FACULTADE DE MATEMÁTICAS

Traballo Fin de Grao

Resolución numérica del problema no lineal de mínimos cuadrados. Aplicaciones a la estimación de parámetros de modelos matemáticos.

Dídac Blanco Morros

Curso Académico

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

GRAO DE MATEMÁTICAS

Traballo Fin de Grao

Resolución numérica del problema no lineal de mínimos cuadrados. Aplicaciones a la estimación de parámetros de modelos matemáticos.

Dídac Blanco Morros

Febrero, 2022

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Trabajo propuesto

Área de Coñecemento:
Título:
Breve descrición do contido
Recomendacións
Outras observacións

Índice

Resumen

Abstract

Introducción

Capítulo 1

Título del Capítulo 1

1.1. Fundamentos de la optimización sin restricciones

El problema de los mínimos cuadrados es un caso particular de optimización sin restricciones, y es por ello que comenzaremos introduciendo sus fundamentos. Ya que el problema de mínimos cuadrados es usado en multitud de campos para estimar parámetros, este es de los más utilizados dentro de los problemas de optimización sin restricciones.

Un problema de optimización sin restricciones tiene la forma

$$\min_{x} f(x) \tag{1.1}$$

donde $x \in \mathbb{R}^n$ y $f : \mathbb{R}^n \to \mathbb{R}$ es continuamente diferenciable, la llamamos **función objetivo**. La dificultad de un problema como este viene de no conocer el comportamiento global de f, normalmente solo disponemos de la evaluación de f en algunos puntos, y a lo mejor de algunas de sus derivadas. El trabajo de los algoritmos de optimización es identificar la solución sin usar demasiado tiempo ni almacenamiento computacional.

Notar que podemos usar la formulación (1.1) para referirnos tanto a los problemas de minimización como de maximización, basta sustituir f por $-f$.

Tenemos dos tipos de solución. Un punto x^* se dice **mínimo global** si $f(x^*) \leq f(x)$ para todo $x \in \mathbb{R}^n$. Como no se suele tener un conocimiento a gran escala de f debido a su coste, la mayoría de algoritmos solo encuentran mínimos locales, lo cual es suficiente para muchos casos prácticos. Un punto x^* se dice **mínimo local** si existe una vecinidad $\mathcal{V}$ de x^* tal que $f(x^*) \leq f(x)$ para todo $x \in \mathcal{V}$.

Aún así, los algoritmos para encontrar mínimos globales se suelen construír a partir de una secuencia de otros algoritmos de optimización local. También podemos aprovechar características fáciles de detectar en la función objetivo, como la convexidad, que nos asegura que un mínimo local será también global.

1.2. Generalidades de los algoritmos

Todo algoritmo de optimización sin restricciones comienza con un punto de partida, denotado normalmente como x_0. Aunque generalmente el usuario introduce una estimación razonable, el punto puede ser elegido por el algoritmo, tanto de forma sistemática como aleatoria. El algoritmo itera sobre x_0, creando una sucesión $\{x_k\}_{k=0}^{n}$ la cual termina cuando no pueda continuar o cuando ya se haya acercado razonablemente a la solución. Para decidir como se avanza de un x_k al siguiente, los algoritmos utilizan información sobre $f(x_k)$ o incluso en los puntos anteriores $x_0, x_1, \ldots, x_{k-1}$ con el objetivo de que $f(x_{k+1}) < f(x_k)$. Hablaremos de las dos estrategias fundamentales que se utilizan para avanzar de x_k a x_{k+1}, *búsqueda de línea* y *región de confianza*.

1.2.1. Búsqueda de línea

En este caso el algoritmo tiene dos tareas a partir de cada iteración, primero elige una *dirección* d_k y tomando el punto de partida busca en esa dirección el nuevo valor. Es decir, dado x_k

$$x_{k+1} = x_k + \alpha_k d_k \tag{1.2}$$

para un d_k elegido previamente, y un *paso* α_k obtenido solucionando otro problema de minimización más simple por ser unidimensional:

$$\min_{\alpha_k > 0} f\left(x_k + \alpha_k d_k\right). \tag{1.3}$$

Si se toma el α_k óptimo se le llama búsqueda de línea *exacta* u *óptima*. Para evitar el gran coste computacional que puede llegar a tomar, lo más común es tomar un α_k que aporte un descenso aceptable, en cuyo caso se le llama búsqueda de línea *inexacta* o *aproximada*. Desde el nuevo punto se busca otra dirección y paso para repetir el proceso. Veamos brevemente cómo se eligen d_k y α_k.

La mayor parte de algoritmos de este tipo necesitan que d_k sea una dirección descendente, esto es, $d_k^T \nabla f_k < 0$, lo cual asegura que en esa dirección se podrá reducir el valor de f. Esta suele tener la forma

$$d_k = -B_k^{-1} \nabla f_k \tag{1.4}$$

con B_k una aproximación de la matriz Hessiana $\nabla^2 f(x_k)$ simétrica y no singular. Según lo que acabamos de decir, necesitamos que B_k sea definida positiva. En las tres corrientes principales se elige un B_k distinto, en el *método del descenso máximo* o *descenso del gradiente*, se usa la matriz identidad I. En el *método de Newton* se usa la matriz exacta, mientras que en los *métodos Quasi-Newton* la matriz Hessiana es aproximada para cada x_k.

En el caso de la elección de α_k, el caso ideal sería encontrar el óptimo en **??**, pero esto es en general demasiado costoso. Debido a ese coste, se suelen utilizar búsquedas inexactas probando una serie de puntos hasta que alguno cumpla unas condiciones preestablecidas con las que se acepta el paso dado. Estas condiciones son por ejemplo las condiciones *Wolfe* o las condiciones *Goldstein*. Esta elección se hace en dos fases, primero un proceso elige un intervalo conteniendo los pasos deseables y una segunda fase donde se va reduciendo el intervalo por técnicas de interpolación o bisección.

1.2.2. Región de confianza

Esta estrategia enfoca el problema de otro modo, primero se fija una distancia máxima Δ_k para definir la región, que generalmente es de la forma

$$\Omega_k = \{x : \|x - x_k\| \leq \Delta_k\} \tag{1.5}$$

y luego ya se busca la dirección y paso. A partir de la información conocida de f, para cada x_k se modela una función m_k que se comporte de manera similar a f cerca de este punto. Generalmente se utiliza el modelo cuadrático de la forma

$$m_k := q^{(k)}(p) = f(x_k) + g_k^T p + \frac{1}{2} p^T G_k p, \tag{1.6}$$

donde $g_k = \nabla f(x_k)$ y $G_k = \nabla^2 f(x_k)$. Este modelo cuadrático es el utilizado en los métodos de búsqueda de línea para determinar la dirección de búsqueda, mientras que en este caso lo usamos para tener una representación adecuada de la función objetivo y así elegir el mínimo dentro de esta región. Este método nos evita el problema de que la Hessiana no sea definida positiva. En cada iteración, una vez elegido Δ_k se resuelve el siguiente problema:

$$\begin{aligned} \min_p \quad & q^{(k)}(p) = f(x_k) + g_k^T p + \frac{1}{2} p^T B_k p \\ \text{s.a.} \quad & \|p\| \leq \Delta_k. \end{aligned} \tag{1.7}$$

Notamos que en el modelo se escribe B_k en lugar de G_k, pues no siempre se usa esta última. Debido al coste computacional, como vimos en la elección de la dirección de búsqueda, a veces se prefiere aproximar de alguna manera más o menos eficiente, e incluso puede ser aceptable tomar la matriz 0.

También se puede variar qué norma define la región de confianza, cambiando así la forma de esta y ofreciendo distintos resultados, aunque generalmente se utiliza la bola definida por $\|p\|_2 \leq \Delta$.

El tamaño de la región es crítica para que una iteración sea efectiva y cuanto lo sea

Anexo I

Título del Anexo I

Anexo II

Título del Anexo II

Bibliografía

[1] Nocedal, J., & Wright, S. (2006). *Numerical Optimization* (2nd ed.). Springer.

[2] Sun, W., & Yuan, Y.-X. (2006). *Optimization theory and methods: Nonlinear programming* (2006th ed.). Springer.