Diego Hernandez Rodriguez
@eu_sou_dieguinho
C311

Esta tarea representa un analisis del modelo $f(x,y) = (x^2+1)^{(y^2+1)} + e^y + e^{-y}$ mediante el uso de los algoritmos, aprendidos en las conferencias de Algoritmos para problemas Irrestrictos y con restricciones, de Maximo Descenso y de Newton.

1 Problema a analizar

El problema a analizar es la ecuacion:

$$f(x,y) = (x^2+1)^{(y^2+1)} + e^y + e^{-y}$$

Su dominio son los reales ya que no tiene punto de indefinicion y es continua en todo su dominio por composicion de funciones continuas elementales.
Esta representa la siguiente funcion en 3D:

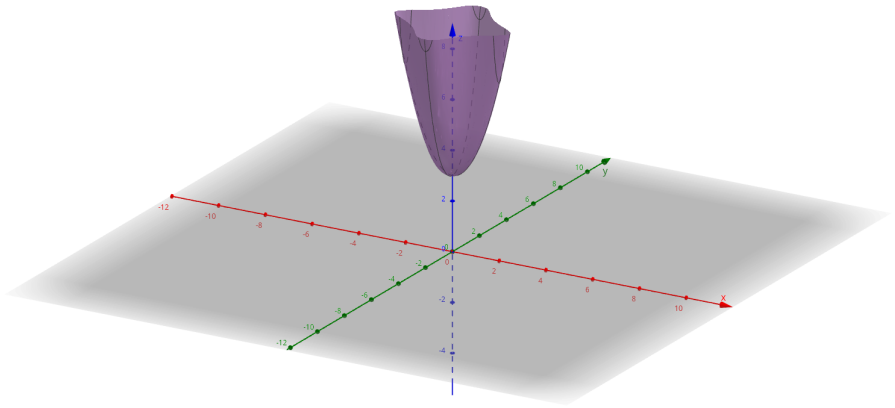


Figura que muestra la continuidad de dicha funcion.

2 Analisis del modelo, soluciones y convexidad

Analizando el problema mas a profundidad, aqui vemos las derivadas parciales de la funcion:

$$\frac{\partial f}{\partial x} = 2x(y^2+1)(x^2+1)^{y^2}$$

$$\frac{\partial f}{\partial y} = 2y\ln(x^2+1)(x^2+1)^{(y^2+1)} + e^y - e^{-y}$$

Sus segundas derivadas parciales:

$$\frac{\partial^2 f}{\partial x^2} = (y^2+1)(2(x^2+1)^{y^2} + (2xy)^2(x^2+1)^{(y^2-1)})$$

$$\frac{\partial^2 f}{\partial y^2} = 2\ln(x^2+1)(x^2+1)^{(y^2+1)}(1+2y^2\ln(x^2+1)) + e^y + e^{-y}$$

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x} = 4xy(x^2+1)^{y^2}(1+(y^2+1)\ln(x^2+1))$$

Y dadas las segundas derivadas no creo que sea necesario representar la matriz Hessiana.

Dicha funcion es continua y estrictamente positiva en todo su dominio.
Ahora para ver la existencia de sus minimos locales vamos a analizar sus derivadas parciales para ver cuando el vector gradiente se hace igual al (0,0).

$\frac{\partial f}{\partial x} = 0$ si y solo si $x = 0$, puesto que $(y^2+1) > 0$ para todo y, $(x^2+1) > 0$ para todo x y $(x^2+1)^{(y^2)} > 0$ para todo x, por tanto $2x$ debe ser igual a 0, por tanto $x = 0$.
Ahora viendo $\frac{\partial f}{\partial y} = 0$ con $x = 0$, tenemos que $\ln(x^2+1) = \ln(1) = 0$, por ende e^y–e^-y tiene que ser igual a 0, osea $y = -y$, lo que implica que y tambien es igual a 0, el punto $P(0,0)$ es un minimo local.
Ahora analizando las derivadas parciales de segundo orden podemos ver que:

$(x^2+1)^{(y^2)} > 0$ para todo x, y

$y^2(x^2+1)^{(y^2-1)} > 0$ para todo x, y

$(y^2+1) > 0$ para todo y

$(2x)^2 > 0$ para todo x

por tanto $\frac{\partial^2 f}{\partial x^2} > 0$ para todo x, y

$\ln(x^2+1) > 0$ para todo x

$(x^2+1)^{(y^2+1)} > 0$ para todo x, y

$\ln(x^2+1)y^2 > 0$ para todo x, y

$e^y + e^y > 0$ para todo y

Por tanto $\frac{\partial^2 f}{\partial y^2} > 0$ para todo x, y

Para el analisis de convexidad de la funcion use las librerias de simpy y numpy

de python, las cuales tienen metodo de calculo de la matriz hessiana y en dependencia de esta empieza a ver convexidad en todos los puntos en un rango especifico, lo cual determino no convexidad en numerosos puntos, para verificar se puede compilar el archivo check_convexity.py y ver los resultados.

3 Descripcion de los algoritmos

Los metodos utilizados son el de Newton y el de Maximo descenso:

El metodo de Maximo descenso es un algoritmo iterativo que busca el minimo siguiendo la direccion opuesta del vector gradiente, el maximo descenso de la funcion, consiste en comenzar con un punto x_0 y luego para cada k iteracion $x_{(k+1)} = x_k - a_k df(x_k)$. Este algoritmo es facil de implementar pero tiene una convergencia relativamente lenta, dada la cantidad de iteraciones necesarias.

El Metodo de Newton en cambio usa informacion de las derivadas de segundo orden, de la matriz Hessiana, aproximando la funcion localmente por una cuadratica, comienza en un punto x_0 inicial y luego en cada iteracion $x_{(k+1)} = x_k - (d^2 f(x_k))^{-1} df(x_k)$, dadas sus caracteristicas tiene una convergencia muy rapida.

En el archivo config.json tengo un conjunto de categorias, cada una con 24 puntos para la experimentacion con los algoritmos, a continuacion un breve resumen de lo visto con ambos algoritmos en cada categoria:

3.1 Cerca del minimo

Ambos tuvieron 10 de 24 puntos convergentes, ambos fallaron cuando $x < 0$, aunque Newton tuvo una convergencia en menor cantidad de iteraciones.

3.2 Lejos del minimo

Mucha mayor robustez con el metodo de Newton, puesto que tuvo muchos mas puntos convergentes, solo que a veces diverge al infinito, Maximo descenso solo tuvo un punto convergente.

3.3 Valores grandes en x

5 puntos mas que convergieron con Newton que con Maximo descenso, puesto que este metodo se maneja mejor en valores grandes de x y en menor cantidad de iteraciones.

3.4 Valores grandes en y

Ambos tuvieron dificultades con valores grandes de y, pero Newton tuvo mejores resultados, aunque tuvo mayor cantidad de iteraciones.

3.5 Zona Plataforma

Maximo descenso tuvo convergencia en casi todos los puntos y en no muchas mas iteraciones que Newton, que fue un poco mas rapido.

3.6 Puntos criticos estacionarios

Ambos fallan en los puntos sobre los ejes donde el gradiente es 0, por lo cual tuvieron pocos puntos convergiendo.

3.7 Direcciones diagonales

Newton claramente superior en cuestion de robustez en direcciones diagonales, aunque Maximo descenso tuvo muchas menos iteraciones

3.8 Cerca de los ejes coordenados

Los dos metodos tienen bastantes fallos cerca de los ejes pero Newton es mas rapido.

3.9 Valores intermedios

En este caso Newton es mas rapido pero Maximo descenso converge mas a menudo.

3.10 Patrones especiales

Newton es mucho mas complejo en patrones complejos.

3.11 Regiones en cuadrantes

Newton es la unica opcion viable en cuadrantes con valores grandes, dado que Maximo descenso no tuvo valores convergentes.

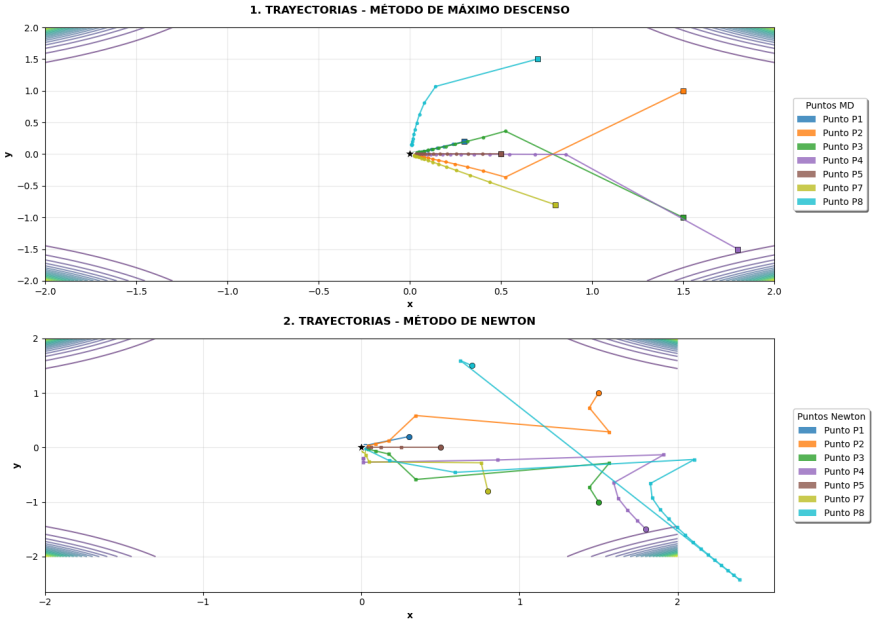
3.12 Escala logaritmica

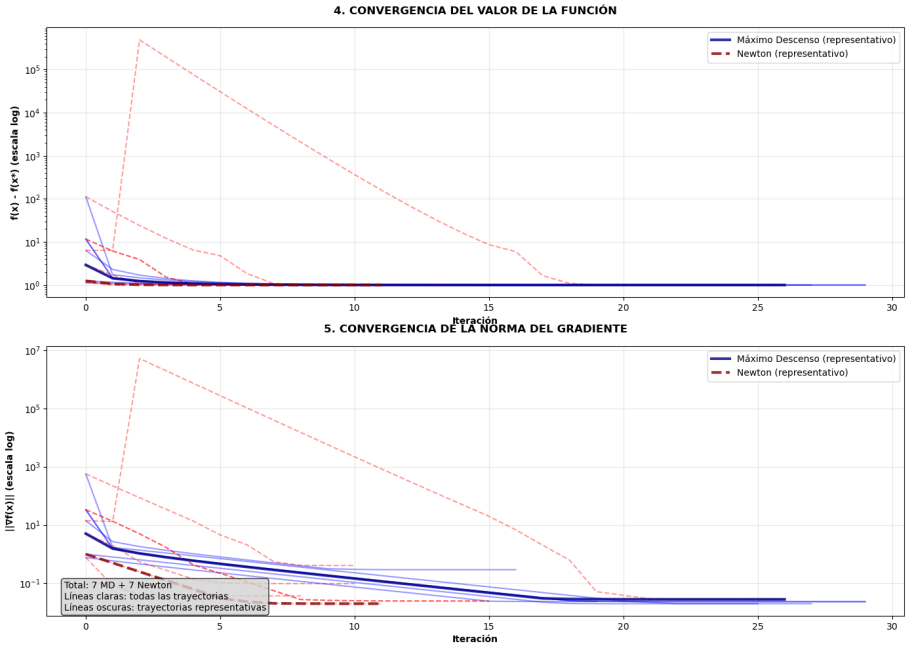
Como los gradientes son nulos en la mayoria de los casos ambos fallan, teniendo solo 1 punto convergente cada uno.

En el init_points.json se puede encontrar un conjunto de puntos los cuales estan seleccionados de la siguiente forma:

- El punto 1 es un punto cercano al minimo
- Los puntos del 2 al 4 son puntos lejos del minimo
- Los 5 y 6 se encuentran en los ejes
- Los 7 y 8 son puntos cerca de puntos potenciales criticos

Aqui se evidencian los resultados obtenidos con estos puntos en las graficas siguientes:




4. CONVERGENCIA DEL VALOR DE LA FUNCIÓN
Máximo Descenso (representativo)
Newton (representativo)
f(x) - f(x*) (escala log)
Iteración
5. CONVERGENCIA DE LA NORMA DEL GRADIENTE
Máximo Descenso (representativo)
Newton (representativo)
||∇f(x)|| (escala log)
Iteración
Total: 7 MD + 7 Newton
Líneas claras: todas las trayectorias
Líneas oscuras: trayectorias representativas

4 Conclusiones

Los resultados de dichos algoritmos pueden verse con el codigo que puedes encontrar en el `https://github.com/IamSkale/tarea_MO` donde en el config.json puedes modificar los puntos que quieras verificar y obtener resultados distintos a los mostrados en este informe.

A modo de resumen pudimos ver con este codigo que el metodo de Newton tiene una convergencia a resultados en mucha menor cantidad de iteraciones que el metodo de maximo descenso, pero a su vez es mas dificil de implementar y de entender lo cual lo hace menos escalable.