РАЗДЕЛ 2. ВЫЧИСЛИТЕЛЬНЫЕ АЛГОРИТМЫ ПАРАМЕТРИЧЕСКОЙ ИДЕНТИФИКАЦИИ

Рассмотрим динамическую систему, описываемую системой обыкновенных дифференциальных уравнений:

$$\mathbf{u}' = \mathbf{f}(t, \mathbf{u}, \boldsymbol{\theta}) \quad (1)$$

$$\mathbf{u}(0) = \mathbf{u}_0 \quad (2)$$

Здесь t-независимая переменная (время), $\mathbf{u} = (u_1, u_2, ..., u_n)^T$ - вектор фазовых переменных, $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_m)^T$ - вектор параметров, $\mathbf{f} = (f_1, f_2, ..., f_n)^T$ - вектор правых частей уравнений системы.

Прямая задача для этой системы заключается в нахождении вектора фазовых переменных как функций независимой переменной $\mathbf{u}(t)$ по известным начальным условиям $\mathbf{u}_0$ при заданном наборе параметров $\boldsymbol{\theta}$. Однако возможна и постановка обратной задачи, когда по известной информации о фазовых переменных $\tilde{\mathbf{F}}(\mathbf{u})$ (например, полученной экспериментальным путем) необходимо определить (идентифицировать) значения определенных параметров из набора $\boldsymbol{\theta}$. Таким образом, в обратных задачах, в отличие от прямых, необходимо по следствию определить причину. В частности, в качестве известной информации могут выступать непосредственно компоненты вектора $\mathbf{u}$, измеренные в определенные моменты времени t_k, т.е. $\tilde{\mathbf{F}}(\mathbf{u}) = \tilde{\mathbf{u}}(t_k)$. Отметим, что обратные задачи часто бывают некорректными. Признаками некорректности являются отсутствие решения, наличие более одного решения, сильная зависимость решения от погрешности входных данных (важность этого фактора связана с тем, что информация о фазовых переменных, как правило, поступает с некоторой погрешностью).

В качестве примера приведем простую динамическую систему «потребитель - возобновляемый ресурс» (модель «хищник - жертва»):

$$u' = -\alpha_u u + \gamma_u uv$$

$$v' = \alpha_v v - \gamma_v uv$$

$u(0) = u_0$

$v(0) = v_0$

Здесь в роли фазовых переменных выступают u, v, т.е. $\mathbf{u} = (u, v)^T$, а компонентами вектора параметров являются α_u, α_v, γ_u, γ_v, т.е. $\boldsymbol{\theta} = (\alpha_u, \alpha_v, \gamma_u, \gamma_v)^T$.

Прямая задача для данной системы заключается в вычислении неизвестных функций $u(t)$, $v(t)$ (фазовых переменных) по известным начальным условиям u_0, v_0 и заданным значениям параметров α_u, α_v, γ_u, γ_v. Эта задача рассматривалась нами в предыдущем разделе. Однако часто возникает другая задача, когда по известной информации о значениях функций в определенные моменты времени $\tilde{u}(t_k)$, $\tilde{v}(t_k)$ требуется определить значения некоторых параметров и/или начальных условий. Таким образом, ставится задача идентификации параметров по известной (например, полученной экспериментальным путем) информации о фазовых переменных.

Как правило, такие задачи сводятся к минимизации функции потерь $J(\boldsymbol{\theta})$, выражающей отличие получаемых в результате решения прямой задачи значений $\mathbf{u}(t_k, \boldsymbol{\theta})$ от эталонных (измеренных) величин $\tilde{\mathbf{u}}(t_k)$, т.е. вектор искомых параметров определяется как $\arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$. Заметим, что не всегда требуется определять полный набор параметров $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_m)^T$, какие-то параметры могут быть известными, тогда оптимизация ведется по меньшему количеству переменных. Наиболее распространенным подходом к формированию функции $J(\boldsymbol{\theta})$ является использование квадратичного отклонения $J(\boldsymbol{\theta}) = \sum_k \left\| \mathbf{u}(t_k, \boldsymbol{\theta}) - \tilde{\mathbf{u}}(t_k) \right\|^2$ или в покомпонентном виде $J(\boldsymbol{\theta}) = \sum_k \sum_i (u_i(t_k, \boldsymbol{\theta}) - \tilde{u}_i(t_k))^2$. Часто используется не абсолютная разность рассчитанных и эталонных значений, а относительная (процентное

отклонение), например, $J(\boldsymbol{\theta}) = \sum_k \sum_i ((u_i(t_k, \boldsymbol{\theta}) - \tilde{u}_i(t_k)) / \tilde{u}_i(t_k))^2$, что позволяет нивелировать разницу в размерностях между компонентами вектора $\mathbf{u}$. Таким образом, в оптимизационном цикле, направленном на поиск минимума функции $J(\boldsymbol{\theta})$, необходимо многократно решать прямую задачу. Поскольку процедура решения такой задачи может быть вычислительно затратной, актуальным вопросом является экономичность оптимизационного процесса. С точки зрения технологии построения вычислительного процесса (и программного обеспечения) наиболее простым вариантом является применение методов оптимизации нулевого порядка, не требующих вычисления градиента функционала. В этом случае на каждом шаге оптимизационного цикла вычисляются лишь значения минимизируемой функции. Как правило, подобные методы обладают довольно низкой скоростью сходимости. Этот недостаток в особой мере проявляется в задачах с большой размерностью искомого вектора (более 100).

К настоящему времени создано большое количество методов нулевого порядка, с которыми можно познакомиться в учебных пособиях и монографиях, посвященных методам оптимизации. Здесь приведем метод Нелдера-Мида (деформируемого многогранника), который хорошо зарекомендовал себя при решении задач оптимизации в параметризованной форме.

Метод Нелдера-Мида заключается в последовательном перемещении и деформировании симплекса вокруг точки экстремума. Симплекс состоит из $n_u + 1$ вершин, где n_u - размерность искомого вектора. Алгоритм строится следующим образом.

1. Задается начальный симплекс с координатами $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$,…, $\boldsymbol{\theta}_{n_u+1}$, коэффициенты отражения α, сжатия β, растяжения γ, редукции σ, точность $\varepsilon > 0$.

2. Определяется лучшая точка симплекса $\boldsymbol{\theta}_b$ (доставляющая минимальное значение функции $J(\boldsymbol{\theta})$), худшая $\boldsymbol{\theta}_h$ (с максимальным значением функции) и вторая после худшей точка $\boldsymbol{\theta}_s$.
3. Вычисляется центр тяжести всех вершин, кроме худшей $\boldsymbol{\theta}_m = \frac{1}{n_u}\left(\sum_{j=1}^{n_u+1} \boldsymbol{\theta}_j - \boldsymbol{\theta}_h\right)$
4. Если $\sqrt{\frac{1}{n_u+1}\sum_{j=1}^{n_u+1}\left(J(\boldsymbol{\theta}_j) - J(\boldsymbol{\theta}_m)\right)^2} \le \varepsilon$, то выполнение алгоритма завершается.
5. Осуществляется отражение худшей вершины относительно центра тяжести $\boldsymbol{\theta}_r = \boldsymbol{\theta}_m + \alpha(\boldsymbol{\theta}_m - \boldsymbol{\theta}_h)$
6. Если $\boldsymbol{\theta}_b \le \boldsymbol{\theta}_r < \boldsymbol{\theta}_s$, то $\boldsymbol{\theta}_h = \boldsymbol{\theta}_r$, осуществляется переход на шаг 2.
7. Если $\boldsymbol{\theta}_r < \boldsymbol{\theta}_b$, то направление выбрано удачно, делается растяжение многогранника в этом направлении $\boldsymbol{\theta}_e = \boldsymbol{\theta}_m + \gamma(\boldsymbol{\theta}_m - \boldsymbol{\theta}_r)$. Если $\boldsymbol{\theta}_e < \boldsymbol{\theta}_r$, то $\boldsymbol{\theta}_h = \boldsymbol{\theta}_e$, в противном случае $\boldsymbol{\theta}_h = \boldsymbol{\theta}_r$, осуществляется переход на шаг 2.
8. Если $\boldsymbol{\theta}_r \ge \boldsymbol{\theta}_s$, то осуществляется сжатие $\boldsymbol{\theta}_c = \boldsymbol{\theta}_m + \beta(\boldsymbol{\theta}_m - \boldsymbol{\theta}_h)$. Если $\boldsymbol{\theta}_c < \boldsymbol{\theta}_r$, то $\boldsymbol{\theta}_h = \boldsymbol{\theta}_c$, осуществляется переход на шаг 2.
9. Формируется новый симплекс с уменьшенными сторонами (сжатие симплекса относительно лучшей вершины)
   $\boldsymbol{\theta}_j = \boldsymbol{\theta}_b + \sigma(\boldsymbol{\theta}_j - \boldsymbol{\theta}_l)$, $j = 1,..., n_u + 1$, осуществляется переход на шаг 2.

Типичные значения параметров для этого алгоритма $\alpha = 1$, $\beta = 0.5$, $\gamma = 2$, $\sigma = 0.5$.

Существенного ускорения оптимизационного процесса можно добиться с помощью методов оптимизации первого порядка. При этом требуется вычисление градиента оптимизируемой функции. Применительно к рассматриваемым задачам принципиальным моментом является то, что функция $J(\boldsymbol{\theta})$ явно не задана, поскольку не заданы явно функциональные

зависимости $\mathbf{u}(t,\boldsymbol{\theta})$. Они рассчитываются в результате решения прямой задачи (1)-(2).

Рассмотрим один из подходов к вычислению градиента оптимизируемой функции. Начнем рассмотрение с простейшего варианта, когда система описывается одним уравнением с одним параметром. Задача принимает вид

$$u' = f(t,u,\theta) \tag{3}$$

$$u(0) = u_0 \tag{4}$$

Продифференцируем уравнение (1) по параметру :

$$\frac{\partial u'}{\partial \theta} = \frac{\partial f(t,u,\theta)}{\partial \theta} + \frac{\partial f(t,u,\theta)}{\partial u}\frac{\partial u}{\partial \theta}$$

Введем обозначение $\frac{\partial u}{\partial \theta} = u_\theta$, тогда уравнение примет вид

$$u'_\theta = a_\theta(t,u,\theta)u_\theta + b_\theta(t,u,\theta) \tag{5}$$

где $a_\theta(t,u,\theta) = \frac{\partial f(t,u,\theta)}{\partial u}$, $b_p(t,u,\theta) = \frac{\partial f(t,u,\theta)}{\partial \theta}$

Таким образом, получено уравнение для определения зависимости производной $u_\theta = \frac{\partial u}{\partial \theta}$ от времени t. Значения u_θ в различные моменты времени характеризуют чувствительность решения прямой задачи к изменению параметра, поэтому уравнение (5) можно называть уравнением чувствительности. По виду полученное уравнение аналогично (1). Дифференцируя начальное условие (2) по параметру p, получим начальное условие для уравнения (5):

$$u_\theta(0) = 0 \tag{6}$$

Зная зависимость $u_\theta(t)$, можно вычислить производную оптимизируемой функции по параметру. Если используется суммарное квадратичное отклонение $J(\theta) = \sum_k (u(t_k,\theta) - \tilde{u}(t_k))^2$, то

$$\frac{\partial J(\theta)}{\partial \theta} = 2\sum_k (u(t_k,\theta) - \tilde{u}(t_k))u_\theta(t_k)$$

Таким образом, чтобы определить производную функции потерь по параметру необходимо наряду с задачей (3), (4) решать уравнения (5), (6). Иными словами, необходимо решать систему уравнений:

$$u' = f(t,u,\theta)$$

$$u'_\theta = a_\theta(t,u,\theta)u_\theta + b_\theta(t,u,\theta)$$

$$u(0) = u_0$$

$$u_\theta(0) = 0$$

Отметим, что первое уравнение в этой системе не зависит от второго, поэтому можно решать задачи последовательно: сначала решается задача (3), (4), потом - задача (5), (6).

Пусть теперь исходное уравнение содержит группу параметров $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_m)^T$.

$$u' = f(t,u,\boldsymbol{\theta})$$

$$u(0) = u_0$$

Дифференцируя по j-му параметру получаем:

$$\frac{\partial u'}{\partial \theta_j} = \frac{\partial f(t,u,\boldsymbol{\theta})}{\partial \theta_j} + \frac{\partial f(t,u,\boldsymbol{\theta})}{\partial u}\frac{\partial u}{\partial \theta_j}$$

Или с учетом $\frac{\partial u}{\partial \theta_j} = u_{\theta j}$:

$$u'_{\theta j} = a_\theta(t,u,\boldsymbol{\theta})u_{\theta j} + b_\theta(t,u,\boldsymbol{\theta})$$

где $a_\theta(t,u,\boldsymbol{\theta}) = \frac{\partial f(t,u,\boldsymbol{\theta})}{\partial u}$, $b_\theta(t,u,p) = \frac{\partial f(t,u,\boldsymbol{\theta})}{\partial \theta_j}$

$$u_{\theta j}(0) = 0$$

Т.е. для нахождения производной по каждому параметру требуется дополнительно решать задачу Коши: $u'_{\theta j} = a_\theta(t,u,\boldsymbol{\theta})u_{\theta j} + b_\theta(t,u,\boldsymbol{\theta})$, $u_{\theta j}(0) = 0$. В результате будет получен градиент $\nabla_{\boldsymbol{\theta}} u$ в зависимости от времени t - $(u_{\theta 1}(t), u_{\theta 2}(t), ..., u_{\theta m}(t))^T$, что позволит рассчитать градиент функции потерь

$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$. Если используется суммарное квадратичное отклонение $J(\boldsymbol{\theta}) = \sum_k (u(t_k, \boldsymbol{\theta}) - \tilde{u}(t_k))^2$, то

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_j} = 2\sum_k (u(t_k, \boldsymbol{\theta}) - \tilde{u}(t_k)) u_{\theta j}(t_k)$$

Пример.

Для примера рассмотрим задачу параметрической идентификации для уравнения, описывающего логистическое поведение:

$$u' = \alpha u - \beta u^2$$

$$u(0) = u_0$$

Уравнение содержит два параметра α и β. Дифференцирование по параметрам дает:

$$u'_\alpha = u + (\alpha - 2u\beta) u_\alpha$$

$$u'_\beta = u^2 + (\alpha - 2u\beta) u_\beta$$

Решая эти уравнения, дополненные нулевыми начальными условиями, наряду с основным уравнением, получим зависимости $u_\alpha(t)$, $u_\beta(t)$, после чего несложно рассчитать градиент целевой функции: $\nabla J(\alpha, \beta) = \left(\frac{\partial J}{\partial \alpha}, \frac{\partial J}{\partial \beta} \right)^T$.

Наконец, если рассматривается общая система (1) с начальными условиями (2), то действия аналогичны:

$$\mathbf{u}' = \mathbf{f}(t, \mathbf{u}, \boldsymbol{\theta})$$

$$\mathbf{u}(0) = \mathbf{u}_0$$

$$\frac{\partial u'_i}{\partial \theta_j} = \frac{\partial f_i(t, \mathbf{u}, \boldsymbol{\theta})}{\partial \theta_j} + \sum_l \frac{\partial f_i(t, \mathbf{u}, \boldsymbol{\theta})}{\partial u_l} \frac{\partial u_l}{\partial \theta_j}$$

Или с учетом обозначения $\frac{\partial u_i}{\partial \theta_j} = u^i_{\theta j}$:

$$\frac{du^i_{\theta j}}{dt} = \sum_l a^i_l(t, \mathbf{u}, \boldsymbol{\theta}) u^l_{\theta j} + b^i_j(t, \mathbf{u}, \boldsymbol{\theta})$$

где $a_l^i(t,\mathbf{u},\theta)=\dfrac{\partial f_i(t,\mathbf{u},\mathbf{\theta})}{\partial u_l}$, $b_j^i(t,u,\mathbf{\theta})=\dfrac{\partial f_i(t,\mathbf{u},\mathbf{\theta})}{\partial \theta_j}$

$u_{\theta j}^i(0)=0$

Запишем полученную систему в векторном виде:

$\mathbf{u}'_{\theta j}=\mathbf{A_j}\mathbf{u}_{\theta j}+\mathbf{b_j}$,

где $\mathbf{A_j}$ - *n*x*n* матрица с компонентами $a_l^i(t,\mathbf{u},\mathbf{\theta})$, $\mathbf{b_j}$- вектор с компонентами $b_j^i(t,\mathbf{u},\mathbf{\theta})$.

Таким образом, для нахождения производной по каждому параметру требуется дополнительно решать задачу Коши: $\mathbf{u}'_{\theta j}=\mathbf{A_j}\mathbf{u}_{\theta j}+\mathbf{b_j}$, $\mathbf{u}_{\theta\, j}(0)=0$.

В результате будут получены градиенты компонент вектора $\mathbf{u}$ - $\nabla_{\mathbf{\theta}}u^i$ в зависимости от времени t - $(u_{\theta 1}^i(t),u_{\theta 2}^i(t),...,u_{\theta m}^i(t))^T$, что позволит рассчитать градиент функции потерь $\nabla_{\mathbf{\theta}}J(\mathbf{\theta})$. Если используется суммарное квадратичное отклонение $J(\mathbf{\theta})=\sum_i\sum_k(u^i(t_k,\mathbf{\theta})-\tilde{u}^i(t_k))^2$, то

$$\frac{\partial J(\mathbf{\theta})}{\partial \theta_j}=2\sum_i\sum_k(u^i(t_k,\mathbf{\theta})-\tilde{u}^i(t_k))u_{\theta j}^i(t_k)$$

Наличие градиента оптимизируемой функции открывает возможность применения градиентных методов оптимизации.

Здесь дадим описание одного из наиболее эффективных методов первого порядка - метода сопряженных градиентов. При использовании данного метода для решения задач оптимизации важен тот факт, что минимум квадратичной нормы теоретически (без учета погрешностей различного рода) находится за *n* (размерность пространства) шагов при поиске вдоль сопряженных относительно матрицы Гессе направлений (**H**-сопряженных направлений). На практике, в окрестности минимума большой класс функций может быть достаточно точно представлен в виде квадратичной аппроксимации, что позволяет эффективно применять метод для различных функций.

Вычислительный процесс строится следующим образом:

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \alpha_k \mathbf{q}_k,\ \mathbf{q}_{k+1} = -\nabla J(\boldsymbol{\theta}^{(k+1)}) + \beta_k \mathbf{q}_k.$$

В качестве начального приближения $\mathbf{q}_0$, как правило, выбирается шаг метода наискорейшего градиентного спуска.

С точки зрения эффективности вычисления принципиальными являются три момента: экономичный расчет градиента целевой функции $\nabla J(\boldsymbol{\theta}^{(k+1)})$, выбор коэффициента β_k, определяющего направление спуска, выбор коэффициента α_k, определяющего шаг спуска вдоль выбранного направления.

Метод вычисления градиента целевой функции рассмотрен выше. Для расчета коэффициента β предложен целый ряд подходов. Здесь приведем исторически первую для данного метода формулу Флетчера - Ривса и формулу Хагера - Чанга (Hager, Zhang), которая хорошо зарекомендовала себя при решении широкого класса оптимизационных задач.

Формула Флетчера - Ривса имеет вид:

$$\beta_k = \frac{\|\mathbf{g}_{k+1}\|^2}{\|\mathbf{g}_k\|^2}, \quad \mathbf{g}_k = \nabla J(\boldsymbol{\theta}^{(k)})$$

Наилучшим образом эта формула проявляет себя при оптимизации квадратичных функций.

Более универсальной является формула Хагера-Чанга:

$$\beta_k = \max\{\tilde{\beta}_k, \eta_k\}$$

$$\tilde{\beta}_k = \left(\mathbf{y}_k - 2\mathbf{q}_k \frac{\|\mathbf{y}_k\|^2}{\mathbf{q}_k^{\mathrm{T}}\mathbf{y}_k}\right)^T \frac{\mathbf{g}_{k+1}}{\mathbf{q}_k^{\mathrm{T}}\mathbf{y}_k}, \quad \mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$$

$$\eta_k = \frac{-1}{\|\mathbf{q}_k\| \min\{\eta, \|\mathbf{g}_k\|\}},$$

где $\eta > 0$ – заданная константа. Авторы предлагают значение константы $\eta = 0{,}01$. Такой подход позволяет избежать негативных эффектов, связанных с потерей свойства **H**-сопряженности направлений вследствие ошибок

вычислений. Реализация метода сопряженных градиентов с таким выбором направления спуска получила название CG_decent (метод сопряженных градиентов с гарантированным спуском).

Одним из важных элементов реализации методов сопряженных градиентов является эффективная процедура подбора оптимального шага (коэффициента α). Выбор шага, по сути, должен удовлетворять двум противоречащим условиям: с одной стороны, учитывая, что каждое вычисление функционала связано с решением прямой задачи, важно, чтобы данный подбор был осуществлен с минимальными накладными расходами, с другой стороны, требуется, чтобы значение целевой функции существенно уменьшалось. В этом плане можно отметить методы неточного линейного поиска. Типичный алгоритм подобного типа состоит из 2 фаз: во-первых, определяются интервалы допустимых величин шагов (*bracketing*), во-вторых, выполняются процедуры бисекции и/или интерполяции для выбора конкретного значения квазиоптимального шага из интервалов. Окончание процедуры неточного поиска шага вдоль направления определяется, как правило, условиями Вульфа (Wolfe): $\phi(\alpha_k) \le \phi(0) + c_1\alpha_k\phi'(0)$, $\phi'(\alpha_k) \ge c_2\phi'(0)$, $0 < c_1 < c_2 < 1$, $\phi(\alpha_k) = J(\boldsymbol{\theta}^{(k)} + \alpha_k \mathbf{q}_k)$. Первое условие требует, чтобы при выбранном шаге α_k существенно уменьшалось значение целевой функции. Второе условие соответствует существенному уменьшению наклона целевой функции. Типичные значения для методов сопряженных градиентов: $c_1 = 10^{-4}$, $c_2 = 0.1$.

Рассмотренные методы позволяют найти локальный минимум целевой функции, для поиска глобального минимума применяются методы глобальной оптимизации. Один из простейших подходов – метод рестартов со случайным выбором начального приближения. Существенным достоинством данного метода является высокая степень распараллеливания.

Отметим, что в плане глобальной оптимизации заслуживают внимания получающие все большее развитие метаэвристические методы, такие, как генетические алгоритмы, метод имитации отжига, метод частиц в стае и т.д.

Задания к лабораторным работам по разделу 2

Провести идентификацию параметров модели «хищник - жертва», постепенно наращивая количество идентифицируемых параметров от 1 до 4-х. Исследование провести в режиме квазиреального эксперимента, т.е., задав набор параметров, провести решение прямой задачи, получить зависимости $u(t), v(t)$, и, проведя их зашумление, использовать полученные значения в качестве эталонных: $\tilde{u}(t_k) = u(t_k) + \delta_u(t_k)$, $\tilde{v}(t_k) = v(t_k) + \delta_v(t_k)$, где δ - Гауссов белый шум с заданной дисперсией (предпочтительнее величину дисперсии брать пропорциональной текущему значению функции).

1. Провести идентификацию с использованием метода оптимизации нулевого порядка (например, метод Нелдера-Мида).

2. Провести идентификацию с использованием градиентного метода, решив дополнительные задачи для определения функций чувствительности решения к изменению параметров.