SESOP-TN: Combining Sequential Subspace Optimization with Truncated Newton method

Michael Zibulevsky

Department of Computer Science

Technion–Israel Institute of Technology

Email: mzib@cs.technion.ac.il

September 29, 2008

Abstract

We present a method for very large scale unconstrained optimization of smooth functions. It combines ideas of Sequential Subspace Optimization (SESOP) [4, 2] with those of the Truncated Newton (TN) method. Replacing TN line search with subspace optimization, we allow Conjugate Gradient (CG) iterations to stay matched through consequent TN steps. This resolves the problem of TN sensitivity to an early break of the CG process. For example, when an objective function is quadratic, the SESOP-TN trajectory coincides with the trajectory of CG as applied directly to the objective. Standard TN lacks this property and converges more slowly. Numerical experiments illustrate the effectiveness of the method. Matlab code is available at http://ie.technion.ac.il/~mcib/sesoptn.html

1 Derivation of the method

Consider a problem of unconstrained minimization of a smooth function of a very large number (possibly, millions) of variables

$$\min f(x), \quad x \in R^n.$$

Truncated Newton [1, 6] is one of the methods of choice for such problems. At every outer iteration it approximately minimizes a quadratic Taylor expansion $q_k(x)$ around the current iterate x^k, using limited number of CG steps. The expansion $q_k(x)$ is given by

$$q_k(x) = f(x^k) + {g^k}^T(x - x^k) + \frac{1}{2}(x - x^k)^T H_k(x - x^k), \tag{1}$$

where $g^k = \nabla f(x^k)$ is the gradient and $H_k = \nabla^2 f(x^k)$ – the Hessian of f at x^k. The outer iteration of TN is accomplished with a line search, in order to guarantee function decrease. The overall effectiveness of the TN method is rather sensitive to the choice of stopping rule for the internal CG optimization. We attempt to overcome this difficulty, replacing the line search with subspace optimization. In this way we allow the CG iterations to stay mathced through consequent TN steps.

Subspace optimization Suppose that TN step k was truncated after l CG iterations. Coming back from the quadratic model to the original objective, we would like to imitate the next CG step, using $f(x)$ instead of $q(x)$. CG iteration $l+1$ inside TN would perform optimization of the quadratic model $q(x)$ over the affine subspace S_{kl}, passing through the current inner iterate x^{kl} and spanned by the last CG step $x^{kl} - x^{k,l-1}$ and the current gradient $\nabla q(x^{kl})$. Instead, the next SESOP iteration will minimize f over the extended affine subspace $S_k \supset S_{kl}$. In order to provide monotone descent of f, we add to S_k the TN direction

$$d_{TN} = x^{kl} - x^k.$$

Now $x^k \in S_k$, and any monotone method used for the subspace optimization over S_k starting from x^k, will reduce the objective relatively to $f(x_k)$. Optionally, we include several previous outer steps and gradients of f into S_k, in order to improve the function descent, when the TN directions are not good enough.

Next Truncated Newton step After performing the subspace optimization, we start a new TN iteration. At this stage, in order to keep alignment through the global CG sequence, we perform the first new CG step as an optimization of the new quadratic model $q_{k+1}(x)$ over the 2D subspace spanned by $x^{k+1} - x^{kl}$ and $g(x_{k+1})$.

Summary of SESOP-TN algorithm: outer iteration k

1. **TN step** Solve approximately Newton system $\nabla^2 f(x^k) d_{TN}^k = -\nabla f(x^k)$, i.e. minimize quadratic model $q_k(x)$ in (1), using l steps of CG. Denote the last CG iterate as x^{kl}.

2. **Subspace optimization step** $x^{k+1} \approx \arg\min_{x \in S_k} f(x)$,

   where affine subspace S_k passes through x^k and is spanned by:
   * TN Direction $d_{TN}^k = x^{kl} - x^k$;
   * Last value of the gradient of quadratic model $\nabla q_k(x^{kl})$ used in TN;
   * Last used CG direction in TN: $(x^{kl} - x^{k,l-1})$;
   * [Optionally] directions of several previous outer steps and gradients of f.

3. **Goto TN step**, while performing the first new CG step as an optimization of quadratic model $q_{k+1}(x)$ over 2D subspace spanned by $(x^{k+1} - x^{kl})$ and $\nabla f(x^{k+1})$.

The presented procedure resolves the problem of TN sensitivity to early break of the CG process. For example, when the objective function is quadratic, SESOP-TN trajectory coincides with the trajectory of CG applied directly to the objective function, independently of the stopping rule in the TN step. Standard TN lacks this property and converges more slowly when truncated too early. Note also that subspace optimization can be carried out very efficiently if

$$f(x) = \varphi(Ax) + \psi(x),$$

where computing the linear map Ax is expensive relatively to the nonlinear functions $\varphi(\cdot)$ and $\psi(\cdot)$ (see [4] for the details).

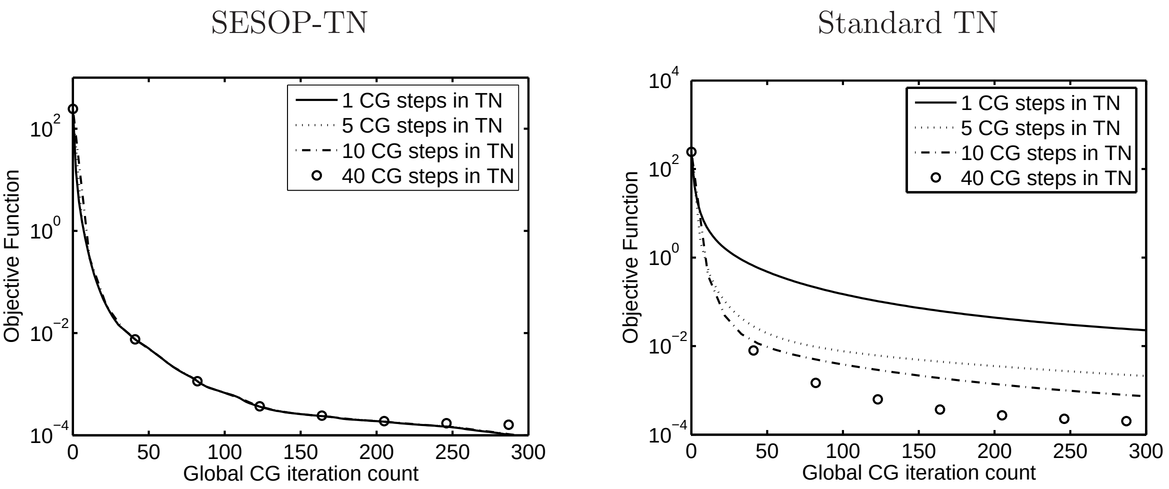


Figure 1: Solving linear least squares (2), with 400 variables. The SESOP-TN trajectory does not depend on the number of CG iterations in TN step. Standard TN converges more slowly, when CG is truncated too early.

2 Preliminary Numerical Experiments

Quadratic function First, let us demonstrate "proof of the concept" using a pure quadratic function. We solve the linear least squares problem

$$\min ||Ax - b||^2 \tag{2}$$

with $n = 400$ variables, where the square random matrix A has zero-mean *i.i.d.* Gaussian entries with variance $1/n$. As we see in Figure 1, SESOP-TN trajectory, as expected, does not depend on the number of CG iterations in the TN step. Standard TN (the right plot) lacks this property.

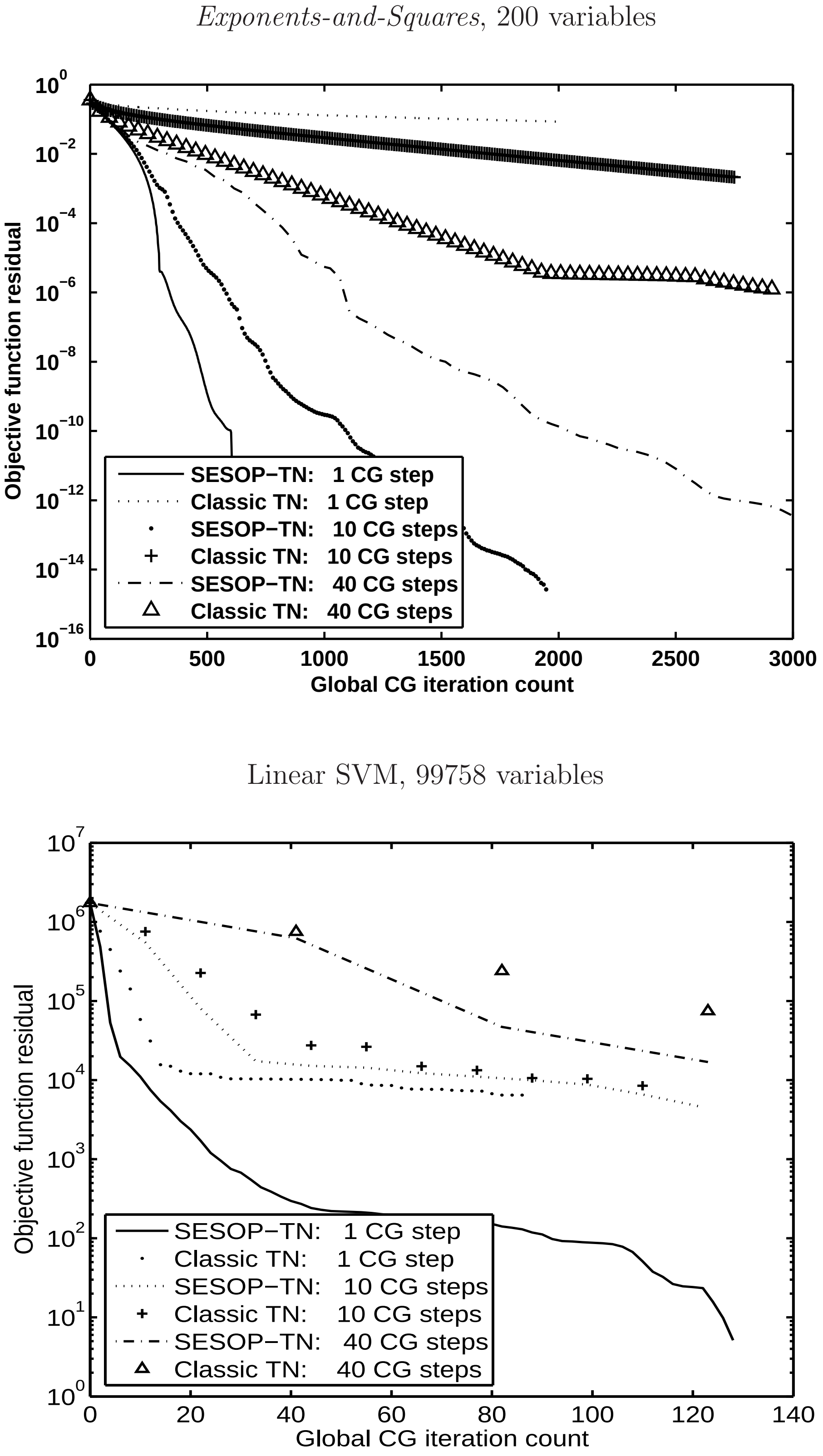
Two non-linear examples The first problem is *Exponents-and-Squares* [7] with $n = 200$ variables:

$$\min \quad e^{-\mathbf{1}^T x} + \frac{1}{2}\sum_{j=1}^{n} j^2 x_j^2 \ .$$

The second example is *Linear Support Vector Machine (SVM)*, see [5] for more details on unconstrained formulation of SVM. We used data set *Astro-physics-29882* [3] with 99758 variables, and selected randomly 1495 training examples from there. In both problems (see Figure 2), SESOP-TN consistently outperformed classic TN, when restricted to 1, 10 or 40 CG iterations in TN step.

Acknowledgements

I am grateful to Thorsten Joachims, S.V.N. Vishwanathan and Choon Hui Teo for the help in managing SVM data.



Exponents-and-Squares, 200 variables

Linear SVM, 99758 variables

Figure 2: Two nonlinear problems. The plots show the residual between the current objective and the optimal value versus CG iteration count.

References

[1] R. S. Dembo and T. Steihaug, *Truncated-Newton algorithms for large-scale unconstrained optimization*, Mathematical Programming, 26 (1983), pp. 190–212.

[2] M. Elad, B. Matalon, and M. Zibulevsky, *Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization*, Applied and Computational Harmonic Analysis, 23 (2007), pp. 346–367.

[3] T. Joachims, *Training linear SVMs in linear time*, in Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), 2006.

[4] G. Narkiss and M. Zibulevsky, *Sequential subspace optimization method for large-scale unconstrained problems*, Technical Report CCIT 559, Technion – Israel Institute of Technology, Faculty of Electrical Engineering, 2005.

[5] ——, *Support Vector Machine via sequential subspace optimization*, Technical Report CCIT 557, Technion – Israel Institute of Technology, Faculty of Electrical Engineering, 2005.

[6] S. G. Nash, *A survey of Truncated-Newton methods*, Journal of Computational and Applied Mathematics, (2000), pp. 45–59.

[7] H. Nielsen, *UCTP - test problems for unconstrained optimization*, Technical Report IMM-REP-2000-18, Department of Mathematical Modelling, DTU, 2000.