Reporte: Convex Optimization for Big Data

Pablo Soria Garcia 111969

El autor presenta el problema de formulaciones convexas y su optimización como uno altamente relevante en el contexto actual, esto derivado del surgimiento de nuevas teorías primero alrededor de datos escasos y problemas de minimización de rangos y segundo al gran uso que se le da hoy en día a problemas de aprendizaje estadístico como por ejemplo las SVM.

Además plantea que existe una gran presión sobre los algoritmos de optimización convexa para que estos sean capaces de aceptar bases de datos que aumentan su tamaño de manera exponencial conforme avanza el tiempo. En respuesta la optimización convexa se está reinventando para acomodarse el Big Data en donde tanto los datos como la cantidad de parámetros son demasiado grandes como para procesarlos de forma local.

El problema básico de optimización en Big Data se plantea de la siguiente manera:

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

Dónde f, g son funciones convexas, el autor plantea tres pilares básicos para entender el problema de optimización que se resumen a continuación.

1 Métodos de primer orden

Estos métodos generalmente obtienen bajo o mediana precisión en soluciones numéricas usando solamente información de primer orden como por ejemplo estimaciones del gradiente. Un problema arquetípico es el de encontrar un estimador de mínimos cuadrados. El primer modelo que explora el autor en esta sección es la regresión regularizada LASSO, esta consiste en incorporar un parámetro lambda a la regularización que realiza la regularización y selección de variables dentro del mismo algoritmo.

Posteriormente, el autor presenta una serie de métodos para optimizaciones en dónde la función objetivo es “suave”, el primero de ellos es el de gradiente, el cual consiste en actualizar la lista de parámetros mediante iteraciones que calculan la dirección del descenso hasta encontrar una convergencia.

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

Algo interesante que propone el autor es que aun cuando el método del gradiente requiere una mayor cantidad de iteraciones que métodos más complejos, el costo computacional de hacer dichas iteraciones en el método del gradiente es muchas veces menor que hacer una sola de métodos complejos y esta es una de las principales razones por las que su uso es bastante extendido.

Existen modificaciones a este método por ejemplo la propuesta por Nesterov, en esta propuesta “acelerada”, que propone que el tamaño de paso del descenso sea 1/L dónde L es:

$$L = \|\Phi\|_2^2$$

Y se agrega un segundo paso al realizar el descenso buscando un parámetro que va en función del número de iteraciones k.

$$\beta_k = \frac{k}{k+3}$$

Este método acelerado alcanza según el autor, la mejor solución del peor caso posible de la tasa de error y es por esto que se habla de una solución de primer orden óptima.

Adicionalmente el trabajo menciona que podemos transformar cualquier problema convexo en uno fuertemente convexo aplicando la misma lógica que la regularización L1 pero con un término cuadrático, a esto se le conoce como regularización L2 o de Ridge, este problema converge de forma geométrica a un minimizador único cuando elegimos el tamaño del paso como 1/L.

El autor también defiende que los métodos de descenso gradiente pueden ser utilizados para problemas “no suaves” aplicando suavizadores a dichos términos y realizando el descenso de forma análoga a lo expuesto anteriormente para el caso “suave”.

2 Aleatorización

El autor sugiere que desde el punto de vista teórico, los métodos de primer orden están bien preparados para resolver problemas de dimensión muy grande, sin embrago en la práctica, con el rápido crecimiento de las fuentes de datos, estos métodos comienzan a ser privativos desde el punto de vista de poder computacional. De esta forma surgen técnicas de aleatorización que maximizan las capacidades y alcance de los métodos de primer orden.

Los métodos de descenso coordinados, seleccionan una coordenada i de forma aleatoria del vector x, y solamente modificar la variable correspondiente xi para mejorar la función objetivo, esto se realiza en cada una de las iteraciones en lugar de realizar el cálculo completo para todo el vector de variables. La idea principal de este método está en la selección de la coordenada en cada paso de la iteración.

Una posible selección de la coordenada es elegir aquella en la que la derivada direccional más grande, otra idea es la de ciclar a lo largo de todos las posibles coordenadas de manera secuencial pero de forma intuitiva, no esperaríamos una velocidad considerable para lograr la convergencia. Los métodos de aleatorización se centran en elegir la coordenada de forma aleatoria para mejorar la forma de escogerla, específicamente el autor propone una aleatorización asumiendo que la coordenada se distribuye de forma uniforme

en un espacio discreto que corre de 1 a p dónde p es el número de parámetros a encontrar.

De manera similar otro enfoque que realiza la actualización de todos los parámetros en cada iteración es el método conocido como descenso estocástico, este método realiza las iteraciones calculando una aproximación al gradiente y actualizando todos los parámetros en el mismo paso.

El autor propone también ciertas mejoras que buscan acelerar la convergencia de este tipo de métodos, como por ejemplo utilizar tamaños de paso grandes al principio y el uso de promedios ponderados a lo largo de las iteraciones.

Adicionalmente el trabajo estipula que recientemente se ha realizado esfuerzos importantes de cara a la aleatorización de las operaciones algebraicas, la idea detrás de esto es generar una representación de bajo rango de la matriz original por medio de un subconjunto de columnas o renglones escogidos de forma aleatoria ganando así control sobre la distribución de los errores.

Computo distribuido

Existen dos problemas clásicos al momento de enfrentarnos a la paralelización de los algoritmos, el primero es el problema de comunicación a lo que el autor sugiere que se produce al no diseñar de forma correcta estos canales dentro de los mismos algoritmos e indica que una posible solución se puede encontrar en a) minimizar la cantidad de comunicación y b) la creación de copias locales de los vectores de variables que eventualmente lleguen a un consenso en convergencia.

El segundo problema es la sincronización para realizar las tareas realmente de forma distribuida, el autor sugiere que el uso de algoritmos asíncronos que utilizan una versión no actualizada del mismo vector.

Algo interesante que siguiere el autor en la sección final es el uso de tecnologías que tienen a la paralelización como filosofía central en su desarrollo como ejemplo esta hadoop, spark, el paradigma MapReduce y el lenguaje PIG, todos ellos vistos en el contexto de esta maestría.

Por último el autor explora el uso de algoritmos asíncronos argumentando que es posible usar versiones no actualizadas del vector de parámetros y en dónde la actualización de estos vectores se realiza de forma independiente en cada procesador. El trabajo finaliza con una predicción interesante:

Con el objetivo de obtener más de los mismos datos, es necesario usar modelos compuestos por los tres grandes rubros que explora el autor es decir soluciones que conjuguen la facilidad de interpretación de las condiciones de primer orden optimizados con aleatorización al momento de elegir las coordenadas y pensado e implementado por medio de una folosofía intrínseca de paralelización.