_k} = -2\sum_i \left(a_i - t_i y - b - \sum_{l=1}^{m} f_l E_l(t_i) \right) E_k(t_i) = 0,$$

$$2\sum_{l=1}^{m} f_l \left(\sum_i E_l(t_i) E_k(t_i) \right) = 2\sum_i (a_i - t_i y - b) E_k(t_i).$$

Таким образом, если не учитывать интеграл гладкости, то имеем следующую систему линейных уравнений относительно f_k, которая решается численно:

$$\sum_{l=1}^{m} A_{kl} f_l = B_k,\ k=1..m,$$

где $A_{kl} = 2\sum_i E_k(t_i) E_l(t_i)$, $B_k = 2\sum_i (a_i - t_i y - b) E_k(t_i)$.

Посчитаем значение интеграла гладкости I:

$$I = \int_{-\infty}^{+\infty} \left(\sum_{k,l=1}^{m} f_k f_l E_k''(t) E_l''(t) \right) dt = \sum_{k,l=1}^{m} \left(\int_{-\infty}^{+\infty} f_k f_l E_k''(t) E_l''(t) dt \right) =$$

$$= \sum_{k,l=1}^{m} f_k f_l \left(\int_{-\infty}^{+\infty} E_k''(t) E_l''(t) dt \right) = \sum_{k,l=1}^{m} f_k f_l I_{kl}, \text{ где } I_{kl} = \int_{-\infty}^{+\infty} E_k''(t) E_l''(t) dt.$$

$$\frac{\partial I}{\partial f_k} = \sum_{l=1}^{m} f_l I_{kl}^*, \text{ где } I_{kl}^* = \begin{cases} I_{kl}, l \neq k, \\ 2I_{kl}, l = k. \end{cases}$$

С учетом интеграла гладкости I, система линейных уравнений, определяющих точку минимума функционала $\Phi=H+\alpha I$, приобретает вид:

$$\sum_{l=1}^{m} A_{kl} f_l = B_k,\ k=1..m,$$

где $A_{kl} = 2\sum_i E_k(t_i) E_l(t_i) + \alpha I_{kl}^*$, $B_k = 2\sum_i (a_i - t_i y - b) E_k(t_i)$.

Значения I_{kl} трудно найти аналитически, но в этом нет необходимости, так как они могут быть численно посчитаны с требуемой точностью с использованием формул численного интегрирования, например, по формуле трапеций или Симпсона [51].

Полученная система линейных уравнений относительно f_k также может быть решена с использованием численных методов решения СЛАУ [51].

II.9. Использование квазилинейных моделей

Процедура использования квазилинейных моделей несколько отличается от аналогичной процедуры в линейном случае, хоть и базируется на ее основе.

Точка на построенной кривой $f(t)$, соответствующая полному ("комплектному") вектору данных a строится как $f((a,y))$. В этом и заключается квазилинейность метода: сначала ищется проекция вектора данных на прямую $\Pr(a)=ty+b$, $t=(a,y)$, а затем строится точка на кривой $f(t)$. Также и для неполных векторов данных – сначала ищется на прямой ближайшая точка $t(a)$ (проекция неполного вектора a), а затем – соответствующая точка на кривой $f(t)$ при $t=t(a)$.

После построения кривой $f(t)$ из данных вычитаются их проекции, то есть матрица данных заменяется на матрицу уклонений от модели. Далее снова ищется наилучшее линейное приближение (к примеру, вида $x_iy_j+b_j$) для матрицы уклонений, вновь строится сглаживание, потом – экстраполяция по Карлеману и т.д., пока уклонения не приблизятся в достаточной степени к нулю. Критерием остановки опять же могут выступать остаточные дисперсии. Таким образом процедура моделирования данных квазилинейными многообразиями малой размерности тоже является итерационной, как и в линейном случае.

В результате исходная таблица предстает в виде Q-факторной модели:

$$a_{ij} \cong \sum_q f_j^q(t_i^q),$$

где аргументом функции служит специальным образом (как это было описано в третьем параграфе) нормированное скалярное произведение исходного данного на линейную основу квазилинейного многообразие или, другими словами, значение проекции данного на прямую.

Заметим, что если $a_{ij}\neq@$, то эта формула аппроксимирует исходные данные, иначе она дает способ восстановления пропущенных значений.

II.10. Механическая интерпретация

Описанные методы построение многообразий малой размерности, наилучшим образом приближающих данные, имеют ясные механические представления, лежащие в их основе.

Пусть в пространство данных помещена прямая жесткая балка (рис. 2.1). И пусть каждая точка (либо линейное многообразие при наличии в векторе данных пробелов) данных соединена с балкой пружинкой, причем конец пружинки может свободно перемещаться вдоль балки (как и вдоль линейного

многообразия вектора данных при наличии в нем пропусков). Зафиксируем начальное положение балки и найдем такое положение пружинок, которое отвечает минимуму энергии растяжения (сумма квадратов растяжений пружинок). После этого зафиксируем положение концов пружинок на балке, освободим балку и дадим ей прийти в механическое равновесие. Далее зафиксируем балку в новом положении и вновь освободим концы пружинок.

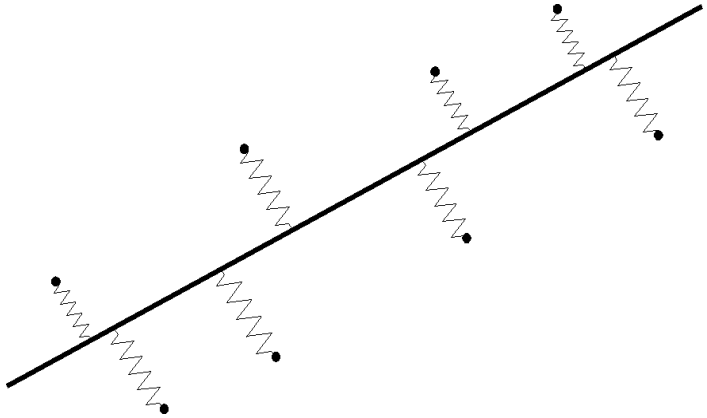

Рис. 2.1

Система монотонно приближается к равновесию – минимуму ее энергии, так как на каждом этапе энергия растяжения уменьшается. Полученная балка будет ни чем иным, как первой главной компонентой. При этом проекция данного определяется местом крепления соответствующей ему пружинки к балке. При наличии пробелов в данных проекция данного определяет его отремонтированное значение, а место крепления пружинки к данному задает его восстановленное значение.

Описанное в предыдущем разделе итерационное нахождение коэффициентов x_i, y_j, b_j полностью аналогично описанному итерационному механическому процессу: положение балки определяется коэффициентами y_j, b_j, а места крепления пружинок – коэффициентами x_i.

Данные с пробелами моделируются твердыми стержнями (один пробел), плоскостями (два пробела) и т.д. Соответствующая пружинка может свободно перемещаться вдоль этих объектов. Это означает, что пружинка эффективно соединяет тень (проекцию) балки на подпространство известных данных с точкой в этом подпространстве.

Если же предположить, что балка может упруго отклоняться от прямого вида, то получим следующую картину (рис. 2.2).

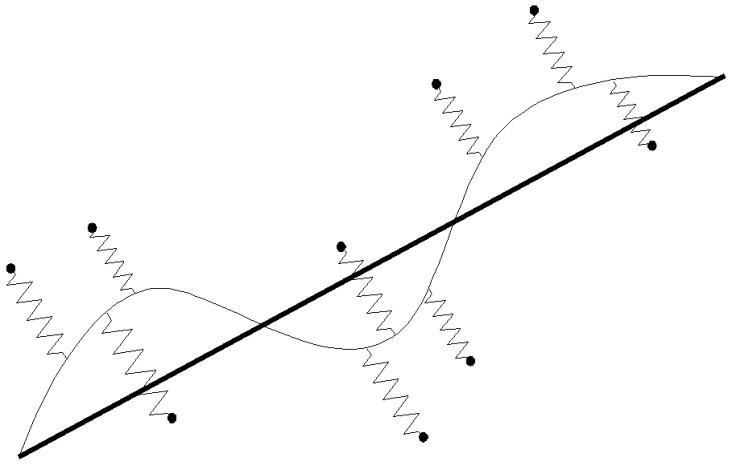

Рис. 2.2

Места крепления пружинок в балке определяются проекциями на прямую балку (т.к. модель квазилинейная).

При этом возникает задача определения поведения концов балки за границами области данных – описанная выше задача экстраполяции. Формула Карлемана в этом случае имеет аналогию с закреплением бесконечных концов балки на прямой.

II.11. НЕЙРОННЫЙ КОНВЕЙЕР ДЛЯ ДАННЫХ С ПРОПУСКАМИ

Если учесть, что построенная квазилинейная модель является суммой квазилинейных вектор-функций, то можно сделать вывод, что указанный алгоритм допускает нейросетевую интерпретацию.

Действительно, построим такую нейронную сеть, где с каждой кривой из (2.17) был бы связан один сумматор (в качестве его весов будем использовать координаты вектора y^q), набор из n свободных слагаемых ("порогов") – координат вектора b^q, и n нелинейных преобразователей, каждый из которых вычисляет одну координату точки на кривой по формуле (2.17).

Действует такой "нейрон" на вектор a входных сигналов (содержащий пробелы) так:

1) по формуле (2.6) вычисляется $t(a)$ (работает сумматор);
2) далее нелинейные элементы вычисляют $f^q(t(a))$;
3) затем разность $f_j^q(t(a))$ ($a_j \neq @$) передается следующему нейрону.

При прохождении a по этому конвейеру одновременно накапливается сумма величин $f_j^q(t(a))$ ($a_j = @$). Они и образуют вектор выходных сигналов – предлагаемые значения пропущенных данных. В случае необходимости провести ремонт данных накапливается сумма величин $f_j^q(t(a))$ для каждой координаты j.

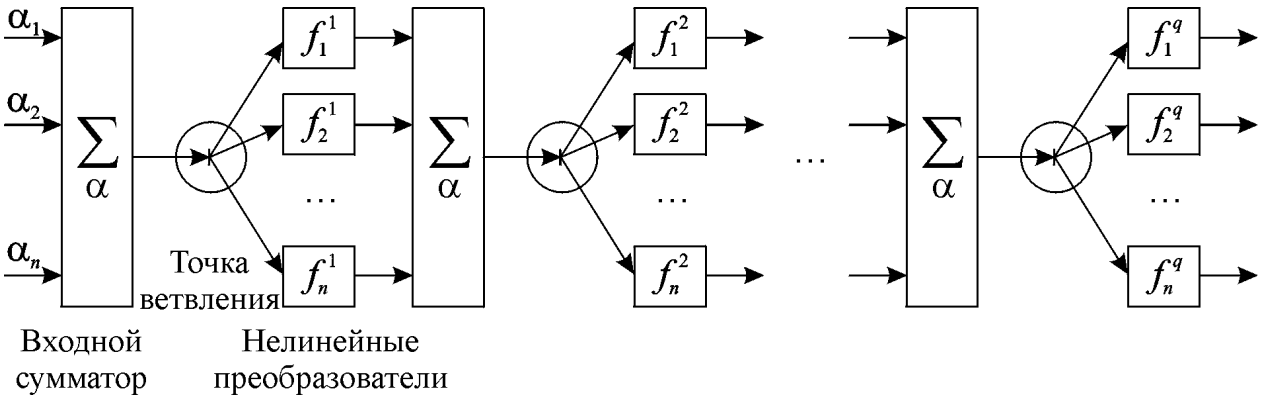

Рис. 2.3

В результате получена следующая нейронная сеть (рис. 2.3). Структура каждого нейрона в ней нестандартна (можно сравнить с [32]) – он имеет один входной сумматор и *n* нелинейных преобразователей (по размерности вектора данных).

Также не вполне обычен способ работы сумматора – для некомплектных векторов данных вычисляется скалярное произведение с имеющимися данными и производится дополнительная нормировка. В этой дополнительной нормировке входных весов сумматора учитываются только тех из них, для которых известны соответствующие значения координат входного вектора.

Примечателен описанный способ построения этих нейронов. Их характеристики вычисляются по очереди, причем сначала строится сумматор (с помощью решения задачи (2.4)), затем – нелинейные преобразователи (по формулам Карлемана), далее сумматор следующего нейрона и т.д.

Все построенные нейроны работают поочередно (в обычном смысле здесь столько же слоев, сколько нейронов), однако они образуют конвейер и освободившиеся нейроны могут переходить к новому вектору данных, поэтому при последовательном поступлении данных время обработки пропорционально числу нейронов, но производительность (количество обработанных векторов данных в единицу времени) определяется временем срабатывания одного нейрона и не зависит от их числа.

Глава III. Самоорганизующиеся многообразия малой размерности

Введение

В первом параграфе вводится понятие главных кривых. Также описывается идея итерационного алгоритма построения главной кривой.

Во втором параграфе вводится понятие самоорганизующихся главных кривых на основе самоорганизующихся карт Кохонена [33, 74].

Третий параграф посвящен самоорганизующимся кривым – одномерным нелинейным многообразиям. Приводится алгоритм их построения. В четвертом же описываются расширения самоорганизующихся кривых – соответствующие самоорганизующиеся карты. Приводятся две основные методики представления карт – с квадратной и гексагональной сетками.

В отличие от линейных и квазилинейных моделей, построение самоорганизующихся кривых и карт связано с проблемой попадания в область локального минимума, возможные решения которой приводятся в пятом параграфе: метод "отжига" и многосеточные методы.

Еще одна проблема, с которой приходится сталкиваться при построении самоорганизующихся многообразий на основе ломаных, заключается в кусочной гладкости полученных многообразий. Поэтому они нуждаются в сглаживании, варианты которого описываются в шестом параграфе.

Последний, седьмой, параграф посвящен механической интерпретации принципа построения самоорганизующихся кривых и карт.

III.1. Определение главных кривых

Главная кривая P_X (Principal Curve – PC) [73] есть обобщение главной компоненты на большой класс нелинейных вектор-функций $f(\lambda)$, $\lambda \in R^1$, предоставляемых для представления данных. Проекция $\lambda(v)$ точки данных v на кривую опять же (как и в случае главных компонент) определяется таким значением λ, для которого $f(\lambda)$ ближайшая к v:

$$\lambda(v) = \arg\min_{\lambda}(v - f(\lambda))^2 .$$

В общем, каждая точка (сегмент в дискретном случае) кривой является проекцией подмножества точек данных, которые называются ее проекторами. В линейном случае проекция совпадает с центром масс проекторов. В этом смысле можно сказать, что главная кривая проходит через середину своих точек данных. Это определение может быть распространено и на нелинейный случай. Следовательно, главная кривая определяется как проходящая через центр масс точек, проецируемых на нее (рис 3.1).

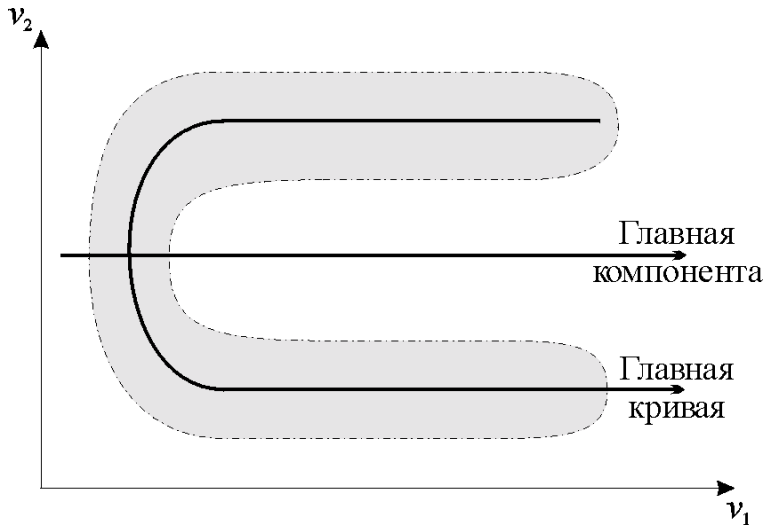

Рис. 3.1.

Также можно сказать, что главной кривой требуется быть центром ее проекторов. Это может быть сформулировано в терминах вариационного принципа, т.е. мы определим среднеквадратическое отклонение между точками данных и их проекциями (ошибка восстановления)

$$E = \int_X (v - f(\lambda))^2 P(v) dv ,$$

и потребуем, чтобы ее изменение относительно $f(\lambda)$ было равно нолю

$$\frac{\partial}{\partial f} E = 0 . \qquad (3.1)$$

Заметим, что мы не должны требовать, чтобы ошибка восстановления была минимальна, как в линейном случае. Вместо этого более слабое свойство стационарности возникает из уравнения (3.1). Причину этого хорошо видно из патологического случая кривой, которая проходит через каждую точку шумящего множества данных X так, что ошибка восстановления равна нолю. Впрочем эта кривая не удовлетворяет назначению главных кривых, поскольку она не исключает шум. Для того, чтобы предотвращать такие патологии, требуется добавочное условие для завершения определения главной кривой. Обычно это условие гладкости, которое ограничивает кривизну PC. Другое условие – это требование, чтобы отображение $v \rightarrow \lambda$ было топографическим лучшим возможным способом. Это точное формулировка тех критериев, которые делают проблему сильно нетривиальной.

Алгоритм Hastie-Stuetzle

Hastie и Stuetzle [68] (здесь и далее HS) описали алгоритм для построения главных кривых, который работает итерационно, начиная с главной компоненты множества данных. На каждой итерации получается новая оценка PC, которая получается вычислением центров масс точек данных относительно текущей оценки PC.

Определение 1: Путь X обозначает случайный вектор в пространстве $\Re^d$ и имеется n векторов данных. Главная кривая $f \subset \Re^d$ – это гладкая (C^∞) кривая в

$\Re^d$, параметризованная параметром $t \in A \subset R^1$, которая проходит через середину d-мерных данных, описываемых X,

$$f(t) = \begin{bmatrix} f_1(t) \\ \vdots \\ f_d(t) \end{bmatrix} = E\left\{X \middle| t_f(X) = t\right\}, \tag{3.2}$$

где:

$t_f(x) = \sup_t \left\{t : \|x - f(t)\| = \inf_\mu \|x - f(\mu)\|\right\}$ – функция проектирования на кривую.

Эта функция определяет такое значение t, для которого дистанция между x и $f(t)$ минимальна.

Таким образом, $f(t)$ есть условное математическое ожидание X среди тех X, к которым $f(t)$ ближе среди любых других точек f (то есть среднее среди всех точек данных, которые проецируются в нее).

Среднеквадратичная ошибка (Mean Square Error – MSE) между точками данных и их ближайшими проекциями на главную кривую на j-ой итерации получается как следующее:

$$MSE^{(j)} = E\left\{\left\|X - f^{(j)}(t_{f^{(j)}}(X))\right\|^2\right\}.$$

Алгоритм построения главной кривой – итерационный, включающий шаг вычисления математического ожидания (3.2) и шаг проектирования на каждой итерации.

Толерантность (TOL) на j-ой итерации определяется как относительное изменение MSE следующим образом:

$$TOL^{(j)} = \frac{\left|MSE^{(j)} - MSE^{(j-1)}\right|}{\left|MSE^{(j-1)}\right|}.$$

Построение прерывается, когда TOL становится ниже заданного порога.

Главная кривая может быть замкнутой или открытой. Для замкнутой кривой во время вычислений добавляется сегмент, включающий конечные точки.

Таким образом, главные кривые являются гладкими self-consistent кривыми, которые проходят через "середину" распределения и обеспечивают хорошее одномерное представление данных.

III.2. Идея самоорганизующихся кривых

Квазилинейные факторы хорошо работают для "средне нелинейных" задач, в которых вклад линейной части относительно велик (порядка 50% или более). Для существенно нелинейных задач естественно использовать вместо линейных главных компонент какие-либо приближения главных кривых.

Предлагается вместо линейных многообразий малой размерности использовать соответствующие самоорганизующиеся карты (Self-Organizing map – SOM) или, другими словами, самоорганизующиеся кривые – SOC (Self-Organizing Curve). Используемый нами алгоритм несколько отличается от

метода карт Кохонена ([74], доступное изложение см. также в [33]) более прозрачной физической интерпретацией и явной формой вариационного принципа.

Предлагается использовать принципиально нелинейный способ. В R^n размещается либо одномерная ломаная, либо двумерная серка, расположение узлов которой удовлетворяет определенным требованиям. И для ломаной, и для сетки этими требованиями являются: а) близость узлов ломаной/сетки к данным; б) не слишком сильная «растянутость» ломаной/сетки; в) не слишком сильная изогнутость ломаной/сетки.

Новшества метода заключается в использовании дополнительных "модулей упругости", а также в использовании кусочно-линейного непрерывного проектора данных на полученную ломаную с последующим сглаживанием по формуле Карлемана.

III.3. SOC

Пусть SOC определяется набором точек (ядер) $Y=\{y_j\}$ ($j=1..m$), последовательно расположенных на кривой (в первом приближении пусть SOC – просто ломаная Y) и требуется отобразить на ней набор точек данных $X=\{x_i\}$. Введем преобразование $П$, которое каждому вектору $x\in X$ сопоставляет ближайшую к нему точку из Y:

$$x \xrightarrow{П} y_j, \left\|y_j - x\right\|^2 \to \min, \tag{3.3}$$

каждому ядру y_j сопоставляется его таксон

$$K_j = \left\{ x \in X \middle| x \xrightarrow{П} y_j \right\}. \tag{3.4}$$

Метод построения SOC напоминает метод динамических ядер, за исключением добавления дополнительных ограничений на связность и нелинейность. Минимизируемая величина строится из следующих слагаемых:

1) мера приближения данных:

$$D_1 = \sum_j \sum_{x\in K_j} \left\|x - y_j\right\|^2; \tag{3.5}$$

Эта мера служит для того, чтобы SOC наилучшим (в некотором определенном смысле) образом приближала исходные данные. Здесь K_j – множество точек из X, для которых узел ломаной y_j является ближайшим.

2) мера связности (близкие точки на кривой должны переходить в близкие в пространстве данных):

$$D_2 = \sum_j \left\|y_j - y_{j+1}\right\|^2; \tag{3.6}$$

Необходимость этой меры вызвана тем, чтобы ломаная не выходила за границы данных – чтобы свободные узлы притягивались к ее "центру". Также эта мера, как будет более подробно описано ниже, используется в различных алгоритмах построения.

3) мера нелинейности:

$$D_3 = \sum_j \left\| 2y_j - y_{j-1} - y_{j+1} \right\|^2 . \qquad (3.7)$$

Эта мера также является мерой равномерности – чтобы расстояния между соседними узлами были примерно одинаковыми.

Таким образом, для построения SOC требуется минимизировать функционал:

$$D = \frac{D_1}{|X|} + \lambda \frac{D_2}{m} + \mu \frac{D_3}{m} \to \min , \qquad (3.8)$$

где λ, μ – параметры связности и нелинейности – "модули упругости" (деление на число точек $|X|$ и число ядер m означает нормировку "на одно слагаемое" и позволяет для выборок разной мощности использовать одинаковые способы изменения λ и μ.)

При фиксированном разбиении множества данных на таксоны SOC строится однозначно – решается простая линейная задача. При фиксированном положении ядер таксоны также легко строятся по формулам (3.3, 3.4). Расщепляя задачу на последовательный поиск ядер – таксонов – ядер … , получаем итерационный алгоритм, сходимость которого гарантируется тем, что на каждом его шаге уменьшается критерий D (3.8).

Алгоритм построения SOC

Пусть задана таблица (a_{ij}) с пробелами (некоторые a_{ij}=@) и набор ядер $Y=\{y^k\}$ (k=0..m-1). Формулы (3.3) – (3.8) перепишутся следующим образом (a_i – i-ая строка таблицы (a_{ij})):

$$a_i \overset{\Pi}{\to} y^k , \sum_i (a_{ij} - y_j^k)^2 \to \min , \qquad (3.9)$$

$$K_k = \left\{ a_i \in (a_{ij}) \middle| a_i \overset{\Pi}{\to} y^k \right\} . \qquad (3.10)$$

$$D_1 = \sum_{k=0}^{m-1} \sum_{i \in K_k} \sum_{\substack{j \\ a_{ij} \neq @}} (a_{ij} - y_j^k)^2 , \qquad (3.11)$$

$$D_2 = \sum_{k=0}^{m-2} \sum_j (y_j^k - y_j^{k+1})^2 , \qquad (3.12)$$

$$D_3 = \sum_{k=1}^{m-2} \sum_j (2y_j^k - y_j^{k-1} - y_j^{k+1})^2 . \qquad (3.13)$$

Требуется решить следующую задачу:

$$D = \frac{D_1}{n} + \lambda \frac{D_2}{m} + \mu \frac{D_3}{m} \to \min , \qquad (3.14)$$

где n – число строк матрицы (a_{ij}).

Значения y_j^k, доставляющие минимум форме (3.14) при заданном разбиении K_k, определяются из системы равенств $\partial D / \partial y_j^k = 0$:

$$\frac{\partial D}{\partial y_j^k} = \frac{1}{n} \cdot \frac{\partial D_1}{\partial y_j^k} + \frac{\lambda}{m} \cdot \frac{\partial D_2}{\partial y_j^k} + \frac{\mu}{m} \cdot \frac{\partial D_3}{\partial y_j^k}, \tag{3.15}$$

$$\frac{\partial D_1}{\partial y_j^k} = -2 \sum_{\substack{i \in K_k \\ a_{ij} \neq @}} (a_{ij} - y_j^k), \tag{3.16}$$

$$\frac{\partial D_2}{\partial y_j^k} = -2(y_j^{k-1} - y_j^k) + 2(y_j^k - y_j^{k+1}), \tag{3.17}$$

$$\frac{\partial D_3}{\partial y_j^k} = -2(2y_j^{k-1} - y_j^{k-2} - y_j^k) + \\ + 4(2y_j^k - y_j^{k-1} - y_j^{k+1}) - 2(2y_j^{k+1} - y_j^k - y_j^{k+2}). \tag{3.18}$$

Подставляя (3.16 – 3.18) в (3.15), для каждого j имеем систему из k линейных уравнений относительно y_j^k:

$$A_{kj}^{-2} y_j^{k-2} + A_{kj}^{-1} y_j^{k-1} + A_{kj}^0 y_j^k + A_{kj}^1 y_j^{k+1} + A_{kj}^2 y_j^{k+2} = B_j^k, \; k=1..m,$$

где A_{kj}^l ($l = -2 \ldots 2$) – пятидиагональная матрица, коэффициенты которой определяются из уравнений (3.15 – 3.18).

Решение может быть получено каким-нибудь численным методом для диагональной матрицы [51].

III.4. SOM

Пусть SOM определяется набором точек (ядер) $Y=\{y_{ij}\}$ ($i,j=1..m$ – будем считать, что сетка квадратная, но для прямоугольной все описанные формулы аналогичны), последовательно расположенных на квадратной сетке и требуется отобразить на ней набор точек данных $X=\{x_i\}$. Введем преобразование $П$, которое каждому вектору $x \in X$ сопоставляет ближайшую к нему точку из Y:

$$x \xrightarrow{П} y_{ij}, \; \|y_{ij} - x\|^2 \to \min, \tag{3.19}$$

каждому ядру y_{ij} сопоставляется его таксон

$$K_{ij} = \left\{ x \in X \middle| x \xrightarrow{П} y_{ij} \right\}. \tag{3.20}$$

Метод построения SOM также, как и SOC, напоминает метод динамических ядер, за исключением добавления дополнительных ограничений на связность и нелинейность.

А так как может быть построено две разновидности SOM – квадратная и гексагональная, то остановимся подробно на каждой из них.

Квадратная SOM

Как и в случае с SOC, минимизируемая величина строится из следующих слагаемых:

1) мера приближения данных:

$$D_1 = \sum_{ij} \sum_{x \in K_{ij}} \left\| x - y_{ij} \right\|^2 ; \tag{3.21}$$

2) мера связности (близкие точки на карте должны переходить в близкие в пространстве данных):

$$D_2 = \sum_{ij} \left[\left\| y_{ij} - y_{i+1,j} \right\|^2 + \left\| y_{ij} - y_{i,j+1} \right\|^2 \right]; \tag{3.22}$$

3) мера нелинейности (или равномерности):

$$D_3 = \sum_{j} \left\| 2y_{ij} - y_{i-1,j} - y_{i+1,j} \right\|^2 + \left\| 2y_{ij} - y_{i,j-1} - y_{i,j+1} \right\|^2 . \tag{3.23}$$

Таким образом, для построения SOC требуется минимизировать функционал:

$$D = \frac{D_1}{|X|} + \lambda \frac{D_2}{m^2} + \mu \frac{D_3}{m^2} \to \min , \tag{3.24}$$

где λ, μ – параметры связности и нелинейности – "модули упругости" (деление на число точек $|X|$ и число ядер m означает нормировку "на одно слагаемое" и позволяет для выборок разной мощности использовать одинаковые способы изменения λ и μ.)

Как и для SOC, процедура построения SOM также является итерационной – то есть задача расщепляется на последовательны поиск ядер – таксонов – ядер … и т.д. И эта процедура также гарантировано сходится.

Гексагональная SOM

С учетом того, что каждый узел SOM в данном случае имеет не 4, а 6 соседей, то минимизируемые функционалы перепишутся в следующем виде:

1) мера приближения данных:

$$D_1 = \sum_{ij} \sum_{x \in K_{ij}} \left\| x - y_{ij} \right\|^2 ; \tag{3.25}$$

2) мера связности (близкие точки на карте должны переходить в близкие в пространстве данных):

$$\begin{aligned} D_2 = & \sum_{ij} \left\| y_{ij} - y_{i,j+1} \right\|^2 + \\ & + \sum_{\substack{ij \\ i-\text{четное}}} \left[\left\| y_{ij} - y_{i+1,j} \right\|^2 + \left\| y_{ij} - y_{i+1,j-1} \right\|^2 \right] + \sum_{\substack{ij \\ i-\text{нечетное}}} \left[\left\| y_{ij} - y_{i+1,j+1} \right\|^2 + \left\| y_{ij} - y_{i+1,j} \right\|^2 \right] \end{aligned} \tag{3.26}$$

3) мера нелинейности (или равномерности):

$$\begin{aligned} D_3 = & \sum_{ij} \left\| 2y_{ij} - y_{i,j-1} - y_{i,j+1} \right\|^2 + \\ & + \sum_{\substack{ij \\ i-\text{четное}}} \left[\left\| 2y_{ij} - y_{i-1,j-1} - y_{i+1,j} \right\|^2 + \left\| 2y_{ij} - y_{i-1,j} - y_{i+1,j-1} \right\|^2 \right] + \\ & + \sum_{\substack{ij \\ i-\text{нечетное}}} \left[\left\| 2y_{ij} - y_{i-1,j} - y_{i+1,j+1} \right\|^2 + \left\| 2y_{ij} - y_{i-1,j+1} - y_{i+1,j} \right\|^2 \right] \end{aligned} \tag{3.27}$$

Алгоритм построения квадратной SOM

Пусть задана таблица (a_{ij}) с пробелами (некоторые a_{ij}=@) и набор ядер $Y=\{y^{kl}\}$ (k,l=0..m–1). Формулы (3.19 – 3.24) перепишутся следующим образом (a_i – i-ая строка таблицы (a_{ij})):

$$a_i \overset{\Pi}{\to} y^{kl},\ \sum_i (a_{ij} - y_j^{kl})^2 \to \min, \tag{3.28}$$

$$K_{kl} = \left\{ a_i \in (a_{ij}) \middle| a_i \overset{\Pi}{\to} y^{kl} \right\}. \tag{3.29}$$

$$D_{1j} = \sum_{k,l=0}^{m-1} \sum_{i \in K_{kl}} \sum_{\substack{j \\ a_{ij} \neq @}} (a_{ij} - y_j^{kl})^2, \tag{3.30}$$

$$D_{2j} = \sum_{k,l=0}^{m-2} \sum_j \left[(y_j^{kl} - y_j^{k+1,l})^2 + (y_j^{kl} - y_j^{k,l+1})^2 \right], \tag{3.31}$$

$$D_{3j} = \sum_{k,l=1}^{m-2} \sum_j \left[(2y_j^{kl} - y_j^{k-1,l} - y_j^{k+1,l})^2 + (2y_j^{kl} - y_j^{k,l-1} - y_j^{k,l+1})^2 \right]. \tag{3.32}$$

Требуется решить следующую задачу:

$$D_j = \frac{D_{1j}}{n} + \lambda \frac{D_{2j}}{m^2} + \mu \frac{D_{3j}}{m^2} \to \min, \tag{3.33}$$

где n – число строк матрицы (a_{ij}).

Значения y_j^{kl}, доставляющие минимум форме (3.33) при заданном разбиении K_{kl}, определяются из системы равенств $\partial D_j / \partial y_j^{kl} = 0$:

$$\frac{\partial D_j}{\partial y_j^{kl}} = \frac{1}{n} \cdot \frac{\partial D_{1j}}{\partial y_j^{kl}} + \frac{\lambda}{m^2} \cdot \frac{\partial D_{2j}}{\partial y_j^{kl}} + \frac{\mu}{m^2} \cdot \frac{\partial D_{3j}}{\partial y_j^{kl}}, \tag{3.34}$$

$$\frac{\partial D_{1j}}{\partial y_j^{kl}} = -2 \sum_{\substack{i \in K_{kl} \\ a_{ij} \neq @}} (a_{ij} - y_j^{kl}), \tag{3.35}$$

$$\frac{\partial D_{2j}}{\partial y_j^{kl}} = -2(y_j^{k-1,l} - y_j^{kl}) + 2(y_j^{kl} - y_j^{k+1,l}) - 2(y_j^{k,l-1} - y_j^{kl}) + 2(y_j^{kl} - y_j^{k,l+1}), \tag{3.36}$$

$$\begin{aligned} \frac{\partial D_{3j}}{\partial y_j^{kl}} = & -2(2y_j^{k-1,l} - y_j^{k-2,l} - y_j^{kl}) + 4(2y_j^{kl} - y_j^{k-1,l} - y_j^{k+1,l}) - \\ & - 2(2y_j^{k+1,l} - y_j^{kl} - y_j^{k+2,l}) - 2(2y_j^{k,l-1} - y_j^{k,l-2} - y_j^{kl}) + \\ & + 4(2y_j^{kl} - y_j^{k,l-1} - y_j^{k,l+1}) - 2(2y_j^{k,l+1} - y_j^{kl} - y_j^{k,l+2}). \end{aligned} \tag{3.37}$$

Для каждого j имеем систему из m^2 линейных уравнений относительно y_j^{kl}:

$$\begin{aligned} & A_j^{k-2,l} y_j^{k-2,l} + A_j^{k-1,l} y_j^{k-1,l} + A_j^{kl} y_j^{kl} + A_j^{k+1,l} y_j^{k+1,l} + A_j^{k+2,l} y_j^{k+2,l} + \\ & + A_j^{k,l-2} y_j^{k,l-2} + A_j^{k,l-1} y_j^{k,l-1} + A_j^{k,l+1} y_j^{k,l+1} + A_j^{k,l+2} y_j^{k,l+2} = B_j^{kl}, \end{aligned}$$

где A_j^{kl} ($k,l = 0 \ldots m$) – коэффициенты при соответствующих y_j^{kl}, которые определяются из уравнений (3.34 – 3.37).

Можно заметить, что матрица имеет диагональный вид, поэтому она с достаточной точностью может быть решена каким-нибудь численным методом, например, оптимизированным для диагональных матриц методом Гаусса, либо методом Гаусса-Зейделя [51].

Алгоритм построения гексагональной SOM

При построении гексагональной SOM претерпевают изменения только функционалы D_2 и D_3:

$$D_{2j} = \sum_{kl} (y_j^{kl} - y_j^{k,l+1})^2 +$$

$$+ \sum_{\substack{kl \\ k-\text{четное}}} \left[(y_j^{kl} - y_j^{k+1,l})^2 + (y_j^{kl} - y_j^{k+1,l-1})^2 \right] + \sum_{\substack{kl \\ k-\text{нечетное}}} \left[(y_j^{kl} - y_j^{k+1,l+1})^2 + (y_j^{kl} - y_j^{k+1,l})^2 \right],$$

$$D_{3j} = \sum_{kl} (2y_j^{kl} - y_j^{k,l-1} - y_j^{k,l+1})^2 +$$

$$+ \sum_{\substack{kl \\ k-\text{четное}}} \left[(2y_j^{kl} - y_j^{k-1,l-1} - y_j^{k+1,l})^2 + (2y_j^{kl} - y_j^{k-1,l} - y_j^{k-1,l-1})^2 \right] +$$

$$+ \sum_{\substack{kl \\ k-\text{нечетное}}} \left[(2y_j^{kl} - y_j^{k-1,l} - y_j^{k+1,l+1})^2 + (2y_j^{kl} - y_j^{k-1,l+1} - y_j^{k+1,l})^2 \right]$$

$$\frac{\partial D_{2j}}{\partial y_j^{kl}} = -2(y_j^{k,l-1} - y_j^{kl}) + 2(y_j^{kl} - y_j^{k,l+1}) +$$

$$+ 2\left[(y_j^{kl} - y_j^{k+1,l}) - (y_j^{k-1,l-1} - y_j^{kl}) + (y_j^{kl} - y_j^{k+1,l-1}) - (y_j^{k-1,l} - y_j^{kl}) \right]_{k-\text{четное}} +$$

$$+ 2\left[(y_j^{kl} - y_j^{k+1,l+1}) - (y_j^{k-1,l} - y_j^{kl}) + (y_j^{kl} - y_j^{k+1,l}) - (y_j^{k-1,l+1} - y_j^{kl}) \right]_{k-\text{нечетное}},$$

$$\frac{\partial D_3}{\partial y_j^{kl}} = -2(2y_j^{k,l-1} - y_j^{k,l-2} - y_j^{kl}) + 4(2y_j^{kl} - y_j^{k,l-1} - y_j^{k,l+1}) - 2(2y_j^{k,l+1} - y_j^{kl} - y_j^{k,l+2}) +$$

$$+ 2\left[2(2y_j^{kl} - y_j^{k-1,l-1} - y_j^{k+1,l}) - (2y_j^{k-1,l-1} - y_j^{k-2,l-1} - y_j^{kl}) - (2y_j^{k+1,l} - y_j^{kl} - y_j^{k+2,l+1}) \right]_{k-\text{четное}} +$$

$$+ 2\left[2(2y_j^{kl} - y_j^{k-1,l} - y_j^{k+1,l-1}) - (2y_j^{k-1,l} - y_j^{k-2,l+1} - y_j^{kl}) - (2y_j^{k+1,l-1} - y_j^{kl} - y_j^{k+2,l-1}) \right]_{k-\text{четное}} +$$

$$+ 2\left[2(2y_j^{kl} - y_j^{k-1,l} - y_j^{k+1,l+1}) - (2y_j^{k-1,l} - y_j^{k-2,l-1} - y_j^{kl}) - (2y_j^{k+1,l+1} - y_j^{kl} - y_j^{k+2,l+1}) \right]_{k-\text{нечетное}} +$$

$$+ 2\left[2(2y_j^{kl} - y_j^{k-1,l+1} - y_j^{k+1,l}) - (2y_j^{k-1,l+1} - y_j^{k-2,l+1} - y_j^{kl}) - (2y_j^{k+1,l} - y_j^{kl} - y_j^{k+2,l-1}) \right]_{k-\text{нечетное}}.$$

Подставляя полученные частные производные в общий функционал и группируя коэффициенты при одинаковых y_j^{kl}, получаем систему из m^2 линейных уравнений относительно y_j^{kl}. Аналогично предыдущему пункту, матрица имеет диагональный вид, поэтому она с достаточной точностью решается численным методом Гаусса, либо методом Гаусса-Зейделя.

III.5. ПРОБЛЕМА ЛОКАЛЬНОГО МИНИМУМА

В отличии от линейного и квазилинейного случаев данная задача минимизации функционала (3.8) не является выпуклой, и поэтому возникают трудности, связанные с попаданием в области локального минимума. Это может привести к неудовлетворительному решению задачи.

Хотя для решения данной проблемы существует множество методов, мы же остановимся на многосеточном методе и методе "отжига".

Метод отжига

Так называемый метод "отжига" состоит в том, что к исходной минимизируемой функции произвольным образом добавляется некоторая функция, в результате чего можно добиться сглаживания локальных минимумов, благодаря чему происходит выход процесса минимизации из локального минимума. Затем добавочная функция устремляется к нулю.

Для нашей задачи предполагается использовать следующий метод "отжига":

За счет увеличений в (3.8) коэффициентов λ, μ система ставится в очень жесткие рамки. Далее они постепенно ослабляются (происходит уменьшение соответствующих коэффициентов). В частности, при больших μ получим отрезки, близкие к первой главной компоненте.

Здесь существует два способа:

1) сначала система стягивается в точку (λ – очень велико), а затем за счет ослабления λ растягивается до нужного уровня;

2) сначала система делается очень большой (λ – очень мало), а затем за счет увеличения λ сжимается до нужного уровня.

Аналогичная процедура может проводиться и с коэффициентом μ.

Многосеточный метод

Опишем обычные многосеточные алгоритмы в абстрактной алгебраической форме [11].

Предположим, что есть последовательность пространств ограниченной размерности

$$M_0, M_1,\ldots,M_k$$

с внутренними произведениями, описываемыми как $(\cdot,\cdot)_i$ для каждого M_i. Предположим также, что имеются операторы "интерполяции"

$$I_i: M_i \to M_{i+1},\ i=0,\ldots,k-1;$$

операторы "сужения"

$$R_i: M_i \to M_{i-1},\ i=1,\ldots,k;$$

и операторы обращения

$$L_i: M_i \to M_i,\ i=0,\ldots,k.$$

Основная цель этих алгоритмов – решить следующую задачу в M_k: дано $f_k \in M_k$, найти $v_k \in M_k$ такое, что

$$L_k v_k = f_k. \qquad (3.38)$$

Особенность многосеточных алгоритмов состоит в необходимости решения вспомогательных задач на нижних уровнях. Поэтому, сформулируем MG_i-алгоритм для приближенного решения задачи:

$$L_i v_i = f_i, \text{ где } f_i \in M_i \qquad (3.39)$$

для любого уровня $i \in [0,k]$. Начнем с некоторого начального приближения w_0 и получим в результате другое приближение (как ожидается, более близкое к v_i), описываемое как $w_1 = MG_i(w_0, f_i)$.

***MG_i*-алгоритм.** Если $i=0$, то

$$w_1 = MG_0(w_0, f_0) = L_0^{-1} f_0,$$

начальное приближение w_0 больше не имеет значения и MG_0 – закончен. Для $i>0$ алгоритм определен как следующее:

A_1 (пре-сглаживание). Пусть $u_0 = w_0$ и определим u_{m_1} следующим образом

$$u_{l+1} = u_l - \Im_{i,l+1}(L_i u_l - f_i),\ l=0,1,\ldots,m_1-1;$$

A_2 (сужение). Пусть

$$g_{i-1} = R_i(L_i u_{m_1} - f_i);$$

A_3 (крупно-сеточное решение). Пусть $\tilde{w}_0 = 0 \in M_{i-1}$ и повторим MG_{i-1}-алгоритм γ раз:

$$\tilde{w}_s = MG_{i-1}(\tilde{w}_{s-1}, g_{i-1}),\ s=1,\ldots,\gamma;$$

A_4 (коррекция). Пусть

$$y_0 = u_{m_1} - I_{i-1}\tilde{w}_\gamma;$$

A_1 (пост-сглаживание). Определим y_{m_2} следующим образом

$$y_{l+1} = y_l - \Im_{i,l+m_1+1}(L_i y_l - f_i),\ l=0,1,\ldots,m_2-1.$$

Окончательно получим

$$w_1 = MG_i(w_0, f_i) = y_{m_2}$$

как результат MG_i-алгоритма.

$\Im_{i,l}$ ($l=1,\ldots,m_1+m_2$) – некоторый линейный оператор. Его структуру можно посмотреть в [11].

Три шага A_2–A_4 вместе взятые обычно называются крупно-сеточной коррекцией.

Полный многосеточный *FMG*-алгоритм для решения задачи (3.39), включающий верхний уровень (3.38) на заключительном этапе:

***FMG*-алгоритм.**

1. Пусть $\tilde{u}_0 = L_0^{-1} f_0$.
2. Для $i=1,2,\ldots,k$ делается следующее:
   2.1. Установить $\tilde{u}_i = I_{i-1}\tilde{u}_{i-1}$;
   2.2. Повторить MG_i-алгоритм t раз:
   $$\tilde{u}_i := MG_i(\tilde{u}_i, f_i).$$

На k-ом шаге алгоритма получается окончательный результат $\tilde{u}_k$, который является некоторым приближением решения v_k задачи (3.38) для уровня k.

Многосеточные или, другими словами, иерархические методы основаны на том, что в процессе вычисления шаг сетки каким-либо образом изменяется. Например, в каскадном методе изначально берется достаточно большой шаг, а затем он постепенно уменьшается. Методы же V-цикла и W-цикла имеют более сложные законы изменения шага сетки.

В данной задаче построения SOC и SOM предлагается использовать каскадный метод, при котором, как уже написано выше, изначально берется сетка с очень большим шагом. На этой сетке строится интерполирующая функция. Получается начальное и весьма грубое приближение. Далее сетка подвергается постепенному дроблению, то есть размер сетки шаг за шагом уменьшается. При этом происходит постепенное исправление начальной аппроксимирующей функции. Сетку можно дробить не равномерно, а в зависимости от требуемой на разных участках точности – то есть для всей области данных можно использовать сетку с достаточно крупным шагом, а в некоторых интересующих местах сетка дробится до достижения требуемой точности.

В результате для имеющейся задачи построения SOC многосеточный метод принимает следующую форму:

Изначально ломаная состоит из двух точек. После минимизации функционала (3.8) получим главную прямую (аналог описанных ранее линейных моделей). Затем этот отрезок делится на две части – т.е. добавляется новый узел. Опять минимизируется функционал (3.8). Далее каждый отрезок ломаной дробится на две части, минимизируется функционал и т.д. до достижения требуемой точности.

Вариант метода с делением только тех отрезков ломаной, которые превышают заданную для данного шага длину, оказывается более экономичным.

III.6. Сглаживание

Проблема сглаживания

Полученная ломаная $\{y_j\}$ (j=1..m) может быть сглажена различными способами, например, с использованием кубических сплайнов. Здесь, однако, возникают определенные трудности, связанные с нахождением проекций данных на сглаженную кривую, т.к. для этого требуется решать алгебраические уравнения 5-ой степени.

Для сглаживания возможно использовать также сглаживающий фильтр, но он также не дает возможности для нахождения проекций данных на ломаную.

Поэтому возникает необходимость в создании специального проектора данных на SOC или SOM.

Пусть M – самоорганизующееся многообразие (SOC или SOM). Требуется построить отображение P: $\boldsymbol{x}\in X \rightarrow \boldsymbol{r}\in M$, которое будем называть правилом проектирования или *проектором*. Для рассматриваемых задач желательно, чтобы проектор обладал следующими качествами:

а) проектор должен сохранять отношения соседства, то есть желательно, чтобы близким точкам в R^n соответствовали близкие точки на M (по крайней мере, в некоторой окрестности M);

б) проектор, по крайней мере, в некоторой окрестности M должен быть однозначным;

в) проектор должен быть по возможности непрерывным, чтобы плавным изменениям состояния системы в X соответствовали непрерывные изменения положения образа в M;

г) проекция должна не слишком сильно отличаться от ближайшей вершины ломаной или сетки.

Создателем SOM Кохоненом был применен самый очевидный и в некотором смысле естественный вариант проектирования – кусочно-постоянный. При этом каждой точке из X сопоставляется та вершина сетки, которая является ближайшей к этой точке.

Достоинством такого проектирования являются его логическая прозрачность и простота, очевидный его недостаток – разрывность, что не позволяет наиболее полно смоделировать многообразием M данные X.

Для предложенной технологии построения самоорганизующихся многообразий такой кусочно-постоянный способ проектирования малопригоден, поскольку, в отличие от Кохоненовских SOM, каждый из узлов M, вообще говоря, не располагается в центре локального сгущения точек данных. Напротив, M представляет собой более-менее равномерно натянутую на данные ломаную/сетку, и поэтому существенная часть данных может быть расположена в промежутках между узлами. В этой работе предлагается вариант кусочно-линейного непрерывного проектирования многомерных данных как на SOC, так и на SOM.

Рассмотрим для начала вариант кусочно-линейного непрерывного проектора на SOC. Концам ломаной при этом соответствуют значения (–1) и 1 соответственно, а проекции узлов на отрезок [–1,1] определяются узлами равномерной (учитывается только число узлов ломаной) или неравномерной (учитываются также расстояния между узлами ломаной) сетки.

Окончательное сглаживание производится с использованием формул Карлемана. Ломаная $\{y_j\}$ (j=1..m) при этом заменяет главную компоненту классического метода, а процесс сглаживания аналогичен построению квазилинейной модели.

Кусочно-линейная проекция на ломаную

Требуется построить отображение линейных многообразий $a=a_t \in (a_{ij})$ на ломаную, определяемую набором вершин $\{y^k\}$ (k=0…m–1), то есть каждому x сопоставить некоторое t, определяющее его проекцию на ломаную.

Пусть y_k – ближайшая к a вершина ломаной. Тогда возможны два варианта – либо эта вершина является крайней вершиной ломаной, либо она является внутренней. Процедуры проектирования для каждого из этих случаев различаются, поэтому рассмотрим на каждом из них отдельно. Но перед этим введем дополнительные обозначения:

Полученную ближайшую вершину y_k будем обозначать как y_0, а соседние с ней две вершины – y_1 и y_2 соответственно (если y_k – крайняя, то берем только одну соседнюю – y_1). И в дальнейшем координаты всех точек y_0, y_1, y_2 и a будем считать относительно точки y_0, т.е. их координатами будут:

$$y_1 = y_1 - y_0,\ y_2 = y_2 - y_0,\ a = a - y_0,\ y_0 = 0. \qquad (3.40)$$

А теперь вернемся к двум вариантам расположения ближайшей вершины.

1. Пусть y_k – крайняя вершина ломаной, т.е. k=0 или k=m–1 (рис. 3.2).

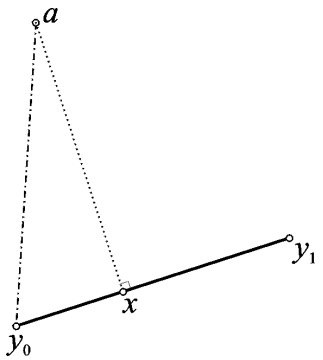

Рис. 3.2.

В этом случае проводится обычная ортогональная проекция точки a на вектор y_1 (см. вторую главу). Т.е. требуется найти такую точку x, лежащую на y_1, что вектор, соединяющий a и x, будет ортогонален y_1.

Так как x лежит на y_1, то ее координата будет равна x=x_1y_1, а условие ортогональности запишется в виде скалярного произведения. Таким образом имеем следующую систему:

$$\begin{cases} x = x_1 y_1, \\ (x - a, y_1)_a = 0. \end{cases}$$

Подставив первую строчку системы во вторую, получим:

$$(x_1 y_1 - a, y_1)_a = 0, \Rightarrow x_1 (y_1, y_1)_a = (a, y_1)_a, \Rightarrow$$

$$x_1 = \frac{(a, y_1)_a}{\|y_1\|_a^2}. \qquad (3.41)$$

Полученное значение x_1 определяет проекцию на одно ребро ломаной, но чтобы получить окончательное значение проекции на ломаную, его надо подвергнуть обработке. Поэтому, для определенности, если k=0, то значение x_1 берется "как есть", а для k=m–1 оно берется с противоположным знаком.

2. Пусть y_k – внутренняя вершина ломаной, т.е. 0<k<m–1 (рис. 3.3).

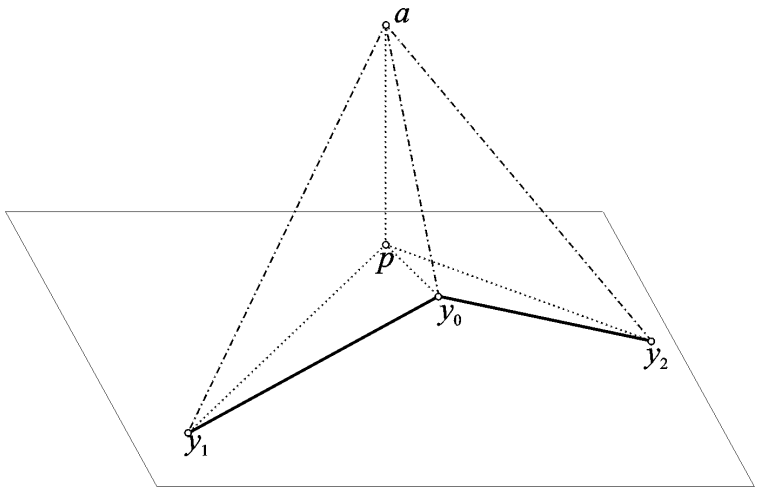

Рис. 3.3.

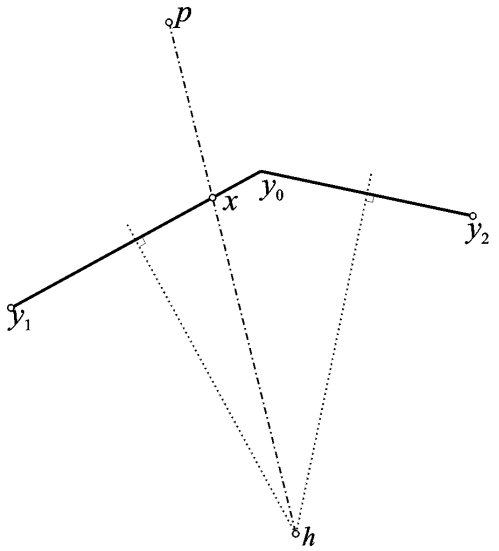

Рис. 3.4.

Из *n*-мерной задача легко сводится к двумерной за счет проецирования точки *a* на плоскость, образованную двумя ближайшими ребрами ломаной (рис. 3.3). При этом легко доказать, что отношения близости между полученной точкой *p* и вершинами ломаной y_0, y_1 и y_2 не изменятся.

Таким образом, первая задача заключается в нахождении такой точки *p*, на плоскости, образованной векторами y_1 и y_2, то вектор, соединяющий *p* и *a*, будет ортогонален обоим векторам y_1 и y_2.

Так как *p* лежит на плоскости, образованной y_1 и y_2, то ее координата будет равна $p=p_1y_1+p_2y_2$, а условие ортогональности запишется в виде скалярного произведения. Таким образом для нахождения координат точки *p* требуется решить следующую систему уравнений:

$$\begin{cases} p = p_1 y_1 + p_2 y_2; \\ (a-p, y_1)_a = 0; \\ (a-p, y_2)_a = 0, \end{cases} \Rightarrow \begin{cases} p_1 (y_1, y_1)_a + p_2 (y_1, y_2)_a = (a, y_1)_a; \\ p_1 (y_1, y_2)_a + p_2 (y_2, y_2)_a = (a, y_2)_a. \end{cases} \quad (3.42)$$

Решение может быть получено любым численным способом. Заметим, что система совместна, если векторы y_1 и y_2 не лежат на одной прямой. В противном случае проводится описанная выше ортогональная проекция на одно из ребер ломаной.

Далее требуется спроецировать полученную точку *p* на ребро ломаной. Для определенности сначала найдем значение проекции на ребро y_1, а затем, по аналогии, на y_2.

Определим проекцию точки *p* на ребро ломаной (рис. 3.4) как точку пересечения ребра ломаной и прямой, соединяющей *p* и точку пересечения серединных перпендикуляров к ребрам ломаной *h*.

Так как искомая точка *h* лежит в плоскости векторов y_1 и y_2, а векторы, соединяющие ее с серединами векторов y_1 и y_2, соответственно ортогональны этим векторам, то точку *h* легко находим из системы уравнений:

$$\begin{cases} h = h_1 y_1 + h_2 y_2, \\ (h - 0.5 y_1, y_1) = 0, \\ (h - 0.5 y_2, y_2) = 0. \end{cases}$$

Подставив первое уравнение системы в остальные два, получим следующую систему линейных уравнений с неизвестными h_1 и h_2:

$$\begin{cases} h_1(y_1, y_1) + h_2(y_1, y_2) = 0.5(y_1, y_1); \\ h_1(y_1, y_2) + h_2(y_2, y_2) = 0.5(y_2, y_2). \end{cases}$$

Она также легко решается численными методами. Будем считать, что векторы y_1 и y_2 не коллинеарные, поэтому система имеет единственное решение.

Координаты проекции x точки p находится из условий, что искомая точка одновременно лежит на прямой, образованной вектором h–p, и на прямой, образованной вектором y_1. Поэтому координаты проекции x на ребро y_1 определяются из системы уравнений:

$$\begin{cases} x - p = \alpha(h - p), \\ x = x_1 y_1. \end{cases}$$

Поставляя первое уравнение во второе, получим:

$$x_1 y_1 = \alpha h_1 y_1 + \alpha h_2 y_2 + p_1 y_1 (1 - \alpha) + p_2 y_2 (1 - \alpha) = 0,$$

и после группировки слагаемых при y_1 и y_2 получим следующее уравнение:

$$y_1(\alpha h_1 + p_1(1 - \alpha) - x_1) + y_2(\alpha h_2 + p_2(1 - \alpha)) = 0,$$

А так как векторы y_1 и y_2 линейно независимы, то последнее равенство выполняется только в том случае, когда коэффициенты при y_1 и y_2 равны нолю:

$$\begin{cases} (h_1 - p_1)\alpha - x_1 = -p_1, \\ (h_2 - p_2)\alpha = -p_2. \end{cases}$$

В результате, проекция на ребро y_1 равна:

$$x_1 = (h_1 - p_1)\frac{p_2}{(p_2 - h_2)} + p_1.$$

Аналогично, проекция на ребро y_2 равна:

$$x_2 = (h_2 - p_2)\frac{p_1}{(p_1 - h_1)} + p_2.$$

Здесь возможна неоднозначность, когда точка пересечения серединных перпендикуляров h и проецируемая точка p совпадают. В этом случае в качестве значения проекции берем середину того ребра ломаной, которое ближе к точке p. Если же расстояния равны (что возможно лишь в том случае, если длины ребер совпадают), то проекция выбирается случайным образом.

Учитывая, что проекция должна лежать на ребре, окончательно берем то из значений x_1 или x_2, которое лежит в отрезке [0, 1]. Для определенности, значение x_1 берется с противоположным знаком, а x_2 без изменений, т.е.:

$$x_1 = \begin{cases} -x_1, \text{ если } x_1 \in [0,1], \\ x_2, \text{ если } x_2 \in [0,1]. \end{cases}$$

В итоге проекция на ломаную в зависимости от типа сетки может определяться двумя способами.

1. Для равномерной сетки.

$$x = -1 + 2\frac{k + x_1}{m - 1}.$$

2. Для неравномерной сетки (с учетом расстояния между узлами).

Пусть l_i – длина i-го ребра ломаной, т.е.:

$$l_i = \|y_{i+1} - y_i\|,\ i=0..m-2.$$

Тогда узлу y_k ломаной будет соответствовать проекция:

$$x(y_k) = -1 + 2\sum_{i=0}^{k-1} l_i \Big/ \sum_{i=0}^{m-2} l_i .$$

И окончательным значением проекции будет (для определенности будем считать, что $l_{-1}=l_0$ и $l_{m-1}=l_{m-2}$):

$$x = -1 + 2 \cdot \begin{cases} \left(\sum_{i=0}^{k-1} l_i + \frac{l_k}{2} x_1 \right) \Big/ \sum_{i=0}^{m-2} l_i , \text{ если } x_1 \ge 0, \\ \left(\sum_{i=0}^{k-1} l_i - \frac{l_{k-1}}{2} x_1 \right) \Big/ \sum_{i=0}^{m-2} l_i , \text{ если } x_1 < 0. \end{cases}$$

Напомним, что значение $x = \pm 1$ соответствуют концам ломаной.

Остается уточнить, что в формулах (3.41) и (3.42) используется обобщенное на случай данных с пробелами скалярное произведение и норма, которые были определены в предыдущей главе. Таким образом, построенный кусочно-линейный проектор данных на ломаную работает как с полными данными, так и с данными, которые содержат пробелы. Действительно, в (3.41) производится проектирование на прямую, а в (3.42) – на плоскость. А это есть ни что иное, как проектирование данных на одномерные и двумерные линейные многообразия, которое и было рассмотрено во второй главе.

В принципе, возможны и другие варианты проецирования данных с пробелами. К примеру, отсутствующие компоненты вектора данных могут быть восстановлены за счет ближайшей вершины, а уж потом уже "комплектный" вектор проецируется на ломаную.

Кусочно-линейная проекция на квадратную и гексагональную сетки

Для квадратной сетки предварительно проводится триангуляция, что переводит ее в гексагональную; таким образом, процесс проецирования для квадратной и гексагональной сетки один и тот же (за исключением произвола в выборе способа триангуляции квадратной сетки).

Проецирование производится либо в ближайшую вершину, либо на ближайшее ребро, либо на ближайшую грань – аналогично одномерному случаю.

III.7. Механическая интерпретация

В описанных линейных и квазилинейных моделях содержится очень сильное ограничение – балка была либо жесткой, либо она гибкая, но все равно проложена вдоль прямой, причем это оказывается существенным, например, в том случае, когда данные расположены не вдоль какой-либо прямой, а вдоль окружности (или хотя бы вдоль сильно изогнутой дуги).

Чтобы избежать этого, балку следует сделать упругой (определяется не прямой, а какой-либо кривой). Но тут возникают сложности в виде определения расстояния от точки до кривой в пространстве (тем более, что вместо точки может выступать линейное многообразие).

При использовании описанного выше метода, близкого методу самоорганизующихся карт Кохонена, искомая упругая балка (кривая – SOC) представляется в виде ломаной, узлы которой свободно соединены с данными (рис. 3.5).

Аналогично жесткому случаю, система через несколько итераций придет в равновесие. Причем их число будет конечно, т.к. на каждом шаге суммарная энергия растяжения пружинок уменьшается, а число возможных состояний (способов крепления узлов ломаной пружинками к данным) конечно.

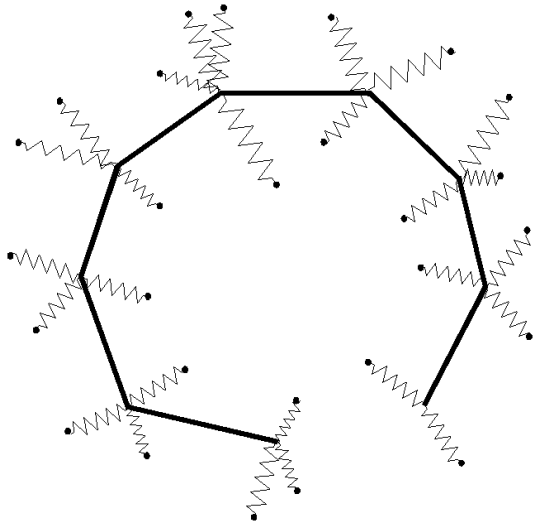

Рис. 3.5.

Введенные модули упругости представляют собой соответственно степень притяжения узлов ломаной друг к другу и степень сопротивления изгибу в узлах.

Глава IV. Экспериментальные результаты

Введение

В первом параграфе приводятся результаты использования метода при факторном и кластерном анализе административных территорий Красноярского края по показателям здоровья и здравоохранения.

Результаты тестирования метода по таблице выборов президентов США приводятся во втором параграфе.

Третий параграф посвящен использованию программных продуктов FAMaster и ModelAnalyzer в задачах гелиофизики.

Использование метода моделирования данных с пробелами многообразиями малой размерности при обнаружении факторов, влияющих на течение и прогноз заболевания у больных со сложными нарушениями ритма и проводимости сердца, описано в четвертом параграфе.

IV.1. Факторный и кластерный анализ административных территорий Красноярского края по показателям здоровья и здравоохранения

Эффективное централизованное управление здравоохранением такого крупного региона, как Красноярский край, представляет собой трудную задачу, в решении которой необходимо учитывать множество различающихся факторов, используя дифференцированный подход к планированию мероприятий и затрат.

Для эффективного управления здравоохранением актуальной является задача типизации отдельных регионов края с учетом взаимосвязей между различными факторами, с большей или меньшей силой оказывающими влияние на планирование и результат управленческих решений. Имеющиеся статистические данные, безусловно, помогают в принятии управленческих решений, однако обилие показателей далеко не всегда позволяет осмыслить ситуацию в целом.

Целью данного исследования стала типизация и выделение относительно похожих групп в системе из 49 регионов (административных территорий) Красноярского края с помощью кластерного анализа.

В расчетах использовались восходящая и нисходящая иерархические классификации по методу полной связи (дальнего соседа), а так же разбиение на классы с использованием метода динамических ядер [32]. Для анализа полученных разбиений использовались методы итерационного моделирования данных линейными и самоорганизующимися многообразиями малой размерности [14]. Все расчеты проводились с помощью программы "Model Analyzer 2.0", разработанной в рамках данной работы.

В системе здравоохранения края устанавливается деление всей территории края на 49 административных единиц (7 городов и 42 района). Для их кластеризации были использованы относительные показатели за 1999 год,

группирующиеся в 2 основных блока: 1) "Здоровье населения" (общая и первичная заболеваемость по возрастным группам и по классам заболеваний – 849 показателей); 2) "Кадры" (обеспеченность врачами разных специальностей) – 100 показателей). Кроме того, использовался параметр "Затраты на одного жителя в системе здравоохранения". Таким образом, общее число показателей составило 950.

Для анализа данных первого и второго блоков были использованы методы итерационного моделирования данных линейными и самоорганизующимися многообразиями малой размерности [14]. Цель применения данных методов заключалась в упорядочивании объектов (регионов) по главной компоненте в пространстве признаков каждого блока. Это позволило отсортировать регионы на условной шкале “хорошо – плохо” по каждому блоку данных. Согласно расположению регионов в пространстве признаков вдоль главной компоненты регионы были упорядочены отдельно по 1 и 2 блокам. Таким образом, для каждого региона были получены значения на 3 шкалах: 1) уровень здоровья населения; 2) обеспеченность кадрами; 3) затраты на здравоохранение в расчете на одного жителя (эти данные имелись исходно).

Наибольший интерес представляет соотношение затрат на одного жителя и уровнем здоровья населения (рис. 4.1), что, по нашему мнению, достаточно четко может характеризовать эффективность управления здравоохранением региона.

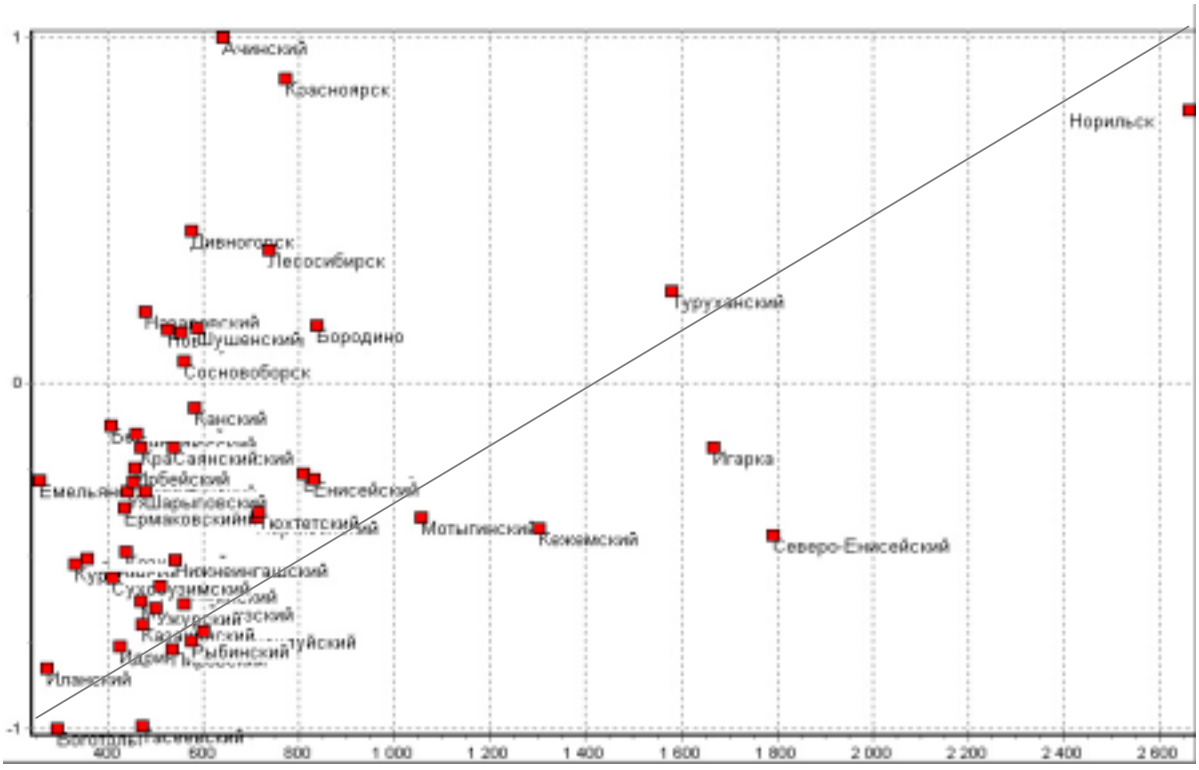

Рис. 4.1. Взаимосвязь между затратами на 1 жителя (ось X) и здоровьем населения (ось Y).

По всем трем показателям была проведена кластеризация регионов. Оптимальным стало разделение регионов на 5 кластеров (таб. 4.1 и рис. 4.2).

Таблица 4.1. Кластеризация регионов по трем показателям (совокупный показатель "Здоровье", совокупный показатель "Кадры", показатель "Затраты на 1 жителя").

№	Кол-во объектов	Объекты
1	10	г. Бородино, г. Дивногорск, г. Лесосибирск, г. Сосновоборск, Енисейский, Канский, Минусинский, Назаровский, Новоселовский, Шушенский р-ны
2	5	г. Игарка, Кежемский, Мотыгинский, Северо-Енисейский, Туруханский р-ны.
3	2	г. Красноярск, Ачинский р-н.
4	1	г. Норильск.
5	31	Абанский, Балахтинский, Березовский, Бирилюсский, Боготольский, Богучанский, Большемуртинский, Большеулуйский, Дзержинский, Емельяновский, Ермаковский, Идринский, Иланский, Ирбейский, Казачинский, Каратузский, Козульский, Краснотуранский, Курагинский, Манский, Нижнеингашский, Партизанский, Пировский, Рыбинский, Саянский, Сухобузимский, Тасеевский, Тюхтетский, Ужурский, Уярский, Шарыповский р-ны

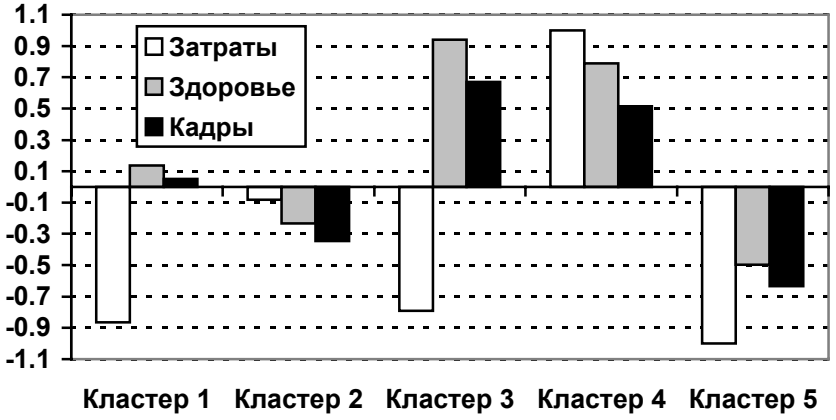

Рис. 4.2. Значения кластерных средних по совокупным показателям ("Здоровье" и "Кадры") и по показателю затрат на 1 жителя. Для удобства восприятия диаграммы значения показателя «Затраты на 1 жителя» нормированы на диапазон [-1..1].

Выводы:

1. Кластеризация регионов с учетом главных компонент позволила выделить 5 кластеров, четко различающихся по соотношению затрат, обеспеченности кадрами и здоровья населения.
2. Корреляционный анализ совокупных показателей здоровья и обеспеченности кадрами с показателем затрат на 1 жителя выявил гораздо большую взаимосвязь здоровья населения с обеспеченностью кадрами

(коэффициент корреляции 0.75) по сравнению с уровнем затрат (коэффициент корреляции 0.38).

3. Полученные результаты позволяют планировать дальнейшее изучение типичных групп регионов с целью более детального выяснения причин, приведших к наблюдающейся ситуации и прицельно искать методы повышения эффективности управления системой здравоохранения в Красноярском крае.
4. Использованная в работе технология, включающая комбинацию двух математических методов, может быть применена для анализа систем здравоохранения в крупных регионах России, а также, возможно, для анализа функционирования различных объектов системы здравоохранения (ЛПУ, город).

IV.2. Таблица результатов выборов президентов США

Проиллюстрируем процесс моделирования данных с пробелами на основе таблицы результатов выборов президентов США, которая содержит 31-у предвыборную ситуацию (с 1860 по 1980 гг.), а также одну тестовую ситуацию (1992 г.). Для каждых выборов в таблице содержатся данные по 12-ти бинарным признакам [77], которые перечислены ниже.

1. Правящая партия была у власти более одного срока? (*More1*)
2. Правящая партия получила более 50% голосов на прошлых выборах? (*More50*)
3. В год выборов была активна третья партия? (*Third*)
4. Была серьезная конкуренция при выдвижении кандидата правящей партии? (*Conc*)
5. Кандидат от правящей партии был президентом в год выборов? (*Prez*)
6. Был ли год выборов временем спада или депрессии? (*Depr*)
7. Был ли рост среднего национального валового продукта на душу населения более 2.1%? (*Val2.1*)
8. Произвел ли правящий президент существенные изменения в политике? (*Chan*)
9. Во время правления были существенные социальные волнения? (*Wave*)
10. Администрация правящей партии виновна в серьезной ошибке или скандале? (*Mist*)
11. Кандидат от правящей партии национальный герой? (*R.Hero*)
12. Кандидат от оппозиционной партии национальный герой? (*O.Hero*)

Также в таблице содержится информация о результатах выборов (победе правящей или оппозиционной партии). Значения входных бинарных признаков равны 0 (ответ "Нет") и 1 (ответ "Да"). Значение выходного признака равно 1 (победа правящей партии) и 2 (победа оппозиции).

Построенные по этой таблице модели уверенно предсказывали результаты вторых выборов Рейгана, победу Буша над Дукакисом, обе победы Клинтона [66]. Причем для уверенного разделения на два класса достаточно одной главной кривой. Проиллюстрируем это на следующем примере.

Построим первые две главные кривые (на основе SOC) и заметим, что они образуют пространство, в котором каждая точка имеет свои координаты (x и y). Используя эти координаты, каждую точку можно поместить на плоскость (с обычными декартовыми координатами) – получаем визуализацию исходного множества данных (рис. 4.3).

На этом рисунке квадратики соответствуют победе правящей партии, а кружочки – победе оппозиции. Треугольник соответствует пропущенному значению (победа оппозиции). Видно, что исходного облако данных четко разбивается на две части, одна из которых соответствует победе правящей партии, а другая – победе оппозиции. Поэтому для уверенного предсказания достаточно одного нелинейного фактора (в линейном случае это не так).

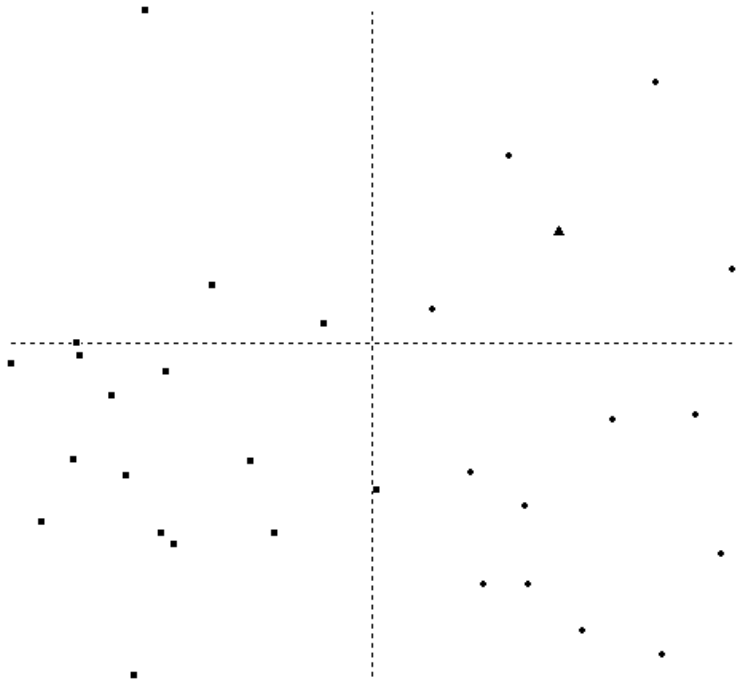

Рис. 4.3. Результаты выборов в пространстве первых двух главных кривых (SOC). Квадратики – победа правящей партии, кружочки – победа оппозиции. Треугольник – отсутствующее значение.

Аналогичный результат получается и другим способом. Если закрыть известные значения результатов выборов и спрогнозировать их значения исходя из модели, то для каждой строчки таблицы получим некоторую ошибку предсказания. Для первой главной кривой распределение этой ошибки представлено в таблице 4.2, из которой видно, что ошибка предсказания не превышает 0.18, что более чем достаточно для уверенного прогноза.

Таблица 4.2.

Интервал	Частота	Проценты
[0.0000, 0.0342)	28 из 31	90.32 %
[0.0342, 0.0683)	1 из 31	3.23 %
[0.0683, 0.1025)	1 из 31	3.23 %
[0.1025, 0.1367)	0 из 31	0.00 %
[0.1367, 0.1708]	1 из 31	3.23 %

Следующим этапом сравнивались степени приближения (в %) исходных данных в зависимости от числа используемых факторов и типов моделей. Средние результаты представлены в таблице 4.3.

Таблица 4.3.

№	Признак	Степень приближения (%) в зависимости от числа факторов								
		Линейная модель			Квазилинейная			SOC		
		1	4	10	1	4	10	1	4	10
1.	Ответ	69.22	69.96	81.78	92.27	92.85	96.72	97.82	98.43	99.50
2.	*More1*	11.88	59.98	77.36	25.37	63.75	95.91	53.49	80.81	96.85
3.	*More50*	9.07	61.10	79.43	14.99	73.69	95.18	30.77	75.89	95.12
4.	*Third*	29.66	44.89	91.56	31.73	66.97	97.45	32.94	76.83	96.93
5.	*Conc*	62.30	63.28	77.84	69.51	77.12	90.24	72.63	78.86	95.42
6.	*Prez*	31.72	59.68	80.27	45.58	68.01	93.03	56.08	74.85	95.18
7.	*Depr*	32.17	52.43	93.38	37.95	71.08	95.56	58.86	80.53	95.62
8.	*Val2_1*	4.12	37.67	94.19	6.27	69.22	96.53	28.80	72.23	95.44
9.	*Changes*	2.33	49.87	86.19	16.81	61.15	94.95	13.01	72.01	93.77
10.	*Wave*	25.13	62.33	80.34	33.18	66.82	95.65	32.68	63.96	96.71
11.	*Mist*	50.34	61.05	86.17	64.83	70.52	97.55	60.80	81.35	96.90
12.	*R_Hero*	33.35	48.12	90.69	54.86	66.27	97.52	27.30	83.67	95.76
13.	*O_Hero*	36.55	50.07	92.03	45.69	68.41	97.55	52.22	76.42	96.17

Если ошибка в вычисленном значении признака меньше 50%, то это означает, что его точное значение (признаки качественные, поэтому при ошибке, меньшей 50%, знак предсказания определяет точное значение).

Для удовлетворительного предсказания линейными моделями достаточно 4-факторов, что говорит о том, что эта задача четырехфакторная (в обычном понимании этого слова). Квазилинейным же моделям, а особенно моделям на основе SOC для удовлетворительного предсказания, как уже отмечалось выше, достаточно всего одного фактора.

Далее проводилось тестирования полученных наборов факторов, которое можно представить в виде двух основных процедур.

1. *Тестирование факторов.* По полной таблице строилась модель, далее в таблицу случайным образом вносились пропуски, а затем запускалась процедура заполнения пропусков. В результате сравнивались получившиеся значения с исходными.

Тестирование показало удовлетворительное заполнения пропусков линейными и квазилинейными моделями при их количестве до 25% от общего числа исходных данных. Тот же показатель для моделей на основе SOC равен 50%. То есть из таблицы можно выкинуть каждое второе данное и все равно их можно будет с удовлетворительной точностью восстановить. Средние значения числа правильно предсказанных примеров для каждого из признаков представлены в таблице 4.4.

Таблица 4.4.

№	Признак	Число правильно предсказанных	Процент
1.	Ответ	11 из 14	78.6 %
2.	*More1*	10 из 17	58.8 %
3.	*More50*	13 из 16	81.2 %
4.	*Third*	9 из 16	56.2 %
5.	*Conc*	11 из 13	84.6 %
6.	*Prez*	9 из 12	75 %
7.	*Depr*	14 из 18	77.7 %
8.	*Val2_1*	12 из 21	57.1 %
9.	*Changes*	11 из 17	64.7 %
10.	*Wave*	10 из 17	58.8 %
11.	*Mist*	11 из 15	73.3 %
12.	*R_Hero*	12 из 16	75 %
13.	*O_Hero*	12 из 15	80 %

2. *Тестирование метода*. В таблицу случайным образом вносились пропуски, далее по "дырявой" таблице строилась модель, а затем запускалась процедура заполнения пропусков. В результате сравнивались получившиеся значения с исходными.

Удовлетворительное заполнение наблюдалось при количестве пробелов до 10% от общего числа исходных данных.

IV.3. ВЕРИФИКАЦИЯ СВЯЗЕЙ МЕЖДУ ДВУМЯ ДИНАМИЧЕСКИМИ СИСТЕМАМИ

Как указано в [16], проблема верификации связей между двумя динамическими системами представляет особый интерес в гелиофизике. Для этого проводились эксперименты с различными временными рядами. Результаты одного из них приведены на рис. 4.4.

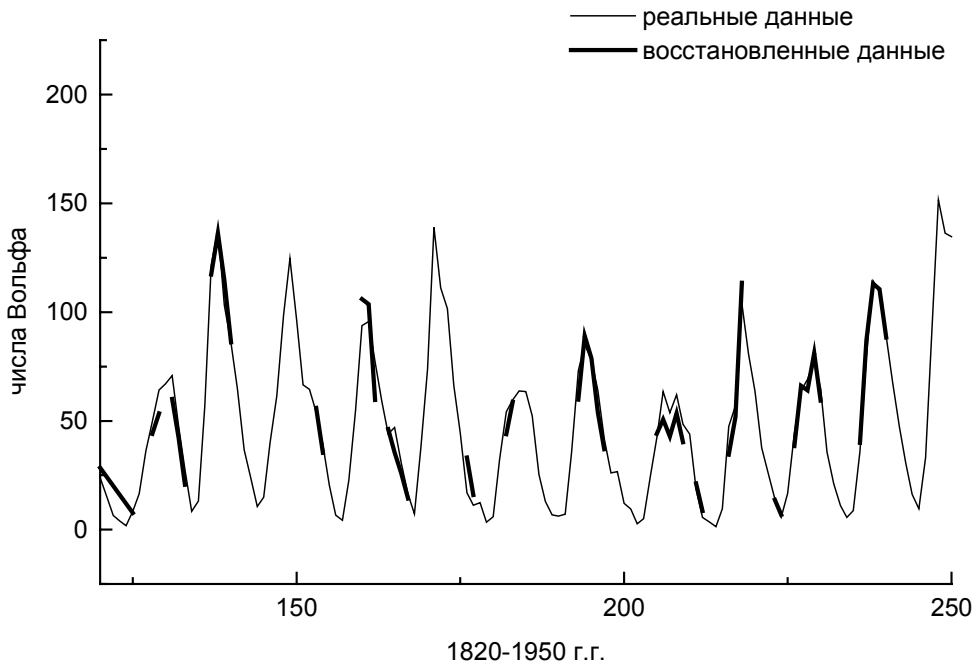

Рис. 4.4. Фрагмент годовых значений ряда Вольфа. SOC, число узлов – 10

Из полного ряда годовых значений чисел Вольфа (для солнечной активности) было удалено 50% точек. Для восстановления пропусков использовалась модель *СОК*. Строки исходной таблицы представляли собой *m*-мерные запаздывающие векторы Такенса, с *m*=6, вида: $a_{kj} = x_k, x_{k+1}, ..., x_{k+5}$, так что удаление одного отсчета в таблице приводит к удалению соответствующей диагонали. Нейронный конвейер удовлетворительно восстановил даже вершины циклов. На рис. 4.5 приведен фрагмент временного ряда космогенного изотопа ^{14}C. Полный ряд имеет диапазон с 5995 г.ВС по 1945г.AD. Мы даем результаты восстановления 30% удаленных точек фрагмента временного ряда датированного периодом с 5995 г. ВС по 10 г. AD. Результаты восстановления очень хорошие.

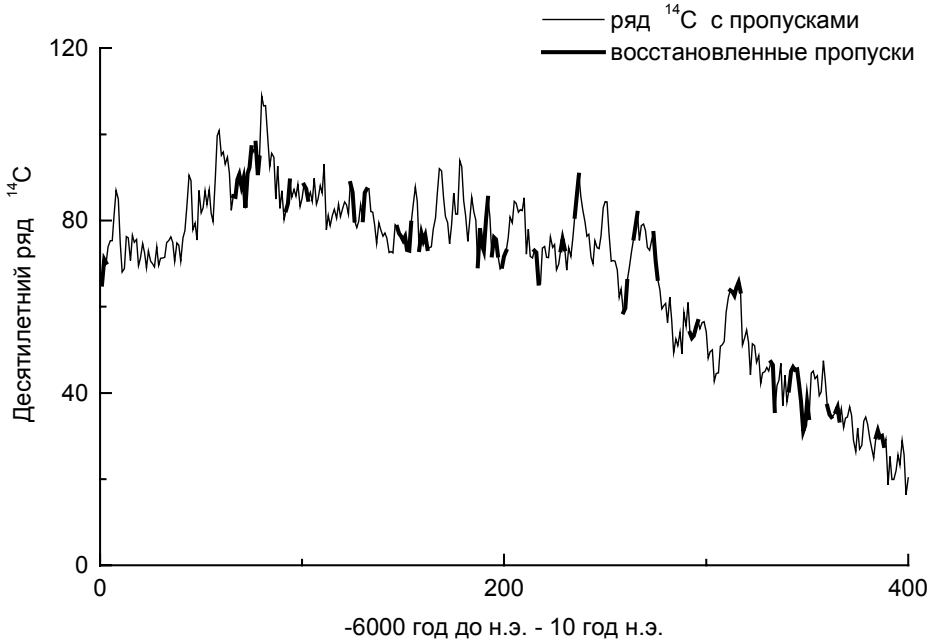

Рис. 4.5. Фрагмент десятилетних значений ряда ^{14}C.
Квазилинейная модель, число узлов – 8

Заметим, что составление исходной таблицы по Такенсу, в отличие от произвольного способа приведенного в [1, 14, 78], существенно меняет ситуацию. Действительно, пропущенное значение *y*-компоненты одного из векторов (L_1) на рис. 4.6 индуцирует пропуск *x*-компоненты в следующем (по Такенсу) векторе (L_3).

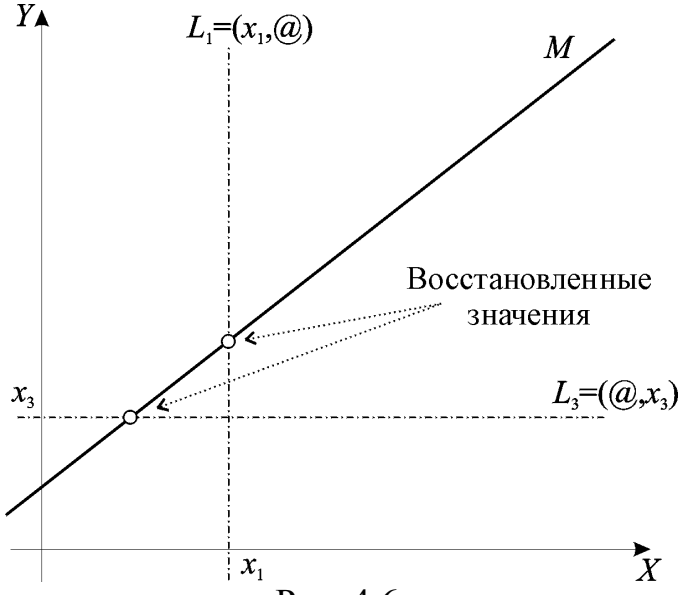

Рис. 4.6.

Пересечение 2-х прямых однозначно восстанавливает пропущенное значение. Для многомерного случая задача сводится к поиску *трансверсального* [45] пересечения плоскостей. Таким образом, метод восстановления пропусков получает формальный контекст.

Наши эксперименты показали, что метод заполнения пробелов с использованием нейронных сетей вполне приемлем для работы с реальными рядами.

IV.4. СОЧЕТАННЫЕ ПОРАЖЕНИЯ ПРОВОДЯЩЕЙ СИСТЕМЫ СЕРДЦА

В клинике главного кардиолога Красноярского края профессора Шульмана Владимира Федоровича программный продукт ModelAnalyzer нашел применение для выяснения прогноза у пациентов со сложными нарушениями ритма и проводимости сердца [46]. Под наблюдением сотрудников этой клиники в течении 15-20 лет находились пациенты с соответствующей патологией. Сотрудники клиники ежегодно проводят обследования данной группы пациентов, выясняя клиническое течение заболеваний, частоту обострения данной патологии, качество жизни данной группы пациентов. При столь длительном наблюдении постоянно накапливаются данные для выяснения, какие же факторы в большей степени влияют на прогноз данной патологии.

Существенную помощь в решении этого вопроса оказала методика многомерного факторного анализа. С ее использованием было выяснено, что у больных с сочетанными поражениями проводящей системы сердца в сравнении с пациентами с изолированным поражением проводящей системы сердца значительно чаще наблюдается застойная сердечная недостаточность, тромбоэмболические осложнения, некоторые формы нарушения сердечного ритма.

Кроме того были выявлены факторы, способствующие возникновению ряда осложнений. В частности, возникновению и развитию застойной сердечной недостаточности способствует наличие пароксизмальной формы фибрилляции предсердий и органические заболевания сердца в анамнезе. Возникновению тромбоэмболии способствует наличие фибрилляции и трепетания предсердия в анамнезе пациента. Возникновению фибрилляций предсердий способствует застойная сердечная недостаточность и нарушение прироста частоты сердечных сокращений при физической нагрузке.

Методика многомерного факторного анализа помогла выяснить, что у пациентов со сложными нарушениями ритма не эффективны некоторые режимы постоянной электрокардиостимуляции, что безусловно послужит темой дальнейших исследований в этой области. И, наконец, были выявлены факторы, способствующие возникновению летального исхода: у больных с сочетанными поражениями проводящей системы сердца наиболее неблагоприятно для прогноза наличие тромбоэмболий, застойной сердечной недостаточности в анамнезе, а также возникновение постоянной или пароксизмальной формы фибрилляции предсердий.

Таким образом, методика многомерного факторного анализа может с успехом использоваться для выявления факторов, оказывающих влияние на

течение и прогноз заболевания у больных со сложными нарушениями ритма и проводимости сердца.

Заключение

Разработан метод, который применим для заполнения пробелов и ремонта данных с пробелами. Представлены три различные вариации метода, начиная с простейших линейных моделей и заканчивая методом главных кривых для данных с пробелами. Нейросетевая реализация метода позволяет легко строить его параллельные реализации.

Приведенный алгоритм заполнения пробелов не требует их предварительного априорного заполнения – в отличие от многих других алгоритмов, предназначенных для той же цели. Однако он требует предварительной нормировки ("обезразмеривания") данных – перехода в каждом столбце таблицы к "естественной" единице измерения. Важное замечание – в задаче обработки данных с пробелами невозможно перейти к однородной задаче центрированием данных.

Большой интерес вызывает вопрос: сколько слагаемых (главных кривых) следует брать для обработки данных? Существует несколько вариантов ответов, но большинство из них подчиняется эвристической формуле: *число слагаемых должно быть минимальным среди тех, что обеспечивают удовлетворительное (терпимое) тестирование метода на известных данных.* Такой принцип "минимальной достаточности" характерен для многих нейросетевых приложений [32, 33, 47].

Разработанный метод выступает в форме некоторого "анзатца" – предложения, а не в виде серии теорем. Это не случайно – предлагается технология построения *правдоподобных* оценок пропущенных данных, а не их неизвестного истинного значения. Практическая ценность методов такого рода должна оцениваться и взвешиваться пользователями технологии. Разработано соответствующее программное обеспечение. Оно хорошо зарекомендовало себя при решении трудных задач с большим числом пропущенных данных, а в более простых (стандартных) случаях приводит к тем же результатам, что и классические методы статистического анализа.

В итоге основными результатами работы можно считать следующие.

1. Для решения задачи заполнения пропусков и ремонта искаженных данных разработан метод итерационного моделирования неполных данных с помощью многообразий малой размерности. Приведены три вариации метода: начиная с простейших линейных многообразий, продолжая построенными на их основе квазилинейными многообразиями и заканчивая методом главных кривых для данных с пробелами.

2. Для параллельной реализации метода итерационного моделирования данных с пробелами разработан способ построения нейронного конвейера, решающего задачи заполнения пробелов и ремонта данных.

3. Разработаны программные продукты FAMaster и ModelAnalyzer, реализующие предложенные технологии.

4. Численные эксперименты показали высокую эффективность итерационного моделирования неполных данных с помощью многообразий малой размерности. Метод хорошо зарекомендовал себя при решении трудных

задач с большим числом пропущенных данных, а в более простых (стандартных) случаях приводит к тем же результатам, что и классические методы статистического анализа.

Литература

1. Россиев А.А. Моделирование данных при помощи кривых для восстановления пробелов в таблицах // Методы нейроинформатики: сборник научных трудов / Под ред. А.Н. Горбаня. – Красноярск: КГТУ, 1998, – С. 6–22.
2. Горбань А.Н., Макаров С.В., Россиев А.А. Нейронный конвейер для восстановления пробелов в таблицах // Нейронные сети и модели: Труды международной НТК «Нейронные, реляторные и непрерывнологические сети и модели» (19-21 мая 1998 г.), Т.1 / Под ред. Л.И. Волгина. – Ульяновск: УлГТУ, 1998. – с.3.
3. Россиев А.А. Моделирование данных для восстановления пробелов в таблицах // Материалы конференции молодых ученых Института вычислительного моделирования СО РАН, апрель 1998 г. – Красноярск: ИВМ СО РАН, 1998, – с. 46–61.
4. Горбань А.Н., Макаров С.В., Россиев А.А. Нейронный конвейер для восстановления пробелов в таблицах и построения регрессии по малым выборкам с неполными данными // Математика. Компьютер. Образование. Вып. 5. Часть II. Сборник научных трудов / Под ред. Г.Ю. Ризниченко. М.: Изд-во Прогресс-Традиция, 1998. С. 27–32.
5. Горбань А.Н., Макаров С.В., Россиев А.А. Итерационный метод главных компонент для таблиц с пробелами // Третий сибирский конгресс по прикладной и индустриальной математике (ИНПРИМ-98), 22-27 июня 1998. Тезисы докладов. Ч.5. Новосибирск: Изд-во Института математики СО РАН, 1998. – с.74.
6. Горбань А.Н., Макаров С.В., Россиев А.А. Применение линейного и нелинейного факторного анализа, мозаичной регрессии и формул Карлемана для предобработки данных с пробелами // Нейроинформатика и ее приложения: Тезисы докладов VI Всероссийского семинара, 2-5 октября 1998 г. – Красноярск: КГТУ, 1998, – с.197–198.
7. Россиев А.А. FAMaster – программный продукт для моделирования неполных данных и заполнения пробелов в них // Нейроинформатика и ее приложения: Тезисы докладов VI Всероссийского семинара, 2-5 октября 1998 г. – Красноярск: КГТУ, 1998, – с.155.
8. Gorban' A.N., Rossiev A.A. Neural Network Iterative Method of Principal Curves for Data with Gaps. Journal of Computer and System Sciences International, 1999, Vol. 38, No. 5, P. 825–850.
9. Горбань А.Н., Макаров С.В., Россиев А.А. Заполнение пробелов в данных при помощи линейного и нелинейного факторного анализа, мозаичной регрессии и формул Карлемана // Всеросс. научно-техн. конф. Нейроинформатика-99. Сборник научных трудов. В 3 частях. Ч.1. – М.: МИФИ. 1999. 276 с. – С. 25–31.
10. Россиев А.А. Нейросетевая реализация метода главных кривых для данных с пробелами. // "Студент и научно-технический прогресс": Информационные

технологии. Материалы XXXVII международной научной студенческой конференции. –Новосибирск: НГУ, – 1999. С. 90–91.

11. Горбань А.Н., Россиев А.А. Итерационный метод главных кривых для данных с пробелами // Проблемы нейрокибернетики. Материалы XII Международной конференции по нейрокибернетике. – Ростов-на-Дону: Изд-во СКНЦ ВШ. 1999. 323 с. С. 198–201.
12. Россиев А.А. Нейросетевой подход к итерационному методу главных кривых для данных с пробелами // Материалы конференции молодых ученых Института вычислительного моделирования СО РАН, март 1999 г. – Красноярск: ИВМ СО РАН, 1999. – с. 92–94.
13. Горбань А.Н., Россиев А.А. Самоорганизующиеся кривые и нейросетевой итерационный метод главных кривых для данных с пробелами // Нейроинформатика и ее приложения: Тезисы докладов VII Всероссийского семинара, 1999 / Под ред. А.Н. Горбаня. Красноярск. КГТУ. 1999. – 167 с. – С. 32–33.
14. Горбань А.Н., Россиев А.А., D.C. Wunsch II, Самоорганизующиеся кривые и нейросетевое моделирование данных с пробелами // 2-я Всероссийская научно-техническая конференция “Нейроинформатика-2000”. Сборник научных трудов. Ч.1. М.: МИФИ.– 2000. С.40–46.
15. Зиновьев А.Ю., Питенко А.А., Россиев А.А. Проектирование многомерных данных на двумерную сетку. // 2-я Всероссийская научно-техническая конференция “Нейроинформатика-2000”. Сборник научных трудов. Ч.1. М.: МИФИ.– 2000. С.80-88.
16. Дергачев В.А., Макаренко Н.Г., Куандыков Е.Б., Горбань А.Н., Россиев А.А., Восстановление пробелов методами нейроинформатики, International Conference on Problems of Geocosmos, St. Peterburg, 2000, Book of Abstracts.
17. Gorban A.N., Rossiev A.A. Wunch II D.C. Neural Network Modelling of Data with Gaps // Радіоелектроніка. Інформатика. Управління, Запорожье. 2000, № 1, С. 47–55
18. Головенкин С.Е, Матюшин Г.В., Россиев А.А. Выявление факторов, влияющих на течение и прогноз заболевания у больных с сочетанными поражениями проводящей системы сердца // Нейроинформатика и ее приложения. Тезисы докладов VIII Всероссийского семинара. – Красноярск: КГТУ. – 2000. – С.44.
19. Горбань А.Н, Россиев А.А. Нейросетевое итерационное моделирование данных с пробелами самоорганизующимися многообразиями малой размерности // Нейроинформатика и ее приложения. Тезисы докладов VIII Всероссийского семинара. – Красноярск: КГТУ. – 2000. – С.45–48.
20. Артюхов И.П., Виноградов К.А., Россиев А.А., Россиев Д.А. Кластерный анализ регионов Красноярского края по показателям здоровья и здравоохранения // Моделирование неравновесных систем – 2000: Материалы III Всероссийского семинара. – Красноярск: КГТУ. – 2000. – С. 208–210.

21. Айвазян С.А., Енюков И.С., Мешалкин Л.Д. Прикладная статистика: Основы моделирования и первичная обработка данных. М.: Финансы и статистика, 1983. – 471с.
22. Айвазян С.А., Енюков И.С., Мешалкин Л.Д. Прикладная статистика: Исследование зависимостей. М.: Финансы и статистика, 1985. – 488с.
23. Айвазян С.А., Бухштабер В.М., Енюков И.С., Мешалкин Л.Д. Прикладная статистика: Классификация и снижение размерности. М.: Финансы и статистика, 1989. – 607с.
24. Айзенберг Л.А. Формулы Карлемана в комплексном анализе. Первые приложения. Новосибирск: Наука, 1990. 248 с.
25. Афифи А., Эйзен С. Статистический анализ. Подход с использованием ЭВМ. – М.: Мир, 1982. – 488с.
26. Вапник В.Н. Восстановление зависимостей по эмпирическом данным. – М.: Наука, 1979. – 448с.
27. Горбань А.Н., Миркес Е.М., Свитин А.П. Метод мультиплетных покрытий и его использование для предсказания свойств атомов и молекул // Журнал физической химии. – 1992. – Т.66, №6. – с.1503-1510.
28. Горбань А.Н., Миркес Е.М., Свитин А.П. Полуэмпирический метод классификации атомов и интерполяции их свойств. Препринт ВЦ СО АН СССР №19, Красноярск, 1989, 29с.
29. Горбань А.Н., Миркес Е.М., Свитин А.П. Полуэмпирический метод классификации атомов и интерполяции их свойств // Математическое моделирование в биологии и химии. Новые подходы. – Новосибирск: Наука. Сиб. отделение, 1992. – с.204-220.
30. Горбань А.Н., Новоходько А.Ю. Нейронные сети в задаче транспонированной регрессии, Второй Сибирский Конгресс по Прикладной и Индустриальной Математике, Тезисы докладов. Новосибирск, 1996. С.160-161.
31. Горбань А.Н., Новоходько А.Ю., Царегородцев В.Г. Нейросетевая реализация транспонированной задачи линейной регрессии, Нейроинформатика и ее приложения: Тезисы докладов IV Всероссийского семинара. Красноярск, 1996, с.37–39.
32. Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере. Новосибирск: Наука, 1996.
33. Ежов А.А., Шумский С.А. Нейрокомпьютинг и его приложения в экономике и бизнесе. М.: МИФИ, 1998. – 224 с.
34. Енюков И.С. Методы, алгоритмы, программы многомерного статистического анализа. – М.: Финансы и статистика, 1986.
35. Жамбю М. Иерархический кластер-анализ и соответствия: Пер. с фр. – М.: Финансы и статистика, 1988. – 342 с., ил.
36. Жанатаусов С.У. Методы прогностических переменных. – Машинные методы обнаружения закономерностей, Новосибирск: 1981, вып. 88, Вычислительные системы. С. 151-155.
37. Загоруйко Н.Г. Методы обнаружения закономерностей

38. Загоруйко Н.Г., Ёлкина В.Н., Лбов Г.С. Алгоритмы обнаружения эмпирических закономерностей. – Новосибирск: Наука, 1985. – 110с.
39. Загоруйко Н.Г., Елкина В.Н., Лбов Г.С., Емельянов С.В. Пакет прикладных программ ОТЭКС. М.: Финансы и статистика, 1986.
40. Загоруйко Н.Г., Ёлкина В.Н., Тимеркаев В.С. Алгоритм заполнения пропусков в эмпирических таблицах (алгоритм “ZET”) // Вычислительные системы. – Новосибирск, 1975. – Вып. 61. Эмпирическое предсказание и распознавание образов. – С. 3-27.
41. Кендалл М., Стьюарт А. Многомерный статистический анализ и временные ряды. – М.: Наука, 1976. – 736 с.
42. Кендалл М., Стьюарт А. Статистические выводы и связи. – М.: Наука, 1973. – 900 с.
43. Лбов Г.С. Методы обработки разнотипных экспериментальных данных. – Новосибирск: Наука, 1981. – 157с.
44. Литл Р.Дж.А., Рубин Д.Б. Статистический анализ данных с пропусками. М.: Финансы и Статистика, 1991.
45. Макаренко Н.Г. Многообразия, погружения и трансверсальность//сб. Проблемы солнечной активности, Ленинград, 1991, 13-28
46. Матюшин Г.В. Сочетанные поражения проводящей системы сердца (распространенность, клинико-электрокардиографические варианты, клиническое течение, прогноз) // Диссертация на соискание ученой степени доктора медицинских наук, КрасГМА, 2000.
47. Нейроинформатика / А.Н. Горбань, В.Л. Дунин-Барковский, Е.М. Миркес и др. Новосибирск: Наука (Сиб. отделение), 1998.
48. Пфанцагль И. Теория измерений. – М.: Мир, 1976. – 246с.
49. Рао С.Р. Линейные статистические методы. – М.: Наука, 1968. – 548 с.
50. Растригин Л.А., Пономарев Ю.П. Экстраполяционные методы проектирования и управления. – М.: Машиностроение, 1986. – 120 с.
51. Самарский А.А. Введение в численные методы. М.: Наука. Главная редакция физико-математической литературы, 1982. – 272 с.
52. Справочник по прикладной статистике. В 2-х т., под ред Э.Ллойда, У.Ледермана, Ю.Н.Тюрина – М.: Финансы и статистика, 1989, 1990.
53. Тюрин Ю.Н., Макаров А.А. Статистический анализ данных на компьютере / Под ред. В.Э.Фигурнова – М.: ИНФРА-М, 1998. – 528 с., ил.
54. Факторный, дискриминантный и кластерный анализ. – М.: Финансы и статистика, 1989. – 215 с.
55. Afifi A.A., Elashoff R.M. Missing observations in multivariate statistics. – J. Amer. Statist. Assoc., 1966, vol. 61. pp. 595-604.
56. Beale E.M., Little R.J. Missing values in multivariate analysis. – J. Roy. Statist. Soc. B., 1975, vol. 37. pp. 129-145.
57. Buck S.F. A method of estimation of missing values in multivariate data. – J. Roy. Statist. Soc. B., 1960, vol. 22. pp. 202-206.
58. Delicado P., Principal Curves and Principal Oriented Points, Tech. rep. 309, Department d'Economia i Empresa, Universitat Pompeu Fabra, 1998.

59. Dempster A.P., Laird N.M., Rubin D.B. Maximum likelihood from incomplete data via the EM-algorithm. – J. Roy. Statist. Soc. B., 1977, vol. 39. pp. 1-38.
60. Dodge Y. Analysis of experiments with missing data. – New York, Wiley, 1985. 498 p.
61. Engelman L. An efficient algorithm for computing covariance matrices from data with missing values. – Communs Statist. B., 1982, vol. 11. p. 113-121.
62. Frane G.M. Some simple procedures for handling missing values in multivariate analysis. – Psychometrika, 1976, vol. 41. pp. 409-415.
63. Glasser M. Linear regression analysis with missing observations among the independent variables. – J. Amer. Statist. Assoc., 1964, vol. 59. p. 834-844.
64. Gleason T.C., Staelin R. A proposal for handling missing data. – Psychometrika, 1975, vol. 40. pp. 229-252.
65. Gorban A.N., Novokhodko A.Yu. Neural Networks In Transposed Regression Problem, Proc. INNS WCNN ’96.
66. Gorban A.N., Waxman C. Neural Networks for Political Forecast. Proceedings of the WCNN'95 (World Congress on NeuralNetworks'95, Washington DC, July 1995), PP.176- 178.
67. Hartley H.O., Hocking R.R. The analysis of incomplete data. – Biometrics, 1971, vol. 27. pp. 783-808.
68. Hastie T. and Stueltze, Principal Curves, Journal of the American Statistical Association, vol. 84, no. 406, pp. 502-516, 1989.
69. Hastie T. Principal Curves and Surfaces, PhD thesis, Stanford University, 1984.
70. Hocking R.R., Marx D.L. Estimatiom with incomplete data: an improved computational method and the analysis of nested data. – Communs Statist. A., 1979, vol. 8. pp. 1151-1181.
71. Huseby J.R., Schwertman N.C., Allen D.M. Computation of the mean vector and dispersion matrix for incomplete multivariate data. – Communs Statist. B., 1980, vol. 9. pp. 301-309.
72. Kegl B., Krzyzak A., Linder T. and Zeger K. Principal Curves: Learning and convergence, in Proceedings of IEEE International Symposium on Information Theory, p. 387, 1998.
73. Kegl B., Krzyzak A., Linder T., and Zeger K., Learning and Design of Principal Curves, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1999.
74. Kohonen T. Self-Organizing Maps. Springer: Berlin – Heidelberg, 1997.
75. Kramer M.A. Nonlinear principal component analysis using autoassociative neural networks. AIChE Journal. 1991. V.37, No. 2. PP. 233-243.
76. LeBlank M., Tibshorany N. Adaptive principal surfaces. Journal of the American Statistical Association. 1994, Mar. V. 89, No. 425. PP. 53-64.
77. Lichtman A.J., Keilis-Borok V.I., Pattern Recognition as Applied to Presidential Elections in U.S.A., 1860-1980; Role of Integral Social, Economic and Political Traits, Contribution N 3760. 1981, Division of Geological and Planetary Sciences, California Institute of Technology.
78. Little R.J., Rubin D.B. Statistical analysis with missing data. – New York, Wiley, 1987. 430 p.

79. Little R.J., Schlushter M.D. Maximum likelihood estimation for mixed continuous and categorical data with missing values. – Biometrika, 1985, vol. 72. pp. 497-512.
80. Little R.J., Smith P.J. Editing and imputation for quantitative survey data. – J. Amer Statist. Assoc., 1987, vol. 82. pp. 58-68.
81. Mardia K.V., Kent J.T. and Bibby J.M. Multivariate Analysis. London: Academic Press, 1979.
82. Srivastava M.S. Multivariate data with missing observations. – Communs Statist. Theory and Method, 1985, vol. 14. pp. 775-792.
83. Tibshirani R., Principal Curves revisited, Statistics and Computation, vol. 2, pp. 183-190, 1992.
84. Titterington D.M., Jiang J.M. Recursive estimation procedures for missing data problems. – Biometrika, 1983, vol. 70. pp. 613-624.
85. Walsh J.E. Computer-feasible method for handling incomplete data in regression analysis. – J. of ACM, 1961, vol. 18. pp. 201-211.
86. Wilks S.S. Moments and distributions of estimates of population from fragmentary samples. – Ann. Math. Statist., 1932, vol.3. pp. 163-195.