Final Exam

Subhrangshu Bit

July 21, 2020

1 Problem

Consider the problem -

$$min \;\; f(x) = 2x_1^2 + 3x_2^2 - 3x_1x_2 + 2x_1 - 4x_2$$

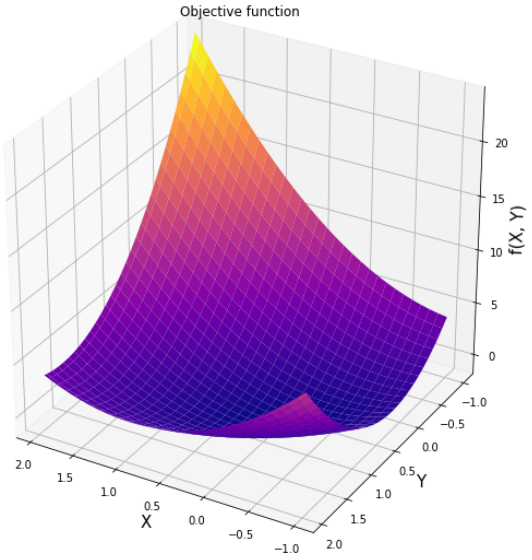
Starting from the initial point $x_1 = 0, x_2 = 0$ solve the problem using two methods -

- Davidon-Fletcher-Powell (DFP) Method
- Fletcher-Reeves (FR) Conjugate Gradient method

The first method corresponds to a Quasi-Newton method which is to be implemented with the initial approximation of the inverse of the hessian as identity : $D_1 = I_2$.
Further show that the directions generated by the two methods at every iteration are identical and explain the reason behind this.

2 Solution

2.1 Visualizing the function



The above plot along with the form of the objective function $f(x)$ makes it very clear that it is convex and moreover it is quadratic in nature. Before implementing the two methods mentioned above we compute the expressions for the gradient and hessian of the objective function.

$$\nabla f(x) = \begin{bmatrix} 4x_1 - 3x_2 + 2 \\ 6x_2 - 3x_1 - 4 \end{bmatrix}$$

$$H = \nabla^2 f(x) = \begin{bmatrix} 4 & -3 \\ -3 & 6 \end{bmatrix}$$

Unconstrained minimzation of a function involves two parts -

- Line Search
- Direction Search

In this problem we have used the backtracking line search method based on the Armijo's rule to obtain an optimum step length given a direction.

2.2 Davidon-Fletcher-Powell Method

Given in the problem the initial approximation of the inverse of the hessian is $D_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Since the objective function is quadratic the DFP algorithm restarts after every 2 iterations. Furthermore, D_3 obtained at the end of the iteration is precisely the inverse of the Hessian matrix H.
If we can show that the directions obtained from the DFP method are $H - conjugate$ then a part of our problem statement is resolved. After implementing the DFP update method in python we have the obtained the following iterations -

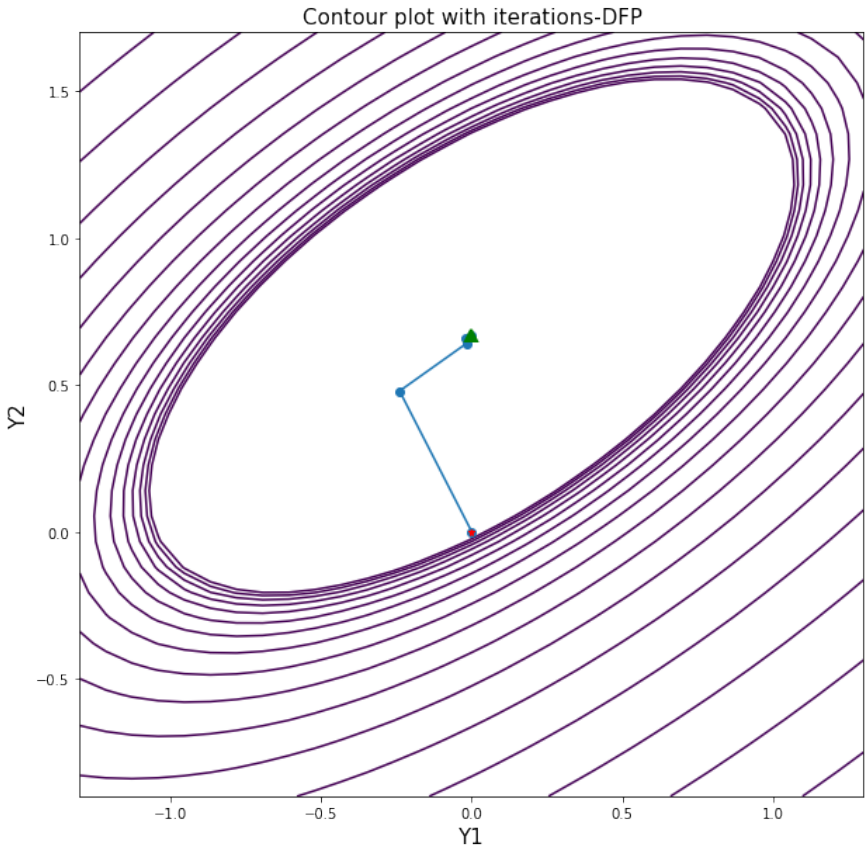
```
      Y1        Y2
0   0.000000  0.000000
1  -0.239361  0.478721
2  -0.017156  0.640839
3  -0.018666  0.658484
4  -0.000793  0.667150
```

```
                                    Directions
0                                   [-2.0, 4.0]
1      [0.4508990635339303, 0.32897110370136995]
2    [-0.00885860110306158, 0.10349590165459244]
3    [0.03626748089654302, 0.017585432875661566]
4  [0.004623322800861196, -0.005280462049949364]
```

```
                                              D
0                          [[1.0, 0.0], [0.0, 1.0]]
1  [[0.7173076923076922, 0.41153846153846146], [0...
2                          [[1.0, 0.0], [0.0, 1.0]]
3  [[0.7791058086976662, 0.4025184701081798], [0...
4                          [[1.0, 0.0], [0.0, 1.0]]
```

Observe that after every two iterations the approximation of the inverse of the hessian D_j is reverted to I_2.

Plotting the iterations on the contours of the objective function the following diagram is obtained-



Observations:
Note that the method precisely reaches the optimal point [0, 0.667] in just two iterations. This verifies the result from conjugate gradient that in quadratic objective functions the algorithm is supposed to terminate in just n (here 2) iterations. The DFP method also mimics this result in the above plot thereby approving to our problem that the directions are identical to those of the conjugate gradient method.

2.3 Fletcher-Reeves Conjugate Gradient Method

The conjugate gradient method is primarily based on the n H-conjugate directions where H is the Hessian matrix of an objective function. Here since the objective function is quadratic we have a fixed Hessian matrix H, as mentioned above, which is symmetric and positive definite. We then obtain the conjugate gradient directions from H and update the direction at every iteration according to the rule -

$$d_{j+1} = -\nabla f(y_{j+1}) + \alpha_j d_j$$

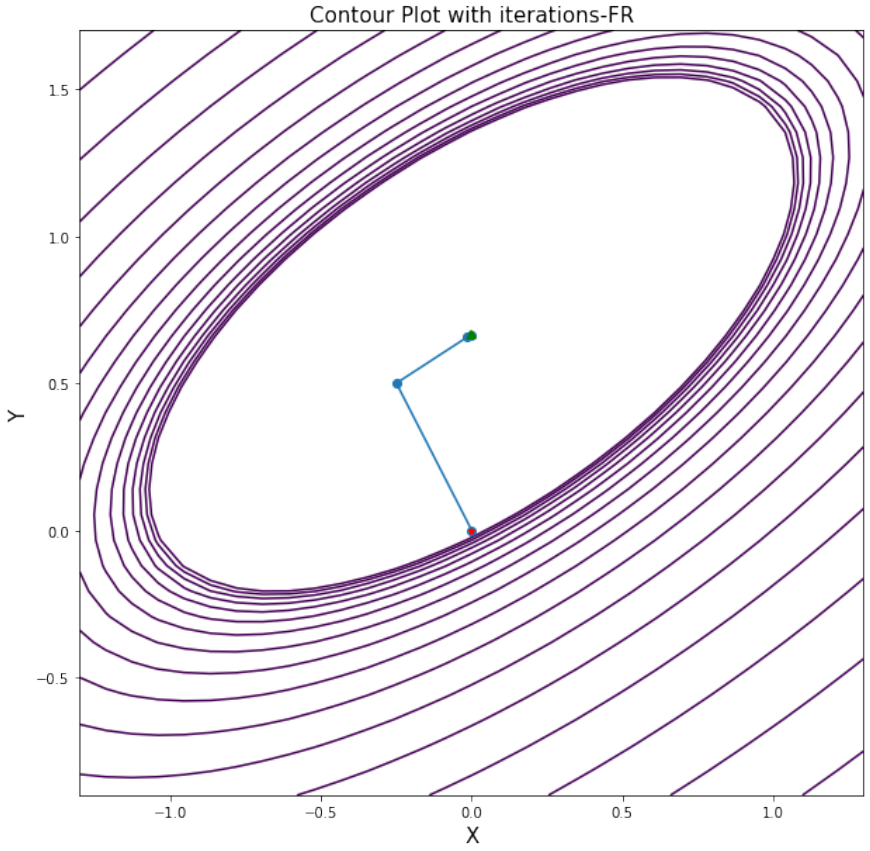
Based on the value of α_j there are many variants of the CG algorithm and one such is the Fletcher-Reeves for which -

$$\alpha_j = \frac{\| \nabla f(y_{j+1}) \|^2}{\| \nabla f(y_j) \|^2}$$

Using python if we implement the above method for our problem the iterations are tabulated below -

	X	Y	f(X)	Norm	Directions
0	0.000000	0.000000	0.000000	4.472136	[-2, 4]
1	-0.250000	0.500000	-1.250000	0.559017	[0.46875, 0.3125]
2	-0.015625	0.656250	-1.333008	0.034939	[0.03125, 0.015625]
3	0.000000	0.664062	-1.333313	0.017469	STOP

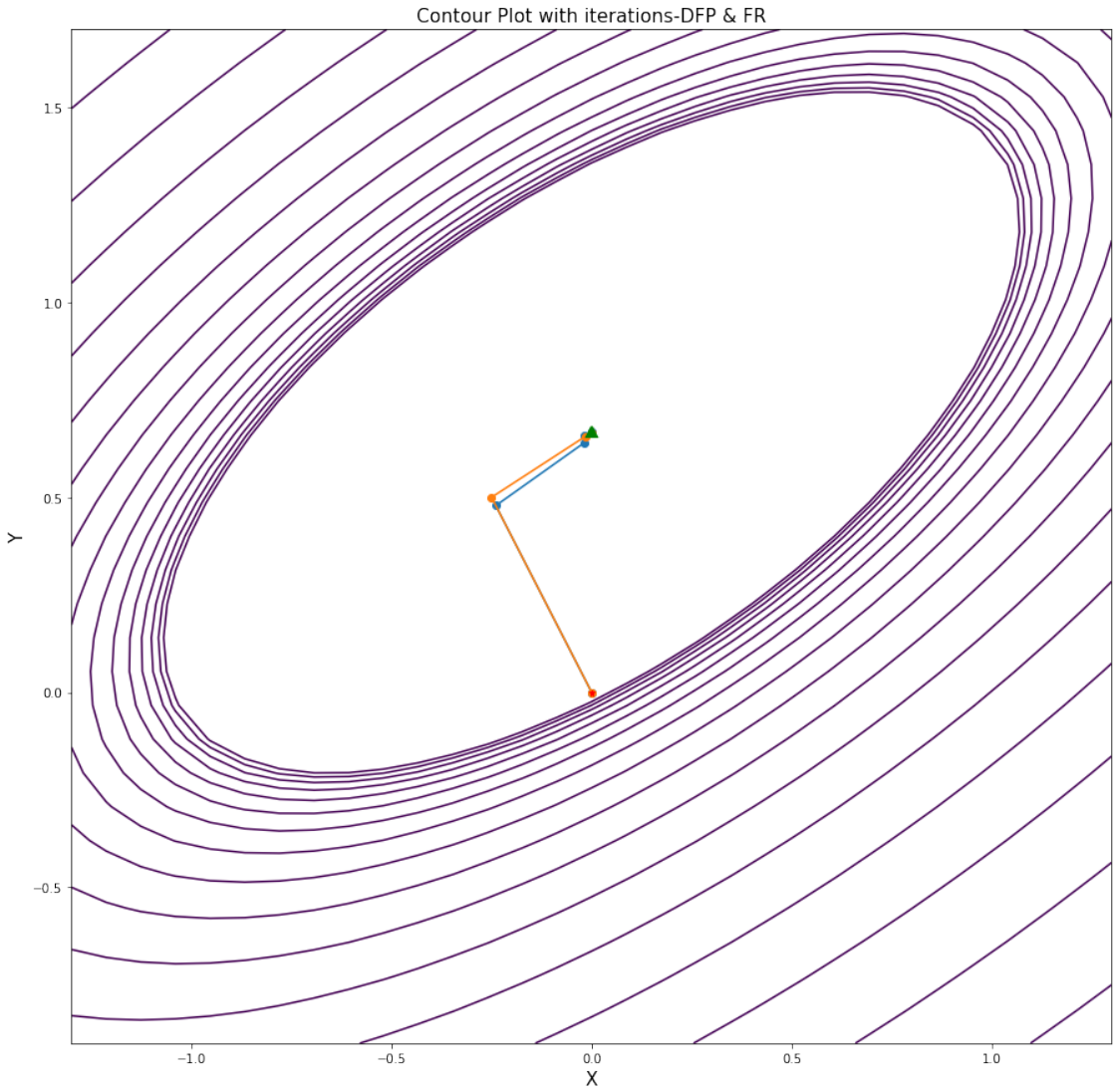
Plotting the iterations on the contours of the objective function-



Observations: The algorithm almost reaches the optimal point in just two iterations which verifies the result that the algorithm converges in n (here 2) iterations in case of quadratic functions.
Now in order to compare the directions obtained by the two methods - DFP and FR, we compute the norm of the difference of the two and have obtained the following result -

```
0.0
0.02428895201232323
0.09659190048878616
```

To get a visualization of the similarity of the directions obtained in the two methods we plot the iterations of both the methods onto one contour plot as provided in the following figure -



Observations:

- The directions are almost identical.
- Both the algorithms reach the optimal point precisely in just 2 iterations.
- There is a slight variation in the direction obtained starting from the second point. This is due to the step length computed iteratively using the backtracking line search algorithm. The step length obtained in each case is not the optimal, it includes some computational errors. Moreover the matrix computations also involve some errors. If these errors can be minimized by exact computations the directions will be exactly same in both the methods.

3 Explanation

After getting a hold of the practical view of similarity it requires an explanation of the results obtained. As mentioned earlier we now prove that the directions obtained from the DFP method are $H - conjugate$.

Let H be ab $n \times n$ symmetric positive definite matrix, and consider the problem to minimize -

$$f(x) = c^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T H\mathbf{x}$$

subject to $\mathbf{x} \in \mathbb{R}^n$. We solve the problem by the DFP method starting with an initial point $\mathbf{y_1}$ and a symmetric positive definite matrix D_1 as the approximation of the inverse of the Hessian. In particular for $j = 1, 2, \cdots, n$, let λ_j be an optimal solution to the problem to minimize $f(y_j + \lambda d_j)$ subject to $\lambda > 0$ and $y_{j+1} = y_j + \lambda_j d_j$, where $d_j = -D_j \nabla f(y_j)$ and D_j is determined by -

$$D_{j+1} = D_j + \frac{p_j p_j^T}{p_j^T q_j} - \frac{D_j q_j q_j^T D_j}{q_j^T D_j q_j}$$

$$where \;\; p_j = \lambda_j d_j = y_{j+1} - y_j$$
$$q_j = \nabla f(y_{j+1}) - \nabla f(y_j)$$

If $\nabla f(y_j) \neq 0$ for each j, then the directions $d_1, d_2, \cdots, d_n$ are H-conjugate.

We need to show that for any j with $1 \leq j \leq n$ -

$$d_i^T H d_j = 0 \;\; for \;\; i \neq k; \;\; k \leq j$$

For j = 1 this is obvious since we have only one direction and thus trivially any d_i where $i < j$ may be considered as zero thereby satisfying the above. To show that this holds for j+1 we use the method of induction. Say the above is true for $j \leq (n-1)$. Now for $j > 1$ first consider the expression -

$$Hp_k = H(\lambda_k d_k) = H(y_{k+1} - y_k) = Hy_{k+1} - Hy_k = \nabla f(y_{k+1} - \nabla f(y_k) = q_k$$

Then $Hp_1 = q_1$. Therefore,

$$D_2 H p_1 = (D_1 + \frac{p_1 p_1^T}{p_1^T q_1} - \frac{D_1 q_1 q_1^T D_1}{q_1^T D_1 q_1}) q_1$$

$$\implies D_2 H p_1 = D_1 q_1 - p_1 + D_1 q_1 = p_1$$

The terms can be cancelled out since they are scalars. Since $p_1 = \lambda_1 d_1$ we can equivalently write $D_2 H d_1 = d_1$.
By the induction hypothesis this holds for $i \leq j$ -

$$D_{j+1} H d_i = d_i \tag{1}$$

Since we are interested in a quadratic problem so $f(y_j + \lambda d_j)$ achieves a minimum at λ_j only if the gradient with respect to λ -

$$\nabla f(y_j + \lambda d_j)^T d_j = 0$$
$$\implies \nabla f(y_{j+1})^T d_j = 0$$

So $d_i^T \nabla f(y_{i+1}) = 0$. Now we are interested in the term $d_i^T \nabla f(y_{j+1})$ where $i \leq j$. We can write -

$$\nabla f(y_{j+1}) = c + Hy_{j+1} = c + H(y_{i+1} + \sum_{k=i+1}^{j} \lambda_k d_k)$$

$$= c + Hy_{i+1} + \sum_{k=i+1}^{j} \lambda_k H d_k$$

Since we have by induction hypothesis assumed that for $j \leq (n-1)$ the directions are conjugate gradient so -

$$d_i^T \nabla f(y_{j+1}) = d_i \nabla f(y_{i+1} + \sum_{k=i+1}^{j} \lambda_k d_i H d_k$$
$$\implies d_i^T \nabla f(y_{j+1}) = 0$$

Thus for any $i \leq j$ we have $d_i^T \nabla f(y_{j+1}) = 0$. Then -

$$0 = d_i^T \nabla f(y_{j+1}) = (D_{j+1} H d_i)^T \nabla f(y_{j+1}) \qquad [using \ \ (1)]$$
$$= d_i^T H(D_{j+1} \nabla f(y_{j+1})) = -d_i^T H d_{j+1}$$

This completes our induction proof that $d_i^T H d_k = 0$ for all $i \neq k$; $i, k \leq j$ and $1 \leq j \leq n$. Hence the directions obtained from the DFP method are H-conjugate which proves a part of our result that the directions obtained from CG method using Fletcher-Reeves are identical to those from DFP method.

Before moving to the similarity of the directions obtained from the two methods a restart strategy used in both the methods and it's justification is of utmost importance.

Restart Strategy:

In the explanation above we have seen the result -

$$D_{j+1} H d_i = d_i$$

where $i \leq j$

This implies that d_i is an eigenvector of $D_{j+1}H$ corresponding to the eigenvalue 1. Hence, at each step of the method, the revised approximation of the inverse of the Hessian accumulates one additional linearly independent eigenvector, with a unit eigenvalue for the product $D_{j+1}H$, until $D_{n+1}H$ finally has all its n eigenvalues equal to 1, giving -

$$D_{n+1} H P = P$$

where P is the matrix of eigenvectors of $D_{n+1}H$. Hence -

$$D_{n+1} H = I_n$$

$$\implies D_{n+1} = H^{-1}$$

Therefore in quadratic case we will have - $D_3 = H^{-1}$.
Similarly the restart strategy is also implemented in the conjugate gradient method. Essentially in a quadratic objective function the method is expected to reach the optimal in just 2 steps and the need for restart in not applicable, however due to inexact line search methods used and floating point errors this becomes helpful.

So, now the question arises in a quadratic case how are the directions obtained from these two methods related? The first direction for CG method is -

$$d_1^{CG} = -\nabla f(x_1)$$

whereas that of DFP method with the starting approximation $D_1 = I_2$ -

$$d_1 = -D_1 \nabla f(x_1) = -\nabla f(x_1)$$

Thus $d_1^{CG} = d_1$. Now moving on to the next iteration -
For CG method -

$$d_2^{CG} = -\nabla f(x_2) + \beta_j d_1$$

where β_j is the FR multiplier.
For DFP method -

$$d_2 = -D_2 \nabla f(x_2) = -(I_2 + \frac{p_1 p_1^T}{p_1^T q_1} - \frac{D_1 q_1 q_1^T D_1}{q_1^T D_1 q_1}) \nabla f(x_2)$$

Now as we proved earlier the DFP directions are also $H - conjugate$ we must have -

$$d_2^{CG} H d_1^{CG} = d_2 H d_1 = 0$$

Here since our objective function is quadratic the space generated by the directions will be of dimension 2 and the directions will be $H - orthogonal$ i.e, orthogonal in that space (This space generated by conjugate gradient directions is known as Krylov space κ).

$$\implies d_1^{CG}, d_2^{CG} \in \kappa_2(d_1^{CG}, H)$$

and also

$$\implies d_1, d_2 \in \kappa_2(d_1, H)$$

In this space d_1^{CG} is orthogonal to d_2^{CG} and also $d_1 = d_1^{CG}$ so d_2 is orthogonal to d_1 and d_1^{CG} implying -

$$d_2^{CG} = \delta d_2$$

This delta is a scalar > 0 (since we know the directions are descent) and thus $d_2^{CG} \parallel d_2$. The parallel directions result due to varying step lengths resulting from inexact line searches, if exact line searches are used without any floating point errors then $\delta = 1$ and then the identical directions - $d_2^{CG} = d_2$.

Now note that since we had the initial approximation of the inverse of the Hessian H to be I_2 lead to $d_1^{CG} = d_1$. In case of any other initial value it wouldn't have been the same. However if we have $D_1 = I_n$ the directions from the two methods can be further extended for n to be parallel using the

method of induction as proved in [2]. The connection of FR with DFP can be further explained for $D_1 = I$ from the following simplification.
We know the FR direction update rule -

$$d_k^{CG} = -\nabla f(x_k) + \beta_k d_{k-1}$$

$$= -\nabla f(x_k) + \frac{\nabla f(x_k)^T \nabla f(x_k)}{\nabla f(x_{k-1}^T \nabla f(x_{k-1})} d_{k-1}$$

$$= -\nabla f(x_k) + \frac{d_{k-1} \nabla f(x_k)^T}{\nabla f(x_{k-1}^T \nabla f(x_{k-1})} \nabla f(x_k)$$

$$= -(I + \frac{d_{k-1} \nabla f(x_k)^T}{\nabla f(x_{k-1}^T \nabla f(x_{k-1})}) \nabla f(x_k)$$

Thus CG method using FR can be viewed as a Quasi-Newton method with the initial approximation being I which is updated according to the above rule.

This completes our explanation and justification of the identical search directions obtained from the CG and DFP methods.

4 Applications

Optimization is one of the core components of any Machine Learning algorithm. The essence of most machine learning models is to build an optimization problem and learn the optimal parameters defining the underlying fuunction from the given data. In today's era of Big Data the efficiency and effectiveness of the numerical optimization algorithms dramatically influence the popularization and application of machine learning models.
From the perspective of gradient information optimization algorithms mainly constitutes :

- First order
- High-order
- derivative-free

The first order algorithms such as stochastic gradient descent, ADAM, AdaGrad etc. are very popular but there are many limitations. The high-order algorithms makes use of the curvature information to address those limitations like when we have highly non-linear and ill-conditioned objective functions, which are very common in deep neural networks and also in deep reinforcement learning models. However including curvature condition exactly incurs huge cost for large data, which can be addressed using the Quasi-Newton methods and Conjugate Gradient methods.
Quasi-Newton methods, specially L-BFGS initially developed from DFP method, is a widely used algorithm in deep neural networks to solve the problem of empirical risk minimization. In recent years a modification of the L-BFGS was developed which uses the idea of trust-regions as an alternative to the gradient descent. The motivation behind trust regions is to find the search direction, d_k, in a region within which they trust the accuracy of the quadratic model of the objective function. These methods not only have the benefit of being independent from the fine-tuning of hyper parameters, but they may improve upon the training performance and the

convergence robustness of the line-search methods. A further reading can be referred in [5]
The reinforcement learning (RL) problem, – a class of machine learning – is that of learning through interaction with an environment and the learner is the agent. This mainly involves finding an optimal policy π^* to get the maximum return. This is a class of solution methods to Markov Decision Processes where the agent has no prior information about the environment models i.e, the state transition probabilities and reward function. Thus based on the experience it tries to learn the underlying parameters which basically can be formulated as an empirical risk minimization which has been found to perform very well using Quasi-Newton methods. This can referred to in [5]
In present world of recommendations, reviews, tv shows machine learning models form the base of any major project. Collaborative Filtering (CF) are one such class of models which is a method of automatic predictions (filtering) about the interests of a user by collecting preferences from many users (collaborating). In these algorithms arises the problem of solving weighted ridge regression (WRR) where the observations are weighted. The CG method is a state-of-the-art approach for the approximate solution WRR problems and is also used in one the most popular streaming service today - Netflix. Further explanation can be referred in [4]
The Conjugate Gradient method is also very popular in many areas. In solving fluid mechanics problem CG method plays an important role to solve the Navier-Stokes equations governing viscous incompressible fluids, which is basically a discrete system of non-linear equations. A wholesome description is in [6]

References

[1] Numerical Optimization: Penn State Math 555 Lecture Notes; Christopher Griffin
[2] ON THE CONNECTION BETWEEN THE CONJUGATE GRADIENT METHOD AND QUASI-NEWTON METHODS ON QUADRATIC PROBLEMS Anders FORSGREN Tove ODLAND
[3] Nonlinear Programming Theory and Algorithms; Mokhtar S. Bazaraa, Hanif D. Sherali, C.M. Shetty
[4] Applications of the conjugate gradient method for implicit feedback collaborative filtering ;Gábor Takács, István Pilászy, Domonkos Tikk
[5] Quasi-Newton Optimization Methods For Deep Learning Applications; Jacob Rafati, Roummel F. Marcia
[6] THE APPLICATION OF QUASI-NEWTON METHODS IN FLUID MECHANICS M. S. ENGELMAN, G. STRANG AND K.J. BATHE