

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 изложение 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 $X=(X^{(1)}, \dots, X^{(d)})$ размерности d и с конечными случайными моментами. Первая линейная главная компонента для X представляет собой прямую линию, ожидаемое Евклидово 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 $v=(v_1, \dots, v_n) \in \mathfrak{R}^d$. 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

$$c, c_0: \frac{\partial}{\partial c} E = 0, \frac{\partial}{\partial c_0} E = 0, \quad (2.1)$$

где

$$E = \int_x \|v - \lambda c - c_0\|^2 P(v) dv. \quad (2.2)$$

$P(v)$ – 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 $f(x) = \lambda x + b$, которая задается направляющим вектором λ 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 $x = \text{Pr}_f(a)$ на полученной прямой.

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

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 $x = \text{Pr}_M(a)$. 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 $x_i y_j$ методом наименьших квадратов:

$$\Phi = \sum_{\substack{i,j \\ a_{ij} \neq @}} (a_{ij} - x_i y_j)^2 \rightarrow \min. \quad (2.3)$$

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 $\partial\Phi/\partial x_i = 0$:

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

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

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 $\partial\Phi/\partial y_j = 0$:

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

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

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:

$$y - \text{случайный, нормирован на 1 (т.е. } \|y\|^2 = \sum_j y_j^2 = 1).$$

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 $\varepsilon, \delta < 0$.

В результате для матрицы A получили наилучшее приближение матрицей P_1 вида $x_i y_j$. 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 нулю.

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

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 $x_i y_j + b_j$.

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

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

$$\Phi = \sum_{\substack{i,j \\ a_{ij} \neq @}} (a_{ij} - x_i y_j - b_j)^2 \rightarrow \min. \quad (2.4)$$

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 $\partial\Phi/\partial x_i = 0$:

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

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

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

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

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 $\partial\Phi/\partial b_j=0$:

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

Представляя полученные уравнения в виде системы, получим:

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

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:

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

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

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

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 :

$$y - \text{случайный, нормирован на 1 (т.е. } \|y\|^2 = \sum_j y_j^2 = 1).$$

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

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 нулю, то вводится критерий остановки – не когда процесс полностью сойдется, а когда относительное уменьшение на каждой итерации станет меньше наперед 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. Вторым критерий – малость самого значения Φ . Процедура останавливается, если $\Delta\Phi/\Phi < \varepsilon$ или $\Phi < \delta$ для некоторых $\varepsilon, \delta < 0$.

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 $x_i y_j + b_j$. Далее, из матрицы A вычитаем полученную матрицу P_1 , и для полученной матрицы отклонений $A - P_1$ 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. $A = P_1 + P_2 + \dots + P_q$. 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 $x_i y_j + b_j$;

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

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 $x_j y_j + b_j$ ($P_q = x_j^q y_j^q + b_j^q$), исчерпывающая исходную матрицу A с заданной точностью. Для каждого q по заданной строке определим число $x^q(a)$ и вектор a_j^q :

$$\begin{aligned}
 a_j^0 &= a_j, (a_j \neq @); \\
 &\dots\dots\dots \\
 x^q(a) &= \left(\sum_{\substack{j \\ a_j \neq @}} (a_j^{q-1} - b_j^q) y_j^q \right) / \left(\sum_{\substack{j \\ a_j \neq @}} (y_j^q)^2 \right); \\
 a_j^q &= a_j^{q-1} - x^q(a) y_j^q - b_j^q, (a_j \neq @); \\
 &\dots\dots\dots
 \end{aligned} \tag{2.5}$$

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

$$\begin{aligned}
 \text{Pr}(a) &= t(a) y_j + b_j, \\
 t(a) &= \left(\sum_{\substack{j \\ a_j \neq @}} (a_j - b_j) y_j \right) / \left(\sum_{\substack{j \\ a_j \neq @}} (y_j)^2 \right).
 \end{aligned} \tag{2.6}$$

То же самое в векторной форме:

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

Для Q -факторного восстановления данных полагаем:

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

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 отклонение от прямой f_1 . Для этого отклонения 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 значения 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:

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

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:

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

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 $\partial\Phi/\partial x_{2i}=0$:

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

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 $\partial\Phi/\partial b_j=0$:

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

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:

$$\Phi = \sum_{\substack{i,j \\ a_{ij} \neq 0}} (a_{ij} - x_{1i}y_{1j} - x_{2i}y_{2j} - x_{3i}y_{3j} - b_j)^2 \rightarrow \min. \quad (2.10)$$

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 $\partial\Phi/\partial x_{3i}=0$:

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

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 $\partial\Phi/\partial b_j=0$:

$$\left\{ \begin{array}{l} \frac{\partial \Phi}{\partial y_{1j}} = -2 \sum_{\substack{i \\ a_{ij} \neq @}} (a_{ij} - x_{1i}y_{1j} - x_{2i}y_{2j} - x_{3i}y_{3j} - b_j)x_{1i} = 0, \\ \frac{\partial \Phi}{\partial y_{2j}} = -2 \sum_{\substack{i \\ a_{ij} \neq @}} (a_{ij} - x_{1i}y_{1j} - x_{2i}y_{2j} - x_{3i}y_{3j} - b_j)x_{2i} = 0, \\ \frac{\partial \Phi}{\partial y_{3j}} = -2 \sum_{\substack{i \\ a_{ij} \neq @}} (a_{ij} - x_{1i}y_{1j} - x_{2i}y_{2j} - x_{3i}y_{3j} - b_j)x_{3i} = 0, \\ \frac{\partial \Phi}{\partial b_j} = -2 \sum_{\substack{i \\ a_{ij} \neq @}} (a_{ij} - x_{1i}y_{1j} - x_{2i}y_{2j} - x_{3i}y_{3j} - b_j) = 0. \end{array} \right.$$

То же самое в векторной форме:

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

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:

$$\Phi = \sum_{\substack{i,j \\ a_{ij} \neq 0}} (a_{ij} - f_j(\sum_k a_{ik} y_k))^2 + \alpha \int_{-\infty}^{+\infty} (f''(t))^2 dt, \quad (2.11)$$

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 $ty + b$).

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

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

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

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

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

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:

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

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

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

$$\begin{aligned} \int_{-\infty}^{+\infty} \left(f_j''(t) \right)^2 dt &= \int_{-1}^{+1} \left(f_j''(t) \right)^2 dt = \sum_{\substack{k,l=2 \\ k+l=2\text{-четное}}}^n f_k^j f_l^j k(k-1)l(l-1) \frac{t^{k+l-3}}{k+l-3} \Big|_{-1}^{+1} = \\ &= \sum_{\substack{k,l=2 \\ k+l=2\text{-четное}}}^n f_k^j f_l^j k(k-1)l(l-1) \frac{2}{k+l-3}. \end{aligned}$$

Продифференцируем полученное выражение по f_k^j :

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

Окончательно, для $k=0..n$, имеем:

$$\begin{aligned} \frac{\partial\Phi}{\partial f_k^j} &= -2 \sum_{\substack{i \\ a_{ij} \neq @}} (a_{ij} - \sum_{l=0}^n f_l^j x_i^l) x_i^k + \\ &+ \alpha \sum_{\substack{l=2 \\ l \neq k \\ k+l=2\text{-четное}}}^n f_l^j k(k-1)l(l-1) \frac{2}{k+l-3} + 2 f_k^j k^2 (k-1)^2 \frac{2}{2k-3} = 0. \end{aligned}$$

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

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

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

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

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 Φ (2.11) на некоторой равномерной 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:

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

где $\alpha > 0$ – параметр сглаживания.

Пусть задан отрезок $[-1, 1]$. Разобьем его на n частей точками $t_s, s=0..n$, где: $-1 = t_0 < t_1 < \dots < t_{n-1} < t_n = 1$, 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:

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

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

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

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

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

$$\begin{aligned}
I &= \int_{-1}^1 (f''(t))^2 dt = \sum_{s=1}^{n-1} \int_{t_{s-1}}^{t_s} \sum_{k,l=2}^3 k(k-1)l(l-1) f_{sk}^j f_{sl}^j t^{k+l-4} dt = \\
&= \sum_{s=1}^{n-1} \left(\sum_{k,l=2}^3 f_{sk} f_{sl} k(k-1)l(l-1) \frac{t^{k+l-3}}{k+l-3} \Big|_{t_{s-1}}^{t_s} \right) = \\
&= \sum_{s=1}^{n-1} \left(\sum_{k,l=2}^3 f_{sk} f_{sl} k(k-1)l(l-1) \frac{h_s^{k+l-3}}{k+l-3} \right).
\end{aligned}$$

Таким образом, исходная задача сглаживания запишется в следующем виде:

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

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

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:

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

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

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

где:

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

есть линейные ограничения, полученные из условий непрерывности.

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

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

т.е. систему из $(7n-3)$ уравнений и с $(7n-3)$ неизвестными.

Первые строчки этой системы можно расписать следующим образом:

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

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

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

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

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 $((7n-3) \times (7n-3))$, 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 $\{t_k\}$ ($k=1,2,\dots$). Требуется построить формулу продолжения с *конечного* множества, наилучшую в следующем смысле: последовательность функций $f_n(t)$, полученных продолжением с множества $\{t_k\}$ ($k=1,2,\dots,n$), 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:

$\Phi = H + \alpha I \rightarrow \min$, где:

$$H = \sum_{\substack{i,j \\ a_{ij} \neq 0}} (a_{ij} - f_j(\sum_k a_{ik} y_k))^2, \quad I = \int_{-\infty}^{+\infty} (f''(t))^2 dt,$$

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

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

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:

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

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

Посчитаем значение интеграла гладкости I :

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

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

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

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

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

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 $x_jy_j+b_j$) 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 многообразия 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 построения 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.

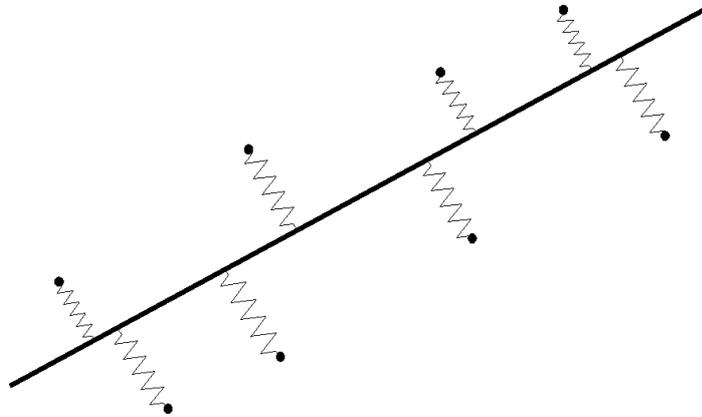


Рис. 2.1

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).

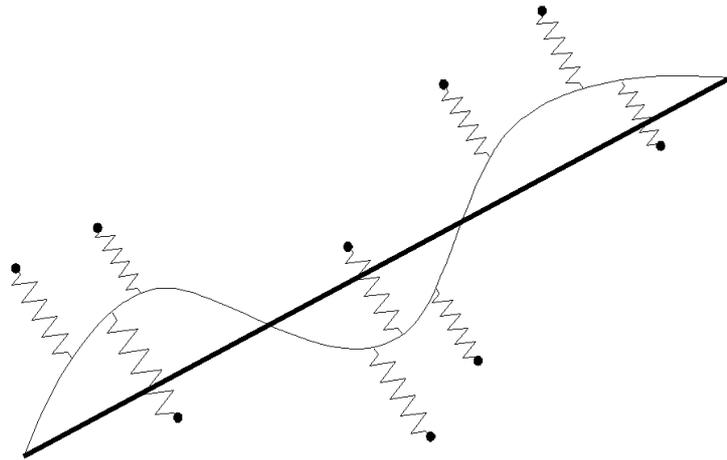


Рис. 2.2

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 .

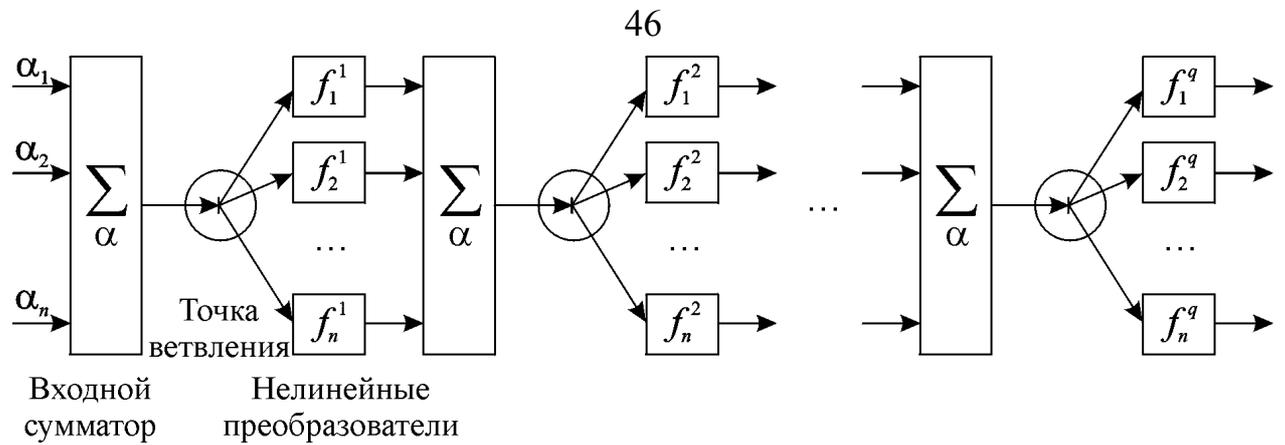


Рис. 2.3

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.