Daniel Sharp 138176

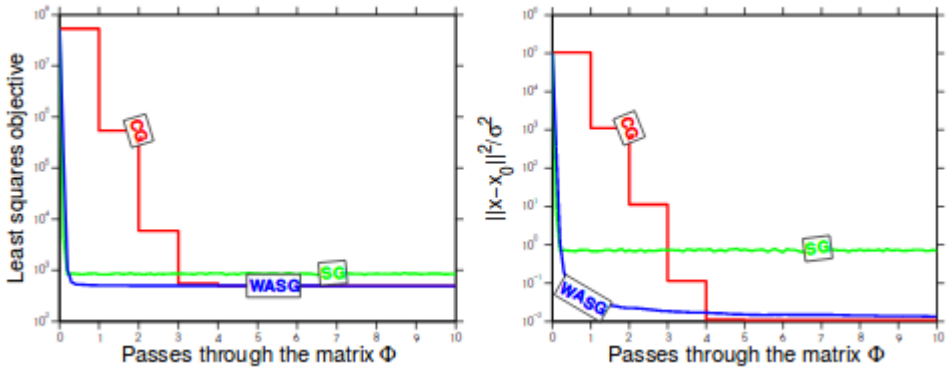
Convex Optimization for Big Data
Volkan Cevher, Stephen Becker y Mark Schmidt

Los autores del articulo introducen explicando que la teoría de optimización convexa se ha puesto a prueba en los últimos años. Esto debido a que es dificil utilizar los mismos algorítmos que han utilizado tradicionalmente para resolver problemas con cantidades masivas de datos, como los que se generan hoy en día. Para contrarrestar estos problemas, se han desarrollado algorítmos adaptados a problemas con grandes datos, que descansan en tres pilares: métodos de primer orden, aleatorización y cómputo en paraelo. Posteriormente los describen, como lo resumiré a continuación:

- Métodos de primer orden

Estos consisten, a grandes rasgos, en obtener estimaciones de baja o mediana exactitud a través de métodos que requieren menor costo computacional. En esta sección, los autores presentan las formulaciones del modelo de mínimos cuadrados con penalización: LASSO, que penaliza los coeficientes de la regresión con un valor absoluto; y Ridge, que penaliza los pesos de los coeficientes al calcular la norma del vector.

Ejemplifican la estrategia de utilizar los métodos de primer orden con el método de gradiente estocástico y, a través de la siguiente gráfica, muestran como este permite una convergencia mucho más rápida con respecto al método del gradiente conjugado.



Es posible observar como los métodos de gradiente estocástico (SG y WASG) alcanzan el mínimo mucho más rápido que con gradientes conjugados. Algo que no sabía, era que incluso es posible obtener un estimado del número de iteraciones que el algoritmo de gradiente estocástico para llegar a una solución con una tasa de error predefinida.

Dentro de los métodos de gradiente estocástico, se ha intentado hacer ligeras modificaciones al modelo básdico para mejorar su desempeño. El modelo básico ess el siguiente:

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

Y sobre el, se han hecho ciertas propuestas, como el método de gradiente acelerado de Nesterov, que consiste en agregar un factor de inercia para acelerar la convergencia:

Algorithm 1 Nesterov's accelerated gradient method for unconstrained minimization ($v^0 = x^0$) [11]

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

2: $v^{k+1} = x^{k+1} + \beta_k (x^{k+1} - x^k)$

Una beneficio clave que ofrece el estimador de Ridge (mencionado anteriormente) es que permite convertir cualquier problema convexo en uno fuertemente convexo, que implica que para el problema existe un único minimizador y ofrece mayor eficiencia de optimización.

Posteriormente, los autores describen el problema canónico compuesto, que consiste en una función f diferenciable y una función convexa g no suave, por ejemplo, cuando se utiliza regularización LASSO, pues el componente de penalización, al ser un valor absoluto no es suave y no es diferenciable en sus vertices. Para este tipo de problemas, los autores presentan los métodos de gradiente próximo, que aprovechan la estructura compuesta para obtener tasas de convergencia similares a los problemas 'suaves'.

- Escalamiento de grandes datos con aleatorización

Los autores presentan una serie de métodos para aprovechar los beneficios de la aleatorización de los datos. Comienza explicando los métodos de descenso en coordenadas, que coniste en tomar unidades individuales de los datos en cada iteración y calcular el gradiente para estos, mejorando así el gradiente global a un mucho menor costo computacional. Existen varias formas de elegir las observaciones, pero la que ha demostrado resultados más satisfactorios es la aleatoriedad, pues el costo de obtener la observación es independiente del taamaño total de la tabla y aun así logra una buena tasa de convergencia. Estas ideas dan pie a los métodos de gradiente estocástico, que realizan esta mismos pasos para iterar sobre unos datos y ajustar los coeficientes de forma tal que se minimice el error.

Después explican como esta misma estrategia puede utilizarse en álgebra matricial para descomposición o multuplicación de matrices. Se puede aproximar a través de los cálculos con sub-matrices de la completa, de forma que los calculos son significativamente más económicos.

- Cómputo distribuido y en paralelo

Finalmente, los autores comentan las bondades de realizar el cómputo en paralelo, que ha probado ser realmente eficiente. Esto es especialmente importante hoy en día que, por la ley de Dennard, las mejoras en eficiencia computacional para chips hacen que requieren cantidades cada vez más grandes de electricidad, lo que no es sostenible. Por lo que para resolver problemas con grandes datos es necesario utilizar cómputo en paralelo. Realizar esto presenta dos principales retos:

- La comunicación entre los nodos puede perjudicar la velocidad de los calculos en caso de una saturación de la red, por lo que hay que diseñar los algoritmos paralelos de tal forma que requieran la menor comunicación entre nodos posible.

- La sincronización entre los nodos es importante para muchos algoritmos para evitar condiciones de carrera, y por lo tanto, errores numéricos en los resultados. El cálculo completo puede verse perjudicado si una sola de las máquinas en la red se demora en el cálculo.

Para evitar estos dos problemas los autores recomiendan utilizar problemas ‘embarazosamente paralelas’ que funcionan de forma ‘distribuir y recolectar’, como lo hace el MapReduce de Hadoop. También proponen métodos de primer orden con comunicación decentralizada entre los nodos o reducida lo más posible. Finalmente, proponen la utilziación de métodos de primer orden de forma asíncrona, lo que permite que los nodos puedan realizar calculos sin sufrir los problemas de que otro nodo en la red sea muy lento, pues pueden actualizar las variables globales sin perjudicar a otros nodos.

- Discusión con proyecto

Durante la ejecución de nuestro proyecto pudimos realizar una pequeña muestra a cada uno de los tres pilares que proponen los autores. Fue interesante leer este artículo después de haber implementado descenso en gradiente estocástico en CUDA, pues vi las formas en las que aprovechamos los beneficios de cada uno de los pilares y me permite pensar en nuevas ideas otras implementaciones.