

УДК 519.651

*М. Г. Беляев*Московский физико-технический институт (государственный университет)
Институт проблем передачи информации им. А.А. Харкевича РАН
ООО «ДАТАДВАНС»

Аппроксимация данных, порожденных декартовым произведением

В работе предложен метод построения аппроксимации для выборок, имеющих структуру декартова произведения произвольного числа многомерных факторов. В подходе используется модель линейного разложения по словарю (некоторому набору) функций, сформированному с помощью тензорного произведения словарей функций меньшей размерности. Для контроля гладкости аппроксимации вводится штраф на изменчивость полученной модели, а коэффициенты разложения по сформированному словарю ищутся оптимальным образом в смысле среднеквадратичной выборочной ошибки. Параметры регуляризации, задающие уровень гладкости модели в каждом из факторов, вычисляются с помощью ошибки скользящего контроля. Специальная структура данных позволяет производить все вычисления со сформированным словарем эффективно. Основанный на предложенном подходе алгоритм успешно используется при решении актуальных прикладных задач.

Ключевые слова: аппроксимация многомерной функции по выборке, тензорное произведение, словари функций.

1. Введение

Работа посвящена задаче аппроксимации, которая заключается в восстановлении некоторой неизвестной зависимости по обучающей выборке — заданному набору пар «точка» — «значение функции в точке» (см. раздел 2). Эта задача широко распространена в инженерных приложениях и является одной из основных составляющих суррогатного моделирования [1]. Для ее решения разработано большое количество подходов [2], которые эффективно работают в широком диапазоне требований прикладной области.

В работе рассматривается специальный случай задачи аппроксимации, в котором множество точек обучающей выборки является декартовым произведением некоторого количества факторов (множеств) произвольной размерности (см. раздел 2.1). Выборки такого рода распространены в практических задачах в силу некоторых особенностей задач инженерного проектирования, обуславливающих выбор именно такого дизайна эксперимента. Большинство методов аппроксимации предназначены для выборок произвольной структуры и не учитывают особенности выборок, порожденных декартовым произведением, что не позволяет в этом случае строить аппроксимацию вычислительно эффективно (обзор методов приведен в разделе 2.2).

Среди многообразия подходов к решению задачи восстановления зависимостей можно выделить классический метод интерполяции двумерных данных с помощью тензорного произведения сплайнов [3], который рассчитан на выборку, порожденную декартовым произведением двух одномерных множеств (двумерная сетка). Использование этого метода, учитывающего структуру данных, позволяет уменьшить вычислительную сложность построения аппроксимации по сравнению с универсальными подходами.

Реальные задачи зачастую обладают существенно более сложной структурой, чем двумерная сетка. Цель работы — предложить метод аппроксимации для выборок, имеющих структуру декартова произведения с несколькими многомерными факторами, сохранив при этом преимущество в меньшей вычислительной сложности построения аппроксимации. В разделе 3 вводятся некоторые операции с многомерными матрицами, необходимые для

дальнейшего изложения. Основные результаты формулируются в разделе 4, где предлагается рассматривать тензорное произведение словарей функций и вводится (в подразделе 4.1) штраф специального вида на изменчивость модели. В подразделе 4.2 представлено утверждение, определяющее условие на существование и единственность решения задачи аппроксимации в рассмотренном классе функций. Предложенный алгоритм построения аппроксимации описан в разделе 4.3, а результаты вычислительных экспериментов приведены в разделе 5.

2. Задача аппроксимации

Рассмотрим задачу аппроксимации функции по конечному набору ее значений в заданных точках. Пусть g — это некоторая непрерывная многомерная функция, $g : (X \subset R^d) \rightarrow R^{d_y}$ (в дальнейшем будем рассматривать случай $d_y = 1$, который легко обобщается на случай $d_y > 1$). Будем называть обучающей выборкой S совокупность множества точек Σ (мощности N) и значений функции g в точках из этого множества:

$$S = \{x_i \in \Sigma, y_i = g(x_i)\}_{i=1}^N = \{\Sigma, g(\Sigma)\}.$$

Задача 1 (аппроксимации). *По заданной обучающей выборке S найти функцию $f^* \in F$ для некоторого функционального класса F , минимизирующую среднеквадратичную ошибку аппроксимации Q , подсчитанную в точках множества Σ :*

$$f^* = \operatorname{argmin}_{f \in F} (Q(f, \Sigma, g(\Sigma))) = \operatorname{argmin}_{f \in F} \sum_{x \in \Sigma} (g(x) - f(x))^2.$$

Отметим, что эта постановка задачи аппроксимации включает в себя интерполяцию как частный случай при $Q(f, \Sigma, g(\Sigma)) = 0$.

Задача 1 позволяет найти такую аппроксимацию $f^* \in F$, которая приближает обучающую выборку наилучшим образом. Однако в этом случае может произойти переобучение [2]: значения аппроксимации f^* будут значительно отличаться от значений функции g в точках, не принадлежащих Σ . Стандартный подход к решению этой проблемы — введение некоторого штрафа, который ограничивает изменчивость модели и позволяет получить более гладкие решения. Например, в качестве такого штрафа может быть выбрана норма вторых производных f (см., например, сглаживающие сплайны [3]) или норма f в некотором гильбертовом пространстве функций (ядерная гребневая регрессия [4]). В этом случае задача аппроксимации также может быть сформулирована как задача минимизации, но с другой целевой функцией.

Задача 2 (аппроксимации со штрафом на изменчивость). *Пусть заданы обучающая выборка S и штрафная функция $P_\lambda : F \rightarrow R_+^1$. Необходимо найти $f^* \in F$ для некоторого заданного класса F , минимизирующую функцию ошибки со штрафом $R(f, \Sigma, g(\Sigma)) = Q(f, \Sigma, g(\Sigma)) + P_\lambda(f)$, а именно:*

$$f^* = \operatorname{argmin}_{f \in F} (Q(f, \Sigma, g(\Sigma)) + P_\lambda(f)) = \operatorname{argmin}_{f \in F} \sum_{x \in \Sigma} (g(x) - f(x))^2 + P_\lambda(f).$$

Заметим, что штрафная функция принадлежит некоторому семейству и задается с помощью параметра λ , который определяет вклад штрафного слагаемого в общую целевую функцию. Как правило, P_λ зависит от λ линейно. Например, задача ядерной гребневой регрессии сводится к следующей оптимизационной задаче:

$$f^* = \operatorname{argmin}_{f \in F} \sum_{x \in \Sigma} (g(x) - f(x))^2 + \lambda \|f\|_F^2.$$

2.1. Структура данных

Опишем более формально структуру данных, которую мы будем рассматривать в рамках этой работы. Пусть заданы некоторые множества точек

$$\sigma_k = \{x_{i_k}^k \in X_k, i_k = 1, \dots, n_k\}, X_k \in \mathbb{R}^{d_k}, k = 1, \dots, K.$$

Определение 1. Будем называть множества σ_k факторами, а размерность d_k каждого элемента множества — размерностью фактора σ_k .

Определим множество точек Σ как декартово произведение всех факторов: $\Sigma = \sigma_1 \times \sigma_2 \times \dots \times \sigma_K$. Тогда элементы множества Σ — это вектора размерности $d = \sum_{k=1}^K d_k$ (а объем этой выборки равен $N = \prod_{k=1}^K n_k$):

$$\Sigma = \{x = [x_{i_1}^1, x_{i_2}^2, \dots, x_{i_K}^K], \{x_{i_k}^k \in \sigma_k\}_{k=1}^K\}.$$

Описанная структура данных встречается в реальных задачах достаточно часто. Например, распространен случай расположения данных на многомерной сетке, поскольку он удобен при планировании эксперимента и обеспечивает равномерное покрытие пространства дизайном точками [1]. В введенных обозначениях такому дизайну соответствует декартово произведение K одномерных факторов.

Также в инженерных приложениях часто встречаются задачи, в которых все переменные можно разбить на две группы: первая из них соответствует условиям эксперимента, а вторая часть определяет конкретное измерение, сделанное в ходе этого эксперимента. Например, при исследовании аэродинамики крыла первой группе переменных может соответствовать его геометрическое описание, второй группе — угол атаки крыла или скорость воздушного потока, а в качестве выходной переменной может выступать подъемная сила, которой обладает крыло. Сложность изменения значений переменных в первой и второй группах может быть существенно разной (в смысле подсчета нового значения выходной характеристики для модифицированного входного вектора). В описанном примере изменение геометрии крыла — это достаточно дорогая операция (особенно при экспериментах в аэродинамической трубе), в то время как скорость потока набегающего воздуха и угол атаки могут быть изменены большое количество раз для каждого крыла. Эта особенность делает целесообразным использование одинаковой, достаточно плотной сетки значений (набора векторов) переменных из второй группы для каждого зафиксированного значения вектора переменных из первой группы. В таком случае выборка данных будет иметь структуру декартова произведения с двумя факторами. В некоторых задачах факторизация имеет более сложную структуру и переменные разбиваются на большее число групп, чем две. Следует заметить, что несимметричная сложность изменения значений переменных из разных групп часто приводит к возникновению анизотропии: переменные из «недорогих» групп переменных (угол атаки) принимают существенно больше различающихся значений, чем переменные из «дорогих» групп (геометрия крыла).

Вернемся к задаче аппроксимации и рассмотрим ее частный случай: пусть множество точек обучающей выборки есть декартово произведение $K > 1$ факторов, $\Sigma = \sigma_1 \times \sigma_2 \times \dots \times \sigma_K$. Тогда область определения функции g также будет декартовым произведением: $X = X_1 \times X_2 \times \dots \times X_K$. Поставим в соответствие задачам аппроксимации 1, 2 аналогичные задачи с ограничением на структуру выборки.

Задача 3. Решить задачу 1 при условии, что множество точек Σ есть декартово произведение: $\Sigma = \sigma_1 \times \sigma_2 \times \dots \times \sigma_K$.

Задача 4. Решить задачу 2 при условии, что множество точек Σ есть декартово произведение: $\Sigma = \sigma_1 \times \sigma_2 \times \dots \times \sigma_K$.

2.2. Обзор методов аппроксимации

Для решения задач 1 и 2 существует множество общих подходов, которые могут быть применены к выборкам с произвольным дизайном, в том числе и в случае, когда Σ есть декартово произведение. Например, это могут быть следующие распространенные методы:

- линейная регрессия [2];
- гауссовские процессы (или кригинг), см. [5];
- разложение по словарю нелинейных параметрических функций, см. [6];
- сплайны (для $d = 1$), см. [3].

Однако универсальность этих методов не позволяет добиться высокой эффективности для частных задач 3 и 4. Сплайны предназначены для решения одномерных задач и по определению не могут быть использованы для размерности $d > 1$. Линейная регрессия предлагает крайне простую модель, которая может быть востребована при малом количестве данных, но, как правило, выборки, порожденные декартовым произведением, имеют достаточно большой объем, который может доходить до нескольких миллионов точек. Большие значения N затрудняют использование и остальных техник.

Гауссовские процессы имеют вычислительную сложность $O(d_y N^3 + d N^2)$, что ограничивает использование этого метода при объемах выборки более 1000 точек. Разреженные гауссовские процессы [7] предназначены для решения этой проблемы и снижают вычислительную сложность до $O(d_y N m^2 + d N m)$ (где m — это размер некоторой подвыборки из обучающей выборки). Как правило, в качестве m надо брать некоторую не стремящуюся к нулю долю от N , что позволяет расширить область применимости гауссовских процессов лишь до объемов выборки порядка 10 000 точек.

Модели, основанные на разложении по словарю нелинейных параметрических функций (включая искусственные нейронные сети), в основном зависят от объема выборки N линейно (за исключением регрессии на основе радиальных базисных функций, сложность которой может доходить до $O(N^3)$). Однако вычислительная сложность этих методов существенно зависит от объема словаря p и размерностей данных d, d_y . В зависимости от метода настройки параметров функций словаря сложность может варьироваться от $O(N p (d + d_y))$ до $O(N p d_y + N p^2 d^2 + p^3 d^3)$. Зачастую p выбирают пропорционально N , что приводит к (как минимум) квадратичной зависимости от объема выборки. В противном случае зависимость остается линейной, но множитель перед ней принимает достаточно большие значения. Кроме того, необходимо учесть итеративный характер обучения (настройки параметров) таких моделей (количество итераций может доходить до нескольких тысяч), что еще сильнее увеличивает масштабный множитель.

Помимо высокой вычислительной сложности следует отметить еще одну проблему универсальных методов в рассматриваемом классе задач аппроксимации. Большинство из этих методов явно или неявно предполагает, что точки множества Σ распределены в пространстве дизайна равномерно. Это условие нарушается в случае анизотропных данных (мощности различных факторов существенно отличаются) и построенные универсальными методами модели могут иметь крайне низкую точность в областях, удаленных от точек обучающей выборки, но находящихся внутри выпуклой оболочки множества Σ . Такая особенность не формализуется строго, однако этот недостаток универсальных нелинейных методов проявляется в подавляющем большинстве реальных задач.

Существует подход, учитывающий наличие структуры декартова произведения в обучающей выборке данных. Этот метод основан на тензорном произведении сплайнов [3] и предназначен для решения задачи интерполяции данных, заданных на двумерной решетке (частный случай задачи 3 для двух одномерных факторов). Вычислительная сложность этого метода составляет $O(n_1^3 + n_2^3 + N(n_1 + n_2)d_y)$, где n_1 и n_2 — это количество точек в соответствующих факторах, $n_1 n_2 = N$. Так как для сплайнов можно явным образом

построить базис, то метод не требует итеративного обучения и константа перед оценкой сложности мала.

В работе будет предложен метод, основанный на тензорном произведении некоторых наборов функций, построенных в каждом из факторов. Такой подход является обобщением тензорного произведения сплайнов и позволяет как строить аппроксимацию для выборок, в которых Σ есть декартово произведение общего вида (произведение K многомерных факторов), так и решать не только задачу 3, но и более общую задачу 4 со штрафом.

3. Тензорное произведение и многомерные матрицы

Для изложения предлагаемого подхода к решению задачи аппроксимации необходимо ввести некоторые определения. В этом разделе будет описано понятие тензорного произведения, а также рассмотрено обобщение матричных операций на K -мерные матрицы.

3.1. Тензорное произведение

Пусть функции $f_1(x^1)$ и $f_2(x^2)$ определены на $X_1 \in \mathbb{R}^{d_1}$ и $X_2 \in \mathbb{R}^{d_2}$ соответственно.

Определение 2. Функцию $f([x^1, x^2])$, определенную на $X_1 \times X_2$, будем называть тензорным произведением функций $f_1(x^1)$ и $f_2(x^2)$ и обозначать $f_1 \otimes f_2$, если $f([x^1, x^2]) = f_1(x^1) f_2(x^2)$.

Несложно показать, что тензорное произведение ассоциативно, коммутативно и билинейно. Кроме того, оно обобщается и на другие математические объекты. В частности, нас будет интересовать тензорное произведение матриц, которое носит название произведение Кронекера и обладает как указанными выше, так и другими свойствами [8].

3.2. Многомерные матрицы и операции с ними

Определение 3. Введем обобщение понятия матрицы. Будем называть \mathcal{Y} K -мерной матрицей с размерами $n_1 * n_2 * \dots * n_K$, если

$$\mathcal{Y} = \{y_{i_1, i_2, \dots, i_K}, \{i_k = 1, \dots, n_k\}_{k=1}^K\}.$$

Введем операцию развертки K -мерной матрицы вдоль направления k , которое ставит в соответствие многомерной матрице \mathcal{Y} матрицу $\mathcal{Y}^{(k)}$ размера $\prod_{l \neq k} n_l * n_k$. Строки матрицы $\mathcal{Y}^{(k)}$ — это срез значений тензора \mathcal{Y} размера $1 * n_k$ при зафиксированных $i_{k+1} = 1, \dots, i_K = 1, i_1 = 1, \dots, i_{k-1} = 1$ и изменяющемся $i_k = 1, \dots, n_k$. В случае обычных матриц ($K = 2$) операция развертки сводится к транспонированию: $\mathcal{Y}^{(1)} = \mathcal{Y}^T$, а $\mathcal{Y}^{(2)} = \mathcal{Y}$.

Стандартные операции (сложение K -мерных матриц и умножение на скаляр) вводятся естественным образом (поэлементно), как и скалярное произведение:

$$\langle \mathcal{Z}, \mathcal{Y} \rangle = \sum_{i_1, i_2, \dots, i_K} z_{i_1, i_2, \dots, i_K} y_{i_1, i_2, \dots, i_K}.$$

Помимо этих операций введем еще одну: умножение K -мерной матрицы на матрицу вдоль направления. Пусть B — это некоторая матрица размера $n_k * n_k^*$. Тогда будем говорить, что K -мерная матрица \mathcal{Z} размеров $n_1 * \dots * n_{k-1} * n_k^* * n_{k+1} * \dots * n_K$ — это результат умножения \mathcal{Y} вдоль направления k на матрицу B , и обозначать $\mathcal{Z} = \mathcal{Y} \otimes_k B$, если $\mathcal{Z}^{(k)} = \mathcal{Y}^{(k)} B$. Отметим, что умножение вдоль отличающихся направлений коммутативно, а вдоль одинаковых может быть сведено к одному умножению:

$$(\mathcal{Y} \otimes_k B) \otimes_l C = \begin{cases} (\mathcal{Y} \otimes_l C) \otimes_k B, & k \neq l; \\ \mathcal{Y} \otimes_k (BC), & k = l. \end{cases} \quad (1)$$

Умножение многомерных матриц вдоль направления тесно связано с произведением Кронекера. Пусть \mathcal{Y} — это многомерная матрица размера $n_1 * \dots * n_K$, а $B_k, k = 1, \dots, K$ — матрицы размера $n_k * p_k$. Рассмотрим операцию vec , которая ставит в соответствие многомерной матрице вектор-столбец, содержащий все ее элементы. Тогда справедливо

$$\text{vec}(\mathcal{Y} \otimes_1 B_1 \dots \otimes_K B_K) = (B_1 \otimes B_2 \otimes \dots \otimes B_K) \text{vec}(\mathcal{Y}). \quad (2)$$

Сравним вычислительную сложность обеих частей равенства (2). Для упрощения предположим, что все матрицы B_k квадратные и имеют размер $n_k * n_k$, а $N = \prod n_k$. Тогда вычисление левой части требует $O(N \sum n_k)$ операций, в то время как для подсчета правой части необходимо $O(N^2)$ операций. Таким образом, использование умножения вдоль направления позволяет получить заметно меньшую вычислительную сложность.

4. Тензорное произведение словарей функций

Для решения задач аппроксимации 3 и 4 необходимо выбрать класс функций F , в котором будет искажаться решение соответствующей оптимизационной задачи. Рассмотрим подход, основанный на тензорном произведении словарей (некоторых наборов функций). Вернемся к методам аппроксимации, перечисленным в разделе 2.2 (сплайны, линейная регрессия, гауссовские процессы, разложение по словарю параметрических функций). Все эти методы аппроксимации объединяет общее свойство: полученную модель можно выписать в виде $f(x) = \sum_{j=1}^p \alpha_j \psi_j(x)$, где вид функций $\psi_j(x)$ зависит от конкретного метода аппроксимации. Таким образом, аппроксимация f для каждого из этих подходов может рассматриваться как разложение по словарю $\{\psi_j(x)\}_{j=1}^p$.

Теперь определим класс функций, в котором будем искать решение задач 3 и 4. Введем следующие обозначения для всех $k = 1 \dots K$:

- $\sigma_k = \{x_{i_k}^k \in X_k\}_{i_k=1}^{n_k}$, $X_k \subset \mathbb{R}^{d_k}$ — фактор декартова произведения;
- $\Delta_k = \{\psi_{j_k}^k\}_{j_k=1}^{p_k}$ — словарь функций, определенных на X_k .

Предположим, что словари функций Δ_k заданы. Сформируем словарь функций, определенных на $X_1 \times \dots \times X_K$, с помощью тензорного произведения функций из Δ_k :

$$\Delta_{\text{tensor}} = \{\psi_{j_1}^1 \otimes \psi_{j_2}^2 \otimes \dots \otimes \psi_{j_K}^K, \{j_k = 1, \dots, p_k\}_{k=1}^K\}.$$

Пусть линейное пространство функций F_{tensor} состоит из всех возможных линейных комбинаций функций из Δ_{tensor} . Выпишем представление некоторой функции $f \in F_{\text{tensor}}$ в общем виде с использованием многомерных матриц:

$$f(x) = \mathcal{A} \otimes_1 \vec{\psi}_1(x^1) \otimes_2 \vec{\psi}_2(x^2) \dots \otimes_K \vec{\psi}_K(x^K), \quad (3)$$

где $\vec{\psi}_k(x)$ — это вектор-столбец значений функций из словаря Δ_k в точке x , а \mathcal{A} — многомерная матрица коэффициентов разложения по этому словарю. Далее мы будем искать решение задач 3 и 4 именно в классе функций F_{tensor} .

4.1. Штрафная функция

Значения функции g в точках из Σ параметризуются индексами i_1, i_2, \dots, i_K и могут быть представлены в виде многомерной матрицы \mathcal{Y} с размерами $n_1 * n_2 * \dots * n_K$:

$$g([x_{i_1}^1, x_{i_2}^2, \dots, x_{i_K}^K]) = y_{i_1, i_2, \dots, i_K} \Rightarrow g(\Sigma) = \mathcal{Y}. \quad (4)$$

Обозначим как Ψ_k матрицу значений функций словаря Δ_k в точках из σ_k (размер матриц Ψ_k будет равен $n_k * p_k$) и выпишем функцию ошибки Q , используя выражения (3) и (4):

$$Q(f, \Sigma, g(\Sigma)) = \sum_{x \in \Sigma} (g(x) - f(x))^2 = \langle \hat{\mathcal{Y}} - \mathcal{Y}, \hat{\mathcal{Y}} - \mathcal{Y} \rangle, \hat{\mathcal{Y}} = \mathcal{A} \otimes_1 \Psi_1 \otimes_2 \Psi_2 \dots \otimes_K \Psi_K. \quad (5)$$

Задача аппроксимации 3 сводится к минимизации (5) по $f \in F_{tensor}$. Рассмотрим более общую задачу со штрафом 4, которая включает задачу 3 как частный случай. Для постановки этой задачи необходимо задать не только класс функций F , в котором будет строиться аппроксимация, но и вид штрафующего слагаемого $P_\lambda(f)$. В выборе $P_\lambda(f)$ есть некоторая свобода, поэтому обычно штрафное слагаемое выбирают таким образом, чтобы оно позволяло контролировать изменчивость аппроксимации f и получить вычислительно-эффективные формулы. Предложим подход, удовлетворяющий этим соображениям.

В качестве иллюстрации рассмотрим штраф, ограничивающий изменчивость аппроксимации в X_1 — первом факторе декартова произведения. В качестве меры изменчивости будем использовать квадрат выборочной нормы второй производной f по x^1 , подсчитанной в точках множества Σ :

$$\tilde{p}(f) = \left\| \frac{\partial^2 f}{(\partial x^1)^2}(\Sigma) \right\|_S^2 = \left(\sum_{i_1, \dots, i_K} \sum_{j_1, \dots, j_K} \alpha_{j_1, j_2, \dots, j_K} \frac{\partial^2 \psi_{j_1}^1}{(\partial x^1)^2}(x_{i_1}^1) \psi_{j_2}^2(x_{i_2}^2) \dots \psi_{j_K}^K(x_{i_K}^K) \right)^2. \quad (6)$$

Теперь обобщим этот штраф и будем ограничивать изменчивость аппроксимации в произведении некоторого числа факторов $X_{k_1} \times \dots \times X_{k_q}$. Введем бинарный вектор θ , ненулевые компоненты которого будут соответствовать индексам k_1, \dots, k_q , определяющим вид штрафа. Штраф из примера (6) эквивалентен $\tilde{p}(f) = p_\theta(f)$, где у θ не равна нулю только первая компонента: $\tilde{\theta}_1 = 1$. Для θ с несколькими компонентами, отличными от 0, штраф определяется аналогично (6), но порядок частной производной f будет равен не 2, а $2\|\theta\|_0$ (под L_0 нормой понимается количество ненулевых элементов). Перепишем выражение (6) для произвольного бинарного вектора θ в терминах многомерных матричных операций:

$$p_\theta(f) = \langle \mathcal{A}, \mathcal{A} \otimes_1 (\theta_1 \Omega_1 + (1 - \theta_1) \Psi_1^T \Psi_1) \dots \otimes_K (\theta_K \Omega_K + (1 - \theta_K) \Psi_K^T \Psi_K) \rangle. \quad (7)$$

Каждая матрица Ω_k (размера $p_k * p_k$) в (7) — это выборочная норма¹ вторых частных производных функций словаря Δ_k . Пусть $\partial^2 \Psi^k / \partial x_l^k \partial x_m^k$ — это матрица значений частных производных функций словаря Δ_k в точках из σ_k по компонентам вектора x^k , тогда

$$\Omega_k = \sum_{l=1}^{d_k} \sum_{m=1}^{d_k} \left(\frac{\partial^2 \Psi^k}{\partial x_l^k \partial x_m^k} \right)^T \left(\frac{\partial^2 \Psi^k}{\partial x_l^k \partial x_m^k} \right).$$

Теперь введем вектор параметров регуляризации λ размера $K * 1$, с помощью которого будем взвешивать $p_\theta(f)$ для разных θ при формировании окончательного штрафа $P_\lambda(f)$. Каждая компонента λ_k будет определять штраф на изменчивость аппроксимации в соответствующем X_k . Пусть Θ — это множество всех бинарных векторов длины K за исключением нулевого. Тогда определим штрафную функцию следующим образом:

$$P_\lambda(f) = \sum_{\theta \in \Theta} \left(p_\theta(f) \prod_{k=1}^K (\lambda_k \theta_k + (1 - \theta_k)) \right). \quad (8)$$

Введение вектора λ дает возможность явно определять относительный вклад ошибки аппроксимации $Q(f, \Sigma, g(\Sigma))$ и изменчивости $P_\lambda(f)$ в общую функцию ошибки. Кроме того, использование именно такой параметризации позволяет штрафовать изменчивость в различных факторах по-разному, что может быть крайне востребовано в задачах с анизотропией (мощности факторов n_k сильно отличаются для разных k). Например, если устремить λ_k к 0, то в штрафной функции $P_\lambda(f)$ устремятся к нулю все коэффициенты перед слагаемыми, включающие частные производные с использованием x_k , и изменчивость в факторе X_k не будет штрафоваться.

¹для легко вычисляемых функций (например, сплайнов) Ω_k может быть и интегральной нормой

4.2. Вычисление оптимальных коэффициентов разложения

Утверждение 1. *Решение задачи 4 в классе F_{tensor} со штрафом $P_\lambda(f)$ вида (8) существует и единственно, если матрицы $\Psi_k^T \Psi_k + \lambda_k \Omega_k$ не вырождены. Кроме того, тензор коэффициентов разложения этого решения по словарю Δ_{tensor} вычисляется по формуле*

$$\mathcal{A}^* = \mathcal{Y} \otimes_1 \Psi_1^T (\Psi_1^T \Psi_1 + \lambda_1 \Omega_1)^{-1} \otimes_2 \Psi_2^T (\Psi_2^T \Psi_2 + \lambda_2 \Omega_2)^{-1} \dots \otimes_K \Psi_K^T (\Psi_K^T \Psi_K + \lambda_K \Omega_K)^{-1}. \quad (9)$$

Доказательство. При заданных словарях Δ_k и, соответственно, известном классе функций F_{tensor} поиск оптимальной аппроксимации в этом классе сводится к минимизации функции ошибки по коэффициентам разложения. В свою очередь, задача поиска многомерной матрицы оптимальных коэффициентов разложения \mathcal{A} может быть сведена к обычной задаче линейной регрессии. Действительно, пусть Ψ — это матрица значений функций из словаря Δ_{tensor} в точках Σ . Тогда выражение (5) для функции ошибки Q может быть переписано в виде $Q = (\text{vec}(\mathcal{Y}) - \Psi \text{vec}(\mathcal{A}))^T (\text{vec}(\mathcal{Y}) - \Psi \text{vec}(\mathcal{A}))$. Штрафное слагаемое из (8) в матричных операциях будет выглядеть как $P = \text{vec}(\mathcal{A})^T \Omega \text{vec}(\mathcal{A})$ для некоторой матрицы Ω (которая подсчитывается на основе (7)). Тогда задача 4 будет переписана как задача поиска оптимальных коэффициентов \mathcal{A}^* :

$$\mathcal{A}^* = \underset{\mathcal{A}}{\text{argmin}} \left[(\text{vec}(\mathcal{Y}) - \Psi \text{vec}(\mathcal{A}))^T (\text{vec}(\mathcal{Y}) - \Psi \text{vec}(\mathcal{A})) + \text{vec}(\mathcal{A})^T \Omega \text{vec}(\mathcal{A}) \right].$$

Это стандартная задача гребневой регрессии и ее решение может быть найдено явно:

$$\text{vec}(\mathcal{A}^*) = (\Psi^T \Psi + \Omega)^{-1} \Psi^T \text{vec}(\mathcal{Y}). \quad (10)$$

Заметим, что матрица Ψ может быть подсчитана с помощью произведения Кронекера: $\Psi = \Psi_1 \otimes \dots \otimes \Psi_K$. Учтем структуру Ψ и перепишем второй сомножитель в (10) по (2):

$$\Psi^T \text{vec}(\mathcal{Y}) = \text{vec}(\mathcal{Y} \otimes_1 \Psi_1^T \otimes_2 \Psi_2^T \dots \otimes_K \Psi_K^T). \quad (11)$$

Выпишем представление штрафной матрицы Ω , используя (7):

$$\Omega = \sum_{\theta \in \Theta} (\theta_1 (\lambda_1 \Omega_1) + (1 - \theta_1) \Psi_1^T \Psi_1) \otimes \dots \otimes (\theta_K (\lambda_K \Omega_K) + (1 - \theta_K) \Psi_K^T \Psi_K). \quad (12)$$

Теперь упростим первый множитель в (10), используя свойства произведения Кронекера:

$$(\Psi^T \Psi + \Omega)^{-1} = ((\Psi_1^T \otimes \dots \otimes \Psi_K^T) (\Psi_1 \otimes \dots \otimes \Psi_K) + \Omega)^{-1} = ((\Psi_1^T \Psi_1) \otimes \dots \otimes (\Psi_K^T \Psi_K) + \Omega)^{-1}.$$

Обратим внимание на вид (12) штрафной матрицы Ω и примем во внимание билинейность произведения Кронекера:

$$\begin{aligned} (\Psi^T \Psi + \Omega)^{-1} &= ((\Psi_1^T \Psi_1) \otimes \dots \otimes (\Psi_K^T \Psi_K) + \Omega)^{-1} = ((\Psi_1^T \Psi_1 + \lambda_1 \Omega_1) \otimes \dots \\ &\dots \otimes (\Psi_K^T \Psi_K + \lambda_K \Omega_K))^{-1} = (\Psi_1^T \Psi_1 + \lambda_1 \Omega_1)^{-1} \otimes \dots \otimes (\Psi_K^T \Psi_K + \lambda_K \Omega_K)^{-1}. \end{aligned} \quad (13)$$

Теперь объединим (11) с (13) и перепишем выражение для оптимальных коэффициентов (10), преобразовав его в многомерную матрицу:

$$\mathcal{A}^* = \mathcal{Y} \otimes_1 \Psi_1^T \dots \otimes_K \Psi_K^T \otimes_1 (\Psi_1^T \Psi_1 + \lambda_1 \Omega_1)^{-1} \dots \otimes_K (\Psi_K^T \Psi_K + \lambda_K \Omega_K)^{-1}.$$

Используя свойство (1) умножения многомерной матрицы вдоль направлений, получаем итоговую формулу (9).

Условия на существование и единственность следуют из аналогичных условия для обычной гребневой регрессии. Действительно, матрица $\Psi^T \Psi + \Omega$ представима в виде произведения Кронекера $(\Psi_1^T \Psi_1 + \lambda_1 \Omega_1) \otimes \dots \otimes (\Psi_K^T \Psi_K + \lambda_K \Omega_K)$. Это означает, что $\Psi^T \Psi + \Omega$ не вырождена тогда и только тогда, когда не вырождены все $\Psi_k^T \Psi_k + \lambda_k \Omega_k$. ■

Рассмотрим вопрос выбора вектора регуляризации λ . Применим для его оценки стандартный подход — скользящий контроль (leave-one-out) [2]. Этот метод позволяет оценить ошибку аппроксимации линейной регрессии на тестовых данных, используя только точки обучающей выборки:

$$Q_{loo} = \sum_{i=1}^N \left(y_i - \vec{\psi}(x_i) \text{vec}(\mathcal{A}_{-i}) \right)^2,$$

где коэффициенты разложения $\text{vec}(\mathcal{A}_{-i})$ оцениваются по выборке, из которой исключена точка $\{x_i, y_i\}$. Эта формула может существенно упрощена с помощью матричных преобразований [2]:

$$Q_{loo} = \sum_{i=1}^N \left(\frac{y_i - \vec{\psi}(x_i) \text{vec}(\mathcal{A})}{1 - L_{i,i}} \right)^2, \quad L = \Psi(\Psi^T \Psi + \Omega)^{-1} \Psi^T. \quad (14)$$

Адаптируем формулу (14) к особенностям нашей задачи, приняв во внимание специальную структуру матрицы регрессоров Ψ . Это позволит нам воспользоваться вычислительно эффективными формулами, основанными на операциях с многомерными матрицами.

Утверждение 2. *Ошибка скользящего контроля при использовании (9) может быть подсчитана по формуле*

$$Q_{loo} = \left\langle \mathcal{Y} - \widehat{\mathcal{Y}}, (\mathcal{Y} - \widehat{\mathcal{Y}}) * \left(\mathcal{I} - \mathcal{I} \otimes_1 \text{diag}(L_1) \dots \otimes_K \text{diag}(L_K) \right)^{-2} \right\rangle; \quad (15)$$

где $L_k = \Psi_k (\Psi_k^T \Psi_k + \lambda_k \Omega_k)^{-1} \Psi_k^T$, $\text{diag}(L_k)$ — диагональная матрица, совпадающая с диагональю L_k , а \mathcal{I} — тензор размера $n_1 * \dots * n_K$.

Доказательство. Найдем выражение для матрицы L из (14) с учетом специфики задачи:

$$L = \Psi(\Psi^T \Psi + \Omega)^{-1} \Psi^T = \Psi_1 \otimes \dots \otimes \Psi_K (\Psi_1^T \Psi_1 + \lambda_1 \Omega_1)^{-1} \otimes \dots \otimes (\Psi_K^T \Psi_K + \lambda_K \Omega_K)^{-1} \Psi_1^T \otimes \dots \otimes \Psi_K^T = \left(\Psi_1 (\Psi_1^T \Psi_1 + \lambda_1 \Omega_1)^{-1} \Psi_1^T \right) \otimes \dots \otimes \left(\Psi_K (\Psi_K^T \Psi_K + \lambda_K \Omega_K)^{-1} \Psi_K^T \right). \quad (16)$$

Для подсчета ошибки скользящего контроля необходимы только диагональные элементы матрицы L , которые согласно (16) выражаются через матрицы L_k из (15): $\text{diag}(L) = \text{diag}(L_1) \otimes \dots \otimes \text{diag}(L_K)$. Используя связь (2) между произведением Кронекера и умножением многомерных матриц, преобразуем (14) к скалярному произведению многомерных матриц и получаем искомую формулу (15). ■

Вычислительная сложность формулы (15) составляет $O(\sum p_k^3 + N)$, что позволяет использовать эту оценку ошибки аппроксимации на тестовой выборке для выбора вектора регуляризации λ . В силу факторизованной структуры функции Q_{loo} использование метода покоординатного спуска для ее минимизации будет достаточно эффективным.

4.3. Алгоритм построения аппроксимации

В предыдущем разделе предполагалось, что словари Δ_k заданы, однако это справедливо только для одномерных факторов, в которых можно явным образом построить базис в пространстве сплайнов. Рассмотрим задачу построения словарей Δ_k по заданной обучающей выборке \mathcal{S} . Как уже было указано выше, универсальные методы аппроксимации могут быть представлены как разложение по некоторому словарю параметрических функций. Построение аппроксимации для этих методов сводится к определению оптимальных (в смысле минимизации некоторой функции ошибки) параметров функций, составляющих словарь. Зафиксируем словари $\Delta_k, k \neq l$ и рассмотрим задачу нахождения словаря Δ_l .

Функция ошибки Q в этом случае может быть переписана следующим образом:

$$\begin{aligned} Q(f, \Sigma, g(\Sigma)) &= \sum_{i_l=1}^{n_l} \sum_{i_k, k \neq l} \left(y_{i_1, \dots, i_K} - \sum_{j_l=1}^{p_l} \sum_{j_k, k \neq l} \alpha_{j_1, \dots, j_K} \vec{\psi}_1(x_{i_1}^1) \dots \vec{\psi}_K(x_{i_K}^K) \right)^2 = \\ &= \sum_{i_k, k \neq l} \sum_{i_l=1}^{n_l} \left(y_{i_l}^{i_k, k \neq l} - \sum_{j_l=1}^{p_l} \beta_{j_l}^{j_k, k \neq l} \vec{\psi}_l(x_{i_l}^l) \right)^2 = \text{trace} \left((\mathcal{Y}^{(l)} - \Psi_l B_l)^T (\mathcal{Y}^{(l)} - \Psi_l B_l) \right). \end{aligned} \quad (17)$$

Задача минимизации (17) по функциям $\{\psi_{j_l}^l\}_{j_l=1}^{p_l} = \Delta_l$ (точнее, по их параметрам) эквивалента решению задачи 1 по выборке $S_l = \{\sigma_l, \mathcal{Y}^{(l)}\}$. Таким образом, построение словарей Δ_k сводится к K задачам аппроксимации, которые могут быть решены с помощью универсальных методов.

Сформулируем полный алгоритм построения аппроксимации, предполагая, что обучающая выборка действительно представляет собой декартово произведение с известной структурой.

- 1) Формируем из значений функции $\{y_i\}_{i=1}^N$ многомерную матрицу $\mathcal{Y} = g(\Sigma)$.
- 2) Для всех $k = 1, \dots, K$ строим аппроксимацию, используя обучающую выборку $S_k = \{\sigma_k, \mathcal{Y}^{(k)}\}$. Решаем эту задачу с помощью одного из стандартных методов аппроксимации. Выбор метода должен быть основан на параметрах получившейся задачи. Например, для $d_k = 1$ логично использовать сплайны, а для больших n_k при $d_k > 1$ — словарь нелинейных параметрических функций.
- 3) Находим оптимальный параметр регуляризации λ^* , минимизируя (15).
- 4) Подсчитываем по (9) коэффициенты разложения \mathcal{A}^* по словарю Δ_{tensor} для λ^* .

Рассмотрим предложенный алгоритм с точки зрения вычислительной эффективности. Оценим сложность нахождения словаря Δ_k в каждом факторе. При решении задачи аппроксимации по выборке $S_k = \{\sigma_k, \mathcal{Y}^{(k)}\}$ размерность y будет достаточно велика и равна N/n_k , а объем обучающей выборки составит n_k . Если в k -м факторе словарь строится с помощью сплайнов, то сложность будет $O(n_k^3 + n_k^2(N/n_k)) = O(n_k^3 + N n_k)$. Заметим, что для повышения вычислительной эффективности можно использовать В-сплайны [3], которые позволят понизить вычислительную сложность до $O(n_k + N)$ за счет ограниченного носителя соответствующих базисных функций. Для гауссовских процессов критически важным является объем выборки, который в силу факторизации существенно сокращается. Вычислительная сложность решения подзадачи аппроксимации в k -м факторе с помощью этого метода будет равна $O((N/n_k)n_k^3) = O(N n_k^2)$. При использовании разложения по словарю параметрических функций в зависимости от метода обучения сложность составляет от $O(n_k p_k (N/n_k)) = O(N p_k)$ до $O(n_k p_k (N/n_k) + n_k p_k^2 d_k^2 + p_k^3 d_k^3) = O(N p_k + n_k p_k^2 d_k^2 + p_k^3 d_k^3)$. Следует заметить, что объем словаря Δ_{tensor} равен $P = \prod p_k$, что позволяет выбирать достаточно малые p_k . Таким образом, учет структуры выборки позволяет получить линейную зависимость от объема выборки для всех методов аппроксимации и уменьшить коэффициент перед некоторыми линейными слагаемыми (в частности, множителями с p_k для последнего метода).

Сложность вычисления оптимальной K -мерной матрицы коэффициентов разложения \mathcal{A} по (9) будет равна $O\left(\sum_{k=1}^K p_k^3 + \sum_{k=1}^K N(n_k + p_k) \prod_{l=1}^k p_l/n_l\right)$. Так как для всех методов $p_k/n_k \leq 1$, то вычислительная сложность не будет превосходить $O(\sum_{k=1}^K (n_k^3 + N n_k))$. При минимизации (15) дополнительная стоимость каждой итерации (по сравнению с вычислением \mathcal{A}) составляет $O(\sum_{k=1}^K n_k \prod_{l=1}^K p_l)$ и не превосходит $O(N \sum_{k=1}^K n_k)$.

Для упрощения анализа будем предполагать, что число точек n_k одинаково во всех факторах и равно $N^{1/K}$. Тогда наиболее тяжелой операцией будет построение регрессии с помощью гауссовских процессов со сложностью $O(N^{1+2/K})$. Если же использовать другие

методы аппроксимации, то сложность не превысит $O(N^{1+1/K})$. Кроме того, если в задаче есть одномерные факторы (а в подавляющем большинстве прикладных задач это условие выполняется), то сложность вычисления \mathcal{U} и минимизации ошибки скользящего контроля будет уменьшаться за счет использования В-сплайнов.

Таким образом, вычислительная сложность метода относительно объема выборки будет несущественно превышать линейную, а константа перед этой зависимостью будет достаточно мала.

5. Экспериментальные результаты

Рассмотрим ряд прикладных задач из области аэрокосмического проектирования. Типичным примером таких задач является зависимость некоторой аэродинамической характеристики от условий полета и конфигурации объекта. Структуры множеств точек обучающей выборки Σ рассматриваемых задач представлены в табл. 1.

Т а б л и ц а 1

Факторизация задач

Задача	K	d	d_k	N	n_k
1	3	3	$1 \times 1 \times 1$	780	$10 \times 26 \times 3$
2	3	5	$1 \times 3 \times 1$	502460	$679 \times 148 \times 5$
3	7	7	$1 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1$	540000	$25 \times 12 \times 4 \times 6 \times 5 \times 3 \times 5$

Сравним предложенный метод аппроксимации с линейной регрессией (оценка коэффициентов с помощью гребневой регрессии), разложением по словарю параметрических функций (однослойная нейронная сеть, обучаемая алгоритмом Левенберга—Марквардта [9]) и регрессией на гауссовских процессах (пакет GPML [10], разработанный автором книги [5]). Для оценки ошибки аппроксимации будем использовать стандартный подход: разобьем выборку данных на обучающую и тестовую части. Первая из этих подвыборок будет использоваться для построения модели, а вторая — для определения качества построенной аппроксимации. Заметим, что тестовую выборку необходимо выделять таким образом, чтобы сохранить структуру декартова произведения точек обучающей выборки. В качестве ошибки аппроксимации будем использовать среднеквадратичную ошибку, нормированную на ошибку аппроксимации константой ($\#\Sigma$ — мощность множества Σ):

$$E(f, \Sigma, g(\Sigma)) = \sqrt{\frac{\sum_{x \in \Sigma} (g(x) - f(x))^2}{\sum_{x \in \Sigma} (g(x) - \bar{g}(\Sigma))^2}}, \quad \bar{g}(\Sigma) = \frac{1}{\#\Sigma} \sum_{x \in \Sigma} g(x).$$

Будем подсчитывать ошибку не только на тестовой выборке, но и на обучающей, а также учтем время обучения модели. Построение аппроксимации с помощью регрессии на гауссовских процессах для задач 2 и 3 не представляется возможным в силу большого объема обучающей выборки (как было отмечено выше, этот метод требует $O(N^3)$ операций и работает разумное время для N порядка тысячи точек). Результаты экспериментов приведены в табл. 2.

Т а б л и ц а 2

Ошибки аппроксимации и время построения модели

	лин. регр.			нейрон. сети			гаус. проц.			предл. подход		
	t	E_{tr}	E_{test}	t	E_{tr}	E_{test}	t	E_{tr}	E_{test}	t	E_{tr}	E_{test}
1	0.003	0.81	0.65	12.5	0.04	0.38	28.0	0.002	0.42	0.11	0.05	0.12
2	0.08	0.83	0.84	9500	0.048	11.6	-	-	-	74.5	0.021	0.022
3	0.02	0.14	0.13	3800	0.002	0.003	-	-	-	2.96	0.0003	0.0009

Заметим, что увеличение ошибки на тестовой выборке по сравнению с обучающей в задачах 1 и 2 для универсальных нелинейных методов аппроксимации связано с анизотропией данных. В силу существенной неравномерности распределения точек по пространству дизайна построенные этими методами модели обладают нерегулярным поведением, см. пример среза аппроксимации гауссовскими процессами для задачи 1 на рис. 1. Схожая проблема возникает и для искусственных нейронных сетей в задачах 1 и 2. В то же время предложенный метод построения аппроксимации позволяет получить качественную аппроксимацию не только для задачи 1 (рис. 2), но и для более сложных зависимостей, см. срез построенной модели для задачи 2 на рис. 3. Истинная функция $f(x)$ в задаче 3 имеет не очень сложную структуру, что косвенно помогло избежать резкого скачка ошибки на тестовой выборке для искусственной нейронной сети, хотя метод все же уступил в точности предложенному подходу. Также следует отметить время построения моделей — для предложенного подхода оно существенно меньше, чем для универсальных нелинейных методов аппроксимации.

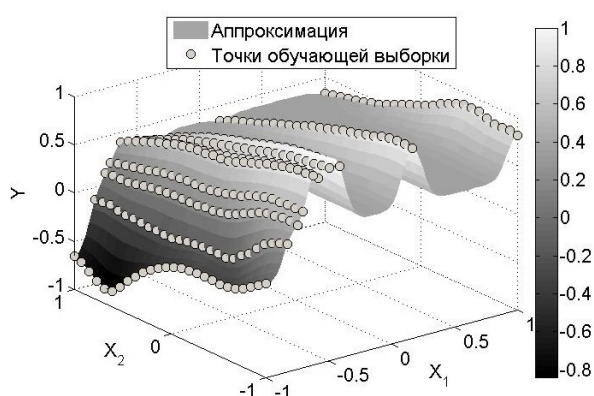


Рис. 1. Регрессия на гауссовских процессах: срез модели в задаче 1

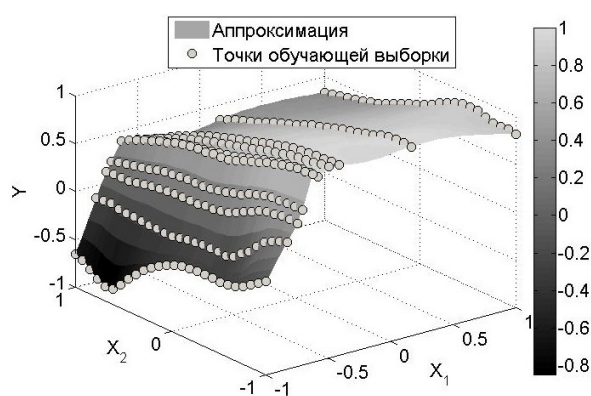


Рис. 2. Предложенный подход: срез модели в задаче 1

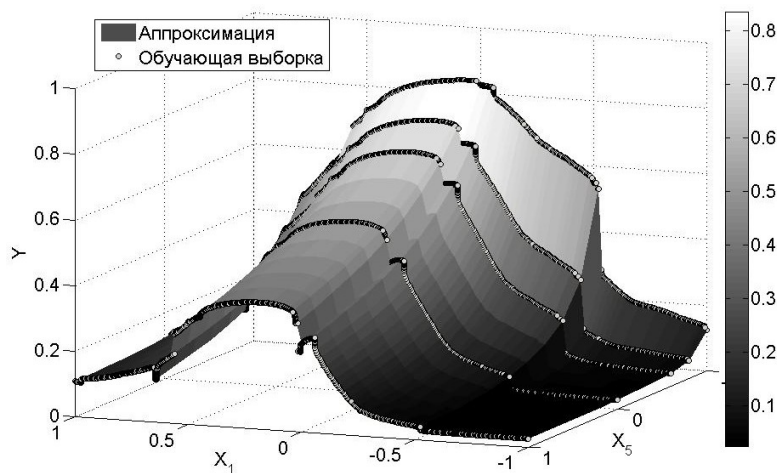


Рис. 3. Предложенный подход: срез модели в задаче 2

6. Заключение

Предложенный в работе метод формирования словаря с помощью тензорного произведения позволяет как свести одну задачу аппроксимации к ряду более простых, так и получить богатый класс функций, способный хорошо приближать сложные многомерные зависимости. Изменчивость модели ограничивается с помощью штрафа специального вида, который построен таким образом, чтобы дать возможность контролировать гладкость

модели в каждом факторе независимо от других. Для нахождения оптимальных коэффициентов разложения получены вычислительно-эффективные аналитические формулы.

Низкая вычислительная сложность алгоритма позволяет быстро строить модели даже для крайне больших объемов выборки, а контроль гладкости полученной модели успешно решает проблему анизотропии, характерную для прикладных задач. В качестве основного недостатка можно отметить существенное ограничение на структуру выборки, которое, впрочем, выполняется для многих задач инженерного проектирования. В частности, предложенный подход был успешно применен для решения ряда актуальных задач из аэрокосмической отрасли.

Работа выполнена при поддержке Лаборатории структурных методов анализа данных в предсказательном моделировании, МФТИ, грант правительства РФ дог. 11.G34.31.0073.

Литература

1. *Forrester A.I.J., Sobester A., Keane A.J.* Engineering design via surrogate modelling: a practical guide. — New York: Wiley, 2008.
2. *Tibshirani R., Hastie T., Friedman J.* The elements of statistical learning: data mining, prediction and inference, 2nd edition. — New York: Springer, 2009.
3. *de Boor C.* A Practical Guide to Splines, 2nd edition. — Berlin: Springer-Verlag, 2001.
4. *Schölkopf B., Herbrich R., Smola A.* A generalized representer theorem // Computational Learning Theory. — 2001. — V. 2111. — P. 416–426.
5. *Rasmussen C.E., Williams C.K.I.* Gaussian processes for machine learning. — Cambridge: MIT Press, 2006.
6. *Беляев М.Г., Бурнаев Е.В., Любин А.Д.* Методика формирования функционального словаря в задаче аппроксимации многомерной зависимости // Сборник докладов конференции «математические методы распознавания образов «ММРО-15». — 2011. — С. 146–149.
7. *Бурнаев Е.В., Янович Ю.А.* Разреженные гауссовские процессы // Труды 54-й научной конференции МФТИ «Современные проблемы фундаментальных и прикладных наук». — 2011. — Т. ФУПМ-2. — С. 85–86.
8. *Graham A.* Kronecker Products and Matrix Calculus: With Applications. — New York: Wiley, 1982.
9. Matlab Neural Network Toolbox.
<http://www.mathworks.com/products/neural-network/index.html>
10. Gaussian processes for machine learning toolbox.
<http://www.gaussianprocess.org/gpml/code/matlab/doc/index.html>

Поступила в редакцию 28.11.2012.