

Аппроксимация многомерных зависимостей по структурированным выборкам

Беляев М.Г.

Аннотация

В работе рассматривается задача многомерной аппроксимации по выборкам с факторным планом (полным или неполным). Заданный план несет в себе знания из предметной области: факторы с бóльшим числом уровней соответствуют группам переменных, которые сильнее влияют на значение функции. При решении задачи аппроксимации необходимо учитывать эти знания: срезы построенной модели должны быть достаточно гладкими для факторов с малым числом уровней. Универсальные методы аппроксимации не учитывают как структуру выборки, так и требования к модели. В работе предложен структурно-ориентированный метод аппроксимации, а именно специальным образом выбраны класс функций и регуляризация. Оптимальное решение в этом классе в случае полного факторного плана может быть эффективно найдено в явном виде с помощью свойств произведения Кронекера. В случае неполного плана вычислительная сложность явных формул недопустимо высока. В работе предложен эффективный метод поиска оптимального решения и доказано, что его сложность линейно зависит от числа пропущенных точек.

1 Введение

В задачах инженерного проектирования часто необходимо изучать сложные физические явления. Эти явления, как правило, моделируются с помощью систем дифференциальных уравнений, которые в большинстве случаев не решаются аналитически. Для нахождения решения применяются громоздкие численные методы, использование которых существенно повышает время моделирования. В результате даже при текущем уровне развития вычислительной техники типичное время одного расчета с использованием физической модели может достигать до нескольких дней [1]. Для уменьшения временных затрат на этапе предварительного проектирования используют математические модели, построенные с помощью методов анализа данных по заданной обучающей выборке, которая состоит из пар “точка” (описание объекта и условий функционирования, входной вектор) – “значение функции в точке” (характеристика объекта, выходной вектор). Такие модели позволяют получить лишь приближенное описание физического явления, но обеспечивают очень высокую скорость моделирования [2].

Переменные задачи (компоненты входного вектора) зачастую можно разбить на несколько групп с разным физическим смыслом [3]. Например, при исследовании аэродинамики самолета к первой группе можно отнести геометрические параметры,

а ко второй — режим полета (угол атаки, скорость). В таких случаях при формировании выборки зачастую используется следующая стратегия: для каждого зафиксированной группы переменных из одного фактора проводится эксперимент для набора заранее выбранных уровней переменных из второго фактора, причем от эксперимента к эксперименту этот набор не меняется. Такая постановка эксперимента приводит к использованию полных факторных планов. Если же часть точек пропущена (эксперимент не завершен, генератор данных не может выдать значения в некоторых точках и т.д.), то план будет неполным факторным.

Количество уровней факторов плана, как правило, существенно различаются для разных факторов. Это связано с тем, что при выборе плана экспериментов принимаются во внимание знания из предметной области, которые говорят о неравнозначной зависимости моделируемой функции от различных групп параметров. В некоторых случаях бывает известен вид зависимости по какому-либо параметру [4]. Например, в случае квадратичной зависимости мощность соответствующего фактора выбирается равной трем, поскольку больший размер будет избыточным. В общем случае явный вид зависимости не известен, однако относительная степень ее сложности может быть оценена на основе знаний из предметной области и затем учтена при выборе плана выборки. Для тех групп переменных, которые вносят существенный вклад в изменчивость функции, используются большие мощности факторов, чем для параметров, от которых функция зависит достаточно гладко. Таким образом, структура множества точек обучающей выборки может нести в себе знания из предметной области о характере моделируемой зависимости.

Наличие гладко влияющих на изменчивость функции факторов, для которых, как следствие, было выбрано малое число уровней, во многих задачах предъявляет дополнительные требования к методу аппроксимации. Построенная модель должна иметь достаточно гладкие срезы вдоль этого фактора (при зафиксированных переменных из всех остальных факторов). Заметим, что малое число уровней может быть выбрано и по другим соображениям. Например, переменные из какого-либо фактора варьировать достаточно дорого (в смысле временных или материальных затрат), поэтому при проведении эксперимента значения соответствующих переменных меняются редко. Зависимость истинной функции от переменных из таких факторов может быть достаточно сложной. Однако и в этом случае целесообразно получить модель с достаточно гладкими срезами, поскольку по малому числу точек можно построить только достаточно простую зависимость.

Универсальные методы аппроксимации, не использующие знания о структуре данных, не позволяют учесть знания о плане эксперимента и зачастую не удовлетворяют требованию о гладкости срезов (как правило, в случае наличия факторов с существенно отличающимся числом уровней). Для схожих постановок задачи (есть некоторые предположения о структуре плана задачи) существуют некоторые достаточно эффективные решения. Хорошо известен подход, основанный на тензорном произведении сплайнов [6] и позволяющий вычислительно-эффективно строить интерполяцию для выборок с полным факторным планом, состоящем только из одномерных факторов. Ограничение на размерность факторов является существенным недостатком методов. Кроме того этот метод теряет вычислительную эффективность работе с неполным декартовым произведением. Также известен подход для работы с так называемыми разреженными решетками [7]. Основное отличие от рассматриваемой постановки задачи заключается в том, что точки обучающей выборки должны

образовывать специальное подмножество полного факторного плана с одномерными факторами, в то время как нас интересует более общий случай произвольного подмножества. Существуют и работы, посвященные произвольной структуре пропущенных точек [8, 9], однако в них предлагаются эвристические итеративные процедуры, которые даже на искусственных примерах не всегда сходятся к решению.

Следует отметить работу [10] по регрессии на гауссовских процессах, в которой предполагается, что план является полным факторным (в том числе с многомерными факторами). Эта работа посвящена разработке ковариационной функции специального вида, учитывающей этот вид данных. Такой подход не позволяет контролировать гладкость модели, но, возможно, введение некоторых дополнительных априорных предположений на параметры ковариационной функции позволило бы неявно внести знания о структуре исходной зависимости в построенную модель. Однако метод не обобщается на случай неполного факторного плана, кроме того, его вычислительная сложность недопустимо высока.

Также следует отметить популярный метод MARS (Multivariate adaptive regression splines) [11]. Этот подход (также как и данная работа) использует словарь функций, сформированный с помощью тензорного произведения. Метод не использует явный контроль гладкости, однако, учитывая адаптивный выбор словаря, можно надеяться, что полученная модель будет соответствовать структуре данных. Коэффициенты разложения (и сам словарь) выбираются с помощью итеративной эвристической процедуры включения-исключения, что обуславливает меньшую точность, чем использование полного словаря. Кроме того, в силу особенностей формирования словаря (используются очень простые функции), полученная модель будет иметь разрывные производные, что не удовлетворяет требованиям большинства задач инженерного проектирования.

В работе мы рассмотрим задачу аппроксимации как задачу поиска оптимального решения в некотором функциональном классе с ограничением на изменчивость модели (раздел 2). Некоторые особенности задач инженерного проектирования, подталкивающие к использованию факторного плана экспериментов (полного или неполного), будут описаны в разделе 2.3. В разделе 4.1 мы предложим специальный метод решения задачи аппроксимации с помощью разложения по структурно-ориентированному словарю функций. В основе этого метода лежит использование тензорного произведения словарей функций и специальный штраф, позволяющий достичь контроля гладкости модели вдоль различных срезов (раздел 4.2). Решение задачи аппроксимации в заданном классе сводится к поиску оптимального тензора коэффициентов разложения по словарю (раздел 5). Выбранный штраф на изменчивость позволяет получить квадратичную по коэффициентам разложения задачу. Для полного факторного плана решение этой задачи может быть получено в явном виде (раздел 5.1). Использование тензорной арифметики и свойств произведения Кронекера позволяет эффективно вычислять тензор коэффициентов со сложностью $O(P \sum p_k + \sum p_k^3)$ вместо $O(\prod p_k^3)$ для стандартной формулы, не учитывающей структуру задачи (здесь p_k - это размеры словарей в каждом из факторов, а P - размер словаря, сформированного посредством тензорного произведения). В случае неполного факторного плана (раздел 5.2) нарушение структуры выборки приводит к невозможности использования явных эффективных формул. Однако, задачу оптимизации можно переформулировать таким образом, что она эффективно решается с помощью метода сопряженных градиентов, причем число необходимых шагов оптимизации на единицу превосходит число

пропусков. Общая сложность метода составляет $O\left(\sum p_k^3 + (\tilde{N} - N + 1) P \sum p_k\right)$ и также значительно превосходит по эффективности стандартный подход (где $\tilde{N} - N$ - число пропущенных точек). Столь низкая вычислительная сложность позволяет решать задачи аппроксимации, требующие использование словарей больших размеров (P может доходить до сотен тысяч).

2 Постановка задачи

В этом разделе мы рассмотрим типичную постановку задачи аппроксимации, а также ее частные случаи, возникающие в задачах инженерного проектирования.

2.1 Задача аппроксимации

Рассмотрим задачу аппроксимации функции по конечному набору ее значений в заданных точках. Пусть g — это некоторая непрерывная многомерная функция, $g : (X \subset R^d) \rightarrow R^1$ (где X — некоторый компакт в R^d). Будем называть обучающей выборкой S совокупность множества точек Σ (мощности N) и значений функции g в точках из этого множества:

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

Рассмотрим задачу в следующей постановке:

Задача 1 Пусть заданы обучающая выборка S , некоторый класс функций F и штрафная функция $P_\lambda : F \rightarrow R_+^1$. Необходимо найти $f^* \in F$, минимизирующую функцию ошибки со штрафом $R(f, \Sigma, g(\Sigma)) = Q(f, \Sigma, g(\Sigma)) + P_\lambda(f)$, а именно:

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

Использование штрафа, ограничивающего изменчивость модели, — это распространенный подход, позволяющий явно контролировать изменчивость модели. Например, в качестве такого штрафа может быть выбрана норма вторых производных f (см., например, сглаживающие сплайны [12]) или норма f в некотором гильбертовом пространстве функций (ядерная гребневая регрессия [13]).

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 (где F - гильбертово пространство):

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

2.2 Факторный план эксперимента

Введем определение полного факторного плана эксперимента. Пусть заданы некоторые множества точек

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

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

Определение 2 Будем называть множество точек Σ_{full} полным факторным планом, если Σ_{full} декартово произведение всех факторов:

$$\Sigma_{full} = \sigma_1 \times \dots \times \sigma_K = \{[x_{i_1}^1, x_{i_2}^2, \dots, x_{i_K}^K], \{i_k = 1, \dots, n_k\}_{k=1}^K\}.$$

Элементы множества Σ_{full} — это вектора размерности $d = \sum_{k=1}^K d_k$, а объем выборки (мощность Σ_{full}) равен $N_{full} = \prod_{k=1}^K n_k$.

2.3 План эксперимента в инженерном проектировании

В задаче 1 рассматривается план эксперимента Σ произвольного вида. Как правило, предполагается, что множество Σ не обладает никакой структурой и равномерно покрывает область определения функции g . Однако, для задач инженерного проектирования характерен план эксперимента специального вида, причем такой выбор Σ обусловлен особенностями этих задач.

В приложениях часто встречаются задачи, в которых все переменные можно разбить на две группы, первая из которых соответствует описанию объекта, а вторая определяет условия эксперимента. Например, при исследовании аэродинамики крыла первой группе переменных может соответствовать его геометрическое описание, второй группе — угол атаки крыла и скорость воздушного потока, а в качестве выходной переменной может выступать подъемная сила, которой обладает крыло. Такая структура делает целесообразным использование одинаковой сетки значений (набора векторов) переменных из второй группы для каждого зафиксированного значения вектора переменных из первой группы. В таком случае множество точек $\Sigma = \Sigma_{full}$ будет полным факторным планом с двумя факторами. В некоторых задачах факторизация имеет более сложную структуру и переменные разбиваются на большее число групп, чем две (например, угол атаки и скорость можно разбить на два фактора).

Не менее часто встречаются задачи, в которых используется неполный факторный план эксперимента, то есть множество точек обучающей выборки представляет собой декартово произведение с пропусками (Σ не совпадает с Σ_{full} , а является его подмножеством). Такой план может быть обусловлен одной из следующих причин:

1. В качестве неизвестной функции g выступает некоторый программный код, численно решающий какие-либо уравнения математической физики. Для такого рода генераторов данных достаточно распространена ситуация, когда в некоторых точках $x \in X$ значение функции не может быть найдено (итеративный процесс поиска решения не сходится или сходится с крайне низкой скоростью). В результате, если предварительный план эксперимента представлял собой декартово произведение, то после выполнения численных экспериментов итоговый план Σ окажется лишь некоторым подмножеством исходного полного факторного плана Σ_{full} .
2. Эксперимент (генерация данных) еще не завершен, поэтому в части точек значения отсутствуют. Такая ситуация может иметь место, если подсчет значения g в одной точке требует значимое время.

3. Выборка описывает поведение некоторого объекта, для которого характерны несколько режимов работы. Для каждого режима план эксперимента строится как декартово произведение, а множество точек обучающей выборки Σ в таком случае является объединением нескольких (как правило, пересекающихся) декартовых произведений. В качестве типичного примера можно рассмотреть самолет, у которого можно выделить три режима эксплуатации: взлет, крейсерский полет и посадка. Для этих режимов характерны существенно отличающиеся наборы пар (угол атаки - скорость полета), выступающие в качестве входной переменной $x \in R^2$.

В работе мы рассмотрим два частных случая задачи аппроксимации, соответствующих описанным выше планам эксперимента.

Задача 2 Решить задачу 1 при условии, что план эксперимента является полным факторным, $\Sigma = \Sigma_{full} = \sigma_1 \times \sigma_2 \times \dots \times \sigma_K$.

Задача 3 Решить задачу 1 при условии, что план эксперимента является неполным факторным, $\Sigma \subset \Sigma_{full} = \sigma_1 \times \sigma_2 \times \dots \times \sigma_K$.

В обоих случаях мы предполагаем, что область определения функции g есть декартово произведение: $X = X_1 \times X_2 \times \dots \times X_K$.

В этой постановке не учитываются характерная особенность факторных планов эксперимента, возникающих в задачах инженерного проектирования, а именно существенно различное число уровней в факторах. Это свойство выборок накладывает определенные требования на метод аппроксимации. Во многих приложениях требуется, чтобы модель была достаточно гладкой вдоль факторов с малым числом уровней. Это требование, которое никак не отражено в постановке задач 2, 3, мы учтем при выборе класса функций F и штрафной функции $P_\lambda(f)$.

3 Тензоры и операции с ними

Дальнейшее изложение удобно вести на языке тензоров, поэтому введем некоторые связанные с ними определения и операции.

Определение 3 Будем называть тензором [14] \mathcal{Y} K -мерный массив размеров $n_1 * n_2 * \dots * n_K$:

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

Стандартные операции (сложение тензоров и умножение на скаляр) вводятся естественным образом (поэлементно), как и скалярное произведение:

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

Несложно проверить, что введенное таким образом скалярное произведение удовлетворяет всем необходимым аксиомам (с нулевым элементом, для которого все z_{i_1, i_2, \dots, i_K} равны 0).

Помимо этих операций введем еще две: умножение тензора на матрицу вдоль направления k и развертку тензора вдоль направления k . Операция развертки 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 $\mathcal{Y}^{(2)} = \mathcal{Y}$.

Теперь рассмотрим операцию умножения тензора на матрицу вдоль направления k . Пусть B — это некоторая матрица размера $n_k * n_k^*$. Тогда будем говорить, что тензор \mathcal{Z} размера $n_1 * \dots * n_{k-1} * n_k^* * n_{k+1} * \dots * n_K$ — 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:

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

Умножение многомерных матриц вдоль направления тесно связано с произведением Кронекера [15]. Пусть \mathcal{Y} — это многомерная матрица размера $n_1 * \dots * n_K$, а B_k — матрицы размера $n_k * p_k$ (для $k = 1, \dots, K$). Рассмотрим операцию vec , которая ставит в соответствие многомерной матрице вектор-столбец, содержащий все ее элементы. Тогда справедливо

$$(1) \quad \text{vec}(\mathcal{Y} \otimes_1 B_1 \dots \otimes_K B_K) = (B_1 \otimes \dots \otimes B_K) \text{vec}(\mathcal{Y}).$$

Сравним вычислительную сложность обеих частей равенства (1). 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 $N \sum n_k$ операций сложения и умножения, в то время как для подсчета правой части необходимо N^2 операций без учета сложности произведения Кронекера (где под операцией подразумевается либо одно сложение, либо умножение, либо последовательное умножение и сложение). Таким образом, использование умножения вдоль направления позволяет уменьшить сложность такого рода вычислений.

4 Структурно-ориентированный подход

По условию задачи аппроксимации должны быть заданы обучающая выборка, класс функций F и штрафная функция $P_\lambda(f)$, однако пока обсуждалась лишь структура обучающей выборки. В этом разделе мы предложим F и $P_\lambda(f)$ специального вида, чтобы с одной стороны эффективно учесть знания о структуре функции, а с другой удовлетворить требованиям о гладкости модели вдоль факторов с малым числом уровней.

4.1 Класс функций

Для решения задачи 1 необходимо выбрать класс функций F , в котором будет искать решение соответствующей оптимизационной задачи. Мы будем использовать предложенный в [16] подход, основанный на тензорном произведении словарей (для многомерных факторов) или базисов (для одномерных факторов) функций. Опишем

более детально этот класс функций. Пусть $\Delta_k = \{\psi_{j_k}^k\}_{j_k=1}^{p_k}$ — словарь функций, определенных на X_k (для всех $k = 1 \dots K$). Словари Δ_k могут быть построены одним из методов аппроксимации, в которых модель может быть представлена как линейное разложение по словарю. Например, это может быть разложение по словарю параметрических функций [17], линейная регрессия, регрессия на гауссовских процессах [18], сплайны [12]. Выбор конкретного метода аппроксимации должен определяться размерностью фактора d_k и количеством точек n_k в этом факторе.

Сформируем словарь функций, определенных на $X_1 \times \dots \times X_K$, с помощью тензорного произведения функций из Δ_k :

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

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

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

где $x = [x^1, x^2, \dots, x^K]$, $\vec{\psi}_k(x^k)$ — это вектор–столбец значений функций из словаря Δ_k в точке x^k , а \mathcal{A} — многомерная матрица коэффициентов разложения по этому словарю. Далее мы будем искать решение задач 2 и 3 именно в классе функций F_{tensor} .

Примем во внимание выбранный класс функций и перепишем формулу для суммы квадратов ошибок аппроксимации в точках из Σ . Значения функции g в точках из Σ параметризуются индексами i_1, i_2, \dots, i_K и могут быть представлены в виде тензора \mathcal{Y} с размерами $n_1 * n_2 * \dots * n_K$:

$$(3) \quad g([x_{i_1}^1, x_{i_2}^2, \dots, x_{i_K}^K]) = y_{i_1, i_2, \dots, i_K} \Rightarrow g(\Sigma) = \mathcal{Y}.$$

Обозначим через 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 (2) и (3):

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

4.2 Регуляризирующая штрафная функция

Штрафное слагаемое $P_\lambda(f)$ необходимо выбирать таким образом, чтобы учесть требования к гладкости модели (изменчивость модели вдоль факторов с малым числом уровней должна быть мала). Также надо принять во внимание достаточно высокие требования по вычислительной сложности (в силу больших объемов обучающей выборки). Опишем предложенный в работе [16] подход, который мы будем использовать в этой работе.

Рассмотрим сначала штраф, ограничивающий изменчивость аппроксимации вдоль фактора X_1 (переменные из всех остальных факторов зафиксированы). В качестве меры изменчивости будем использовать квадрат выборочной нормы второй производной f по x^1 , подсчитанной в точках множества Σ :

$$(4) \quad \left\| \frac{\partial^2 f}{(\partial x^1)^2}(\Sigma) \right\|_S^2 = \sum_{i_1, \dots, i_K} \left(\sum_{j_1, \dots, j_K} \alpha_{j_1, j_2, \dots, j_K} * \frac{\partial^2 \psi_{j_1}^1}{(\partial x^1)^2}(x_{i_1}^1) * \psi_{j_2}^2(x_{i_2}^2) * \dots * \psi_{j_K}^K(x_{i_K}^K) \right)^2.$$

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 (4) эквивалентен $p_{\tilde{\theta}}(f)$, где у $\tilde{\theta}$ не равна нулю только первая компонента: $\tilde{\theta}_1 = 1$. Для θ с несколькими компонентами, отличными от 0, штраф определяется аналогично (4), но порядок частной производной f будет равен не 2, а $2\|\theta\|_0$ (под L_0 нормой понимается количество ненулевых элементов). Перепишем выражение (4) для произвольного бинарного вектора θ в терминах многомерных матричных операций:

$$(5) \quad p_{\theta}(f) = \langle \mathcal{A}, \mathcal{A} \otimes_1 (\theta_1 \Omega_1 + (1 - \theta_1) \Psi_1^T \Psi_1) \dots \otimes_K (\theta_K \Omega_K + (1 - \theta_K) \Psi_K^T \Psi_K) \rangle.$$

Каждая матрица Ω_k (размера $p_k * p_k$) в (5) — это выборочная нормавторных 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:

$$(6) \quad P_{\lambda}(f) = \sum_{\theta \in \Theta} \left(p_{\theta}(f) \prod_{k=1}^K (\lambda_k \theta_k + (1 - \theta_k)) \right).$$

Функция $P_{\lambda}(f)$ основана на ограничении изменчивости во всех возможных срезах (часть переменных зафиксированна) и формируется как взвешенная сумма всех соответствующих штрафных функций. 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 случае сильно отличающегося числа уровней в факторах. 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, в то время как в остальных факторах $X_l, l \neq k$ ограничения на гладкость модели останутся в силе.

5 Вычисление оптимальных коэффициентов разложения

При выбранных классе функций F_{tensor} и штрафном слагаемом $P_{\lambda}(f)$ задача построения аппроксимации сводится к нахождению оптимального тензора коэффициентов разложения по словарю Δ_{tensor} , т.е. минимизации пенализованной функции

ошибки $R(f, \Sigma, g(\Sigma)) = Q(f, \Sigma, g(\Sigma)) + P_\lambda(f)$ по тензору \mathcal{A} . Заметим, что с учетом (5) и (6) штрафная функция может быть представлена как квадратичная функция $P_\lambda(f) = \text{vec}(\mathcal{A})^\top \Omega \text{vec}(\mathcal{A})$, где

$$(7) \quad \Omega = \sum_{\theta \in \Theta} (\theta_1(\lambda_1 \Omega_1) + (1 - \theta_1) \Psi_1^\top \Psi_1) \otimes \cdots \otimes (\theta_K(\lambda_K \Omega_K) + (1 - \theta_K) \Psi_K^\top \Psi_K).$$

Тогда выпишем задачу минимизации в явном виде

$$(8) \quad \mathcal{A}^* = \underset{\mathcal{A}}{\text{argmin}} \left[\text{vec}(\mathcal{A})^\top \Omega \text{vec}(\mathcal{A}) + (\text{vec}(\mathcal{Y}) - \Psi \text{vec}(\mathcal{A}))^\top (\text{vec}(\mathcal{Y}) - \Psi \text{vec}(\mathcal{A})) \right].$$

где Ψ — это матрица значений функций словаря Δ_{tensor} в точках множества Σ .

Задача (8) фактически является задачей гребневой регрессии и может быть решена явно

$$(9) \quad \text{vec} \mathcal{A}^* = (\Psi^\top \Psi + \Omega)^{-1} \Psi^\top \text{vec}(\mathcal{Y}).$$

Вычисление оптимального тензора коэффициентов по (9) требует $O(P^3)$ (где $P = \prod p_k$ — объем словаря Δ_{tensor}) [19]. Для больших словарей $P \gg 10\,000$ (которые характерны для задач высокой размерности) такая вычислительная сложность оказывается слишком велика. Рассмотрим частные задачи аппроксимации 2, 3 и предложим более эффективные способы вычисления тензора коэффициента, учитывающие особенности плана эксперимента Σ .

5.1 Полный факторный план

Если план эксперимента есть полное декартово произведение, то (с учетом способа формирования словаря) матрица регрессоров Ψ есть произведение Кронекера $\Psi_1 \otimes \Psi_2 \cdots \otimes \Psi_K$. Используя свойства произведения Кронекера и его связь с тензорной арифметикой (в частности, уравнение (1)) можно показать [16], что справедливо следующее утверждение.

Утверждение 1 *Решение задачи 2 в классе F_{tensor} со штрафом $P_\lambda(f)$ вида (6) существует и единственно, если и только если матрицы $\Psi_k^\top \Psi_k + \lambda_k \Omega_k$ невырождены. Кроме того, тензор коэффициентов разложения этого решения по словарю Δ_{tensor} вычисляется по формуле:*

$$(10) \quad \mathcal{A}^* = \mathcal{Y} \otimes_1 \Psi_1^\top (\Psi_1^\top \Psi_1 + \lambda_1 \Omega_1)^{-1} \cdots \otimes_K \Psi_K^\top (\Psi_K^\top \Psi_K + \lambda_K \Omega_K)^{-1}.$$

Замечание 1 *Необходимые и достаточные условия на невырожденность $\Psi_k^\top \Psi_k + \lambda_k \Omega_k$, $k = 1, \dots, K$ эквивалентны требованию положительной определенности матрицы гессiana квадратичной функции $R(f, \Sigma, g(\Sigma))$.*

Сложность вычисления оптимального тензора коэффициентов разложения \mathcal{A} по (10) 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 нас интересует случай $n_k = p_k$, то вычислительная сложность будет равна $O(\sum_{k=1}^K (p_k^3 + P p_k))$, что существенно меньше, чем $O(P^3 = \prod p_k^3)$ (сложности подсчета коэффициентов по формуле (9), которая не учитывает структуру выборки).

5.2 Неполный факторный план

При отсутствии хотя бы одной точки из полного факторного плана теряется основное свойство матрицы регрессоров и она больше не является произведением Кронекера меньших матриц. Однако, и в случае неполного произведения данные по-прежнему обладают некоторой структурой, знания о которой можно использовать для эффективного решения задачи минимизации (8). В этом случае не удастся выписать явную вычислительно-эффективную формулу для нахождения оптимального тензора коэффициентов. Мы будем искать тензор коэффициентов с помощью минимизации функции ошибки $R(f, \Sigma, g(\Sigma))$ методом сопряженных градиентов. Известно, что минимизация квадратичной функции с помощью этого метода выполняется не более, чем за P шагов, а вычислительная сложность каждого шага составляет $O(P^2)$ (где объем словаря P — это размерность задачи). Таким образом, прямолинейное использование метода сопряженных градиентов требует $O(P^3)$ операций и не позволяет получить преимущество по сравнению с явной формулой (9). Покажем, что использование структуры задачи позволяет существенно снизить вычислительную сложность нахождения тензора коэффициентов.

5.2.1 Модифицированная функция ошибки

Проведем некоторые предварительные преобразования, а именно заполним пропуски в данных:

- будем подсчитывать матрицу регрессоров $\tilde{\Psi}$ (значения функций словаря) во всех точках декартова произведения Σ_{full} , а не только в точках обучающей выборки Σ ;
- подставим произвольные величины в качестве значений неизвестной функции в пропущенных точках $x \in \Sigma_{full} \setminus \Sigma$.

Такое преобразование позволяет сохранить структуру произведения Кронекера матрицы регрессоров $\tilde{\Psi}$ и переписать множество значений функции в точках обучающей выборки, расширенное произвольными псевдо-значениями, как тензор (обозначим его как $\tilde{\mathcal{Y}}$).

Определим модифицированную взвешенную функцию ошибки $\tilde{R}(\mathcal{A})$ следующим образом:

$$(11) \quad \tilde{R}(\mathcal{A}) = \text{vec}(\mathcal{A})^T \Omega \text{vec}(\mathcal{A}) + (\text{vec}(\tilde{\mathcal{Y}}) - \tilde{\Psi} \text{vec}(\mathcal{A}))^T W (\text{vec}(\tilde{\mathcal{Y}}) - \tilde{\Psi} \text{vec}(\mathcal{A})),$$

где тензор значений $\tilde{\mathcal{Y}}$ и матрица регрессоров $\tilde{\Psi}$ соответствуют расширенной выборке (восстановленному полному факторному плану), а W — некоторая диагональная матрица весов. Будем формировать матрицу W следующим образом:

- $W_{i,i} = 1$, если $x_i \in \Sigma$ (i -я точка из Σ_{full} представлена в обучающей выборке);
- $W_{i,i} = 0$, если $x_i \in (\Sigma_{full} \setminus \Sigma)$ (i -я точка была пропущена в обучающей выборке).

При таком выборе весовой матрицы W справедливо $\tilde{R}(\mathcal{A}) \equiv R(f, \Sigma, g(\Sigma))$ при любом способе заполнения пропущенных значений в тензоре $\tilde{\mathcal{Y}}$. Таким образом, задача минимизации $R(f, \Sigma, g(\Sigma))$ может быть заменена эквивалентной задачей минимизации $\tilde{R}(\mathcal{A})$.

5.2.2 Существование и единственность решения

Утверждение 2 Если матрицы $\Psi_k^T \Psi_k$ и $\lambda_k \Omega_k$ (для $k = 1, \dots, K$) положительно определены, то решение задачи 3 в классе F_{tensor} со штрафом $P_\lambda(f)$ вида (6) существует и единственно.

Доказательство. Матрица вторых производных \tilde{R} согласно (11) равна $\Omega + \tilde{\Psi}^T W \tilde{\Psi}$. Словарь Δ_{tensor} был сформирован с помощью тензорного произведения, а значения матрицы регрессоров подсчитывались в точках из декартова произведения Σ_{full} , поэтому матрица $\tilde{\Psi}$ есть произведение Кронекера $\Psi_1 \otimes \Psi_2 \otimes \dots \otimes \Psi_K$. Так как в общем случае $p_k \leq n_k$, а $\Psi_k^T \Psi_k$ по условию положительно определены, то матрицы Ψ_k имеют полный столбцовый ранг. Тогда в силу свойств произведения Кронекера матрица $\tilde{\Psi} = \Psi_1 \otimes \dots \otimes \Psi_K$ также имеет полный столбцовый ранг. Кроме того, матрица W неотрицательно определена, поэтому $\tilde{\Psi}^T W \tilde{\Psi}$ также неотрицательно определена.

Вернемся к формуле (7) для штрафной матрицы Ω . Так как по условию $\lambda_k \Omega_k$ и $\Psi_k^T \Psi_k$ положительно определены (а θ_k — бинарная величина), то и матрица Ω положительно определена. С учетом неотрицательной определенности матрицы $\tilde{\Psi}^T W \tilde{\Psi}$ получаем, что матрица вторых производных функции \tilde{R} также положительно определена. Так как функция \tilde{R} квадратична по \mathcal{A} , то это означает, что существует единственная точка минимума этой функции. ■ Условия

утверждения 2 фактически означают, что словари Δ_k не вырождены (например, не включают линейно зависимый набор функций), и на практике выполняются для всех разумных словарей.

5.2.3 Нахождение оптимального решения

Утверждение 3 Если выполняются условия утверждения 2, то задача поиска тензора коэффициентов разложения по словарю Δ_{tensor} может быть сформулирована таким образом, что ее решение найдется с помощью метода сопряженных градиентов

1. Не более чем за $\tilde{N} - N + 1$ шагов (где \tilde{N} — мощность полного декартова произведения Σ_{full} , а $\tilde{N} - N$, соответственно, — количество пропущенных в Σ точек).
2. Вычислительная сложность каждого шага не превосходит $O(P \sum p_k)$.
3. Сложность дополнительных вычислений, которые выполняются однократно, составляет $O(P \sum p_k + \sum p_k^3)$.

Доказательство. Рассмотрим задачу минимизации функции $\tilde{R}(f, \Sigma_{full}, \tilde{\mathcal{Y}}, W) = \tilde{R}(\mathcal{A})$ методом сопряженных градиентов. Матрица вторых производных этой функции равна

$$(12) \quad H_A \stackrel{\text{def}}{=} \Omega + \tilde{\Psi}^T W \tilde{\Psi} = \Omega + \tilde{\Psi}^T \tilde{\Psi} + \tilde{\Psi}^T (W - I_P) \tilde{\Psi}.$$

Используя (7), разложение матрицы регрессоров $\tilde{\Psi} = \Psi_1 \otimes \dots \otimes \Psi_K$ и билинейность произведения Кронекера можно показать, что

$$\tilde{\Psi}^T \tilde{\Psi} + \Omega = (\Psi_1^T \Psi_1 + \lambda_1 \Omega_1) \otimes (\Psi_2^T \Psi_2 + \lambda_2 \Omega_2) \cdots \otimes (\Psi_K^T \Psi_K + \lambda_K \Omega_K).$$

Разложим по Холецкому [19] матрицы $\Psi_k^T \Psi_k + \lambda_k \Omega_k$ (мы можем сделать это, поскольку по условию эти матрицы положительно определены):

$$(13) \quad \Psi_k^T \Psi_k + \lambda_k \Omega_k = U_k^T U_k; \quad k = 1, \dots, K.$$

Заметим, что разложение Холецкого произведения Кронекера есть произведение разложений каждого сомножителя:

$$(14) \quad \tilde{\Psi}^T \tilde{\Psi} + \Omega = U^T U; \quad U \stackrel{\text{def}}{=} U_1 \otimes \dots \otimes U_K.$$

Преобразуем выражение (12) гессиана H функции $\tilde{R}(\mathcal{A})$, используя полученное разложение (14)

$$(15) \quad H_A = U^T U + \tilde{\Psi}^T (W - I_P) \tilde{\Psi} = U^T \left(I_P + (\tilde{\Psi} U^{-1})^T (W - I_P) (\tilde{\Psi} U^{-1}) \right) U.$$

Обозначим $H_B = I_P + (\tilde{\Psi} U^{-1})^T (W - I_P) (\tilde{\Psi} U^{-1})$ и перепишем функцию ошибки, используя полученное представление гессиана (15),

$$(16) \quad \tilde{R}(\mathcal{A}) = \text{vec}(\mathcal{A})^T U^T H_B U \text{vec}(\mathcal{A}) - 2 \text{vec}(\mathcal{A})^T \tilde{\Psi}^T \text{vec}(\tilde{\mathcal{Y}} * \mathcal{W}) + \langle \tilde{\mathcal{Y}}, \tilde{\mathcal{Y}} * \mathcal{W} \rangle,$$

где тензор \mathcal{W} — это переписанная в форме многомерного массива диагональ весовой матрицы W , а $*$ используется для обозначения поэлементного умножения тензоров.

Введем замену переменных $\mathcal{B} = \mathcal{A} \otimes_1 U_1 \otimes_2 U_2 \dots \otimes_K U_K$. Заметим, что согласно уравнению (1), связывающему тензорное умножение с произведением Кронекера, и с учетом (15) справедливо $\text{vec}(\mathcal{B}) = U \text{vec}(\mathcal{A})$. Тогда выражение (16) для функции ошибки будет переписано в новых переменных следующим образом:

$$\tilde{R}(\mathcal{B}) = \text{vec}(\mathcal{B})^T H_B \text{vec}(\mathcal{B}) + \langle \tilde{\mathcal{Y}}, \tilde{\mathcal{Y}} * \mathcal{W} \rangle - 2 \text{vec}(\mathcal{B})^T (\tilde{\Psi} U^{-1})^T \text{vec}(\tilde{\mathcal{Y}} * \mathcal{W}).$$

Введем еще одно обозначение $\Phi = \tilde{\Psi} U^{-1}$ и выпишем окончательное представление функции ошибки в новых переменных

$$(17) \quad \begin{aligned} \tilde{R}(\mathcal{B}) = & \text{vec}(\mathcal{B})^T (I_P + \Phi^T (W - I_P) \Phi) \text{vec}(\mathcal{B}) - \\ & - 2 \text{vec}(\mathcal{B})^T \Phi \text{vec}(\tilde{\mathcal{Y}} * \mathcal{W}) + \langle \tilde{\mathcal{Y}}, \tilde{\mathcal{Y}} * \mathcal{W} \rangle. \end{aligned}$$

Таким образом, матрица H_B является гессианом квадратичной функции ошибки $\tilde{R}(\mathcal{B})$ и представима в виде $I_P + \Phi^T (W - I_P) \Phi$. Покажем, что у матрицы H_B есть не более чем $\tilde{N} - N + 1$ различных собственных чисел.

В силу определения W на диагонали $(W - I_P)$ будут находиться -1 (точка не представлена в обучающей выборке) и 0 (точка представлена в обучающей выборке). Тогда ранг $(W - I_P)$ будет равен количеству ненулевых элементов, то есть $\tilde{N} - N$ (где \tilde{N} — мощность полного факторного плана). Матрица $\Phi = \tilde{\Psi} U^{-1}$ имеет полный столбцовый ранг (так как тем же свойством обладает $\tilde{\Psi}$, а U невырождена). Соответственно, ранг $\Phi^T (W - I_P) \Phi$ будет равен рангу матрицы $(W - I_P)$, то есть $\tilde{N} - N$. Тогда у гессиана $H_B = I_P + \Phi^T (W - I_P) \Phi$ не более $\tilde{N} - N$ собственных чисел отличается от 1, то есть всего у этой матрицы не более $\tilde{N} - N + 1$ различных собственных чисел. Известно, что метод сопряженных градиентов для квадратичных

функций сходится за число итераций, не превышающее число различных собственных чисел матрицы гессиана (см., например, теорему 5.4 в [20]). Таким образом, для нахождения решения потребуется не более $\tilde{N} - N + 1$ шагов.

Рассмотрим теперь вопрос вычислительной сложности минимизации функции $R(\mathcal{B})$, которая складывается из двух частей: сложности предварительных вычислений (переход к задаче вида (17)) и затрат на оптимизацию. Рассмотрим сначала вычислительную сложность метода сопряженных градиентов, на каждом шаге которого необходимо вычислять

- градиент функции ошибки;
- длину оптимального шага вдоль сопряженного направления.

Градиент функции $\tilde{R}(\mathcal{B})$ может быть легко получен из (17):

$$\text{vec} \left(\nabla \tilde{R}(\mathcal{B}) \right) = 2 \left(\mathbf{I}_P + \Phi^T (\mathcal{W} - \mathbf{I}_P) \Phi \right) \text{vec}(\mathcal{B}) - 2\Phi \text{vec}(\tilde{\mathcal{Y}} * \mathcal{W}).$$

Так как $\Phi = \tilde{\Psi} \mathbf{U}^{-1}$, а $\tilde{\Psi}$ и \mathbf{U} раскладываются в произведение Кронекера, то

$$(18) \quad \Phi = \Phi_1 \otimes \Phi_2 \otimes \dots \otimes \Phi_K, \quad \Phi_k \stackrel{\text{def}}{=} \Psi_k \mathbf{U}_k^{-1}.$$

Перепишем выражение для градиента в терминах тензорных операций с учетом (18).

$$(19) \quad \begin{aligned} \nabla \tilde{R}(\mathcal{B}) = & 2 \left((\mathcal{B} \otimes_1 \Phi_1 \dots \otimes_K \Phi_K) * \tilde{\mathcal{W}} \right) \otimes_1 \Phi_1 \dots \otimes_K \Phi_K + \\ & + 2\mathcal{B} - 2 \left(\tilde{\mathcal{Y}} * \mathcal{W} \right) \otimes_1 \Phi_1 \dots \otimes_K \Phi_K, \end{aligned}$$

где $\tilde{\mathcal{W}} = \mathcal{W} - \mathcal{I}$, а \mathcal{I} — тензор, состоящий из всех 1.

Слагаемое $2 \left(\tilde{\mathcal{Y}} * \mathcal{W} \right) \otimes_1 \Phi_1 \dots \otimes_K \Phi_K$ не зависит от текущего значения \mathcal{B} и может быть подсчитано один раз на предварительном этапе. Сложность умножения тензора размера $p_1 * p_2 * \dots * p_K$ на K матриц размера $p_k * p_k$ составляет $P \sum_k p_k$ операций. Поэлементное умножение и сложение тензоров требует, очевидно, P операций. Тогда на вычисление тензора $\nabla \tilde{R}(\mathcal{B})$ потребуется $3P + 2P \sum p_k$ операций.

Оптимальный шаг метода сопряженных градиентов также может быть посчитан по вычислительно-эффективной формуле

$$\alpha = \frac{0.5 \langle \nabla \tilde{R}(\mathcal{B}), \nabla \tilde{R}(\mathcal{B}) \rangle}{\langle \tilde{\mathcal{D}} * \tilde{\mathcal{W}}, \tilde{\mathcal{D}} \rangle + \langle \mathcal{D}, \mathcal{D} \rangle},$$

где тензор \mathcal{D} — это направление шага сопряженного градиента на текущей итерации, а $\tilde{\mathcal{D}} = \mathcal{D} \otimes_1 \Phi_1 \dots \otimes_K \Phi_K$. По аналогии с (19) несложно показать, что сложность подсчета оптимального шага составляет $4P + P \sum p_k$.

Для завершения оценки сложности каждой итерации осталось рассмотреть этап обновления тензора направления поиска и текущего приближения решения, которые включают только операции сложения и умножения на константу, сложность которых равна P . Таким образом, общая вычислительная сложность каждой итерации составляет $O(P \sum p_k)$.

Теперь кратко рассмотрим сложность предварительных вычислений. Подсчет постоянного слагаемого в формуле (19) требует $P + P \sum p_k$ операций. Переход к функции ошибки $\tilde{R}(\mathcal{B})$ состоит из следующих этапов:

- подсчет K разложений Холецкого в (13) с общей сложностью $O(\sum p_k^3)$;
- вычисление матриц Φ_k в (18) с общей сложностью $O(\sum p_k^3)$;
- вычисление постоянного слагаемого в градиенте (19) функции ошибки, требующее $P + P \sum p_k$ операций.

Суммарная сложность перехода к задаче минимизации функции $R(\mathcal{B})$ составляет $O(P \sum p_k + \sum p_k^3)$.

Наконец, после нахождения точки минимума функции $R(\mathcal{B})$ необходимо вернуться в исходные переменные, вычислив тензор коэффициентов разложения \mathcal{A} по формуле $\mathcal{A} = \mathcal{B} \otimes_1 U_1^{-1} \dots \otimes_K U_K^{-1}$. При найденных на предварительном этапе U_k^{-1} это требует $P \sum p_k$ операций. ■

Следствие 1 *Общая вычислительная сложность алгоритма поиска оптимальных коэффициентов разложения \mathcal{A} составляет $O(\sum p_k^3 + (\tilde{N} - N + 1) P \sum p_k)$.*

Помимо высокой вычислительной эффективности предложенных методов решения задач 2 и 3 следует отметить экономичное использование памяти. Матрица регрессоров размера $N * P$ никогда не формируется в явном виде, а размер максимального массива не превышает наибольшего из $\max(p_k^2)$ (размер матриц Ψ_k) и $N = \prod n_k$ (тензор значений $\tilde{\mathcal{Y}}$).

6 Выбор параметра регуляризации

Мы получили оптимальное решение при зафиксированной штрафной функции $P_\lambda(f)$. Однако на практике значение параметра регуляризации λ неизвестно и должно оцениваться по данным. Традиционный подход к оценке параметров регуляризации заключается в использовании некоторого критерия качества аппроксимации (например, ошибки на проверочной выборке) [21]. Мы воспользуемся одним из таких критериев, ошибкой скользящего контроля. Для этой ошибки по аналогии с выражением для оптимальных коэффициентов (9) можно выписать явную формулу, не использующее структурные особенности задачи:

$$(20) \quad Q_{loo} = \sum_{i=1}^N \frac{(y_i - \hat{y}_i)^2}{(1 - L_{ii})^2}, \quad L = \Psi(\Psi^T \Psi + \Omega)^{-1} \Psi^T.$$

Сложность использования этой формулы составляет $O(P^3)$, что в большинстве случаев недопустимо высоко.

Для полного факторного плана Q_{loo} может быть эффективно подсчитана (сложность $O(\sum p_k^3 + N)$) с помощью тензорных операций [16]:

$$(21) \quad Q_{loo} = \left\langle \mathcal{Y} - \hat{\mathcal{Y}}, (\mathcal{Y} - \hat{\mathcal{Y}}) * \left(\mathcal{I} - \mathcal{I} \otimes_1 \text{diag}(L_1) \dots \otimes_K \text{diag}(L_K) \right)^{-2} \right\rangle;$$

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 $n_1 * \dots * n_K$.

В случае неполного плана матрица $(\Psi^T \Psi + \Omega)^{-1}$ не формируется явно при нахождении тензора оптимальных коэффициентов \mathcal{A}^* , соответственно, ошибка скользящего контроля эффективно не вычисляется. Мы воспользуемся следующим алгоритмом выбора параметра λ для неполного факторного плана, который более привлекателен с вычислительной точки зрения, чем использование явной формулы (20).

0. Выбираем начальное значение параметра регуляризации $\lambda = \lambda^0$, $\lambda_k^0 = 10^{-7}$.
1. Строим модель $f(x)$, решая задачу с регуляризацией $P_\lambda(f)$ для выбранного λ .
2. Оцениваем значения $f(x)$ для пропущенных точек $x \in \Sigma_{full} \setminus \Sigma$.
3. Используя эти значения получаем задачу с полным факторным дизайном. Основываясь на (21) оцениваем новое значение параметра регуляризации и переходим к пункту 1.

В качестве критерия остановки итерационного процесса мы используем норму разности между последовательными оценками пропущенных значений и максимальное число итераций.

Описанный алгоритм может давать оценки ошибки скользящего контроля, значительно отличающиеся от ее истинного значения (что, конечно, может повлиять на окончательное значение параметра регуляризации λ). Проведенные эксперименты (в том числе и на данных из реальных задач инженерного проектирования) показывают, что предложенный подход позволяет добиться приемлемого качества аппроксимации. Оценка параметра регуляризации оказывается достаточно близка к оптимальной (с позиции минимальной ошибки скользящего контроля), особенно в случае, когда число пропусков относительно невелико. Тем не менее, в последующих исследованиях необходимо предложить эффективный способ вычисления ошибки скользящего контроля (или другой оценки обобщающей способности) для выборок с неполным факторным планом.

7 Вычислительные эксперименты

Сравним предложенный метод аппроксимации с двумя наиболее популярными универсальными методами аппроксимации, которые не используют знания о специальной структуре выборки: нейронными сетями (реализация в пакете Matlab [22]) и регрессии на гауссовских процессах (реализация [23] авторов книги [18]). Проведем три эксперимента на выборках с неполным факторным планом (эксперименты на выборках с полным факторным планом см. в [16]).

1. На функции Розенброка с примерно однородной сложностью зависимости от каждой из $d = 4$ переменных:

$$f(x) = \sum_{k=1}^{d-1} ((1 - x_k)^2 + 100 * (x_{k+1} - x_k^2)^2); \quad x \in [-2.048, 2.048]^4.$$

Объем выборки $N = 1800$, факторизация с одинаковым числом уровней в каждом из факторов $n_k = 7, k = 1, \dots, 4$ ($N_{full} = 7^4 = 2401$).

2. На искусственной функции, моделирующей типичную аэродинамическую зависимость, с неоднородной сложностью зависимости от каждой из $d = 3$ переменных:

$$f(x) = \arctg(x_1^{0.5} + 0.5x_3^{0.5})(-\arctg(50(x_2 + 0.2)^4(x_1 - 0.3 - 0.2x_2)) + 2x_1(1 - x_1) + (1 + x_2)x_2), x \in [0, 1]^3.$$

Объем выборки $N = 1800$ с неодинаковым числом уровней в каждом из факторов $n_1 = 41, n_2 = 10, n_3 = 6$ ($N_{full} = 2460$). Количество уровней выбрано исходя из сложности зависимости относительно этой переменной. Заметим, что выборка также отражает еще одну типичную особенность плана экспериментов из задач инженерного проектирования: в областях с большей сложностью функции генерируется более плотная сетка значений.

3. На реальной задаче инженерного проектирования, связанной с вычислением аэродинамических характеристик некоторого летательного объекта в зависимости от режима полета. Размерность задачи $d = 3$, объем обучающей выборки $N = 1840$, число уровней в факторах: $n_1 = 10, n_2 = 51, n_3 = 5$ ($N_{full} = 2550$).

Объемы выборок и размерность задач выбраны достаточно небольшими, чтобы иметь возможность построить аппроксимацию с помощью регрессии на гауссовских процессах (при больших объемах выборки вычислительные требования становятся слишком высоки). Неполный факторный план в случае искусственных задач выбирался как случайная подвыборка полного плана.

Будем сравнивать время построения модели (в секундах), точность аппроксимации на обучающей и тестовых выборках. В качестве меры ошибки воспользуемся корнем из средне- квадратичной невязки, нормированной на аналогичную величину для аппроксимации с помощью среднего значения (как правило, в прикладных задачах приемлемым считаются значения этой ошибки порядка 10^{-2}):

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

	Эксперимент 1			Эксперимент 2			Эксперимент 3		
	E об.	E т.	Вр.	E об.	E т.	Вр.	E об.	E т.	Вр.
Нейр. сеть	1.5e-4	0.023	129	0.015	0.107	333	0.012	0.124	255
Гаусс. пр.	2.7e-5	0.005	67	0.001	0.098	98	3.2e-4	0.213	72
Предл. подх.	8.2e-4	0.016	0.21	0.005	0.011	0.62	0.009	0.035	2.1

Таблица 1: Ошибки аппроксимации и время обучения

Также сравним изменчивость моделей в терминах выборочной нормы частных вторых производных модели по каждой из компонент входного вектора.

Результаты сравнения представлены в таблицах 1, 2. Отметим, что в первом эксперименте (с одинаковым числом уровней в факторах) все методы показали достаточно высокую точность и примерно одинаковый уровень гладкости. Преимущество предложенного подхода в этом случае заключается лишь в более низком времени обучения. В то же время в экспериментах 2, 3, в которых число уровней в факторах

	Эксперимент 1				Эксперимент 2			Эксперимент 3		
	x_1	x_2	x_3	x_4	x_1	x_2	x_3	x_1	x_2	x_3
Нейр. сеть	4.1e4	4.3e4	5.1e4	3.3e3	288.0	17.16	26.76	10.2	9.63	18.9
Гаусс. пр.	4.7e4	5.0e4	4.9e4	3.5e3	802.9	17.21	3.52	13.2	12.3	1.52
Предл. подх.	4.2e4	4.1e4	4.5e4	3.3e3	239.6	11.80	1.31	2.42	9.85	1.56

Таблица 2: Изменчивость моделей

существенно отличается, универсальные методы аппроксимации не позволили получить достаточно гладкую модель в факторах с малым числом уровней. Об этом же говорит и достаточно высокая ошибка аппроксимации в тестовых точках.

8 Заключение

В работе рассмотрена задача аппроксимации по выборкам, факторный план которых (полный или неполный) несет знания из прикладной области о моделируемой зависимости. Предложены новый метод аппроксимации, основанный на разложении по полученному с помощью тензорного произведения словарю функций, и метод введения регуляризации, штрафующий изменчивость модели вдоль различных срезов. Разработан метод нахождения оптимального тензора коэффициентов разложения по словарю для случая неполного факторного плана и доказана его высокая вычислительная эффективность.

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

Список литературы

- [1] Grihon S., Alestra S., Burnaev E., Prikhodko P. *Optimization of Composite Structure based on Surrogate Modeling of Buckling Analysis* // Тр. конф. Информационные Технологии и Системы. 2012. С. 41–47.
- [2] Кулешов А.П. *Когнитивные технологии в адаптивных моделях сложных объектов*, Информационные технологии и вычислительные системы. 2008. № 1. С. 18–29.
- [3] Bayarria M.J., Bergera J.O., Kennedy M. C. et al. *Predicting Vehicle Crashworthiness: Validation of Computer Models for Functional and Hierarchical Data* // Journal of the American Statistical Association. 2009. V. 104(487). P. 929–943.
- [4] Отчет по проекту “Быстрый аэродинамический расчет компоновки пассажирского самолета”. МНИИПУ. 2007.
- [5] Беляев М.Г. *Аппроксимация и интерполяция на основе тензорного произведения параметрических словарей* // Тр. конф. Информационные Технологии и Системы. 2012. С. 32–40.

- [6] Stone C.J., Hansen M.H., Kooperberg C., Truong Y. K. *Polynomial splines, their tensor products in extended linear modeling*. Annals of Statistic. 1997. V. 25. N. 4. P. 1371–1470.
- [7] Barthelmann V., Novak E., Ritter K. *High dimensional polynomial interpolation on sparse grids*. Advances in Computational Mathematics. 2000. V. 12. N. 4. P. 273–288.
- [8] Dierckx P. *Computation of least-squares spline approximations to data over incomplete grids*. Computers & Mathematics with Applications. 1984. V. 10(3). P. 283–289.
- [9] Pisinger G., Zimmermann A. *Linear least squares problems with data over incomplete grid*. BIT Numerical Mathematics. 2007. V. 47. P. 809–824.
- [10] Paulo R. *Default Priors for Gaussian Processes*. The Annals of Statistics. 2005. V. 33(2). P. 556–582.
- [11] Friedman J.H. *Multivariate adaptive regression splines*. Annals of Statistic. 1991. V. 19. N. 1. P. 1–141.
- [12] de Boor C. *A Practical Guide to Splines, 2nd edition*. Springer-Verlag. Berlin. 2001.
- [13] Schölkopf B., Herbrich R., Smola A. *A generalized representer theorem*. Computational Learning Theory. 2001. V. 2111. P. 416–426.
- [14] Kolda T. G., Bader B. W. *Tensor Decompositions and Applications*. SIAM Review. 2009. V. 51(3). P. 455–500.
- [15] van Loan C.F. *The ubiquitous Kronecker product*. Journal of Computational and Applied Mathematics. 2000. V. 123(1-2). P. 85–100.
- [16] Беляев М.Г. *Аппроксимация данных, порожденных декартовым произведением*. Труды МФТИ. 2013. Т.5. № 3. С. 11–23.
- [17] Беляев М.Г., Бурнаев Е.В., Любин А.Д. *Методика формирования функционального словаря в задаче аппроксимации многомерной зависимости // Сборник докладов конференции ММРО-15 (2011)*. С. 146–149.
- [18] Rasmussen C.E., Williams C.K.I. *Gaussian processes for machine learning*. MIT Press. Cambridge. 2006.
- [19] Golub G.H., van Loan C.F. *Matrix Computations, 3rd edition*. The Johns Hopkins University Press, 1996.
- [20] Nocedal J., Wright S. *Numerical Optimization, 2nd Edition*. Springer. 2006.
- [21] Tibshirani R., Hastie T., Friedman J. *The elements of statistical learning: data mining, prediction and inference, 2nd edition*. Springer. New York. 2009.
- [22] Matlab Neural Network Toolbox.
<http://www.mathworks.com/products/neural-network/index.html>
- [23] Gaussian processes for machine learning toolbox.
<http://www.gaussianprocess.org/gpml/code/matlab/doc/index.html>