Semana 07 – Exercício Semanal

Relatório da Implementação de Algoritmos de Otimização Não-Linear

PRO6006 - Métodos de Otimização Não Linear com aplicações em aprendizado de máquina

Nathan Sampaio Santos – 8988661

O objetivo deste relatório é realizar uma análise comparativa detalhada de uma família de algoritmos clássicos de otimização: **BFGS**, **SR1**, **DFP** e **Gradiente Conjugado de Fletcher-Reeves**. A avaliação busca contrastar as características fundamentais de cada método com base em resultados empíricos obtidos a partir de um conjunto de seis funções de teste padrão: **Sphere**, **Rotated Hyper-Ellipsoid**, **Bohachevsky**, **Sum of Different Powers**, **Zakharov** e **Matyas**.

1. ANÁLISE DA IMPLEMENTAÇÃO

A análise da implementação revela um claro trade-off entre a complexidade e o uso de memória dos métodos Quasi-Newton e do método do Gradiente Conjugado.

Os métodos Quasi-Newton (BFGS, DFP, SR1) são inerentemente mais complexos e intensivos em memória. Sua característica fundamental é a necessidade de alocar e atualizar uma matriz n×n (H) que aproxima a inversa da Hessiana, inicializada tipicamente como uma matriz identidade. Este requisito impõe um custo de memória de $O(n^2)$, tornando-os inviáveis para problemas de grande escala. A lógica de atualização, como a do BFGS, H_new = np.dot(term1, np.dot(H, term2)) + term3, envolve múltiplas operações com matrizes, o que aumenta a complexidade do código.

Em contrapartida, o método do Gradiente Conjugado (Fletcher-Reeves) é drasticamente mais leve e simples. Ele não armazena nenhuma matriz, operando apenas com vetores como a direção de busca, inicializada com p = -g. Consequentemente, seu custo de memória é de apenas O(n). Sua lógica central para encontrar a próxima direção de busca resume-se a poucas linhas, como o cálculo de beta = g_dot_g_new / g_dot_g e a atualização p = -g_new + beta * p.

Em resumo, o método do Gradiente Conjugado oferece uma implementação mais simples e muito mais eficiente em termos de memória, enquanto os métodos Quasi-Newton exigem mais recursos computacionais e um código mais elaborado para gerenciar a aproximação da matriz Hessiana.

2. ANÁLISE DE DESEMPENHO

A análise do desempenho, avaliando tanto a eficiência quanto a robustez, revela o comportamento de cada algoritmo. Em termos de eficiência, medida pelo número de iterações, **BFGS** e **SR1** se destacam como os mais rápidos. O SR1 chega a ser o mais veloz em alguns testes, como na função Matyas (3 iterações), mas o BFGS demonstra uma consistência exemplar em todos os cenários. Por outro lado, **DFP** e **Fletcher-Reeves CG** são marcadamente ineficientes. O DFP chega a levar quase 5 vezes mais iterações que o BFGS na função "Sum of Different Powers" (103 contra 22). O Fletcher-Reeves CG confirma sua tendência a travar em determinados pontos, necessitando de um número excessivo de iterações (71, 100, etc.) na maioria dos testes. Apesar de matematicamente ele deveria convergir em até n interações, na prática, isso não se demonstrou real, uma vez que pode haver problemas de má condicionamento das matrizes, bem como problemas numéricos de implementação.

Ao se falar de robustez, o **BFGS** se prova o algoritmo mais robusto, resolvendo todos os problemas com sucesso. A velocidade do **SR1** vem com um custo: ele falha na função "Sum of Different Powers", onde a busca linear não consegue encontrar uma direção de descida válida. Isso expõe sua fragilidade e o torna uma escolha arriscada para problemas genéricos. Finalmente, a função **Bohachevsky** revela uma limitação fundamental de todos os métodos testados: nenhum deles encontrou o mínimo global verdadeiro, ficando presos em um mínimo local. Isso demonstra de forma conclusiva que são **otimizadores locais**, uma característica que deve sempre ser considerada ao se deparar com funções não convexas.

A análise combinada de implementação e desempenho permite as seguintes recomendações:

1. Para Uso Geral e Confiabilidade (Problemas de Médio Porte): o método BFGS oferece o melhor equilíbrio entre robustez, eficiência e simplicidade de uso, apesar de seu maior custo de memória.
2. Para Problemas de Grande Escala (Memória é o Fator Crítico): O método Gradiente Conjugado deve ser o escolhido.