Convex Optimization for Big Data

Volkan Cevher, Stephen Becker & Mark Schmidt

El artículo plantea como el avance en el poder de computo, la existencia de algoritmos eficientes y el uso de la geometría convexa ha generado una mayor demanda en el uso de estas técnicas. Sin embargo, la cantidad de información que se genera hoy en día es tal que incluso el cómputo en paralelo y distribuido no son suficientes para procesar la información de una manera eficiente. Por tal motivo, los autores consideran que es necesario exista una evolución de la optimización convexa que permita trabajar con el "Big Data".

Los autores consideran que la optimización convexa del Big Data se puede clasificar en tres grandes pilares:

1. Métodos de primer orden
2. Aleatoriedad
3. Cómputo paralelo y distribuido

Cabe mencionar que esto no significa que sean caminos independientes, en realidad lo consideran tres principios que al combinarlos permiten obtener buenos resultados en los cálculos del Big Data.

Los métodos de primer orden tienen su motivación en toda la investigación que se ha realizado para resolver y problemas lineales, así como la optimización de los algoritmos que los resuelven. Sin embargo, estos problemas de primer orden pueden tener relacionadas funciones con diversas características:

- Funciones convexas y diferenciables; en este caso el descenso por gradiente es usado ampliamente debido a su relativamente fácil escalabilidad a dimensiones mayores.
- Funciones convexas con funciones diferenciables y una función convexa no diferenciable; en este caso el descenso en gradiente próximo (proximal-gradient) es una elección para buscar las soluciones ya que la convergencia de este método aprovecha las "estructura compuesta" de ambas funciones.
- Ninguno de los anteriores; en este caso son problemas que no son ni diferenciable ni tipo "Lipschitz", sin embargo, se usan el Lagrangiano y su descomposición dual para resolverlos.

Resolver problemas de Big Data mediante la aleatoriedad implica usar algunos métodos de primer orden, pero adaptándolos a realizar cálculos aproximados. Esto implica buscar un punto de equilibrio entre rapidez y precisión. Algunos de los métodos utilizados para esto son:

- Descenso coordenado; este método selecciona un conjunto aleatorio de coordenadas y calcula el gradiente únicamente respecto a estas coordenadas. Logrando así una convergencia similar al método completo de usar todas las coordenadas.
- Gradiente estocástico; este método actualiza las todas las coordenadas a la vez, pero realiza una aproximación del cálculo del gradiente, reduciendo así los cálculos necesarios.
- Métodos algebraicos aleatorios; se basan en general cálculos aproximados de las operaciones algebraicas (producto de matrices, producto de vectores, etc) o extraer submatrices de forma aleatoria para generar los cálculos necesarios.

El computo en paralelo y distribuido toma ventaja del incremento exponencial en la capacidad de almacenamiento y cómputo a lo largo del tiempo. Por tal motivo es posible mejorar métodos de primer orden usando mayor capacidad de cómputo, para esto existen algunos métodos:

- Algoritmos embarazosamente paralelizables; son aquellos algoritmos de primer orden que se benefician del cómputo en paralelo al distribuir diversas operaciones que no dependen unas de otras y por lo tanto ejecutarlas simultáneamente, por ejemplo, el cálculo del gradiente.
- Métodos de primer orden con comunicación descentralizada; el objetivo de estos métodos es evitar el "cuello de botella" que se genera al tener que enviar toda la información a un mismo procesador. En este caso, el algoritmo de descenso coordenado toma ventaja al hacer que cada unidad de procesamiento procese con una única coordenada y al final cada unidad tenga que enviar información asociada a una única coordenada en lugar de todas a la vez.
- Métodos de primer orden asíncronos con comunicación descentralizada; un algoritmo que puede tomar ventaja de esto es el gradiente estocástico debido a que únicamente calcula una aproximación del gradiente. Bajo condiciones adecuadas, es posible que este método mantenga la convergencia que ya es robusto ante información que no esté actualizada del todo (ya que siempre es una aproximación) y por lo tanto los procesos asíncronos no le impactan significativamente.

Como conclusión entiendo que los autores plantean que mejorar los tiempos y la precisión de los algoritmos en la era del Big Data no debe depender únicamente de tener mayor poder de cómputo. Es un proceso donde el desarrollador debe buscar optimizar los algoritmos desde la base, i.e., que el algoritmo use de manera eficiente los recursos disponibles. Una vez en este paso, la idea es evaluar la posibilidad de ceder precisión y exactitud del cálculo a cambio de obtener los resultados de una forma más rápida, i.e., incluir aleatoriedad en los cálculos para disminuir el número de operaciones a realizar. Y una vez se llegue a un óptimo en este aspecto, entonces comenzar a evaluar si es factible y la forma en que se pueda aprovechar el gran poder de cómputo que los clusters y las GPU's proporcionan para distribuir y paralelizar los cálculos.

Víctor Augusto Samayoa Donado

CVU: 175750