Golden Section Search over Traditional Line search variants for Large Markov Networks

Mihir Kulkarni
Computer Science and Engineering
University at Buffalo
Buffalo, NY-14214
UB Person number: 50168610
mihirdha@buffalo.edu

Pramod Rangaraju
Computer Science and Engineering
University at Buffalo
Buffalo, NY-14214
UB Person number: 50169514
prangara@buffalo.edu

***Abstract*— This project highlights the use of Golden Section search over traditional line search variants for parameter estimation of Markov networks with large number of factors. This Project compares performance of Golden section search over various gradient descent models viz. stochastic gradient descent, Direct Line search such as Exact Line search, Backtracking Line Search. The comparison is carried out on metrics such as Speed of convergence and Computation on some particular unimodal function.**

***Index Terms*— Golden Section Search, Gradient Descent, Markov Networks, Line Search.**

I. Introduction

Markov Networks are inherently global models as opposed to Bayesian Network. This poses significant computational ramifications in the parameter estimation of Markov Networks. Due to lack of a closed form solution, Markov Networks resort to iterative methods such as Stochastic Gradient Descent and Line search for the estimation of parameters. Unfortunately, each step in the iterative algorithm requires inference. This makes the estimation for Markov Networks with large number of factors nearly impossible. Although we have not been able to speed up the inference up to some extent, it is still not at par with the complexity of the modern problems.

Golden section search attempts to minimize the number of times the inference is calculated by reaching the global optimum in minimum number of steps. Traditional line searches attempt to reach minimum in logarithmic number of steps. Although Golden section search also takes logarithmic number of steps, it attempts to reduce the steps by carefully choosing the step size.

We are comparing the computational complexity required by Golden section search as opposed to other Stochastic Gradient Descent and Line search variants.

We have performed analysis on them for metrics such as Speed of Convergence and Computation of the various unimodal functions of the error during the optimization process and present the result of various searches.

II. Dataset

Likelihood functions of Markov Networks are concave i.e. unimodal. This means that they have a single global optimum. For this project we have used several unimodal functions to evaluate the performance of various algorithms presented in the project.

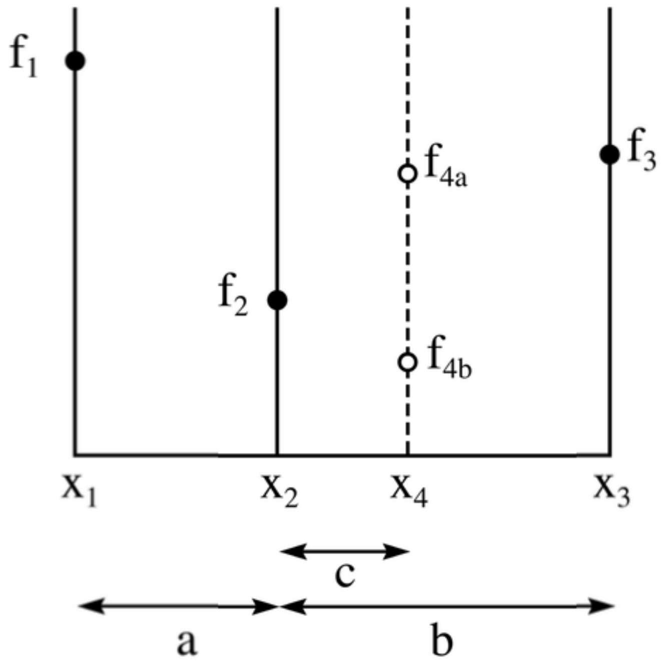
III. Algorithms

The following are some of the algorithms we have implemented and used to compare their performances.

- Golden Section Search Algorithm
- Stochastic Gradient Descent
- Exact Line Search
- Backtracking Line Search

A. Golden Section Search Algorithm

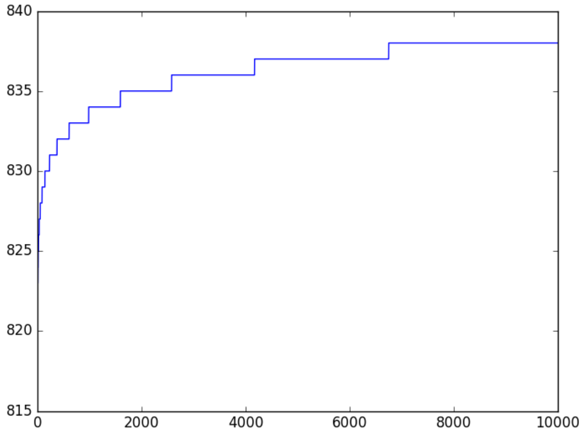
The golden section search is a technique for finding the extremum (minimum or maximum) of a strictly unimodal function by successively narrowing the range of values inside which the extremum is known to exist. The technique derives its name from the fact that the algorithm maintains the function values for triples of points whose distances form a golden ratio



The diagram above illustrates a single step in the technique for finding a minimum. The functional values of f(x) are on the vertical axis, and the horizontal axis is the x parameter. The value of f(x) has already been evaluated at the three points: x_1, x_2 and x_3. Since f_2 is smaller than either f_1 or f_3, it is clear that a minimum lies inside the interval from x_1 to x_3 (since f is unimodal).

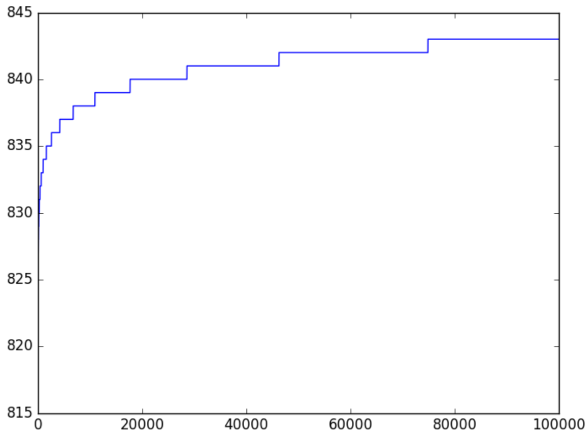
The next step in the minimization process is to "probe" the function by evaluating it at a new value of x, namely x_4. It is most efficient to choose x_4 somewhere inside the largest interval, i.e. between x_2 and x_3. From the diagram, it is clear that if the function yields f_{4a} then a minimum lies between x_1 and x_4 and the new triplet of points will be x_1, x_2, and x_4. However if the function yields the value f_{4b} then a minimum lies between x_2 and x_3, and the new triplet of points will be x_2, x_4, and x_3. Thus, in either case, we can construct a new narrower search interval that is guaranteed to contain the function's minimum.

In our experiments, we could see that the number of computation required i.e. number of times the inference needed to be calculated quickly stabilized and followed a logarithmic curve.



X axis- search space
Y axis- Number of times inference is called

In the second experiment, even after increasing the size of search space by a factor of 10, the number of computations only increased from 838 to 842.



X axis- search space
Y axis- Number of times inference is called

B. Stochastic Gradient Descent

Many numerical learning algorithms amount to optimizing a cost function which can be expressed as an average over the training examples. The loss function measures how well the learning system performs on each example. The cost function is then the average of the loss function measures on all training examples, possibly augmented with capacity control terms.

Computing such an average takes a time proportional to the number of examples n. This constant always appears in the running time of all optimization algorithm that considers the complete cost function. **Stochastic gradient descent** updates the learning system on the basis of the loss function measured for a single example. Such an algorithm works because the averaged effect of these updates is the same. Although the convergence is much noisier, the elimination of the constant n in the computing cost can be a huge advantage for large-scale problems. In this case, $Q_i(w)$ is the value of the error function at i-th example, and Q(w) is the error.

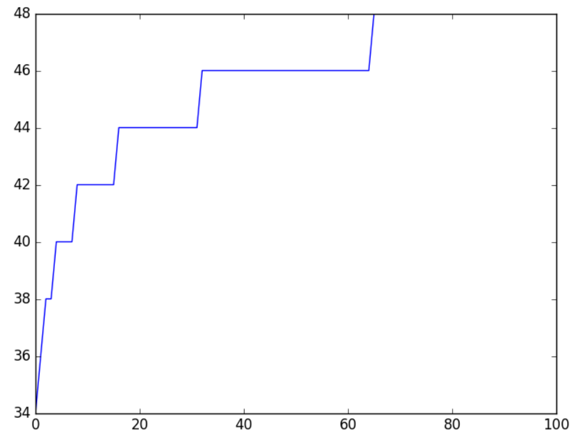
$$Q(w) = \sum_1^n Q_i(w)$$

Where w is the parameter which minimizes the Q(w) needs to be estimated. To minimize the above function, a standard gradient descent method would perform the following iterations

$$w := w - \eta VQ(w) = w - \eta \sum_1^n VQ_i(w)$$

where η is the step size or the learning rate

The computation in Stochastic gradient descent, slthough logarithmic, was increasing quite rapidly



C. Exact Line Search

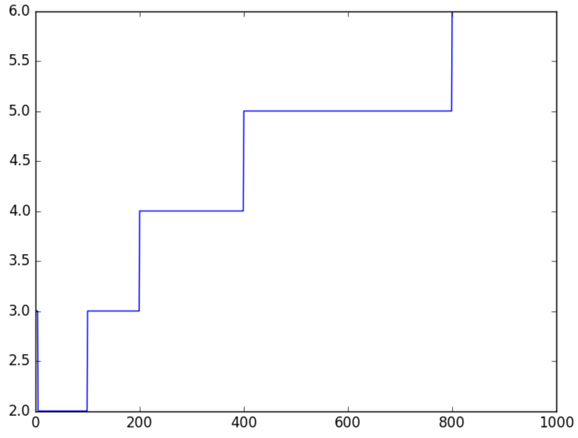
The line search approach first finds a descent direction along which the objective function f will be reduced and then computes a step size that determines how far a unimodal function should move along that direction.

Line Search algorithm.

1. Set iteration counter k = 0, and make an initial guess, x0 for the minimum
2. Repeat:
3. Compute a descent direction Pk
4. Choose αk to 'loosely' minimize h(α) = f(xk + αPk) over α ϵ R+
5. Update Xk+1 = Xk + αkpk, and k = k+1
6. Until ||Vf(xk)|| < tolerance

At the line search step (4) the algorithm might either exactly minimize h, by solving h’(α) = 0 or loosely, by asking for a sufficient decrease in h.

In our experiment The line search did not converge in reasonable amount of time for large search area.



X axis- search space

Y axis- Number of times inference is called

In small search space however, the number of computations, although still logarithmic, were increasing rapidly.

D. Backtracking Line Search

Backtracking Line Search is a search scheme based on the Armijo–Goldstein condition, is a line search method to determine the maximum amount to move along a given search direction. It involves starting with a relatively large estimate of the step size for movement along the search direction, and iteratively shrinking the step size (i.e., "backtracking") until a decrease of the objective function is observed that adequately corresponds to the decrease that is expected, based on the local gradient of the objective function.

The backtracking line search starts with a large estimate of α and iteratively shrinks it. The shrinking continues until a value is found that is small enough to provide a decrease in the objective function that adequately matches the decrease that is expected to be achieved, based on the local function gradient f(x)

Define the local slope of the function of α along the search direction P as

$$m = p^T \mathbf{V} f(x).$$

It is assumed that P is a unit vector in a direction in which some local decrease is possible, i.e., it is assumed that m<0.

Based on a selected control parameter c ϵ (0, 1), the Armijo–Goldstein condition tests whether a step-wise movement from a current position x to a modified position x + p achieves an adequately corresponding decrease in the objective function. The condition is fulfilled if

$$f(x+ \alpha p) < f(x) + \alpha c m$$

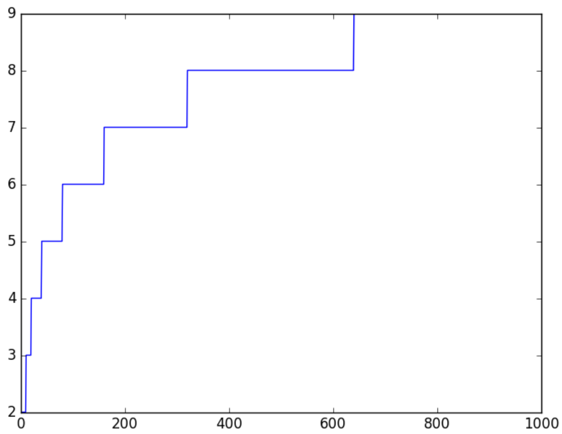
This condition, when used appropriately as part of a line search, can ensure that the step size is not excessively large. However, this condition is not sufficient on its own to ensure that the step size is nearly optimal, since any value of α that is sufficiently small will satisfy the condition.

Thus, the backtracking line search strategy starts with a relatively large step size, and repeatedly shrinks it by a factor Γ ϵ (0, 1) until the Armijo–Goldstein condition is fulfilled.

The search will terminate after a finite number of steps for any positive values of c and Γ that are less than 1.

In our experiment, even if the number of steps were small in backtracking line search it did not always reach the global optimum. It also did not converge for large search space which golden section search handled easily.

So we are showing results for only small search space



X axis- search space
Y axis- Number of times inference is called

IV. Result

We found that although there is little difference in Golden Section Search and traditional Line search algorithms, the difference becomes apparent and often very large when dealing with large search space. Hence, golden section search is an useful line search method when dealing with Markov Networks with large number of factors.

References

[1] Bishop, Christopher M. (2006). Pattern Recognition and Machine Learning. Springer. pp. 359–422. ISBN 0-387-31073-8. MR 2247587

[2] https://en.wikipedia.org/wiki/Stochastic_gradient_descent

[3] https://en.wikipedia.org/wiki/Golden_section_search

[4] Bordes, A., Bottou, L., Gallinari, P.: SGD-QN: Careful quasi-Newton stochastic gradient descent. Journal of Machine Learning Research 10, 1737–1754 (July 2009), with erratum, JMLR 11:2229-2240

[5] Bottou, L., Bousquet, O.: The tradeoffs of large scale learning. In: Platt, J., Koller, D., Singer, Y., Roweis, S. (eds.) Advances in Neural Information Processing Systems, vol. 20, pp. 161–168. NIPS Foundation (http://books.nips.cc)

[6] Bottou, L.: Online algorithms and stochastic approximations. In: Saad, D. (ed.) Online Learning and Neural Networks. Cambridge University Press, Cambridge, UK

[7] Bousquet, O.: Concentration Inequalities and Empirical Processes Theory Applied to the Analysis of Learning Algorithms. Ph.D. thesis, Ecole Polytechnique, Palaiseau, France

[8] Cortes, C., Vapnik, V.: Support-vector network. Machine Learning 20(3), 273–297

[9] Dennis, J., Schnabel, R.B.: Numerical Methods For Unconstrained Optimization and Nonlinear Equations. Prentice-Hall, Inc., Englewood Cliffs, New Jersey

[10] Joachims, T.: Training linear SVMs in linear time. In: Proceedings of the 12th ACM SIGKDD International Conference. New York

[11] Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Brodley, C.E., Danyluk, A.P. (eds.) Proceedings of the Eighteenth International Conference on Machine Learning (ICML). pp. 282–289. Morgan Kaufmann, Williams College, Williamstown

[12] Lee, W.S., Bartlett, P.L., Williamson, R.C.: The importance of convexity in learning with squared loss. IEEE Transactions on Information Theory 44(5), 1974–1980

[13] Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: RCV1: A new benchmark collection for text categorization research. Journal of Machine Learning Research 5, 361–397