Triple Jump Acceleration for the EM Algorithm

Han-Shen Huang[a] Bou-Ho Yang[a,b] Chun-Nan Hsu[a]
Institute of Information Science[a]
Academia Sinica
Nankang, Taipei, Taiwan
Department of Electrical Engineering[b]
Chang Gung University
Taoyuan, Taiwan
{*hanshen,ericyang,chunnan*}*@iis.sinica.edu.tw*

Abstract

This paper presents the triple jump framework for accelerating the EM algorithm and other bound optimization methods. The idea is to extrapolate the third search point based on the previous two search points found by regular EM. As the convergence rate of regular EM becomes slower, the distance of the triple jump will be longer, and thus provide higher speedup for data sets where EM converges slowly. Experimental results show that the triple jump framework significantly outperforms EM and other acceleration methods of EM for a variety of probabilistic models, especially when the data set is sparse. The results also show that the triple jump framework is particularly effective for Cluster Models.

1. Introduction

The Expectation-Maximization (EM) algorithm [5] is one of the most popular algorithms for learning probabilistic models from incomplete data. However, when applied to large real world data sets, the EM algorithm is slow to converge. Previously, Bauer et. al. [1] have proposed *parameterized EM* to accelerate EM for Bayesian Networks. Ortiz and Kaelbling [6] have proposed a similar method for Mixtures of Gaussians. Salakhutdinov and Roweis [7] showed that with the learning rate (i.e., the extent of the extrapolation) within a certain interval, parameterized EM is guaranteed to converge. However, since the learning rate in such an interval is too small, the speedup will not be significant. Therefore, they proposed *adaptive overrelaxed EM* [7], which tries greater learning rates without the convergence guarantee and switches back to regular EM if the new data likelihood is not increased. In this way, the data likelihood will still monotonically increase and adaptive overrelaxed EM is guaranteed to converge.

We propose the *triple jump* acceleration framework to accelerate the EM algorithm based on the Aitken acceleration method [2]. The idea is to extrapolate the third search point based on the previous two search points. The extrapolation can reach a point far away from the previous two points, like hop, step and jump in the triple jump. Experimental results show significant speedup for Bayesian Networks, Hidden Markov Models, and AUTOCLASS-like Cluster Models [3].

2. Bound Optimization Methods

Given an incomplete data set $\mathcal{D}$, suppose we want to learn the parameter vector θ of a probabilistic model that maximizes the log-likelihood $L_{\mathcal{D}}(\theta)$ of the data [1]. The EM algorithm is one of the bound optimization methods [7], which exploits a bound $G(\theta)$ on the objective function $L(\theta)$ and proceeds by optimizing $G(\theta)$. Bound optimization methods assume that for any θ there exists $G(\theta)$ such that $G(\theta) \leq L(\theta)$ and it is easy to solve $\arg\max_\theta G(\theta)$. In iteration t, a bound optimization method finds $\theta^{(t+1)}$ by creating $G^{(t)}(\theta)$ such that $G^{(t)}(\theta^{(t)}) = L(\theta^{(t)})$ and then solve this optimization problem:

$$\theta^{(t+1)} = \arg\max_\theta G^{(t)}(\theta). \quad (1)$$

Since $L(\theta^{(t)}) = G^{(t)}(\theta^{(t)}) \leq G^{(t)}(\theta^{(t+1)}) \leq L(\theta^{(t+1)})$, $L(\theta)$ is guaranteed to increase monotonically and converge to a local maximum.

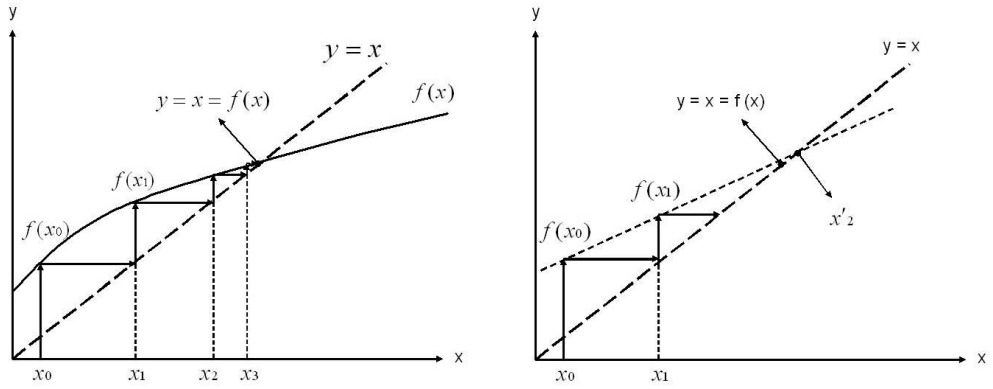
Overrelaxed bound optimization methods [7] accelerate regular bound optimization methods by an extrapolation to the direction of bound optimization:

$$\theta^{(t+1)} = \theta^{(t)} + \eta(\arg\max_\theta G^{(t)}(\theta) - \theta^{(t)}), \quad (2)$$

where η is the learning rate. The *parameterized EM* [1] (pEM) algorithm is an instantiation of the overrelaxed bound optimization for EM. It has been shown that parameterized EM converges within the neighborhood of a local

[1] We will abbreviate $L_{\mathcal{D}}(\theta)$ and use $L(\theta)$ in the context where $\mathcal{D}$ is obvious.

Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05)
1550-4786/05 $20.00 © 2005 IEEE



(a) Illustration of fixed-point iteration method (b) Illustration of the triple jump framework

Figure 1. Bound optimization methods can be considered as fixed-point iteration, as shown in 1(a). Therefore, bound optimization methods can be accelerated by extrapolation, as shown in 1(b).

maximum when $0 < \eta < 2$ [8]. However, this learning rate is too small to provide significant speedup. Salakutdinov and Roweis [7] proposed the adaptive overrelaxed bound optimization methods to dynamically change the learning rate to provide significant speedup. η is multiplied with a given increasing rate if the likelihood is still improving. Otherwise, η will be reset to one.

3. Accelerating Bound Optimization Methods

Equation (2) can be considered as a fixed-point iteration which has a general form $x = f(x)$. Figure 1 illustrates an example of the fixed-point iteration when x is a scaler. The idea is to find the intersection x^* of $y = f(x)$ and $y = x$ by iteration, as illustrated in Figure 1(a). Aitken's acceleration suggests an extrapolation from (x_0, x_1) to (x_1, x_2) to find the next search point. That is, let $\gamma = \frac{|x_2 - x_1|}{|x_1 - x_0|}$, we compute a new $\hat{x_2}$ by extrapolation with $\hat{x_2} = x_1 + \gamma(x_2 - x_1)$.

Assuming that $\gamma < \kappa$ where $0 < \kappa < 1$ [2], and the errors of x_0, x_1 and x_2 with x^* are in the same direction and reduced at the same rate [2], we can perform the extrapolation along the direction as illustrated in Figure 1(b), by repeatedly applying the extrapolation with the same γ:

$$\hat{x^*} = x_1 + \sum_{s=0}^{\infty} \gamma^s (x_2 - x_1) = x_1 + \frac{(x_2 - x_1)}{1 - \gamma} = x_1 + \eta(x_2 - x_1). \tag{3}$$

4. The Triple Jump Framework

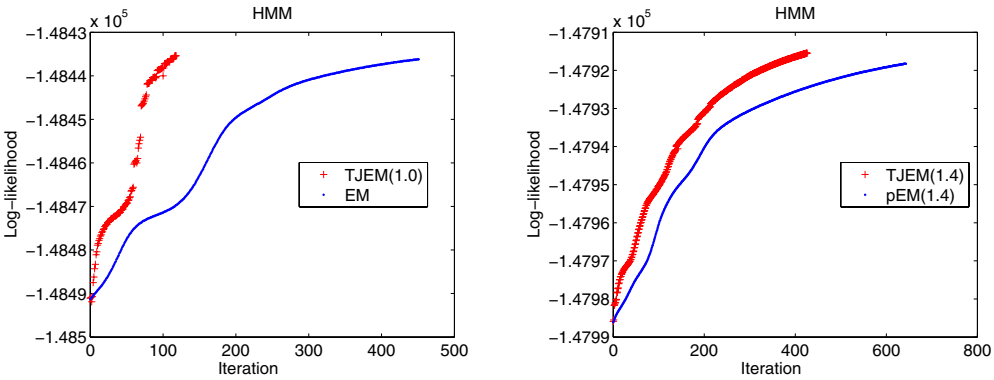
The basic idea of the triple jump framework is to perform two iterations of bound or overrelaxed bound optimization to obtain γ, and compute the next search point with a large η. Several issues must be resolved to realize this basic idea. First, if an overrelaxed bound optimization method is applied in the first two iterations, they must use the same learning rate. Otherwise, the two iterations will become irrelevant and γ will not be reasonable. The second issue is that the extrapolations by the triple jump may not always converge. We adopt the strategy of adaptive overrelaxed bound optimization methods, which resort to the regular bound optimization method. At last, the triple jump might lead to illegal points. We can bring back the third jump to a legal place by interpolation with the second jump.

Algorithm 1 gives the algorithm that computes the third jump in the triple jump framework. The input consists of three parameter vectors — initial, hop and step estimates, computed in advance using Equation (2) with the same learning rate. Note that the parameter vectors are divided into independent sub-vectors and the jump is computed for each sub-vector independently. Updating sub-vectors reduces the chance of making illegal jumps.

Algorithm 1 Jump

input: initial estimate θ_a, hop estimate θ_b, and step estimate θ_c
output: jump estimate θ_d.
for each sub-vector set $(\theta'_a, \theta'_b, \theta'_c)$ **do**
  $\gamma \leftarrow \frac{\|\theta'_c - \theta'_b\|}{\|\theta'_b - \theta'_a\|}$.
  if $(\gamma < \kappa)$
    $\theta'_d \leftarrow \theta'_b + \frac{1}{1-\gamma}(\theta'_c - \theta'_b)$.
    while θ'_d is illegal **do** $\theta'_d \leftarrow 0.5\theta'_c + 0.5\theta'_d$
  else
    $\theta'_d \leftarrow \theta'_c$
  end if
end while

Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05)
1550-4786/05 $20.00 © 2005 IEEE



(a) Learning curves of TJEM(1.0) and EM

(b) Learning curves of TJEM(1.4) and pEM(1.4)

Figure 2. Comparing TJEM with EM and pEM(1.4) for training HMMs.

Algorithm 2 gives the pseudo code of the triple jump framework, which is a greedy state-space search algorithm that searches for a local optimal parameter vector θ^*. For conciseness, we define a two-valued function $[L^{(t)}, \hat{\theta}_{\text{BO}}^{(t+1)}] = \textbf{BO1}(\theta^{(t)})$, which returns the log-likelihood $L^{(t)} = L(\theta^{(t)})$ and $\hat{\theta}_{\text{BO}}^{(t+1)}$ by executing Equation (1). This function computes one iteration of bound optimization. Three operators expands the search space, resulting in three possible candidates for the next iteration :

- $\hat{\theta}_1^{(t)}$: invoking Algorithm 1 to make the third jump,
- $\hat{\theta}_2^{(t)}$: extrapolating as in overrelaxed bound optimization with Equation (2), or
- $\hat{\theta}_3^{(t)}$: applying an iteration as in regular bound optimization methods using Equation (1).

In iteration t, those candidates will be tested in turn to see whether they increase the data likelihood and whether they are legal. The first legal parameter vector that increases the likelihood more than δ is selected as $\theta^{(t)}$, and the unvisited candidates are discarded. The search stops when no candidate can be selected.

Algorithm 2 Triple jump

randomly initialize $\theta^{(0)}$, $t \leftarrow 0$, *num_algs* = 3
$[L^{(0)}, \hat{\theta}_3^{(1)}] \leftarrow \textbf{BO1}(\theta^{(0)})$, $\hat{\theta}_2^{(1)} \leftarrow$ Equation (2) with η
repeat
  $i \leftarrow 1$, $t \leftarrow t+1$
  repeat
    $[\hat{L}_i^{(t)}, \hat{\theta}_3^{(t+1)}] \leftarrow \textbf{BO1}(\hat{\theta}_i^{(t)})$
    if $((\hat{L}_i^{(t)} - L^{(t-1)}) > \delta)$ $\theta^{(t)} \leftarrow \hat{\theta}_i^{(t)}$
    else $i \leftarrow i+1$ **end if**
  until ($i >$ *num_algs* **or** $\hat{L}_i^{(t)} - L^{(t-1)}) > \delta$)
  if ($i \leq$ *num_algs*) // $\hat{L}_i^{(t)} - L^{(t-1)} > \delta$
    adjust η, $\hat{\theta}_2^{(t+1)} \leftarrow$ Equation (2) with η
    if ($i \neq 1$) $\hat{\theta}_1^{(t+1)} \leftarrow \text{Jump}(\theta^{(t-1)}, \theta^{(t)}, \hat{\theta}_i^{(t+1)})$ **end if**
  end if
until ($i >$ *num_algs*) // *converge*

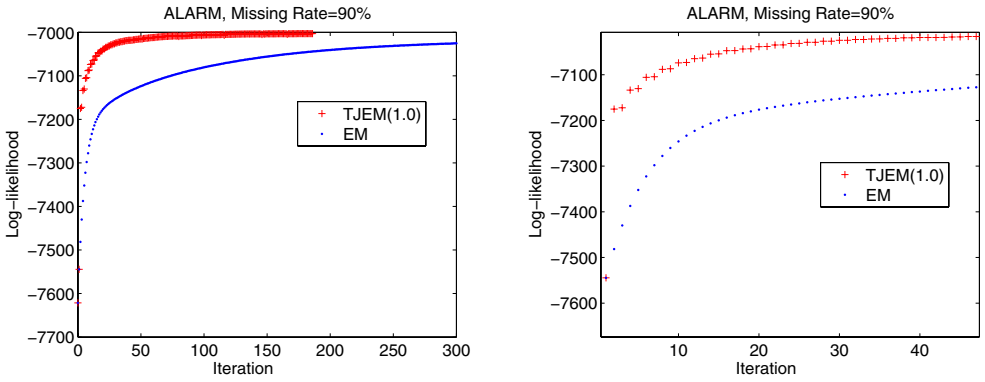
5. Experimental Results

We experimentally evaluated the performance of the EM instantiation of the triple jump framework for different probabilistic models. We use EM and pEM(η) to represent the EM algorithm and parameterized EM algorithm with learning rate η, respectively. AEM denotes the EM instantiation of adaptive overrelaxed bound optimization. TJEM(η) is the EM instantiation of the triple jump framework with a fixed learning rate η for preparing hop and step in a triple jump, and TJEM(AEM) the triple jump EM with adaptive learning rate. We measured the performance of the different methods by counting the number of times that Equation (2) is executed because it is the most time-consuming part and may be executed more than once to select the next parameter in TJEM and AEM. We will simply refer to this measure as *iterations* in the following discussion.

5.1. Hidden Markov Models

First, we evaluated the acceleration of TJEM with Hidden Markov Models (HMM). We synthesized data sets from a five-state, 20-symbol HMM with randomly assigned state transition probabilities. Each data set contains 500 sequences of length 100. We used the data sets to train HMM with different methods and compared their convergence performance. We compared EM versus TJEM(1.0) and pEM(1.4) versus TJEM(1.4). Note that TJEM(1.0) is TJEM using EM to prepare hop and step. Figure 2 shows the results. TJEM outperforms its counterparts by reaching

Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05)
1550-4786/05 $20.00 © 2005 IEEE



(a) Learning curves of TJEM(1.0) and EM

(b) Zoom-in view of early iterations

Figure 3. Learning curves of TJEM and EM for training a large BN with sparse data.

local maxima faster with much less iterations. We had tried other learning rates for pEM and obtained similar results.

5.2. Bayesian Networks

We also evaluated the performance of TJEM with the ALARM model [4], a real world Bayesian Network with 37 nodes. We randomly assigned conditional probabilities and synthesized data sets with 1,000 examples. In addition, we randomly removed 90% of data from the data set to evaluate the performance of TJEM for sparse data. Figure 3(a) shows the learning curves of EM versus TJEM(1.0) for the data set with 90% of missing data. The curves show that TJEM converges much faster than EM as expected. We can zoom in to the early iterations as shown in Figure 3(b) to observe the change of the log-likelihood obtained in each iteration. The figure shows clear consecutive "triple jumps" — two small hops followed by a far jump, by TJEM.

5.3. Cluster Models

This experiment shows that TJEM is particularly effective for Cluster Models, which are AUTOCLASS-like Bayesian Networks [3] consisting of a latent cluster node with independent children as the features. We created a Cluster Model that clusters feature vectors with 50 binary features into 10 groups. Each data set contains 1,000 synthesized examples with 30%, 60% and 90% missing rates for the data in the feature vectors. It turns out that pEM, AEM, and EM all requires more iterations to converge for more missing data. Moreover, TJEM requires nearly the same iterations to converge regardless of the proportion of missing data. For example, pEM(1.2) and AEM required averagely 14 and 10 iterations respectively to converge with 60% missing rate, and 49 and 21 iterations respectively with 90% missing rate. But TJEM took only five iterations for almost all cases.

6. Conclusion and Future Work

This paper presents a new framework for accelerating the EM algorithm by performing "triple jump" on the surface of log-likelihood of data. This new framework can also be applied to other bound optimization methods. Experimental results show that our method converges faster than previous ones for several probabilistic models, and is particularly effective for Cluster Models.

References

[1] E. Bauer, D. Koller, and Y. Singer. "Update rules for parameter estimation in Bayesian networks." In *Proc. of the 13th Conference on Uncertainty in Artificial Intelligence*, pages 3–13, 1997.

[2] R. L. Burden and D. Faires. *Numerical Analysis*. PWS-KENT Pub Co., 1988.

[3] P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor, and D. Freeman. "Autoclass: A Bayesian classification system." In *Proc. of the 5th International Conference on Machine Learning*, pages 54–56, 1988.

[4] G. F. Cooper and E. Herskovits. "A Bayesian method for the induction of probabilistic networks from data." *Machine Learning*, 9:309–347, 1992.

[5] A. Dempster, N. Laird, and D. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." *Journal of the Royal statistical Society*, B39:1–37, 1977.

[6] O. Luis and K. Leslie. "Accelerating EM: An empirical study." In *Proc. of the 15th Conference on Uncertainty in Artificial Intelligence*, pages 512–521, 1999.

[7] R. Salakhutdinov and S. Roweis. "Adaptive overrelaxed bound optimization methods." In *Proc. of the 20th International Conference on Machine Learning*, pages 664–671, 2003.

[8] R. Salakhutdinov, S. Roweis, and Z. Ghahramani. "On the convergence of bound optimization algorithms." In *Proc. of the 19th Conference on Uncertainty in Artificial Intelligence*, pages 509–516, 2003.