AMSI
VACATION
RESEARCH
SCHOLARSHIPS
2017-2018

Numerical optimisation methods for big data analytics

Drew Mitchell

Supervised by Professor Hans De Sterck

Monash University

Vacation Research Scholarships are funded jointly by the Department of Education and Training and the Australian Mathematical Sciences Institute.

Australian Government
Department of Education and Training

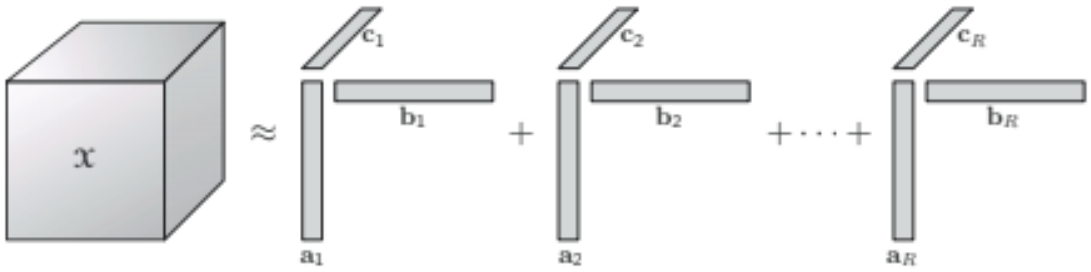
AMSI AUSTRALIAN MATHEMATICAL SCIENCES INSTITUTE

Abstract

With the ubiquitous use of technology in the modern era the rise of massive data sets with complex structure is not only inevitable but largely already prevalent. In this report we investigate and propose an adaption and extension of Nesterov's method for convex problems into a method capable of accelerating alternating-type optimisation methods for non-convex problems such as canonical tensor decomposition. This method is determined to be comparable with much more difficult to implement accelerations such as the L-BFGS method. However, unpredictable behaviour in convergence and a reliance on user-input-constants detracts from the versatility of the algorithm.

1 Introduction

In this project we looked at various optimization methods for the problem of CP tensor decomposition. This problem requires the computation of a rank-R canonical tensor minimising the Frobenius distance to a given tensor. Figure 1 shows this decomposition expressed pictorially, Kolda & Bader (2009).



(a) CP tensor decomposition of the third-order tensor $\mathcal{X}$ into a sum of the outer product of column vectors; a_i, b_i, c_i. Image from Kolda & Bader (2009).

CP tensor decomposition may be formalised into the optimisation problem given by:

$$f(\mathcal{A}_R) = \frac{1}{2}||\mathcal{A}_R - \mathcal{X}||_F^2, \tag{1}$$

in which we are trying to find the rank-R tensor $\mathcal{A}_R$ to minimise $f(\mathcal{A}_R)$. Here $||.||_F$ represents the Frobenius norm. At any local minima of this problem the following first-order optimality

equation must hold,

$$\nabla f(\mathcal{A}_R) = g(\mathcal{A}_R) = 0. \tag{2}$$

This equation allows us to monitor the convergence of our methods by inspecting the relative gradient norm.

1.1 Alternating Least-Squares

The method of *alternating least-squares* (ALS) is often used for decomposition style problems. The method works by minimising one factor of the current decomposition while holding the other factors constant and then alternating through the factors. When applied to the above problem, it works by holding constant all of the b_i and c_i vectors while minimising the a_i vectors. After this the a_i and c_i vectors would be held constant while minimising the b_i vectors. Finally, the c_i vectors would be minimised. Unfortunately the ALS method converges slowly for difficult problems. As such we have the incentive to look for faster algorithms or ways to accelerate the ALS method itself, Sterck (2012).

1.2 Nesterov's method

The Nesterov method, created by Yurii Nesterov was developed to have quadratic convergence for convex problems, Nesterov (2004). This method is easy to implement and can be described simply in the following two step process,

$$y_{t+1} = x_t - \alpha_t \nabla f(x_t), \tag{3}$$

$$x_{t+1} = y_{t+1} + \beta_t (y_{t+1} - y_t). \tag{4}$$

Here $\alpha_t = \frac{1}{L}$ with L the Lipschitz constant of the function f. The sequence β_t is defined in terms of the sequence of λ_t terms that give the method its characteristic convergence. With

$$\lambda_0 = 0, \lambda_t = \frac{1 + \sqrt{1 + 4\lambda_{t-1}^2}}{2} \text{ and } \beta_t = \frac{\lambda_t - 1}{\lambda_{t+1}}.$$

This method is renowned as an excellent method for solving convex problems and is a significant improvement on basic gradient-descent methods. The speed of the method is greatly improved by the inclusion of the $\beta_t(y_{t+1} - y_t)$ term which is often called 'momentum', O'donoghue

& Candès (2015). This name is apt: when significant progress is being made the term is large and large steps are taken, and when little progress is made the steps become smaller.

Since the problem of CP tensor decomposition is non-convex, however, we lose the provable convergence speed and the ability to use the fixed sequence of β_t values. We note that using these values is possible but convergence is rarely possible. The appearance of multiple local minima means that we are not guaranteed convergence using the Nesterov method and as such adaptation is required.

2 Acceleration of ALS using Nesterov's Method

It is possible to accelerate the ALS algorithm using other methods and this acceleration can in turn provide robustness and increased convergence speed in solving non-convex problems, Sterck (2012). We introduce the accelerated Nesterov method given by the simple one step procedure:

$$y_{t+1} = ALS(y_t + \beta_{t-1}(y_t - y_{t-1})) \tag{5}$$

Our new iterate is given by the old iterate onto which we add a step of some length β_t in the direction $y_t - y_{t-1}$, and we then take an ALS step using the result to give us the new iterate. Here we have replaced the steepest descent gradient step with a single step of the ALS algorithm. This guarantees that the next iterate will lower the gradient. Since, in the non-convex setting, there is no reason to intuitively use the β_t values from the fixed sequence above, the value for β_t is given via a line search. This line search returns the β_t that minimises the gradient along a given search direction.

This method has been previously utilised for an ALS accelerated NGMRES method, Sterck (2012). Using the line search and ALS acceleration, this version of Nesterov's method is more numerically robust than the basic Nesterov method. Unfortunately, using the line search to determine the best step length is costly as we must evaluate the function and gradient values at various points along the search direction to determine the best β_t. As such, we consider that replacing the line search with some fixed step length may improve the convergence speed if there is some way to keep the algorithm from becoming erratic.

2.1 Accelerated Nesterov with restart

Restarting has been implemented in other numerical methods before with positive benefits, O'donoghue & Candès (2015),Goldstein et al. (2014). By inspecting the sequence of β_t values from the original Nesterov method and observing that the sequence rapidly converges to one, and under the assumption that close to a local minima the problem behaves in a convex manner, we adopted one as our step size. To keep the method from diverging we introduce the concept of restarting. In this case a restart refers to completing only an ALS step after a fixed number of accelerated steps as described in equation (5). The method is outlined below as algorithm 1.

Algorithm 1 Accelerated Nesterov with fixed restart

procedure NESTEROVFIXEDRESTART($y0, maxit, tol, rest$)
▷ y0 - initial guess
▷ maxit - maximum number of iterations
▷ tol - relative gradient norm tolerance
▷ rest - number of iterations until restart

$\beta \leftarrow 1$
$y_{old} \leftarrow y0$
$y_{new} \leftarrow y0$
while $iterations < maxit \;\&\&\; ||\nabla f(y_{new})|| > tol$ **do**
$d \leftarrow y_{new} - y_{old}$
$y_{old} \leftarrow y_{new}$
if $mod(iterations, rest) \; = 0$ **then**
$y_{new} = ALS(y_{new} + \beta * d)$
else
$y_{new} = ALS(y_{new})$
end

In this algorithm we check to see if the 'iterations' number is a multiple of the input variable 'rest'. If this condition is true then we complete only an ALS step without the accelerated

Nesterov step. Since the single ALS step must decrease the gradient the presence of this restarting should keep the gradient from diverging. Unfortunately simply including this single ALS step does not control the growth of the gradient enough for the Accelerated Nesterov with fixed restart algorithm to converge quickly. Figure 2 shows the possible divergence using these fixed restarts for different 'rest' values.

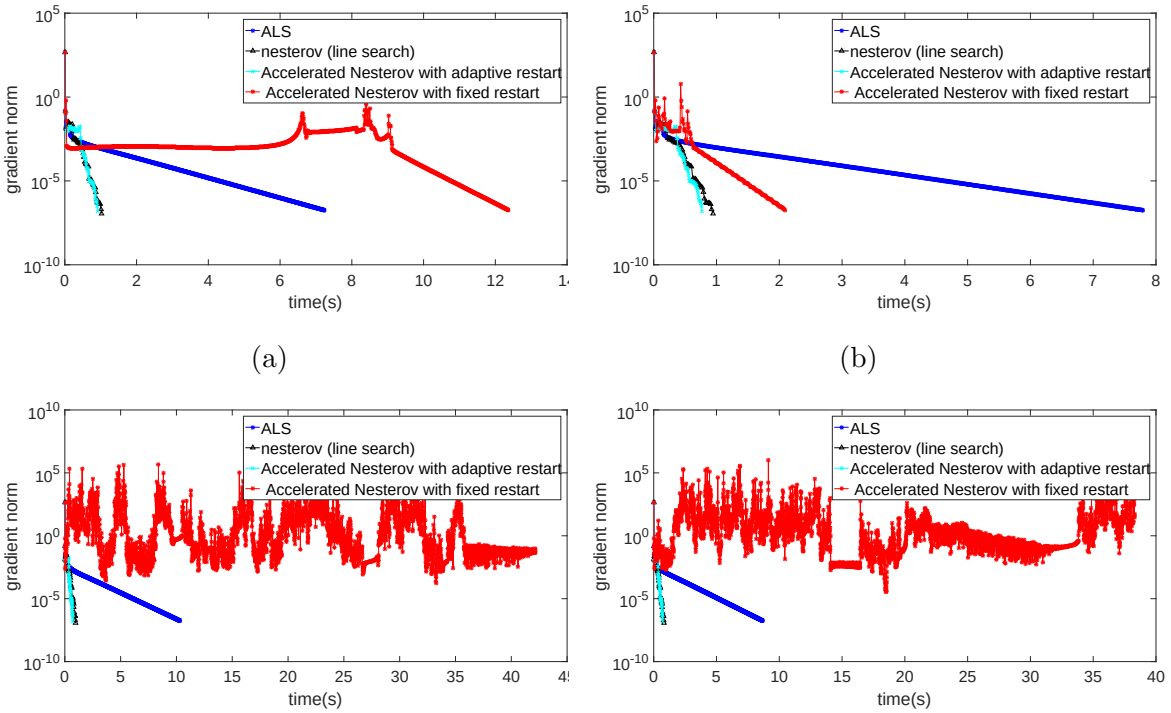


Figure 2: Typical convergence plots for ALS, Nesterov with Line search, Accelerated Nesterov with adaptive restart, Accelerated Nesterov with fixed restart (algorithm 1). a) ALS step every 5th Nesterov step, b) ALS every 10th Nesterov step, c) ALS every 20th Nesterov step, d) ALS every 40th Nesterov step.

The above figure shows typical convergence plots for the methods. Included in this plot is the 'Nesterov Restart' (light blue) convergence data which corresponds to the method described in algorithm 2. The other methods included in this plot are ALS (blue), 'Nesterov with line search' described by equation (5) (black) and the Accelerated Nesterov with fixed restart (red). We can see that although in some cases, with rest as 5 and 10, the method is able to converge, for larger rest values we have a complete lack of convergence. Furthermore, in no plot is this

method comparable to the time for convergence displayed by the Nesterov with line search algorithm. Clearly an alternating approach does not work to keep the Nesterov acceleration method converging. So we need to introduce an adaptive way to restart the algorithm when the gradient grows too rapidly.

2.2 Accelerated Nesterov with adaptive restart

Intuitively it makes sense to utilise the acceleration of ALS with Nesterov when the step is in the right direction and to otherwise simply take a safe step with just ALS. This type of adaptive restarting has been used before, Goldstein et al. (2014) and we implement this with the purpose of only using an ALS step to control gradient growth. The following method describes adaptively resetting the accelerated Nesterov method to limit the growth of the gradient at any new step.

Our means to control the gradient consists of introducing the constant η. Using this we may constantly monitor the growth of the gradient in each new iteration and if the limit set by $\eta||\nabla f(y_{old})||$ is broken we may reset the algorithm by throwing away the current iteration (line 17). Within the next iteration of the method we only take a step using the ALS method such that we can avoid the growth in the gradient. We also allow for a larger growth during the following iterations by increasing η, which then decreases over the next few iterations (line 21). We include this increase in the η value to limit constant restarting which results in simply using the ALS method as each Nesterov step is forgotten.

Algorithm 2 Accelerated Nesterov with adaptive restart

procedure
  $\eta \leftarrow 1.15$
  $\beta \leftarrow 1$
  $restart \leftarrow false$
  $y_{old} \leftarrow y0$
  $y_{new} \leftarrow y0$
  while $iterations < maxit \ \&\& \ ||\nabla f(y_{new})|| > tol$ **do**
    $d \leftarrow y_{new} - y_{old}$
    $y_{old} \leftarrow y_{new}$
    if $restart$ **then**
      $y_{new} = ALS(y_{old})$
      $restart = false$
      $\eta = 1.25$
    else
      $y_{new} = ALS(y_{new} + \beta * d)$
      if $||\nabla f(y_{new})|| \geq ||\nabla f(y_{old})\eta||$ **then**
        $y_{new} \leftarrow y_{old}$
        $restart \leftarrow true$
      end
      if $\eta \neq 1.15$ **then**
        $\eta \leftarrow \eta - 0.02$
      end
    end
  end

3 Numerical Testing

In order to determine the effectiveness of the ALS accelerated Nesterov with restart algorithm the following trials were undertaken for the problem of CP tensor decomposition. Table 1 shows the parameters used in each trail with; 's' as the size of the tensor, 'c' as the collinearity of the tensor, 'R' as the rank of the tensor and 'l_1' and 'l_2' as the noise factors added into the tensor. These trials are broken up into easy problems and hard problems based on the collinearity of the tensor. The test problem and trials have been used to test CP tensor decomposition for similar algorithms, Sterck (2012).

Table 1: List of trials, Sterck (2012)

Trial	s	c	R	l_1	l_2
1	20	0.5	3	1	1
2	20	0.5	5	10	5
3	50	0.5	3	1	1
4	50	0.5	5	10	5
5	100	0.5	3	1	1
6	100	0.5	5	10	5
7	20	0.9	3	0	0
8	20	0.9	5	1	1
9	50	0.9	3	0	0
10	50	0.9	5	1	1
11	100	0.9	3	0	0
12	100	0.9	5	1	1

Each trial was run ten times with random initial guesses, seeded in matlab. Since this problem is non-convex with multiple local minima any method tested had no guarantee of convergence to a the global minimum. For our testing, we did not consider the minima that the tests converged to only that the relative gradient norm was reduced to the desired tolerance. Three different levels of accuracy (desired tolerance) were used; 10^{-5}, 10^{-9} and 10^{-10}.

3.1 Other methods

In order to properly test the speed of the Nesterov with restart algorithm we compared it to other known methods; ALS, and ALS accelerated by N-GMRES, NCG and L-BFGS. Each of the other methods except for the L-BFGS method, (for which code was supplied by Hans De Sterck, Monash University), have been tested previously, Sterck (2012). We did briefly look at implementing restarting in these other methods, however, the length of the project did not permit sufficient time for this and as such these results are omitted.

3.2 Accelerated Nesterov with adaptive restart comparison

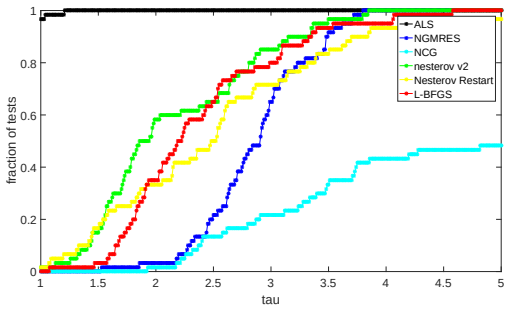
Using the above trial parameters, tests of the algorithms were conducted resulting in the τ plots in figure 3. These τ plots provide a simple visual means for comparing the speed of each method. In these plots the x-axis corresponds to τ which represents a multiplication factor relative to the fastest time. For example, for τ value of two the figures plot the function of runs with runtime within twice the fastest runtime for each test. The y-axis thus corresponds to the fraction of tests within a certain factor, τ, of the fastest time. For example in plot a) of figure 3 we can see that in 60% of the tests 'nesterov v2', which is our Nesterov with line search algorithm, is within a factor of two of the fastest time.

These plots have been broken up with respect to difficulty and tolerance of the problems. Within the easy problems, for each test in each trial the time to convergence of each algorithm was compared with the fastest time giving the τ values. The fraction of tests within certain τ ranges was then determined to create the plots. We note that for trial 12, $tol = 10^{-10}$, and using the 3th seed the program would not complete on the Nesterov with restart method and as such the 11th seed was substituted. The NCG method is not included in the trials for which we have a tolerance of $tol = 10^{-10}$ as it cannot reach a gradient this small.

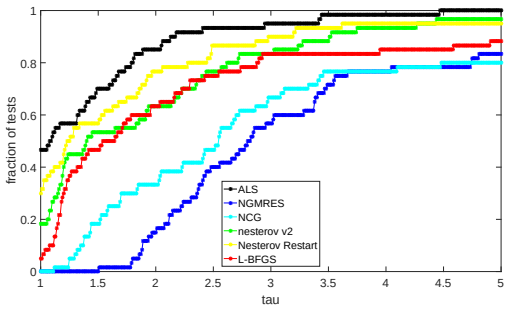
Our initial observation from these plots is that the ALS algorithm is by far the best for easy problems and low tolerances. Even for high tolerances, $tol = 10^{-9}$, we see that ALS is still the fastest algorithm in 80% of our tests. For these easy problems we can see that Nesterov with line search algorithm is competitive with more complicated methods like the L-BFGS method.

We also observe that the Nesterov with adaptive restart (yellow) seems to be slower than our line search Nesterov algorithm for these easy problems.

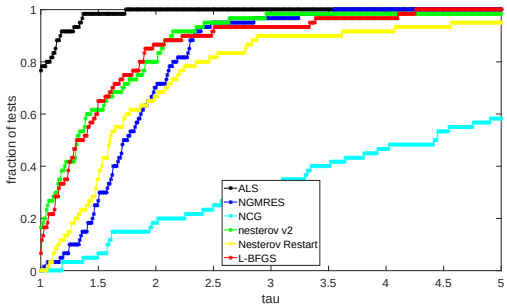
The hard problems however highlight the competitiveness of our Nesterov with restart algorithm. Even for the low tolerance trials, $tol = 10^{-5}$ we see that this method is the second best, and is actually the fastest method 30% of the time. The Nesterov with line search and L-BFGS methods are both competitive in this low tolerance hard set of tests as well. For the high tolerance, $tol = 10^{-9}$ and $tol = 10^{-10}$ we see that L-BFGS becomes the clear fastest method, beating all other methods 60% of the time. In these tests, however, we also see our Nesterov with adaptive restart method is a competitor being the fastest method in the remaining 40% of trials.



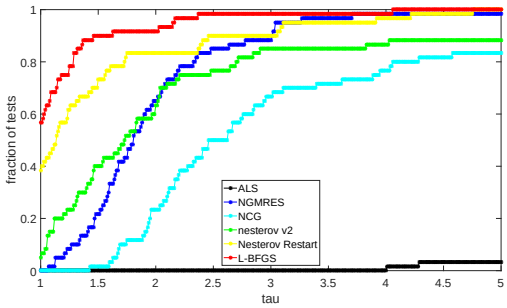
(a) Tau plot of easy problems with 10^{-5} gradient norm tolerance



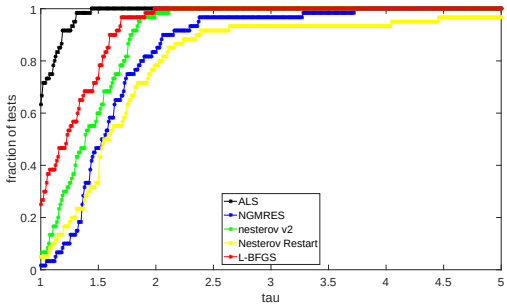
(b) Tau plot of hard problems with 10^{-5} gradient norm tolerance



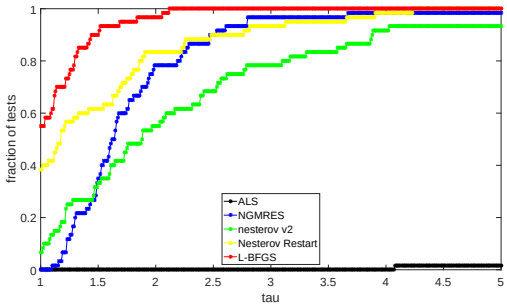
(c) Tau plot of easy problems with 10^{-9} gradient norm tolerance



(d) Tau plot of hard problems with 10^{-9} gradient norm tolerance



(e) Tau plot of easy problems with 10^{-10} gradient norm tolerance



(f) Tau plot of hard problems with 10^{-10} gradient norm tolerance

Figure 3: Tau plots for easy and hard problems over a range of gradient norm tolerances: $tol = 10^{-5}, 10^{-9}, 10^{-10}$

4 Conclusion

We have developed a new extension for Nesterov's method to accelerate simple optimisation methods for non-linear, non-convex problems. The method is an acceleration of the alternating

least squares method and utilises adaptive restarting to control unpredictable gradient growth. This new algorithm is capable of competing with more complicated algorithms for difficult problems in the case where a high level of accuracy is required. Even in the case of low accuracy for difficult problems, this new method is more effective than more complicated methods, however, ALS remains the fastest. For easy problems ALS is by far the best method to use even at high accuracy. There are drawbacks to using this new method: due to the volatility of Nesterov's scheme in the non-convex setting there is still the possibility for divergence. Also, since our choice of η directly affects the frequency of restarting, this constant must be chosen carefully to avoid constant restarting or a complete lack of convergence. The reliance on such a 'magic number' hinders the usability of this new method and as such more work should be done to remove the need for the factor η. In the future we would also like to expand restarting techniques to the other methods used in this report, to fairly test each method. We would also like to expand testing to new problems such as movie-ranking.

References

Goldstein, T., O'Donoghue, B., Setzer, S. & Baraniuk, R. (2014), 'Fast alternating direction optimization methods', *SIAM Journal on Imaging Sciences* **7**(3), 1588–1623.
URL: *https://doi.org/10.1137/120896219*

Kolda, T. G. & Bader, B. W. (2009), 'Tensor decompositions and applications', *SIAM Review* **51**(3), 455–500.
URL: *https://doi.org/10.1137/07070111X*

Nesterov, Y. (2004), *Introductory Lectures on Convex Optimization: A Basic Course*, Vol. 87 of *Applied Optimization*, 1 edn, Kluwer Academic Publishers, Boston.

O'donoghue, B. & Candès, E. (2015), 'Adaptive restart for accelerated gradient schemes', *Found. Comput. Math.* **15**(3), 715–732.
URL: *http://dx.doi.org/10.1007/s10208-013-9150-3*

Sterck, H. D. (2012), 'A nonlinear GMRES optimization algorithm for canonical tensor decomposition', *SIAM J. Scientific Computing* **34**.