Convex Optimization for Big Data

*by* *Volkan Cevher, Stephen Becker, and Mark Schmidt*

El artículo revisa los avances en los algoritmos dedicados a la optimización convexa para Big Data que buscan disminuir costos computacionales como almacenamiento y *bottlenecks.* Primero se describe de manera general este nuevo campo de estudio, luego se explican técnicas contemporáneas como métodos de primer orden y por último, se compara las ventajas y desventajas entre computación distribuida y paralela.

**Optimización convexa en el surgimiendo de Big Data**

La importancia de las formulaciones de convexidad y optimización ha aumentado drásticamente en los últimos 10 años dado el auge de modelos de aprendizaje estadística.

El autor describe dos razones principales que causaron el auge en este campo. Por un lado, se encuentra la eficiencia en los algoritmos para computer soluciones globales y, por el otro, la capacidad de usar geometría convexa para probar condiciones útiles sobre la solución. Asimismo, la formulación puede fácilmente ser usada en varias disciplinas.

Sin embargo, al mismo tiempo del auge nos enfrenamos a la principal problemática que conlleva grandes cantidades de datos. Por ejemplo, el flujo de información de las redes sociales, internet, mensajes, mails, son cada vez mayores. En respuesta a esto, optimización convexa se está reinventando para poder satisfacer las nuevas necesidades del Big Data.

**3 pilares de algoritmos de optimización**

La mezcla de los siguientes tres conceptos se complementan uno al otro para ofrecer enorme beneficios de escalabilidad para la optimización del Big Data.

* Aleatorización: Las técnicas de aleatorización son reconocidas entre los métodos de aproximación para mejorar la escalabilidad del modelo de primer orden ya que podemos controlar su comportamiento esperado. Algunos ejemplos son la actualización aleatoria y parcial en las variables de optimización, reemplazar el gradiente determinista y hacer más rápidas las rutinas básicas lineales por medio de la aleatorización.
* Computación en paralelo y distribuido: Métodos de primer orden naturalmente dan un flexible marco teórico para distribuir tareas de optimización y realizar computaciones en paralelo. Asimismo podemos aumentar la eficiencia de estos métodos con aproximaciones para aumentar los niveles de escalabilidad, desde algoritmos sincronizados con comunicaciones centralizadas y hasta gigantes algoritmos asíncronos con Comunicaciones desentralizadas.
* Métodos de primer orden: Los métodos de primer orden obtienen baja o media precision en las soluciones numéricas. Estos métodos presentan ratios de convergencia de dimensiones casi independientes. Estos son teóricamente robustos a la aproximacion y tipicamente dependen en elementos ideales para la computación distribuida y paralela.

Los problemas de Big Data necesitan una revision exahustiva sobre cómo diseñamos los algoritmos de optimización convexa y proponer decisions no-convencionales. Para resolver mayores problemas con relativamente poco crecimiento en recursos computacionales, este artículo plantea la importancia en identificar los principales fundamentos de la estructura de algoritmos y sus costos computacionales.

A partir de las resticciones en la sincronización y comunicación disponibles por el hardware, surge la necesidad de escoger algoritmos que eficienticen su uso. Se espera que nuevas herramientas de aproximación van a continuar siendo descubiertas para adaptar los algoritmos de convexidad a la heterogeneidad de los problemas computacionales.

Además, se predice el auge en el uso de modelos complejos y sus correspondientes principios de mapeado próximo para problemas de Big Data para tartar con ruido y otras restricciones. Por ejemplo, la formulación de la estimación LASO garantiza que se cumplen propiedades más Fuertes que la del estimación de LS. En otras palabras, para obtener más de la misma base, debemos usar métodos complejos para obtener la misma información pero de manera más rápida.