**Resumen:**

El artículo revisa los avances más recientes en optimización convexa que buscan reducir los cuellos de botella de almacenamiento, cómputo y comunicación para los datos de gran escala. El artículo da un panorama de las técnicas como los métodos de primer orden, aleatorización para escalabilidad, y el cómputo en paralelo y distribuido.

Aunque la optimización convexa ha existido por muchos años su importancia ha aumentado en la última década por la mejora en eficiencia de algoritmos de soluciones óptimas y la habilidad del uso de geometría convexa para probar propiedades de las soluciones. Sin embargo, el autor comenta que esta popularidad ha hecho que la optimización convexa sea utilizada en contextos con una gran cantidad de datos. Para responder a esto, la optimización convexa está encontrando nuevas herramientas en aquellos problemas donde es imposible procesar localmente en donde rutinas sencillas se vuelven prohibitivos pero donde también no es necesario obtener soluciones de alta precisión.

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

Los autores toman la formulación anterior como la descripción fundamental de una optimización para big data, siendo f y g funciones convexas se buscan métodos numéricos para obtener soluciones para x*.

El autor describe 3 conceptos importantes que él considera son pilares fundamentales para entender la optimización de algoritmos en el contexto de big data ordenando con ello las secciones del paper.

**• First-order methods:**

Los métodos de primer orden obtienen soluciones numéricas de baja y media precisión, En śu mayor parte son independientes al número de dimensiones y al ser implementados por “primitivos” computacionales son fácilmente paralelizables y pueden también soportar variantes de optimización convexa "smooth" y "non smooth" al usar el principio de mapeo proximal.

1.1 Smooth objectives

Los autores comparan entre estos modelos a las regresiones lineales implementadas con mínimos cuadrados y el modelo Lasso regularizado en donde el segundo tiene la ventaja de producir soluciones “sparse” dado que la lambda de regularización es un término “non-smooth”. Los autores evalúan cómo en algunos casos aunque distintos métodos puedan ser costosos en número de iteraciones como el método de gradientes el costo por iteración es tan bajo que puede competir con métodos con menores iteraciones pero que son más complejos tomando más tiempo en llegar a la misma precisión.

También se menciona la propiedad estructural de la convexidad fuerte (que puede encontrarse tanto con funciones smooth como con non-smooth)  al agregar dentro de un problema convexo un término de regularización cuadrada que ofrece beneficios importantes como la existencia de un minimizador único.

**2 Big Data scaling via randomization:**

Los autores comentan que los métodos de primer orden en teoría son capaces de trabajar con problemas de gran escala, sin embargo en la práctica demandan mucha capacidad computaciones conforme la dimensión aumenta. Afortunadamente éste tipo de métodos son robustos con el uso de aproximaciones como es el caso de gradiente y cálculos proximales.

Las técnicas de aleatorización son particularmente buenas en aumentar la escalabilidad de los métodos de primer orden dado que ayudan a controlar su comportamiento. Entre las ideas más importantes se encuentra la actualización aleatoria parcial de las variables de optimización reemplazando los cálculos deterministas de los gradientes con estimadores estadísticos.

2.1 Coordinate descent methods
Un ejemplo que se menciona que ejemplifica la demanda computacional es la formulación de pagerank que requiere una operación matriz-vector en cada iteración. Los autores hablan sobre el uso de la familia de métodos descenso cardinales en este problema y que esta estrategia puede llegar a ser incluso más costosa que la calculación de gradiente misma. En este caso la elección aleatoria de la coordenada obtiene la misma tasa de convergencia.

2.2 Stochastic gradient methods
A diferencia de los métodos de descenso de coordenadas que actualizan una coordenada por etapa al igual que el de gradiente los métodos estocásticos actualizan todas las coordenadas de forma simultánea ésto hace al método potencialmente más rápido aunque lo vuelve difícil de hacer el setup.

**3 The role of parallel and distributed computation:**

Aunque la ley de moore se espera que continúe aún durante algunos años mejorando la capacidad computacional y de almacenamiento el escalamiento de estos recursos están demandando niveles de consumo de energía muy altos, esto hace que sea cada vez más claro que el desarrollo de este tipo de algoritmos debería de tender a la paralelización y computación distribuida. Los autores enfatizan el papel de la paralización al optimizar los métodos de primer orden distribuyendo la optimización en tasks y también utilizando algoritmos asíncronos con comunicación descentralizada.

Aunque los métodos de primer orden parecen ser óptimos para paralelizar existen dos problemas fundamentales que requieren solución:

• Communication: Un primer problema que se encuentra es comunicación poco eficiente entre las computadoras y la memoria local, esto puede reducir de forma significativa la eficiencia numérica de los métodos de primer orden. Los autores proponen enfocarnos en el desarrollo de algoritmos que busquen minimizar la comunicación y , segundo, eliminar los vectores maestros y en su lugar trabajar con copias locales en cada máquina.

• Synchronization: El segundo problema que mencionan es la sincronización ya que los métodos de primer orden deben coordinar las actividades de distintas computadoras para las que sus primitivos numéricos dependen del mismo vector en cada iteración. Éste procedimiento alenta todo el proceso si alguna computadora le toma más que a las demás haciéndolas esperar. Para resolver este problema los autores proponen el desarrollo de algoritmos asíncronos que vayan actualizando distintas versiones de los parámetros sin detener los procesos.

Conclusión:
Los autores concluyen que es muy claro que para poder resolver problemas de optimización convexa cada vez más grandes bajo el ritmo de crecimiento modesto de capacidades computacionales es necesario identificar de forma inteligente los trade offs que existen en el uso de aproximaciones algorítmicas. Así también mencionan que los algoritmos deben de estar pensados a poder adaptarse al tipo de plataformas computacionales con las que se cuenta hoy en día y por último concluyen la importancia de los modelos “compuestos” para obtener más información de los datos que tenemos u obtener los resultados de forma más rápida.