

Instituto Tecnológico Autónomo de México
Enero-Mayo 2018
Métodos Numéricos y Optimización
Resumen sobre lectura '*Convex Optimization for Big Data*'
Prof. Erick Palacios Moreno

Alumno: Federico Riveroll
Clave Unica: 105898

30 de mayo de 2018

Introducción

Esta lectura nos habla sobre optimización convexa aplicada a datos contemporáneos y lo hace en tres grandes capítulos. Primeramente nos habla de métodos de primer orden para optimización convexa suave (y no suave). Después profundiza con el escalamiento de Big Data vía la aleatorización. Finalmente, nos habla sobre el rol que juegan y jugarán tanto el cómputo en paralelo como el cómputo distribuido.

Métodos de primer orden

En este capítulo se muestran varias técnicas para realizar soluciones en funciones “no suaves” con la técnica de ‘gradiente próximo’ y métodos de primer orden, específicamente algoritmos que garantizan encontrar el mínimo global (o máximo). El gradiente es muy simple de actualizar:

$$x^{k+1} = x^k - \alpha_k \nabla f(x^k),$$

Si bien la pérdida en número de iteraciones que el gradiente ‘pierde’, lo recupera con creces con el bajo costo en tiempo de cada iteración.

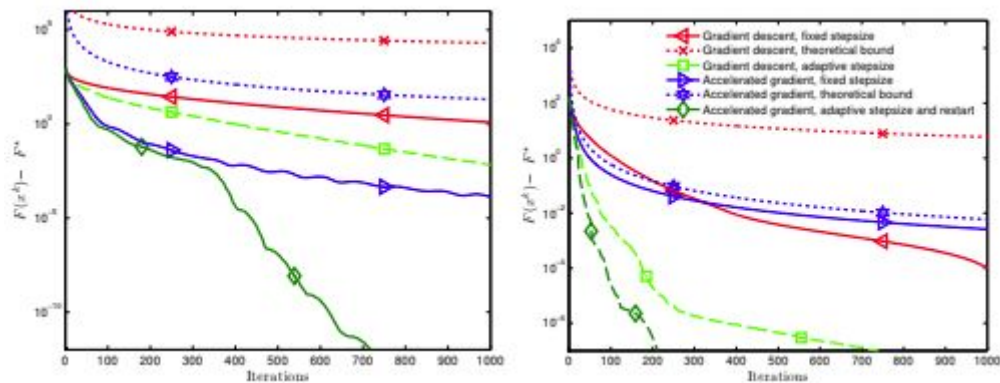
El capítulo menciona otros métodos (o variaciones de este método) como el gradiente acelerado de Nesterov, el cual es considerado un método óptimo de primer orden, y logra la mayor ‘acertividad’ y convergencia óptima con un muy simple detalle:

Algorithm 1 Nesterov’s accelerated gradient method for unconstrained minimization ($v^0 = x^0$) [11]

- 1: $x^{k+1} = v^k - \alpha_k \nabla f(v^k)$
 - 2: $v^{k+1} = x^{k+1} + \beta_k (x^{k+1} - x^k)$
-

Agregar otro paso “momentum step” con parámetro Beta(k).

Los resultados tanto en formulación LASSO como LS mejoran de manera notable con los “ajustes de mejora” como el anterior.



Escalamiento de 'Big Data' vía aleatorización

Otro tema interesante que toca la lectura es el problema que -en práctica, no en teoría- se enfrentan los métodos de primer orden con muchos datos. Dado que los métodos de primer orden son bastante robustos y aproximativos, se pueden reemplazar cálculos determinísticos con estimadores de carácter estadístico y acelerar las rutinas de álgebra lineal básica aleatorizando.

Las grandes dificultades a las que se encuentra el escalamiento de 'Big Data' vía aleatorización son la comunicación entre las computadoras y sobre todo dentro de la memoria local, las cuales pueden reducir significativamente el rendimiento de los métodos de primer orden. La otra dificultad es la sincronización para realizar los cálculos de forma distribuida. Los métodos de primer orden deben coordinar a varios ordenadores en cada iteración que dependen del mismo vector y este proceso le alenta mucho cuando una máquina lleva más tiempo que las otras y por eso la gran necesidad de los algoritmos asíncronos.

Paralelización y cómputo distribuido

Volviendo al problema de los métodos cuya descendencia es coordinada por un gradiente y sus elementos en una ubicación central, el desarrollo de métodos paralelos para desempeñar estas actividades está ahora mismo con alto crecimiento. Una gran parte del problema en el desarrollo de versiones paralelas de estos métodos mencionando el algoritmo de Jacobi, resuelven sistemas lineales aplicando muchas actualizaciones simultáneas en paralelo (en tiempo) de descenso coordinado. De tal modo, que un procesador no necesita llevar todas las actualizaciones sino que cada uno sólo necesita comunicar una actualización y solo necesita recibir las actualizaciones de las coordenadas.

Para llevar a cabo exitosamente el cómputo en paralelo (descentralizado y asíncrono) es fundamental la formulación del problema convexo, por ejemplo, el cálculo de la dirección y magnitud del gradiente cuando el objetivo se descompone naturalmente. Es posible procesar diversas ejecuciones con una máquina usando cómputo local. Cada procesador se

comunica con el central y de ahí se determina la dirección y magnitud de un gradiente final que 'acelera'.

Conclusión

El planteamiento dividido en tres secciones que hacen los autores toca puntos muy interesantes sobre cómo optimizar funciones aunque no sean suaves, cómo escalar dichos métodos con la aleatorización y cómo optimizar con un tema que ha sido clave en el curso que es la paralelización y el cómputo distribuido.

Los conjuntos de datos contemporáneos escalan en tamaño y complejidad de manera creciente dada la cantidad de componentes que ahora generan datos, por ejemplo cualquier proceso electrónico conectado a internet (móviles, aplicaciones, sitios web, electrodomésticos, -Internet of things- etc..). En este contexto, el rol que juega el diseño de la optimización convexa que pueda estar al día con ese crecimiento de tamaño de data sets -que no va paralelo con el avance en hardware y software- (va mucho más rápido), es clave identificar aproximaciones algorítmicas que dependan de estructuras clave.

Se espera que estas nuevas herramientas de aproximación continuarán siendo desarrolladas para poder cubrir estos retos. Lo cual invita a la pregunta de que si podemos usar modelos "composite" para obtener la misma información pero de manera mucho más rápida.