

Постановка задачи

Даны $(t_i, y_i)_{i=1}^n$, где X — одномерные значения, а Y — одномерная целевая переменная.

Рассмотрим p — количество экспоненциальных членов, тогда мы подбираем функцию:

$$f : X \rightarrow Y; f(t, \mathbf{p}) = \sum_{i=1}^p \lambda_i \alpha_i^t$$

где $\mathbf{p} = (\lambda_1, \dots, \lambda_p, \alpha_1, \dots, \alpha_p)$

Предполагая, что $\forall i \alpha_i > 0$, можно переписать $f(t, \mathbf{p})$ как:

$$f(t, \mathbf{p}) = \sum_{i=1}^p \lambda_i \exp(\ln(\alpha_i)t) = \sum_{i=1}^p \lambda_i \exp(\omega_i t),$$

где $\mathbf{p} = (\lambda_1, \dots, \lambda_p, \omega_1, \dots, \omega_p)$ — параметры, которые необходимо подобрать.

Функция потерь для оптимизации — это MSE:

$$L(\mathbf{p}) = \sum_{i=1}^p (y_i - f(t_i, \mathbf{p}))^2.$$

Метод оптимизации — алгоритм Левенберга — Марквардта, встроенный в функцию `curve_fit` библиотеки `scipy`.

Алгоритм Левенберга — Марквардта (LMA)

Подобно другим численным методам минимизации, алгоритм Левенберга — Марквардта является итеративной процедурой. Для начала минимизации необходимо задать начальное приближение для вектора параметров. Начальное значение $\mathbf{p}^T = (1, 1, \dots, 1)$ подходит в большинстве случаев; в задачах с множеством локальных минимумов алгоритм сходится к глобальному минимуму, только если начальное приближение достаточно близко к решению.

На каждом шаге итерации вектор параметров \mathbf{p} заменяется новой оценкой $\mathbf{p} + \Delta$. Чтобы определить Δ , функция $f(t_i, \mathbf{p} + \Delta)$ линеаризуется:

$$f(t_i, \mathbf{p} + \Delta) \approx f(t_i, \mathbf{p}) + \mathbf{J}_i \Delta,$$

где

$$\mathbf{J}_i = \frac{\partial f(t_i, \mathbf{p})}{\partial \mathbf{p}}$$

— это градиент f по параметрам \mathbf{p} .

Таким образом $\forall j \leq p$:

$$\mathbf{J}_{ij} = \frac{\partial f(t_i, \mathbf{p})}{\partial \lambda_j} = \exp(\omega_j t_i),$$

$$\mathbf{J}_{ij+p} = \frac{\partial f(t_i, \mathbf{p})}{\partial \omega_j} = \lambda_j t_i \exp(\omega_j t_i).$$

Функция потерь достигает минимума, когда её градиент по \mathbf{p} равен нулю. Для первого приближения $f(t_i, \mathbf{p} + \Delta)$:

$$L(\mathbf{p} + \Delta) \approx \sum_{i=1}^p [y_i - f(t_i, \mathbf{p}) - \mathbf{J}_i \Delta]^2$$

или в векторной форме:

$$L(\mathbf{p} + \Delta) \approx \|\mathbf{y} - \mathbf{f}(\mathbf{p}) - \mathbf{J}\Delta\|_2^2.$$

Взяв производную от $L(\mathbf{p} + \Delta)$ по Δ и приравняв её к нулю, получим:

$$(\mathbf{J}^T \mathbf{J}) \Delta = \mathbf{J}^T [\mathbf{y} - \mathbf{f}(\mathbf{p})].$$

Выражение выше соответствует методу Гаусса–Ньютона. Матрица Якоби \mathbf{J} обычно не квадратная, а прямоугольная размерности $m \times n$, где n — количество параметров. Перемножение $\mathbf{J}^T \mathbf{J}$ дает квадратную матрицу размерности $n \times n$. Результат — это система из n линейных уравнений, решаемая для Δ .

Вклад Левенберга заключается в использовании регуляризованной версии уравнения:

$$(\mathbf{J}^T \mathbf{J} + \lambda \mathbf{E}) \Delta = \mathbf{J}^T [\mathbf{y} - \mathbf{f}(\mathbf{p})],$$

где λ — коэффициент регуляризации, настраиваемый на каждой итерации. Если снижение L быстрое, значение λ уменьшается, приближая алгоритм к методу Гаусса–Ньютона:

$$\Delta \approx [\mathbf{J}^T \mathbf{J}]^{-1} \mathbf{J}^T [\mathbf{y} - \mathbf{f}(\mathbf{p})],$$

иначе λ увеличивается, приближая шаг к направлению градиентного спуска:

$$\Delta \approx \lambda^{-1} \mathbf{J}^T [\mathbf{y} - \mathbf{f}(\mathbf{p})].$$

Чтобы сделать решение инвариантным к масштабу, алгоритм Марквардта решал модифицированную задачу, в которой каждая компонента градиента масштабировалась в соответствии с кривизной. Это обеспечивает более значительные изменения вдоль направлений с меньшим градиентом, что позволяет избежать медленной сходимости в этих направлениях. Флетчер в своей статье 1971 года *A modified Marquardt subroutine for non-linear least squares* упростил эту формулу, заменив единичную матрицу \mathbf{E} диагональной матрицей, состоящей из диагональных элементов $\mathbf{J}^T \mathbf{J}$:

$$[\mathbf{J}^T \mathbf{J} + \lambda \text{diag}(\mathbf{J}^T \mathbf{J})] \Delta = \mathbf{J}^T [\mathbf{y} - \mathbf{f}(\mathbf{p})].$$

Выбор коэффициента регуляризации

Эффективной стратегией управления λ является "отложенное вознаграждение": увеличение λ после неудачного шага и снижение после успешного. Это позволяет алгоритму быстро сходиться к решению, когда он находится вблизи минимума, и медленно исследовать пространство параметров, когда он находится далеко от минимума.

Детали реализации

Так как производительность реализации, основанной исключительно на теоретических выкладках, оказалась недостаточной для практического применения, в итоговом решении были внесены несколько улучшений.

Функция потерь

Функция потерь была изменена на χ^2 , так как она часто используется в задачах аппроксимации кривых. Она определяется следующим образом:

$$\chi^2(\mathbf{p}) = \sum_{i=1}^n \left(\frac{y_i - f(t_i, \mathbf{p})}{\sigma_i} \right)^2 = [\mathbf{y} - \mathbf{f}(\mathbf{p})]^T \mathbf{W} [\mathbf{y} - \mathbf{f}(\mathbf{p})],$$

где $\mathbf{W} = \text{diag} \left(\frac{1}{\sigma_1^2}, \dots, \frac{1}{\sigma_n^2} \right)$ — матрица весов: $\sigma_i^2 = \mathbb{D}[y_i]$. На практике она используется для увеличения веса измерений с меньшими ошибками.

Формула обновления для Δ была скорректирована, чтобы учитывать изменение функции потерь:

$$(\mathbf{J}^T \mathbf{W} \mathbf{J} + \lambda \mathbf{E}) \Delta = \mathbf{J}^T \mathbf{W} [\mathbf{y} - \mathbf{f}(\mathbf{p})].$$

Принятие шага

Ранее шаг принимался, если функция потерь уменьшалась, иначе он отклонялся, а коэффициент регуляризации увеличивался. Теперь шаг принимается, если метрика ρ больше порогового значения $\epsilon_4 > 0$ (step-acceptance в коде). Эта метрика измеряет фактическое уменьшение χ^2 по сравнению с улучшением, достигаемым шагом метода Левенберга-Марквардта.

$$\begin{aligned} \rho &= \frac{\chi^2(\mathbf{p}) - \chi^2(\mathbf{p} + \Delta)}{|(\mathbf{y} - \hat{\mathbf{y}})^T \mathbf{W} (\mathbf{y} - \hat{\mathbf{y}}) - (\mathbf{y} - \hat{\mathbf{y}} - \mathbf{J} \Delta)^T \mathbf{W} (\mathbf{y} - \hat{\mathbf{y}} - \mathbf{J} \Delta)|} \\ &= \frac{\chi^2(\mathbf{p}) - \chi^2(\mathbf{p} + \Delta)}{|\Delta^T (\lambda \Delta + \mathbf{J}^T \mathbf{W} (\mathbf{y} - \hat{\mathbf{y}}))|} \end{aligned}$$

где $\hat{\mathbf{y}} = \mathbf{f}(\mathbf{p})$.

Эта метрика для принятия шага была предложена Нильсеном в его статье 1999 года [3].

Выбранное значение для ϵ_4 — 10^{-1} .

Стратегия обновления

Коэффициент регуляризации и параметры модели обновляются согласно следующим правилам:

Если $\rho > \epsilon_4$: $\lambda = \max[\lambda/L_{\downarrow}, 10^{-7}]$, $\mathbf{p} \leftarrow \mathbf{p} + \Delta$
иначе: $\lambda = \min[\lambda L_{\uparrow}, 10^7]$

где $L_{\downarrow} \approx 9$ и $L_{\uparrow} \approx 11$ — фиксированные константы (REG_DECREASE_FACTOR и REG_INCREASE_FACTOR в коде). Эти значения были выбраны на основе статьи [2].

Критерии сходимости

Алгоритм останавливается, когда выполняется *одно* из следующих условий:

- Сходимость по норме градиента: $\max |\mathbf{J}^T \mathbf{W}(\mathbf{y} - \hat{\mathbf{y}})| < \epsilon_1$ (gradient_tol в коде)
- Сходимость по коэффициентам: $\max |\Delta/\mathbf{p}| < \epsilon_2$ (coefficients_tol в коде)
- Сходимость по (редуцированному) χ^2 : $\chi_{\nu}^2 = \chi^2/(m - n) < \epsilon_3$ (chi2_red_tol в коде)

где $\epsilon_1 = 10^{-3}$, $\epsilon_2 = 10^{-3}$, $\epsilon_3 = 10^{-1}$ — пороговые значения, заданные пользователем.

Начальное приближение

В задачах нелинейных наименьших квадратов функция потерь $\chi^2(\mathbf{p})$ может иметь множество локальных минимумов. В таких случаях метод Левенберга-Марквардта может сойтись к неудовлетворительному решению. Если это происходит, пользователь может попытаться задать лучшее начальное приближение для параметров, например, с помощью случайного поиска, или поиска по сетке, либо путем анализа данных.

Источники

1. [Wikipedia contributors. Levenberg–Marquardt algorithm. Wikipedia, The Free Encyclopedia..](#)
2. [H.P. Gavin, The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems. 2020.](#)
3. [H.B. Nielson, Damping Parameter in Marquardt's method. 1999.](#)