

Optimización numérica

Proyecto final 1. PCS (GCP+RC)

Santiago Novoa Pérez. 8 de diciembre del 2015.

Introducción

El método de programación cuadrática sucesiva (PCS) es uno de los métodos más efectivos para resolver problemas de optimización no lineales. Para que el método funcione se necesita que tanto la función objetivo como las restricciones sean dos veces continuamente diferenciables.

Los métodos PCS resuelven una secuencia de subproblemas de optimización en la que, en cada subproblema, se intenta resolver un modelo cuadrático de la función objetivo sujeta a una linealización de las restricciones; puede ser utilizado tanto con un marco de regiones de confianza como uno de búsqueda lineal.

En general los métodos PCS se pueden ver como una generalización del método de Newton para optimización sin restricciones dado a la manera en que encuentran una dirección de descenso.

PCS es apropiado para problemas grandes y pequeños con funciones no lineales; en estos casos la solución se alcanza en un número de iteraciones sustancialmente menor que n .

Uno de los problemas que encuentra el método de PCS es que al enfrentarse a regiones donde las matrices no tienen la curvatura deseada (tienen curvatura negativa o cero) el algoritmo se sale de lugar y no puede converger. Existen muchas formas de tratar estos problemas, ya sea con regularización o incorporándole GC al método para siempre tener direcciones de descenso.

Para este proyecto en particular, nos interesará volver el método local de PCS (por problemas de inercia) en un método global incorporando regiones de confianza. Para entender mejor cómo funcionan las regiones de confianza en problemas con restricciones, incorporaremos el algoritmo dogleg a un método de gradiente conjugado proyectado con región de confianza, intentando, así, resolver el problema de la inercia en muchos de los problemas de PCS de una vez por todas.

El problema que buscamos resolver es:

$$\begin{aligned} \text{minimizar} \quad & f(x) \\ \text{s.a.} \quad & c(x) = 0 \end{aligned}$$

$$\begin{aligned} f : \mathbb{R}^n &\longrightarrow \mathbb{R} \\ c : \mathbb{R}^n &\longrightarrow \mathbb{R}^m \\ \Omega = \{x \in \mathbb{R}^n \mid &c(x) = 0\} \end{aligned}$$

Sin embargo, aplicando regiones de confianza, el problema que trataremos será:

$$\begin{aligned} \text{minimizar} \quad & f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla_{xx}^2 \mathcal{L} p \\ \text{s.a.} \quad & A(x)p + c(x) = 0 \\ & \|p\| \leq \Delta \end{aligned}$$

El problema anterior es un problema cuadrático que se puede resolver usando Gradiente Conjugado Proyectado. Lo que hace GCP es resolver el problema de minimizar

Construcción del método

- Condiciones de Optimalidad

Sean

$$\begin{aligned} L(x, \lambda) &= f(x) - \lambda^T c(x) \\ A(x) &= [\nabla c_i^T(x)]_{i=1\dots m} \end{aligned}$$

Si x^* es minimizador local de f , entonces $\exists \lambda^*$ tal que:

$$\begin{aligned} \nabla_x L(x^*, \lambda^*) &= \nabla f(x^*) - \sum_{i=1}^m \lambda_i^* \nabla c_i(x^*) \\ &= 0; \\ \nabla_\lambda L(x^*, \lambda^*) &= c(x^*) \\ &= 0; \end{aligned}$$

Estas ecuaciones resultan de intentar ver en dónde el gradiente de la función lagrangiana se anula para poder encontrar el mínimo de esa función (condiciones necesarias de primer orden).

Si además los gradientes de las restricciones son L.I. (complementariedad estricta), entonces el cono crítico y el espacio nulo de A son el mismo y podemos ver la convexidad de L en:

$$w^T \nabla_{xx} L(x^*, \lambda^*) w \geq 0 \quad \forall w \in C(x^*, \lambda^*)$$

con $C(x^*, \lambda^*)$ el cono crítico en x^* , λ^* , que son todas las direcciones que cumplen con que mantienen las restricciones (en una aproximación lineal) cercanas a factibilidad, es decir:

$$C(x^*, \lambda^*) = \{w \in \mathbb{R}^n \mid \nabla c_i(x^*)^T w = 0 \quad \forall i \in \{1, \dots, m\}\}$$

Lo que esto asegura es que cuando el gradiente sí se anule, o la dirección que se tome sea de descenso o todas las direcciones sean de ascenso (usando la expansión de Taylor y el Lema de Farkas: estar en el cono (con su respectivo λ) o que exista descenso). También se les conoce como condiciones necesarias de segundo orden.

Si queremos asegurar que el problema (convexo) va a encontrar su mínimo en x^* (condiciones suficientes de segundo orden), tenemos que ver que la Hessiana de la matriz Lagrangiana sea positiva definida en la proyección de las direcciones del cono crítico:

$$0 < w^T \nabla_{xx} L(x^*, \lambda^*) w \quad \forall w \in \{C(x^*, \lambda^*) \setminus w = 0\}$$

■ Método cuadrático

La matriz de KKT del problema anterior queda expresada como sigue:

$$K = \begin{bmatrix} W_k & A^T(x_k) \\ A(x_k) & 0 \end{bmatrix}$$

donde W_k es la matriz Hessiana de la función Lagrangiana del problema de optimización.

La K es un resultado de escribir las condiciones de optimalidad descritas anteriormente en un sistema de ecuaciones:

$$\begin{bmatrix} W_k & A^T(x_k) \\ A(x_k) & 0 \end{bmatrix} * \begin{bmatrix} h_x \\ -\lambda_{k+1} \end{bmatrix} = - \begin{bmatrix} \nabla f(x_k) \\ c(x_k) \end{bmatrix}$$

Es fácil verificar que resolver el sistema anterior es equivalente a resolver otro subproblema cuadrático:

$$\begin{aligned} \min \quad & \frac{1}{2} p^T Q_k p + g_k^T p \\ \text{s.a.} \quad & A_k^T p + c_k = 0 \end{aligned}$$

Aquí parece que se ha perdido el significado inicial del problema dado que se empezó hablando de $\nabla_{xx} L(x_k)$ y de si su proyección en el espacio nulo de $A(x_k)$ (todas las direcciones en $C(x_k, \lambda_k)$ con k lo suficientemente grande para que x_k y λ_k se encuentren en el cono factible (una vecindad factible de la solución)) era positiva definida, sin embargo, gracias al resultado de un artículo de Nicolas I.M. Gould ¹, este nuevo problema de optimización tiene

¹**Nicolas I.M. Gould**; *On practical conditions for the existence and uniqueness of solutions to the general equality quadratic programming problem*

propiedades deseables. En primer lugar, si la proyección de la Hessiana de la Lagrangiana sobre el espacio nulo de los gradientes de las restricciones es SPD, la solución de los dos modelos existe y es única. También se prueba que el mínimo del problema de optimización se puede calcular usando una base del espacio nulo. Sin embargo, el resultado más importante (por su practicidad) de la lectura vincula a la inercia de la Hessiana proyectada sobre el nulo con la inercia de K.

$$\begin{aligned}
& Z_k^T \nabla L(x_k, \lambda_k) Z_k = Z_k^T W_k Z_k > 0 \\
\implies & \exists! p \text{ t.q. } p \text{ minimiza el problema cuadrático} \\
& \therefore p \text{ minimiza al problema original} \\
& \text{inercia}(K) = \text{inercia}(Z^T W_k Z^T) + (m, 0, m) \\
& \therefore \text{inercia}(K) = (m, 0, n) \iff Z^T W_k Z^T \text{ es SPD}
\end{aligned}$$

La inercia de una matriz es una ternia que dice el número de valores propios menores a cero, cero y mayores a cero respectivamente :

$$\text{inercia}(A) = (\lambda_-, \lambda_0, \lambda_+)$$

Todos estos resultados nos aseguran que, si la inercia de la matriz K es la correcta $(m, 0, n)$, resolver el problema de minimización con restricciones de igualdad formulado al principio de este trabajo es equivalente a ir resolviendo una serie de subproblemas cuadráticos convexos representados por las ecuaciones de KKT. Paso a paso será necesario resolver, para cada matriz K, el correspondiente vector h que resulte en el lado derecho de la ecuación. La solución (por ser un problema convexo), si las condiciones anteriores se cumplen, existirá paso a paso y será única.

$$K * h_k = LD = - \begin{bmatrix} \nabla f(x_k) \\ c(x_k) \end{bmatrix}$$

Este proceso parece ser igual de complicado que resolver el problema de minimización del inicio, sin embargo, gracias a métodos, como el programado en matlab llamado ldl, que factorizan matrices en matrices de bloques diagonales, es posible encontrar la inercia de una matriz y resolver sistemas de ecuaciones lineales sin que el costo computacional sea tan alto.

Convergencia global

Como dijimos anteriormente, regiones de confianza es una manera muy sencilla de asegurar que haya convergencia global, sin embargo, como estamos tratando con problemas con restricciones, las cosas se pueden complicar un poco.

El enfoque de regiones de confianza hace que al calcular la nueva dirección se haga un acuerdo entre el paso a dar y la región dentro donde se da, si el acuerdo es malo la región se encoge, si el acuerdo es bueno (el paso es de descenso y no sales de factibilidad), la región se agranda. Para mantener a las restricciones en este acuerdo se usa a la función de mérito

para hacer el modelo, sin embargo, para nuestra prueba sólo queremos ver qué pasa con dogleg y GCP.

Otra dificultad que tienen regiones de confianza cuando tratan problemas con restricciones es que muchas veces no son compatibles las regiones con las restricciones. El primer pensamiento que usualmente surge con esto es agrandar la región para que incluya a las restricciones, pero eso sería una contradicción al acuerdo y a cómo trabajan regiones de confianza. Una manera correcta de tratar con este problema es dividir al problema de minimización en 2 subproblemas, uno inicial que nos otorgue una r con la cuál vamos a relajar nuestras restricciones (dogleg hace esto) y otro que resuelva GCP ya con las nuevas condiciones.

$$\begin{aligned} \text{minimizar} \quad & f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla_{xx}^2 \mathcal{L} p \\ \text{s.a.} \quad & A(x)p + c(x) = r_k \\ & r_k = A_k v^* + c_k \\ & \|p\| \leq \Delta \end{aligned}$$

Este sistema relajado tiene que resolver primero el para el valor de v^* :

$$\begin{aligned} \text{minimizar} \quad & \|A_k v + c_k\| \\ & \|v\| \leq 0,8\Delta_k \end{aligned}$$

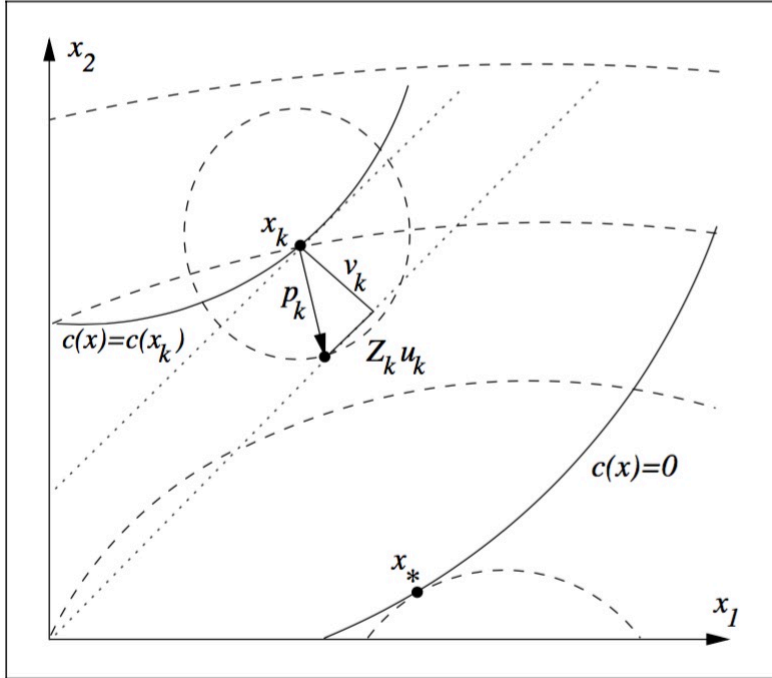


Figure 18.3 The step $p_k = v_k + Z_k u_k$ of the trust-region method.

Para calcular cuánto va a relajar las restricciones, dogleg usa dos direcciones, la dirección de Cauchy (p^U) y la dirección de Newton o el paso completo (p^N). Después de calcular las dos direcciones avanza completamente en la dirección de Cauchy y al final de ella le pega la dirección de Newton formando así una curva parametrizada de x_0 hasta el final de la dirección de Newton. Lo que hace después es cortar ese segmento creado en donde éste corte con la región de confianza:

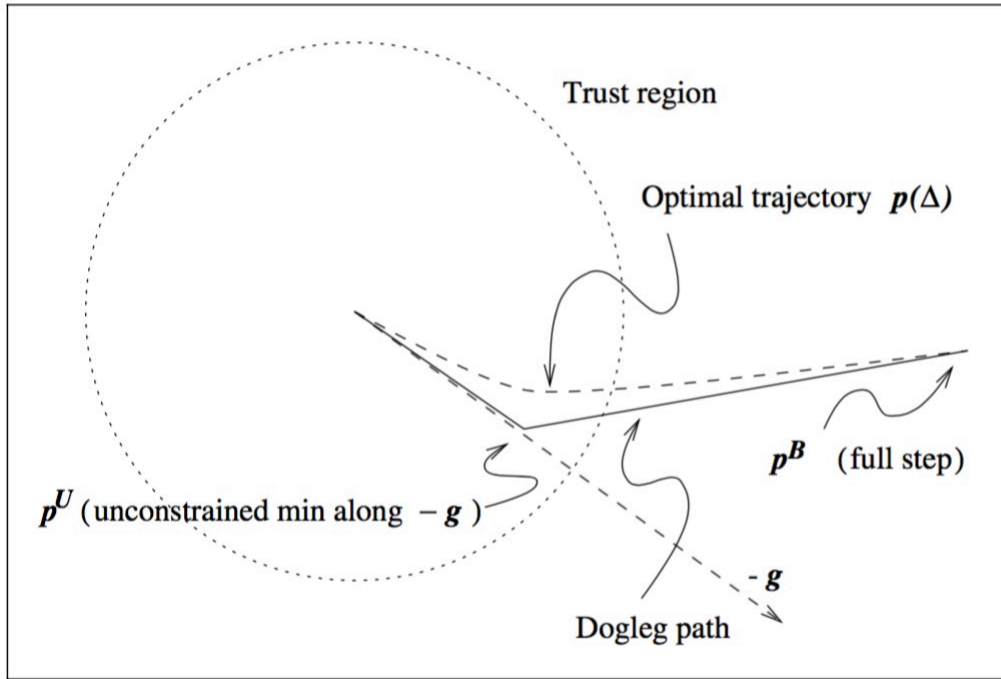


Figure 4.3 Exact trajectory and dogleg approximation.

Este proceso nos deja con 2 posibles resultados:

- La región de confianza corta antes de que empiece el paso de Newton, lo cual quiere decir que el paso de Newton solito está dando un avance mayor a lo que la región de confianza acepta por lo que sería conveniente ver si el acuerdo es bueno y tal vez agrandar la región.
- La región de confianza corta a la dirección después de que el segmento de recta dió vuelta y tal vez las direcciones obtenidas en esta región no son las mejores por lo que podría convenir recortar la región.

Experimento

1. Constantes $c_1 = 10^{-4}$, $maxiter = 1000$, $TOL = 10^{-7}$
2. Condiciones de paro

$$\begin{aligned} k &\geq maxiter \\ \sqrt{|r_k^T * y_k|} &\leq TOL(r_0^T * y_0) \end{aligned}$$

3. Reporte del experimento

Cuadro 1: Matrices Poisson con GCP+RC

n	m	$\ \Delta\ $	$\ res\ $	$\ p\ \geq \ \Delta\ $	$\ p\ $	iter	CPU(s)	err rel	fact pr
10000	100	1	3.40E+00	TRUE	1	0	9.65E-03	1.00E+00	1.59E-17
10000	200	1	3.68E+00	TRUE	1	0	1.40E-02	1.00E+00	2.78E-17
10000	9900	1	7.05E+00	TRUE	1	0	6.02E-01	1.00E+00	2.90E-15
10000	100	25	1.03E+01	TRUE	25	26	2.85E-02	6.72E-02	7.23E-15
10000	200	25	1.04E+01	TRUE	25	27	2.40E-01	6.39E-02	2.45E-14
10000	9900	25	1.14E+01	FALSE	22.3508	33	1.88E+00	1.97E-06	5.90E-14
10000	100	125	1.03E+01	FALSE	49.7498	193	1.91E-01	1.08E-06	8.74E-14
10000	200	125	1.04E+01	FALSE	49.5547	205	1.88E+00	1.00E-06	3.58E-13
10000	9900	125	1.74E+01	FALSE	36.2494	33	1.68E+00	2.00E-06	9.81E-14
10000	100	10000	10.29563	FALSE	49.7498	193	0.1901534	1.0789E-06	8.7359E-14
10000	200	10000	10.3923	FALSE	49.5547	205	1.737791	1.0013E-06	3.576E-13
10000	9900	10000	17.37815	FALSE	36.2494	33	1.75811	1.9995E-06	9.8137E-14

Observaciones

1. Para probar el algoritmo lo único que se hizo fue agarrar diferentes tamaos para la región inicial y ver si alguna de las direcciones chocaba con la región antes de otorgar una respuesta por GCP.
2. Todas las iteraciones tienen curvatura positiva ya que elegimos probar el algoritmo con matrices Poisson.
3. Como todas las iteraciones están en regiones de curvatura positiva, los pasos tomados son grandes en general, es decir, se agrandaría la Δ en el siguiente paso para la mayoría de condiciones iniciales que se consideran en problemas de este tipo ($\Delta = 1$ es una región usual para empezar).
4. Como cortamos el algoritmo una vez que la dirección chocara con la región (para no tener que calcular el acuerdo), podemos ver como baja el error una vez que el algoritmo corre y termina (sólo hay que fijarse en la diferencia entre las primeras 3 iteraciones y las últimas 6 donde el resultado sí es un minimizador local).

5. Las direcciones que dogleg arroja (res) también van creciendo conforme la región crece, sin embargo parece que depende también del preconditionador que se esté usando, es decir, dónde se está proyectando el problema.