Convex Optimization for Big Data (Volkan Cehver, Stephen Becker and Mark Schmidt)

Lorena Malpica 124841

31 de mayo de 2018

El artículo revisa los avances más recientes en algoritmos de optimización convexa para Big Data, éstos tratan de reducir cuellos de botella tanto computacionales, como de almacenamiento y de comunicaciones. Este es un campo emergente y se describen las técnicas contemporáneas de aproximación, como métodos de primer orden o aleatorización para escalabilidad. También se analiza el importante rol que tiene el cómputo distribuido y en paralelo. Estos nuevos algoritmos están basados en principios bastante simples y son capaces de acelerar la resolución de muchos problemas.

La optimización convexa en procesamiento de señales ha estado presente desde que este campo del conocimiento surgió. Sin embargo su importancia se ha incrementado dramáticamente en la última decáda con el surgimiento de teorías nuevas en el área y el desarrollo de métodos estadísticos como las máquinas de Soporte Vectorial. Estas formulaciones ahora se ocupan en una gran variedad de aplicaciones de procesamiento de señales en las áreas de geofísica, imagenología médica y bioinformática.

Hay muchas razones de importancia que explican la explosión en el interés en esta área, siendo las dos más obvias las siguientes:

1) La existencia de algoritmos eficientes para calcular soluciones que alcanzan un óptimo global.
2) La habilidad de utilizar geometría convexa para probar propiedades útiles acerca de la solución.

Una formulación convexa unificada también permite la transferencia de conocimiento útil a través de diferentes disciplinas, como muestreo y computación, que se enfocan en diferentes aspectos del mismo problema matemático. Sin embargo esta nueva popularidad en el área de optimización convexa ha dado como resultado que estos algoritmos tengan la presión constante de poder aplicarse a conjuntos de datos cada vez más grandes y a la resolución de problemas de enormes dimensiones. Problemas de actuales con datos de tráfico de Internet, texto o imágenes producen conjuntos de datos en el rango de terabytes a exabytes.

A pesar del progreso del cómputo en paralelo y distribuido la utilidad de algoritmos clásicos que no vaya más allá de discusiones teoréticas respecto a las posibles soluciones de diversos problemas de optimización. Por esta razón, la optimización convexa se está reinventando para la era de Big Data donde el tamaño de los datos y de los parámetros es demasiado grande para que se procesen de manera local. Adicionalementa incluso las rutinas más sencillas de álgebra lineal como la descomposición de Cholesky o multiplicaciones entre matrices o entre matrices y vectores que son rutinarias en muchos algoritmos se han vuelto problemáticas debido al tamaño de los datos. Por otro lado, muchos algoritmos convexos ya no necesitan alcanzar soluciones de alta precisión dado que muchos modelos de Big Data son intrínsicamente simples o inexactos.

Podemos describir los aspectos fundamentaes de la optimización para Big Data con la siguiente formulación compuesta:

$$F^* \stackrel{\text{def}}{=} \min_x \left\{ F(x) \stackrel{\text{def}}{=} f(x) + g(x) : x \in \mathbb{R}^p \right\}$$

Figure 1:

Donde f y g son funciones convexas. Se revisan métodos numéricos eficientes para obtener una solución óptima para x^* así como las suposiciones que se deben hacer para f y g. Este tipo de problemas de

minimización convexa surgen de manera natural en procesamiento de señales cuando se quieren estimar parámetros desconocidos $x_0 \in R^P$ a partir de datos $y \in R^n$.

Un entendimiento básico de algoritmos de optimización para Big Data recae en 3 pilares:

Métodos de primer orden: Los métodos de primer orden obtienen soluciones numéricas con una precisión entre baja y media, utilizando únicamente información del oráculo de primer orden del objetivo, por ejemplo estimaciones del gradiente. Estos métodos tienen tasas de convergencia que son prácticamente independientes de la dimensión. Típicamente son ideales para cómputo distribuido o en paralelo.

Aleatorización: Las técnicas de aleatorización resaltan particularmente entre otras técnicas de aproximación porque realzan la escalabilidad de los métodos de primer orden dado que podemos controlar su comportamiento esperado. Algunas de las ideas clave incluyen actualizaciones parciales al azar de las variables de optimización, y la aceleración de rutinas básicas de álgebra lineal por medio de la aleatorización.

Cómputo distribuido y en paralelo: los métodos de primer orden proveen de manera natural un marco de trabajo flexible que permite distribuir diversas tareas de optimización y realizar operaciones en paralelo. La escalabilidad se puede incrementar aún más si en lugar de usar algoritmos completamente sincrónicos y con comunicaciones centralizadas se ocupan algoritmos asíncronos con comunicaciones decentralizadas.

Estos tres conceptos se complementan entre si y proveen muchísimos beneficios para la optimización de algoritmos de Bid Data. Por ejemplo métodos aleatorizados de primer orden pueden presentar una aceleración significativa comparados con sus contrapartes determinísticas dado que pueden llegar a una solución de muy buena calidad únicamente inspeccionando una pequeña fracción de los datos.

Una modelo común en diversos problemas de Big Data es el de las observaciones lineales que está presente en muchas disciplinas:

$y = \phi x_0 + z$

donde x_0 es un parámetro desconocido, $\phi \in R^{n \times p}$ es una matriz conocida y $z \in R^n$ representa perturbaciones desconocidas o ruido, que se modelan típicamente con entradas Gaussianas iid con media zaro y varianza σ^2. Observaciones lineares típicamente surgen directamente de fenómenos regidos por las leyes básicas de la física como es el caso de imagenología a través de resonancia magnética o problemas de geofísica.

La formulación convexa clásica en este escenario siempre ha sido el estimador de mínimos cuadrados, que puede ser resuelto de manera eficiente con métodos de subespacios de Krylov usando únicamente multiplicaciones de matrices y vectores. Una variante importante también es el estimador LASSO.

En teoría los métodos de primen orden se encuentran muy bien posicionados para atacar problemas de gran escala. En la práctica sin embargo, el número exacto de cálculos númericos que demandan las iteraciones de estos métodos hacen que aún estos métodos sean difíciles de aplicar conforme crece la dimensión del problema que se quiere resolver. Afortunadamente, los métodos de primer orden son bastante robustos ante el uso de aproximaciones en sus primitivas de optimización. Por lo tanto el uso de aproximaciones por medio de aleatorización incrementa el alcance de los métodos de primer orden.

Un ejemplo de estas aplicaciones son los métodos de descenso por coordenadas, estos métodos tienen una larga historia en el campo de la optimización y se relacionan con métodos clásicos como la estrategia de reducción cíclica de Gauss-Seidel para resolver sistemas lineares. La consideración esencial en todos los métodos de descenso por coordenadas es la elección de cada coordenada i en cada iteración.

Por otro lado además de los métodos de descenso por coordenadas que actualizan una sola coordenada a la vez con su gradiente exacto, los métodos de gradiente estocástico actualizan todas las coordenadas a la vez pero en lugar de usar gradientes exactos ocupan aproximaciones de los gradientes. Al igual que en los métodos de descenso por coordenadas, el aspecto crucial en el diseño de los métodos de gradiente estocástico es la selección de los puntos de datos j seleccionados en cada iteración. De manera análoga se obtienen mejores tasas de convergencia al escogir a j de manera uniforme y al azar en lugar de ir recorriendo todos los datos.

En problemas de Big Data, operaciones básicas de álgebra lineal como la descomposición de matrices (por ejemplo obtener eivenvalores, descomposiciones de Cholesky, y SVD) o multiplicaciones entre matrices generan

grandes cuellos de botella debido a las enormes dimensiones de los datos de entrada. La idea de los métodos aleatorizados de álgebra lineal es o aproximar $M = Q(Q^T M)$ con $Q \in R^{p \times r}$, o construir al azar por filas o por columnas una representación de bajo rango de la matriz, con el objetivo de acelerar el tiempo de cómputo.

Gracias a la ley de Moore, el poder computacional en cuanto a procesamiento y almacenamiento ha ido acelerándose a tasas exponenciales desde mediados de los 2000, esto le ha dado al área de optimización convexa un empujón considerable en terminos de eficiencia. Sin embargo aunque se espera que la ley de Moore siga aplicando en los años venideros, las eficiencias de los transistores se han estancado, por lo tanto debemos de incrementar el uso del cómputo distribuido y en paralelo.

Aunque los métodos de primer orden paracen ser ideales para acelerarse con el uso del cómputo en paralelo hay dos obstáculos principales para lograr estos objetivos:

1) Comunicación: Problemas de comunicación entre computadoras o entre las diferentes jerarquías de la memoria dentro de una misma computadora reducen de manera significativa la eficiencia de estos métodos. Hay dos enfoques principales para atacar este problema, el primero es diseñar algoritmos que minimicen la comunicación. El segundo es eliminar el vector maestro x^k y en lugar de eso trabajar con copias locales en cada máquina y que cada una llegue a un consenso x^* al momento de la convergencia. 2)Sincronización: Para realizar tareas computacionales de manera distribuida, los métodos de primer orden necesitan coordinar las actividades de diversas computadoras cuyas primitivas numéricas dependan del mismo vector x^k en cada iteración. Sin embargo esto alenta mucho el proceso. Para aliviar este problema de sincronización, algoritmos asíncronos permiten actualizaciones usando versiones anteriores de sus parámetros.

Para resolver problemas de optimización convexa de cada vez mayor tamaño el artículo deja claro que es necesario identificar trade-offs en cuanto a las aproximaciones de los algoritmos que estén relacionadas con su estructura central, un ejemplo de esto es la aleatorización tal y como se implementa en muchos de los métodos de primer orden. Otro aspecto importante es mejorar los algoritmos para que puedan realizarse sin tanta necesidad de sincronización y comunicación entre núcleos y/o computadoras, siendo ésta otra área potencial de mejoría. También se predice un gran incremento en el uso de modelos compuestos junto con sus correspondientes principios de mapeo proximal para resolver diversos problemas de Big Data que tengan que ver con el ruido o con otras limitaciones.