TJ^{2}aEM: Targeted Aggressive Extrapolation Method for Accelerating the EM Algorithm

Han-Shen Huang, Bo-Hou Yang, and Chun-Nan Hsu

Abstract The Expectation-Maximization (EM) algorithm is one of the most popular algorithms for parameter estimation from incomplete data, but its convergence can be slow for some large-scale or complex machine learning problems. Extrapolation methods can effectively accelerate EM, but to ensure stability, the learning rate of extrapolation must be compromised. This paper describes the TJ^{2}aEM algorithm, a targeted aggressive extrapolation method that can make much more aggressive extrapolations without causing instability problems. We show that for a wide variety of probabilistic models, TJ^{2}aEM can converge many times faster than other acceleration methods under different data distributions and initial conditions. In addition to EM, TJ^{2}aEM can also be applied to other bound optimization methods, including generalized iterative scaling, non-negative matrix factorization and concave-convex procedure.

Key words: Expectation-Maximization (EM), Aitken Acceleration, Extrapolation, Optimization, Triple-Jump Acceleration

Han-Shen Huang
Institute of Information Science, Academia Sinica, Taipei, Taiwan, e-mail: hanshen@iis.sinica.edu.tw

Bo-Hou Yang
Department of Electrical Engineering, Chang Gung University, Taoyuan, Taiwan
Institute of Information Science, Academia Sinica, Taipei, Taiwan, e-mail: ericyang@iis.sinica.edu.tw

Chun-Nan Hsu
Institute of Information Science, Academia Sinica, Taipei, Taiwan, e-mail: chunnan@iis.sinica.edu.tw

1 Introduction

The Expectation-Maximization (EM) algorithm (Dempster et al., 1977; McLachlan and Krishnan, 1997) is one of the most popular algorithms for parameter estimation of probabilistic models from incomplete data. Suppose we want to estimate the parameter vector θ of a probabilistic model to maximize the log-likelihood $L(\theta)$ from an incomplete data set. The EM algorithm solves the problem by iteratively searching for a local optimal solution θ^* on the data likelihood surface with the guarantee that the likelihood of the estimates increases monotonically.

When applied to complex machine learning problems with large data sets and a large number of parameters to estimate, the EM algorithm may converge slowly. If the data sets also contain a large proportion of missing data or there are a large number of hidden variables in the model to be imputed, the convergence of EM can be even slower. Previously, Bauer et al. (1997) proposed the *parameterized EM* (pEM) algorithm to accelerate EM for Bayesian Networks, and Luis and Leslie (1999) applied pEM for Mixtures of Gaussians. The pEM algorithm accelerates the convergence of the EM algorithm by extrapolating along the direction to the EM estimate with a fixed learning rate η. They showed that pEM converges faster than EM and the convergence is guaranteed when $1 < \eta < 2$. But pEM with a learning rate within this range is usually too conservative to gain significant speedup. Hammerlin and Hoffmann (1991) derived an optimal learning rate for pEM-like extrapolation but in practice it is difficult to obtain this learning rate because it depends on the maximal and minimal eigenvalues of the Jacobian of the EM mapping.

Compared to other numerical optimization methods, conservative extrapolation methods have the advantage that it is easy to implement for any complex probabilistic models and easy to integrate with any existing software package, but is slow due to the lack of informative guidance. In fact, the extrapolation can be made more aggressive to further accelerate the EM algorithm (Salakhutdinov and Roweis, 2003; Hesterberg, 2005; Kuroda and Sakakihara, 2006; Berlinet and Roland, 2007). However, since a large learning rate may lead to likelihood decreasing and thus failure of convergence, an aggressive extrapolation must be interleaved with conservative ones to keep the search stable. Therefore, aggressive extrapolation methods must dynamically adjust their learning rates. Also, to avoid incurring too much overhead, the adjustment must be efficient. Salakhutdinov and Roweis (2003) proposed *adaptive overrelaxed EM* (aEM), which increases η by a constant ratio at every iteration if the pEM extrapolation increases the likelihood and resets η to one otherwise. Hesterberg (2005) proposed *staggered EM*, which estimates the maximal eigenvalue of the Jacobian of the EM mapping to obtain the upper bound of η and then rotates among learning rates within the bounded range in a predefined order or at random. Since these methods confine the range of the adjustment, their extrapolation may not be aggressive enough to achieve substantial speedup in some cases.

An alternative to aggressive extrapolation is targeted aggressive extrapolation methods, which at each iteration compute an informed aggressive extrapolation that targets the local optimum directly. Usually this is achieved by combining two or more consecutive EM estimates. A well-known method is to use estimated eigen-

values as the learning rates based on two consecutive EM estimates (Schafer, 1997). To be precise, we will call this method *the triple jump EM method* (TJEM) (Huang et al., 2005) in this paper and will describe it in details in Section 3. TJEM can be derived from Aitken acceleration and is aimed at approximating the Jacobian by the eigenvalue of its slowest dimension, which dictates the global rate of convergence (Dempster et al., 1977). Since eigenvalues are scalars, both estimation and extrapolation can be computed as efficiently as pEM and aEM. Staggered EM is an extension of TJEM. More recently, Kuroda and Sakakihara (2006) proposed the ε-accelerated EM based on the vector ε algorithm (Wynn, 1962), which was originally designed to accelerate a slowly convergent sequence. Varadhan and Roland (2004) proposed the SQUAREM algorithm. The idea is to extrapolate to a parameter vector on the straight line across two consecutive EM estimates in the parameter space such that this parameter vector is estimated to be the closest to the local optimum. Though these methods can make very aggressive extrapolations, they share a common disadvantage that they favor the acceleration of slow dimensions but may drift away from the optimum along the dimensions already close to the optimum. In contrast, aEM and staggered EM have an advantage here because by applying large and small learning rates by turns, both fast and slow dimensions can be covered. Section 6.1 explains why aEM can be effective. Favoring slow dimensions too much may also cause instability. Therefore, addenda to keep the search stable, such as the restarting test for SQUAREM (Berlinet and Roland, 2007), are required.

This paper describes the TJ^{2}aEM algorithm, a targeted aggressive extrapolation method with no stability problem. Unlike previous targeted extrapolation methods, TJ^{2}aEM rotates its extrapolations to cover all dimensions and applies double extrapolation that proves to stabilize the impact of aggressive extrapolation on fast dimensions. As a result, extrapolations can be made very aggressive to achieve substantial acceleration for the EM algorithm. Experimental results show that TJ^{2}aEM extrapolates more aggressively and converges faster than other acceleration methods. In many cases, two- to three-fold or even higher speedup over other acceleration methods can be achieved for a wide variety of probabilistic models under different data distributions and initial conditions. Furthermore, since TJ^{2}aEM is derived from the fixed-point iteration and Aitken acceleration, TJ^{2}aEM can be directly applied to all of the bound optimization methods defined in (Salakhutdinov and Roweis, 2003), including EM, generalized iterative scaling, non-negative matrix factorization and concave-convex procedure.

This paper is organized as follows. Section 2 reviews pEM and aEM and their convergence properties. Section 3 reviews the TJEM algorithm, which serves as the baseline algorithm for us to derive TJ^{2}aEM. From Section 4 to 6, we describe our step by step derivation of TJ^{2}aEM. Section 4 describes the TJpEM algorithm, which substitutes the EM mapping in TJEM with the pEM mapping. We identify conditions when TJpEM will outperform TJEM, but the conditions also imply that when we choose a large learning rate, TJpEM may converge slower than TJEM. Our solution to the issue is described in Section 5 and is materialized in the TJ^{2}pEM algorithm. The idea is to apply the double extrapolation method to stabilize the ill effect due to a large learning rate. Finally, in Section 6, we describe the TJ^{2}aEM

algorithm, a variant of TJ^{2}pEM with dynamically adjusted learning rates. We show that dynamically adjusting the learning rate can outperform sticking with a fixed optimal learning rate, which also explains why aEM can outperform pEM. We present experimental verifications of our analytical results and report experimental comparisons of the acceleration performance of the above algorithms in Section 7. In the last section, we summarize the conclusions.

2 Accelerating EM by Extrapolation

This section reviews the pEM algorithm and the aEM algorithm. We also present a generic algorithmic framework that can integrate many extrapolation based variants of EM while guarantee convergence. Meanwhile, we introduce the notation in this paper. See Appendix A for the complete notation convention used.

2.1 Parameterized EM

The EM algorithm updates a given parameter vector of a probabilistic model with the guarantee that the data likelihood will be monotonically increased. Let Ω be a parameter space of a probabilistic model, and θ be a n-dimensional parameter vector over Ω. An EM mapping $M : \Omega \to \Omega$ ensures that $L(M(\theta)) \geq L(\theta)$. Starting from an initial parameter, say $\theta^{(0)}$, the EM algorithm applies M to $\theta^{(0)}$ iteratively until convergence. Let $\theta^{(t)}$ denote the output of EM at iteration t, we have $\theta^{(t)} = M(\theta^{(t-1)}) = \cdots = M^t(\theta^{(0)})$, where $M^t(\theta^{(0)})$ denotes applying the EM mapping to $\theta^{(0)}$ for t times. We will abbreviate $M(\theta^{(t-1)})$ as $\theta_{EM}^{(t-1)}$. Note that $\theta^{(t)}$ and $\theta_{EM}^{(t-1)}$ are the same in the EM algorithm, but may be different in EM variants. When $t \to \infty$, the EM algorithm converges to a local optimum θ^* that satisfies $\theta^* = M(\theta^*)$.

The parameterized EM (pEM) algorithm (Bauer et al., 1997) accelerates the EM algorithm by using the pEM mapping $M_\eta : \Omega \to \Omega$ at each iteration:

$$M_\eta(\theta) \equiv \theta + \eta(M(\theta) - \theta). \tag{1}$$

That is, pEM extrapolates along the direction from θ to $M(\theta)$ with a learning rate η. Then, the parameter vector at iteration t of pEM is:

$$\theta^{(t)} = M_\eta(\theta^{(t-1)}) = \theta^{(t-1)} + \eta(\theta_{EM}^{(t-1)} - \theta^{(t-1)}). \tag{2}$$

Similar to EM, we abbreviate $M_\eta(\theta^{(t-1)})$ as $\theta_\eta^{(t-1)}$. If $\eta = 1$, $\theta_\eta^{(t-1)}$ is equivalent to $\theta_{EM}^{(t-1)}$.

The choice of η affects the rate of convergence of pEM. We summarize some convergence properties of EM and pEM in Section 2.2 to show the acceleration of EM by pEM.

2.2 Convergence Properties of EM and pEM

Suppose that we apply the EM algorithm from $\theta^{(t)}$ in the neighborhood of θ^* and EM converges at θ^*. Assuming that the EM mapping M is differentiable. Then we can apply a linear Taylor expansion of M around θ^* so that

$$\theta^{(t+1)} = M(\theta^{(t)}) \approx \theta^* + M'(\theta^*)(\theta^{(t)} - \theta^*) = \theta^* + J(\theta^{(t)} - \theta^*), \tag{3}$$

where J abbreviates $M'(\theta^*)$, the Jacobian of EM at θ^*. We can apply M to $\theta^{(t)}$ consecutively for h times to obtain $\theta^{(t+h)}$. From Equation (3), we have

$$\theta^{(t+h)} \approx \theta^* + J^h(\theta^{(t)} - \theta^*). \tag{4}$$

The eigen decomposition of the Jacobian J at θ^* is

$$J = Q \begin{pmatrix} \lambda_1 & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & \lambda_n \end{pmatrix} Q^{-1} = Q\Lambda Q^{-1}, \tag{5}$$

where $Q = [v_1, \dots, v_n]$ contains the eigenvectors corresponding to eigenvalues $\lambda_1, \dots, \lambda_n$, respectively. Then, J^h in Equation (4) becomes:

$$J^h = Q \begin{pmatrix} \lambda_1^h & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & \lambda_n^h \end{pmatrix} Q^{-1} = Q\Lambda^h Q^{-1}.$$

Since $\theta^{(t+h)} \to \theta^*$ when $h \to \infty$ in EM, it is required that $\lim_{h\to\infty} J^h = 0$ to ensure convergence. It follows that $\lim_{h\to\infty} \lambda_i^h = 0$, and thus, $-1 < \lambda_i < 1$ for all i.

The rate of convergence of M is determined by the largest eigenvalue of J, which is the slowest one among all eigenvalues to converge to 0. More generally, the rate is determined by the spectral radius ρ of J when the eigenvalues can be negative. The spectral radius ρ is defined by $\max\{|\lambda_{max}|, |\lambda_{min}|\}$, where λ_{max} and λ_{min} are the greatest and smallest eigenvalues of J. In previous works, the following assumption on the eigenvalues of J is usually expected to be true, which implies that $\rho = \lambda_{max}$:

Assumption 1 *The eigenvalues of the Jacobian of an EM mapping lie in* $[0,1)$*(Dempster et al., 1977; McLachlan and Krishnan, 1997).*

The convergence rate of pEM can be expressed in terms of the eigenvalues of EM. The relation between the eigenvalues of EM and pEM can be derived by differentiating $M_\eta(\theta^*)$ with respect to θ. By Equation (1), we have

$$\begin{aligned} J_\eta \equiv M'_\eta(\theta^*) &= I + \eta(M'(\theta^*) - I) \\ &= (1-\eta)I + \eta J \\ &= Q[(1-\eta)I + \eta\Lambda]Q^{-1}, \end{aligned} \tag{6}$$

where J_η denotes the Jacobian of M'_η at θ^*, and I the $n \times n$ identity matrix. We describe the results from Equation (6) in Lemma 1.

Lemma 1. *The i-th eigenvalue of J_η, denoted by $\lambda_{\eta i}$, is a linear combination of 1.0 and λ_i, the eigenvalue of J (Hesterberg (2005), for example):*

$$\lambda_{\eta i} = (1-\eta) * 1.0 + \eta \lambda_i. \tag{7}$$

Besides, the eigenvectors of J and J_η are the same.

EM and pEM have the same set of local optima. Although EM and pEM share the same θ^*, not every θ^* can be reached. This is also a convergence issue of the fixed point iteration methods (Burden and Faires, 1988).

Next, we derive the range of η that ensures convergence of pEM based on Lemma 1. The range is determined by the minimal eigenvalue of J_η.

Proposition 1. *Let $\lambda_{\eta max}$ and $\lambda_{\eta min}$ denote the maximal and minimal eigenvalue of J_η. The pEM algorithm with η converges if $0 < \eta < \frac{2}{1-\lambda_{min}}$.*

The Jacobian is not the same for different local maxima, neither are the eigenvalues. For each θ^* that EM can converge to, pEM with a large learning rate η might only converge to some of them. The next corollary describes a strict range for η such that convergence is guaranteed for pEM.

Proposition 2. *Within the neighborhood of θ^*, the pEM algorithm must converge to θ^* if $0 < \eta < 2$.*

The bound in Proposition 2 is too tight because pEM might converge with $\eta > 2$ when $\lambda_{min} > 0$. For example, suppose that the smallest eigenvalue among all Jacobians is 0.1, the upper bound can be relaxed to 2.22, greater than the upper bound given in Proposition 2.

Proposition 3. *The optimal learning rate η^* for pEM is $\frac{2}{2-\lambda_{max}-\lambda_{min}}$(Hammerlin and Hoffmann, 1991).*

Although the mapping and Jacobian of pEM can be expressed via those of EM, we still assign new symbols like M_η and J_η for pEM. That is because pEM is also a fixed point method, and we can directly use M_η to replace M in Aitken acceleration.

2.3 Adaptive Overrelaxed EM

An example of dynamic learning rate methods is the adaptive overrelaxed EM (aEM) algorithm(Salakhutdinov and Roweis, 2003). The aEM algorithm dynamically adjusts the learning rate in pEM by $\eta^{(t)} = 1.1\eta^{(t-1)}$ for iteration t if the likelihood is increased more than a threshold at iteration $t-1$. Otherwise, aEM resets $\eta^{(t+1)} = 1.0$ to guarantee convergence.

The motivation of aEM is that dynamically adjusting η increases the chance of using the optimal learning rate of pEM. In fact, our experimental results in Section 7.6 show that aEM is superior than pEM with even the optimal learning rate because of aggressive extrapolation.

The staggered EM is another aggressive extrapolation method proposed by Hesterberg (2005). The difference of staggered EM and aEM is that, staggered EM estimates the possible range of η every ten iterations, and then choose ten values within the range as the next ten learning rates. We have empirically compared staggered EM and aEM, and found that they are competing with each other. Therefore, we will only use aEM as the representative in our experiments.

2.4 Backtracking for Convergence Guarantee

Accelerating EM by extrapolation with dynamically adjusted learning rates is not guaranteed to converge because the learning rates may exceed the upper bound given in Proposition 1. To ensure convergence, the aEM algorithm applies a simple yet effective method, which drops the resulting parameter vector of the extrapolation $\theta^{(t+1)}$ if it fails to improve the likelihood and replaces it with $\theta_{EM}^{(t)}$, the result obtained by the original EM algorithm. Since we must obtain $\theta_{EM}^{(t)}$ in order to compute the extrapolation, this method incurs tiny overheads while achieves monotone increasing of the likelihood. Therefore, the aEM algorithm is guaranteed to converge.

This backtracking method can be generalized to integrate many EM variants based on extrapolation. The aEM algorithm can be considered as integrating two EM variants, EM (for abbreviation, EM variants include the original EM algorithm) and pEM with dynamic learning rates (Salakhutdinov and Roweis, 2003). The pEM algorithm is the default approach. When pEM fails to improve the likelihood, the result of the EM algorithm is used instead.

When there are many EM variants, we can use Algorithm 1, which searches for a parameter vector that satisfies the condition of monotone increasing of the likelihood. Suppose that there are K variants that generate K candidates $\hat{\theta}_1^{(t)} \dots \hat{\theta}_K^{(t)}$ at iteration t. Candidates are ordered by the aggressiveness of their extrapolation methods. $\hat{\theta}_K^{(t)}$ with the largest K is always generated by the original EM mapping because it is the least aggressive one. Then, the likelihood $L(\hat{\theta}_k^{(t)})$ for each $\hat{\theta}_k^{(t)}$ is computed one by one with the E-step in the order of k. The first $\hat{\theta}_k^{(t)}$ that successfully increases the likelihood more than a threshold δ becomes the final $\theta^{(t)}$, and the M-step is performed to compute $\theta_{EM}^{(t)}$. At last, $\theta^{(0)}, \dots \theta^{(t)}$ and $\theta_{EM}^{(t)}$ are used to generate $\hat{\theta}_1^{(t+1)} \dots \hat{\theta}_K^{(t+1)}$, the candidates for the next iteration. It is clear that Algorithm 1 is guaranteed to converge because it ensures monotone increasing of the likelihood.

The worst case computational cost of Algorithm 1 is $O(TKE)$, where T is the iteration times, K is the number of EM variants, and E is the computational cost to

compute the extrapolation and perform an E-step. In the best case, the most aggressive extrapolation always leads to a parameter vector that improves the likelihood and thus at each iteration, only one E-step will be performed. In the worst case, however, all extrapolation methods fail to improve the likelihood and the algorithm reduces to the original EM algorithm. In this case, K extrapolations and K E-steps will be performed for each extrapolation method. Therefore, integrating an additional variant must be justified by its effectiveness of reducing the number of required iterations, that is, improving the convergence rate. Usually we only consider to have $K \leq 3$. The triple jump methods to be presented in this paper will be integrated with EM or pEM in Algorithm 1 as step 5 to generate new parameter vectors by extrapolation.

Algorithm 1 Integrating EM Variants

1: Randomly initialize $\hat{\theta}_K^{(0)}$, $LL = -\infty$.
2: **repeat** at iteration t (starting from 0)
3: $\theta^{(t)} = \hat{\theta}_k^{(t)}$ with the minimal k such that $L(\hat{\theta}_k^{(t)}) - LL > \delta$ (use E-step to compute the likelihood).
4: $LL = L(\hat{\theta}_k^{(t)})$, and use M-step to compute $\theta_{EM}^{(t)}$.
5: Generate $\hat{\theta}_1^{(t+1)}, \ldots, \hat{\theta}_K^{(t+1)}$ from $\theta^{(0)}, \ldots, \theta^{(t)}$ and $\theta_{EM}^{(t)}$ for the next iteration.
6: **until** no $\hat{\theta}_k^{(t)}$ satisfies $L(\hat{\theta}_k^{(t)}) - LL > \delta$.

3 TJEM: Baseline Targeted Aggressive Extrapolation Algorithm

In this section, we start by reviewing Aitken acceleration for the EM algorithm, then we present the triple jump EM method (TJEM) as the baseline method to be improved. Note that TJEM is not brand-new: it follows the work of Schafer (1997) and Hesterberg (2005) for targeted extrapolation, and follows Algorithm 1, the generic algorithmic framework of Salakhutdinov and Roweis (2003), to discard estimates that fails to improve enough likelihood. We named this method as *triple jump* because its search path is similar to the hop, step and jump phases in triple jump.

3.1 Aitken Acceleration for EM

The EM algorithm is equivalent to solving θ^* by a fixed-point iteration method (Burden and Faires, 1988). That is, EM looks for a parameter vector that satisfies $\theta = M(\theta)$ by iteratively substituting θ on the RHS with that on the LHS until convergence. Therefore, we can use the Aitken acceleration method to speed up EM.

The multivariate version of Aitken acceleration (Louis, 1982; McLachlan and Krishnan, 1997) for EM can be derived as follows. We can express θ^* as

$$\theta^* = \theta^{(t)} + \sum_{h=0}^{\infty} (\theta^{(t+h+1)} - \theta^{(t+h)}). \tag{8}$$

Suppose that $\theta^{(t)}$ is in the neighborhood of θ^*. Based on Equation (4), $\theta^{(t+h+1)} - \theta^{(t+h)}$ can be written as:

$$\theta^{(t+h+1)} - \theta^{(t+h)} \approx \Big[\theta^* + J^h(\theta^{(t+1)} - \theta^*)\Big] - \Big[\theta^* + J^h(\theta^{(t)} - \theta^*)\Big] = J^h(\theta^{(t+1)} - \theta^{(t)}). \tag{9}$$

Applying Equation (9) in Equation (8) gives the multivariate Aitken acceleration:

$$\begin{aligned} \theta^* &\approx \theta^{(t)} + \sum_{h=0}^{\infty} J^h(\theta^{(t+1)} - \theta^{(t)}) \\ &= \theta^{(t)} + (I-J)^{-1}(\theta_{EM}^{(t)} - \theta^{(t)}), \end{aligned} \tag{10}$$

since the eigenvalues of J are between 0 and 1, from Assumption 1. In Equation (10), we replace $\theta^{(t+1)}$ with $\theta_{EM}^{(t)}$ to emphasize that $\theta^{(t+1)}$ is obtained by applying an EM mapping to $\theta^{(t)}$ here.

The multivariate version of Aitken acceleration requires to compute the Jacobian of the EM mapping matrix, which can be intractable for complex models with a high dimensional parameter space. Aitken acceleration also has drawbacks including that it may not always converge and that it may be numerically unstable (Jamshidian and Jennrich, 1997).

3.2 Estimating Eigenvalues of Slowest Dimension

Targeted aggressive extrapolation updates a new estimate based on previous EM estimates. We start from Equation (10) of Aitken acceleration. We substitute J with Equation (5), and then $(I-J)^{-1}$ in Equation (10) becomes:

$$\begin{aligned} (I-J)^{-1} &= \left[Q[I-\Lambda]Q^{-1}\right]^{-1} \\ &= Q[I-\Lambda]^{-1}Q^{-1} \\ &= Q \begin{pmatrix} \frac{1}{1-\lambda_1} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & \frac{1}{1-\lambda_n} \end{pmatrix} Q^{-1}. \end{aligned} \tag{11}$$

With the eigen decomposition of J, we can map θ^* in Equation (10) from the original parameter space to the eigenspace spanned by Q:

$$\begin{aligned}\psi^* = Q^{-1}\theta^* &\approx Q^{-1}\theta^{(t)} + Q^{-1}(I-J)^{-1}(\theta_{EM}^{(t)} - \theta^{(t)}) \\ &= Q^{-1}\theta^{(t)} + [I-\Lambda]^{-1}Q^{-1}(\theta_{EM}^{(t)} - \theta^{(t)}) \\ &= \psi^{(t)} + [I-\Lambda]^{-1}(\psi_{EM}^{(t)} - \psi^{(t)}).\end{aligned}$$

The relation between ψ^* and θ^* can also be written as:

$$\theta^* = \psi_1^* v_1 + \cdots + \psi_n^* v_n,$$

where ψ_i^* with $i = 1,\ldots,n$ denotes the i-th transformed parameter vector ψ^* in the eigenspace. Along with Equation (11), we can observe that the multivariate Aitken acceleration is in fact a series of univariate Aitken acceleration along the direction of v_i:

$$\psi_i^* \approx \psi_i^{(t)} + \frac{1}{1-\lambda_i}(\psi_{EMi}^{(t)} - \psi_i^{(t)}). \tag{12}$$

The global rate of convergence of the EM algorithm is defined as the limit of the ratio of the difference between the next estimated parameter vector to the local maximum and the difference between the current estimated parameter vector to the local maximum:

$$R = \lim_{t\to\infty} R^{(t)} \equiv \lim_{t\to\infty} \frac{\|\theta^{(t+1)} - \theta^*\|}{\|\theta^{(t)} - \theta^*\|} \tag{13}$$

Dempster et al. (1977) have shown that $R = \lambda_{max}$, the largest eigenvalue of J. Therefore, instead of computing the Jacobian, we can simplify Aitken acceleration for EM by replacing every eigenvalue λ_i with a single value $\gamma^{(t)}$ such that $\gamma^{(t)}$ is an approximation of λ_{max} at the t-th iteration:

$$\theta^{(t+1)} = \theta^{(t)} + (1-\gamma^{(t)})^{-1}(\theta_{EM}^{(t)} - \theta^{(t)}). \tag{14}$$

Note that $(1-\gamma^{(t)})^{-1}$ in Equation (14) can be written as $Q\,\mathrm{diag}(1-\gamma^{(t)})^{-1}Q^{-1}$. Compared with Equation (11), we can observe that the extrapolation assumes $\lambda_i = \gamma^{(t)}$ for all i and performs Aitken acceleration accordingly.

We can estimate $\gamma^{(t)}$ as follows. Let $\theta^{(t)} = M(\theta^{(t-1)})$ and $\theta^{(t+1)} = M(\theta^{(t)})$. We have, by Equation (9):

$$J(\theta^{(t)} - \theta^{(t-1)}) \approx \theta^{(t+1)} - \theta^{(t)} = \theta_{EM}^{(t)} - \theta^{(t)}.$$

Then, we substitute J with $\gamma^{(t)}$. Let $\gamma^{(t)}(\theta^{(t)} - \theta^{(t-1)}) = \theta_{EM}^{(t)} - \theta^{(t)}$, we have

$$|\gamma^{(t)}| = \frac{\|\theta_{EM}^{(t)} - \theta^{(t)}\|}{\|\theta^{(t)} - \theta^{(t-1)}\|}. \tag{15}$$

Since λ_i's are non-negative in the EM algorithm by Assumption 1, our estimation of $\gamma^{(t)}$ is defined by (Hesterberg, 2005):

$$\gamma^{(t)} \equiv \frac{\|\theta_{EM}^{(t)} - \theta^{(t)}\|}{\|\theta^{(t)} - \theta^{(t-1)}\|}. \tag{16}$$

$\gamma^{(t)}$ is primarily dominated by the eigenvalues of the slowest dimension in terms of distance to ψ^*. It can be shown that as $t \to \infty$, $\gamma^{(t)} \leq \lambda_{max}$ asymptotically in the neighborhood of θ^*(Hesterberg, 2005). However, when $\theta^{(t)}$ is not in the neighborhood of θ^*, $\gamma^{(t)}$ could be greater than 1.0 and results in unstable extrapolation. We will show how to handle it in Section 7.1.

3.3 The TJEM algorithm

The instantiation of Algorithm 1 that integrates the estimation of γ (i.e., Equation (14)) and the original EM algorithm will be referred to as *the TJEM algorithm*. Other variants to be described later in this paper will be named in a similar manner.

Since our estimation applies two previous estimates of the EM algorithm to obtain $\gamma^{(t)}$, the TJEM algorithm invokes Equation (16) at every other iteration, if all extrapolations successfully improve the likelihood. Starting from $\theta^{(0)}$ in the neighborhood of θ^*, we need to apply EM to obtain $\theta_{EM}^{(0)} = \theta^{(1)}$, again to obtain $\theta_{EM}^{(1)}$, and then we can apply Equation (14), the triple jump extrapolation, to obtain $\theta^{(2)}$. To apply the extrapolation again, we cannot simply use $\theta^{(1)}$, $\theta^{(2)}$ and $\theta_{EM}^{(2)}$ in Equation (16) to obtain $\gamma^{(2)}$, because in Equation (16), $\theta^{(t)}$ must be obtained by the EM algorithm too so that the ratio is a reasonable estimate of the eigenvalue. Therefore, to apply the extrapolation again, we need to apply EM to obtain $\theta_{EM}^{(2)} = \theta^{(3)}$, again to obtain $\theta_{EM}^{(3)}$, and then we can apply the extrapolation to obtain $\theta^{(4)}$, and so on. Therefore, the TJEM algorithm applies the extrapolation at the $2i$-th iteration, $i = 1, 2, \ldots$, assuming that all extrapolations successfully improve the likelihood.

If the improvement is less than a threshold, the extrapolation result will be discarded and the result of EM will be used as the current estimate $\theta^{(t)}$. The triple jump extrapolation can be resumed using $\theta^{(t-1)}$, $\theta^{(t)}$ and $\theta_{EM}^{(t)}$, or postponed for another iteration.

4 TJpEM: Accelerating TJEM by pEM Mapping

One idea to accelerate TJEM is that, since TJEM performs EM mapping and targeted extrapolation by turns, we may have faster convergence if the EM mapping is replaced by the pEM one because pEM can converge faster than EM. The idea can be implemented by replacing M with M_η in the derivation of TJEM, resulting in the triple jump parameterized EM (TJpEM) algorithm.

Let $\theta_\eta^{(t)} \equiv M_\eta(\theta^{(t)})$. Following the derivation in Section 3.2, we have the estimation of the eigenvalue $\gamma_\eta^{(t)}$ for the slowest dimension based on the pEM mapping:

$$\gamma_\eta^{(t)} \equiv \frac{\|\theta_\eta^{(t)} - \theta^{(t)}\|}{\|\theta^{(t)} - \theta^{(t-1)}\|}. \tag{17}$$

In this way, we obtain the targeted extrapolation in TJpEM as:

$$\theta^{(t+1)} = \theta^{(t)} + (1-\gamma_\eta^{(t)})^{-1}(\theta_\eta^{(t)} - \theta^{(t)}). \tag{18}$$

The TJpEM algorithm is an instantiation of Algorithm 1 described in Section 2.4, with $K = 3$ EM variants. In the TJpEM algorithm, $\hat{\theta}_1^{(t)}$ is computed by the targeted extrapolation, $\hat{\theta}_2^{(t)}$ by pEM with a fixed learning rate η, and $\hat{\theta}_3^{(t)}$ by the original EM algorithm.

4.1 Convergence Properties of TJpEM

During its execution, the TJpEM algorithm usually switches between the targeted extrapolation and pEM extrapolation. Therefore, the convergence properties of TJpEM are determined by the Jacobian of the composition of the two mappings at θ^*:

$$M'_{\gamma_\eta}(M_\eta(\theta^*))M'_\eta(\theta^*) = M'_{\gamma_\eta}(\theta^*)M'_\eta(\theta^*) = J_{\gamma_\eta}J_\eta.$$

Since $J_{\gamma_\eta} = Q\Lambda_{\gamma_\eta}Q^{-1}$ and $J_\eta = Q\Lambda_\eta Q^{-1}$, we have $J_{\gamma_\eta}J_\eta = Q\Lambda_{\gamma_\eta}\Lambda_\eta Q^{-1}$ and the eigenvalues are the diagonal elements of $\Lambda_{\gamma_\eta}\Lambda_\eta$. Lemma 2 gives the eigenvalues in $J_{\gamma_\eta}J_\eta$.

Lemma 2. *The i-th eigenvalue of the Jacobian of $M_{\gamma_\eta} \circ M_\eta$ at θ^* with estimated spectral radius $\gamma_\eta^{(t)}$ is*

$$\lambda_{\eta i}\frac{\lambda_{\eta i} - \gamma_\eta^{(t)}}{1-\gamma_\eta^{(t)}}.$$

To compare the spectral radii of TJEM and TJpEM, we assume that Equation (7), the relation between the eigenvalues for the Jacobians of the EM and pEM mappings, holds for the estimated spectral radii $\gamma^{(t)}$ and $\gamma_\eta^{(t)}$:

$$\gamma_\eta^{(t)} = 1 - \eta + \eta\gamma^{(t)}.$$

Now, consider a Jacobian of the EM mapping with 19 distinct eigenvalues λ_i, $i = 1,\ldots,19$. Assume further that $\lambda_i = 0.05 * i$. It follows that $\lambda_{min} = 0.05$ and $\lambda_{max} = 0.95$. Suppose TJEM estimates λ_{max} as $\gamma^{(t)} = 0.83$, an inaccurate approximation. Then according to Equation (7), if we choose $\eta = 1.2$ for TJpEM, we will have $\lambda_{\eta min}$, $\lambda_{\eta max}$, and $\gamma^{(t)}$ to be -0.14, 0.94, and 0.796, respectively, and -0.52, 0.92,

and 0.728, respectively, if we choose $\eta = 1.6$. Figure 1(a) illustrates the absolute eigenvalues of TJEM and TJpEM with these different learning rates. We can clearly observe the tendency that, with the growth of η,

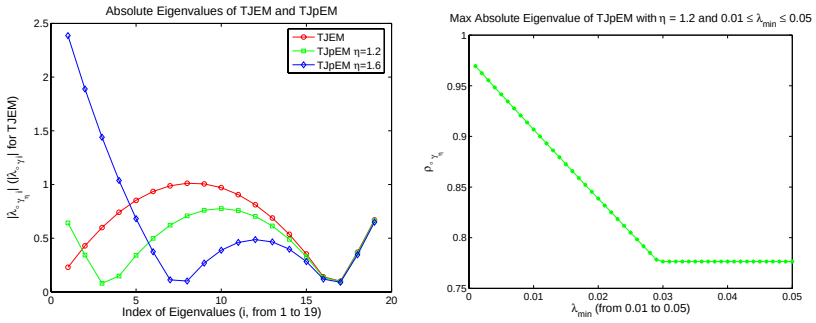
- the peak of the concave curves in the middle in Figure 1(a) decreases gradually, and
- the end of the left tails in Figure 1(a) increases drastically.

Figure 1(a) illustrates that TJpEM can converge faster TJEM with a proper learning rate (e.g. $\eta = 1.2$), while can converge slower or even diverge with a large (e.g. $\eta = 1.6$).

Then, we change λ_{min} and keep the other eigenvalues unchanged to see how sensitive the spectral radius is to λ_{min} and plot the result in Figure 1(b), which shows that when $\lambda_{min} < 0.03$, the spectral radius increases linearly as λ_{min} decreases. The result shows that the spectral radius could also be influenced by tiny difference of λ_{min}.

At last, we derive an upper bound of η for TJpEM to converge faster than TJEM.

Proposition 4. *Within the neighborhood of θ^*, TJpEM with $\eta < \frac{1+\frac{\gamma}{4}}{1-\lambda_{min}}$ can converge faster than TJEM, under the assumption that $\gamma_\eta^{(t)} = 1 - \eta + \eta\gamma^{(t)}$.*



(a) Composite absolute eigenvalues of TJEM and TJpEM

(b) Composite spectral radii of TJpEM with $\eta = 1.2$ and different λ_{min}

Fig. 1 We assume that $\lambda_i = 0.05 * i, i = 1, \ldots, 19$. Accordingly, we plot the composite absolute eigenvalues of the extrapolation in TJpEM in (a), where the peak value of a curve is the spectral radius. (a) shows that in this example, TJpEM has a smaller composite spectral radius than TJEM with $\eta = 1.2$, but a larger one with $\eta = 1.6$. Then, we change λ_{min} and keep the other eigenvalues unchanged to see how sensitive the spectral radius $\rho_{\circ\gamma_\eta}$ is to λ_{min} and plot the result in (b), which shows that when $\lambda_{min} < 0.03$, $\rho_{\circ\gamma_\eta}$ increases linearly as it decreases.

4.2 Impact of Negative $\lambda_{\eta min}$ against TJpEM

Proposition 4 implies that TJpEM converges faster than TJEM with a proper learning rate, but when the learning rate exceeds the proper range, TJpEM might converge slower. Besides, λ_{min} can also influence the spectral radius, especially when $\lambda_{min} \to 0$. This is usually because a large learning rate and a tiny minimal eigenvalue may result in a negative eigenvalue $\lambda_{\eta min}$ for pEM.

A negative eigenvalue brings impact against TJpEM, especially when the value is less than -1. In this case, the extrapolation may bring the search away from the local optimum and fail to improve the likelihood. Recall that to guarantee convergence, Algorithm 1 will discard an extrapolation result if it fails to improve the likelihood. If this occurs often, the rate of convergence will suffer. Consequently, the impact of negative $\lambda_{\eta min}$ against TJpEM is undesirable. A solution for TJpEM is to use a small η. However, it is difficult to determine how to adjust η accordingly and this conservative solution may rarely produce any significant acceleration. Another solution, which will be described in the next section, is the TJ^{2}pEM algorithm. TJ^{2}pEM ensures that all eigenvalues including the minimal one are non-negative, and thus alleviates the impact of negative eigenvalues against TJpEM.

5 TJ^{2}pEM: Stabilizing TJpEM by Double Extrapolation

The TJ^{2}pEM algorithm applies the double extrapolation method that combines two pEM extrapolations into one to prevent the Jacobian from having any negative eigenvalue. The key idea is that, when our mapping is M_η^2, its Jacobian J_η^2 will be $Q\Lambda_\eta^2 Q^{-1}$ and Λ_η^2 will contain no negative eigenvalue. Consequently, the deviation between the real and estimated eigenvalues would be smaller.

Similar to Equation (8), we substitute the EM mapping with M_η, but apply M_η twice at a time for every current estimate to obtain:

$$\begin{aligned} \theta^* &= \theta^{(t-1)} + \sum_{h'=0}^{\infty} (M_\eta^{2h'+2}(\theta^{(t-1)}) - \theta^{(t-1)}) \\ &= \theta^{(t-1)} + \sum_{h'=0}^{\infty} (\theta^{(t+2h'+1)} - \theta^{(t+2h'-1)}). \end{aligned} \tag{19}$$

Note that the superscript t still indicates the number of iterations that pEM has been applied.

Based on Equation (4), $\theta^{(t+2h'+1)} - \theta^{(t+2h'-1)}$ can be written as:

$$\theta^{(t+2h'+1)} - \theta^{(t+2h'-1)} \approx J_\eta^{2h'}(\theta^{(t+1)} - \theta^{(t-1)}).$$

Substituting Equation (19) with the above approximation, we have:

$$\begin{aligned}\theta^* &= \theta^{(t-1)} + \sum_{h'=0}^{\infty} (\theta^{(t+2h'+1)} - \theta^{(t+2h'-1)}) \\ &\approx \theta^{(t-1)} + \sum_{h'=0}^{\infty} J^{2h'} (\theta^{(t+1)} - \theta^{(t-1)}) \\ &= \theta^{(t-1)} + (I - J_\eta^2)^{-1} (\theta^{(t+1)} - \theta^{(t-1)}) \\ &= \theta^{(t-1)} + (I - J_\eta^2)^{-1} (\theta_\eta^{(t)} - \theta^{(t-1)})\end{aligned} \tag{20}$$

Still, we use $\gamma_\eta^{(t)}$ as in TJpEM to approximate J_η and obtain the extrapolation in TJ^{2}pEM as follows:

$$\theta^{(t+1)} = \theta^{(t-1)} + \frac{1}{1 - (\gamma_\eta^{(t)})^2} (\theta_\eta^{(t)} - \theta^{(t-1)}). \tag{21}$$

Note that instead of extrapolating from $\theta^{(t)}$, TJ^{2}pEM extrapolates from $\theta^{(t-1)}$ at the t-th iteration.

5.1 Convergence Properties of TJ^{2}pEM

The extrapolation of TJ^{2}pEM is quite different from that of TJpEM. At iteration t where $\theta^{(t)} = \theta_\eta^{(t-1)}$, the extrapolation in TJpEM uses $\theta^{(t-1)}$, $\theta^{(t)}$, and $\theta_\eta^{(t)}$ to estimate $\gamma_\eta^{(t)}$, and then extrapolates from $\theta^{(t)}$ along $\theta_\eta^{(t)} - \theta^{(t)}$. The mapping in TJpEM from $\theta^{(t-1)}$ to $\theta^{(t+1)}$ is a pEM mapping to $\theta^{(t)}$ and a triple jump extrapolation to $\theta^{(t+1)}$. TJ^{2}pEM as given in Equation (21) estimates $\gamma_\eta^{(t)}$ in the same way as TJpEM, but it extrapolates from $\theta^{(t-1)}$ along the direction of $\theta_\eta^{(t)} - \theta^{(t-1)}$. The mapping in TJ^{2}pEM is a direct mapping from $\theta^{(t-1)}$ to $\theta^{(t+1)}$. In other words, if the extrapolation is a success, we consider it as one update that takes two iterations.

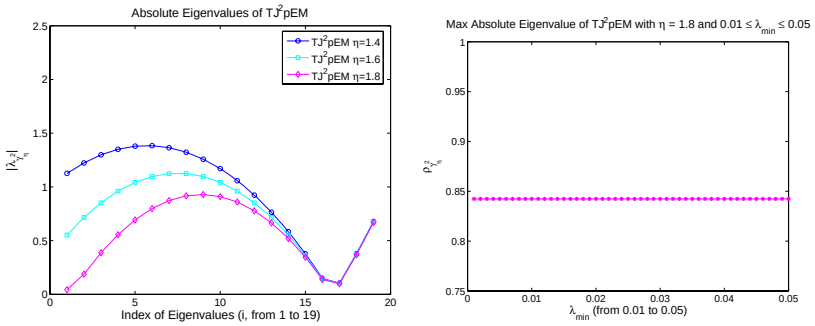
The next lemma gives the eigenvalues of the Jacobian of the TJ^{2}pEM mapping.

Lemma 3. *The i-th eigenvalue $\lambda_{\gamma_\eta^2 i}$ of the Jacobian of the TJ^{2}pEM mapping is:*

$$\lambda_{\gamma_\eta^2 i} = \frac{(\lambda_{\eta i})^2 - (\gamma_\eta^{(t)})^2}{1 - (\gamma_\eta^{(t)})^2}.$$

We use the same example in Section 4.1 to show the eigenvalues of TJ^{2}pEM with different learning rates η in Figure 2(a). We can clearly observe that in this example, with the growth of η,

- $\lambda_{\eta min}$ has a much less impact on the spectral radius than that of TJpEM, where the spectral radius of the Jacobian of the TJpEM mapping may increase drastically due to a small $\lambda_{\eta min}$;



(a) Absolute eigenvalues of TJ^{2}pEM

(b) Spectral radii of TJ^{2}pEM with $\eta = 1.8$ and different λ_{min}

Fig. 2 With the same example as in Figure 1, (a) shows the absolute eigenvalues of TJ^{2}pEM with $\eta = 1.4$, 1.6, and 1.8. The curves have no left tail and look like the curve of TJEM in Figure 1. (b) shows that changes to λ_{min} will not affect the spectral radius $\rho_{\gamma_\eta^2}$ here, suggesting that TJ^{2}pEM is barely affected by λ_{min}.

- TJ^{2}pEM with a large η tends to have a smaller spectral radius than that with a small η, in contrast of TJpEM.

At last, we derive the conditions under which TJ^{2}pEM can converge faster than TJpEM.

Proposition 5. *Given the same learning rate η, TJ^{2}pEM can converge faster than TJpEM if $\lambda_{\eta min} < -\frac{1}{2}$ and $\lambda_{\eta max} \leq \frac{1+\sqrt{2}}{2}\gamma_\eta^{(t)}$.*

5.2 Elimination of Impact of Negative $\lambda_{\eta min}$

Proposition 5 and Figure 2(a) suggest that TJ^{2}pEM will successfully alleviate the impact of negative eigenvalues $\lambda_{\eta min}$ of TJpEM due to a large learning rate η. Another factor that influences $\lambda_{\eta min}$ is λ_{min}, as described in Section 4.2. Here we discuss how λ_{min} may affect $\lambda_{\eta min}$ with an example.

Figure 2(b) plots the spectral radius of the Jacobian of TJ^{2}pEM with $\eta = 1.8$ in the same way as Figure 1(b). We can observe that, when λ_{min} decreases from 0.05 to 0.01, the spectral radius is unchanged. Unlike TJpEM, TJ^{2}pEM is barely affected by the change of λ_{min}. Hence, TJ^{2}pEM will achieve a more stable acceleration performance than TJpEM.

6 TJ^{2}aEM: Accelerating TJ^{2}pEM by Dynamic Learning Rate

The TJ^{2}aEM algorithm is the same as the TJ^{2}pEM except that η in TJ^{2}aEM is dynamically adjusted. Since TJ^{2}pEM is more stable than TJpEM, TJ^{2}pEM is expected to have much less ill effects on the eigenvalues than TJpEM does with dynamic learning rates for M_η. In this section, we explain why dynamic learning rates may help and analyze the convergence properties of TJ^{2}aEM.

6.1 Success of aEM

In this subsection, we use an example to show the advantage of dynamically adjusting the learning rate η over a fixed optimal learning rate η^*. In particular, we will show that using two slightly different learning rates for two consecutive pEM iterations will achieve a higher speedup than using the optimal learning rate η^* at every iteration. Let $\eta^{(1)} = \eta^* + \Delta$ and $\eta^{(2)} = \eta^* - \Delta$, where Δ is an arbitrary constant between 0 and 1. Let M_1 and M_2 be the pEM mappings with the learning rates $\eta^{(1)}$ and $\eta^{(2)}$, respectively. Their Jacobians are $J_1 = Q\Lambda_1 Q^{-1}$ and $J_2 = Q\Lambda_1 Q^{-1}$, respectively. and the eigen matrix of $J_2 J_1$ is $\Lambda_2\Lambda_1$. Hence, the i-th eigenvalue in $J_2 J_1$ is:

$$\begin{aligned}
&(1-\eta^{(1)}+\eta^{(1)}\lambda_i)(1-\eta^{(2)}+\eta^{(2)}\lambda_i)\\
&= (1-\eta^*+\eta^*\lambda_i-\Delta(1-\lambda_i))(1-\eta^*+\eta^*\lambda_i+\Delta(1-\lambda_i))\\
&= (\lambda_{\eta^* i}-\Delta(1-\lambda_i))(\lambda_{\eta^* i}+\Delta(1-\lambda_i))\\
&= (\lambda_{\eta^* i})^2-(\Delta(1-\lambda_i))^2\\
&\leq (\lambda_{\eta^* i})^2.
\end{aligned}$$

Therefore, for each direction along the eigenvector v_i, the eigenvalue of $J_2 J_1$ is smaller than that of $(J_{\eta^*})^2$. It follows that pEM with two different learning rates $\eta^{(1)}$ and $\eta^{(2)}$ may converge faster than pEM with a fixed optimal learning rate η^* in the neighborhood of the local optimum.

We can extend the above result to use several different Δ in turns. Though the optimal learning rate η^* is not known in practice, we can try a wide range of η so that it is virtually equivalent to have several different Δ with η^*. This provides an explanation of why the aEM algorithm (Salakhutdinov and Roweis, 2003) performs better than pEM with η^*.

6.2 Derivation of TJ^{2}aEM

The TJ^{2}aEM algorithm applies dynamic η to TJ^{2}pEM for further acceleration. It is clear that by applying $\eta^{(1)}$ and $\eta^{(2)}$ to Equation (25), we can establish that the spec-

tral radius of TJ^{2}aEM will be smaller than that of TJ^{2}pEM, implying that TJ^{2}aEM may converge faster than TJ^{2}pEM in the neighborhood of the local optimum.

$$
\begin{aligned}
& |\lambda_{\gamma^2_{\eta^{(1)}}i}\lambda_{\gamma^2_{\eta^{(2)}}i}| \\
= & \frac{(\lambda_{\eta^{(1)}i})^2-(\lambda_{\eta^{(1)}\max})^2}{1-(\lambda_{\eta^{(1)}\max})^2}\cdot\frac{(\lambda_{\eta^{(2)}i})^2-(\lambda_{\eta^{(2)}\max})^2}{1-(\lambda_{\eta^{(2)}\max})^2} \\
\propto & \frac{(\lambda_{\eta^{(1)}i}+\lambda_{\eta^{(1)}\max})(\lambda_{\eta^{(2)}i}+\lambda_{\eta^{(2)}\max})}{(1+\lambda_{\eta^{(1)}\max})(1+\lambda_{\eta^{(2)}\max})} \\
= & \frac{(\lambda_{\eta^*i}+\lambda_{\eta^*\max}-\Delta(2-\lambda_i-\lambda_{max}))(\lambda_{\eta^*i}+\lambda_{\eta^*\max}+\Delta(2-\lambda_i-\lambda_{max}))}{(1+\lambda_{\eta^*\max}-\Delta(1-\lambda_{max}))(1+\lambda_{\eta^*\max}+\Delta(1-\lambda_{max}))} \\
= & \frac{(\lambda_{\eta^*i}+\lambda_{\eta^*\max})^2-(\Delta(2-\lambda_i-\lambda_{max}))^2}{(1+\lambda_{\eta^*\max})^2-(\Delta(1-\lambda_{max}))^2} \\
\leq & \frac{(\lambda_{\eta^*i}+\lambda_{\eta^*\max})^2-(\Delta(1-\lambda_{max}))^2}{(1+\lambda_{\eta^*\max})^2-(\Delta(1-\lambda_{max}))^2} \\
\leq & \frac{(\lambda_{\eta^*i}+\lambda_{\eta^*\max})^2}{(1+\lambda_{\eta^*\max})^2}.
\end{aligned}
$$

$|\lambda_{\gamma^2_{\eta^{(1)}}i}\lambda_{\gamma^2_{\eta^{(2)}}i}| = |\lambda_{\gamma^2_{\eta^*}i}|^2$ if $\Delta = 0$. Again, we obtain that $|\lambda_{\gamma^2_{\eta^{(1)}}i}\lambda_{\gamma^2_{\eta^{(2)}}i}|$ with Δ is smaller than $|\lambda_{\gamma^2_{\eta^*}i}|^2$.

TJ^{2}aEM dynamically adjusts η in the range of 1.0 to 2.0 in a zigzag manner to gain further speedup. We choose this range because this is the range that the pEM mapping will always converge. There exist various methods to adjust η. The above method is one of the simplest but works well in our experiments.

7 Experimental Results

This section reports the experimental evaluation of the triple jump acceleration methods by comparing them with aEM. We have also implemented three other algorithms: staggered EM (Hesterberg, 2005), SQUAREM (Berlinet and Roland, 2007), and ε-accelerated EM (Kuroda and Sakakihara, 2006). In our tests on the three algorithms, we found that the performance of staggered EM is similar to aEM, while SQUAREM and ε-accelerated EM usually took longer to converge. Therefore, for the sake of conciseness, it is sufficient for us to show the comparison of our proposed methods with aEM only.

We will compare the numbers of iterations required to converge in our evaluation. More specifically, the number of iterations is the number of times that an E-step is executed, which is the most costly step in EM for the probabilistic models used in our experiments. The ratios of average time for each iteration of EM, TJ^{2}aEM, and

aEM are almost 1 : 1 : 1 after we optimize the codes. Sometimes, EM takes even more time than TJ^{2}aEM and aEM for an iteration. The reasons are

1. the overhead for aggressive extrapolation methods in TJ^{2}aEM and aEM is quite low, and
2. when an estimate fails to improve enough likelihood, TJ^{2}aEM and aEM will discard it and move into the next iteration, without performing the M-step and aggressive extrapolation.

Therefore, it is sufficient to compare the number of iterations to show the speedup in our experiments.

7.1 Implementation of the Triple Jump Extrapolations

The EM variants compared in our experiments include EM, pEM, aEM, TJpEM, TJ^{2}pEM, and TJ^{2}aEM. We will use the number that follows pEM, TJpEM, and TJ^{2}pEM to indicate their learning rates. For example, TJpEM14 stands for the TJpEM algorithm with $\eta = 1.4$. Algorithm 1 described in Section 2.4 is the template of all of these algorithms, with Step 5 instantiated by the corresponding extrapolation methods.

Algorithm 2 shows how we implement the extrapolation methods. It is quite straightforward except that we need to ensure that it is numerically stable. In a difficult EM problem, γ could be close to one and make $\frac{1}{1-\gamma} \to \infty$. An easy fix to this issue is to define an upper bound for γ so that if its value exceeds this upper bound, we will use a value within this upper bound instead. κ in Algorithm 2 is the upper bound of γ. We choose $\kappa = 0.95$ in our experiments. On the other hand, when we have a small γ, its value could be too small to produce any gain by extrapolation. In this case, if γ is less than a constant lower bound κ', we will simply set γ to zero and skip the extrapolation. The lower bound κ' was assigned to 0.5 in our experiments.

The use of κ to avoid unstable extrapolation is quite important in two aspects. First, it avoids aggressive extrapolation with unreasonable learning rates which is very likely to yield worse estimates to be discarded. The consequence is less backtracks and less time in the search. Second, we observed that when γ is greater than we expected, we can usually obtain an estimate with higher likelihood by using a reasonable κ.

Also, we used softmax parameterization for multinomial distributions and Choleski decomposition for covariance matrices to avoid extrapolating to illegal estimates. The stopping condition for all experiments is that the improvement of the likelihood $\delta < 10^{-5}$.

Algorithm 2 Triple Jump Extrapolation

1: **input:** initial estimate θ_a, hop estimate θ_b, and step estimate θ_c
2: **output:** jump estimate θ_d.
3: $\gamma \leftarrow \frac{\|\theta_c-\theta_b\|}{\|\theta_b-\theta_a\|}$;
4: **if** $(\gamma > \kappa)$ **then** $\gamma = \kappa$;
5: **if** $(\gamma < \kappa')$ **then** $\gamma = 0.0$;
6: $\theta_d \leftarrow \theta_a + \frac{1}{1-\gamma^2}(\theta_c - \theta_a)$ for TJ^{2}aEM and TJ^{2}pEM, **or** $\theta_d \leftarrow \theta_b + \frac{1}{1-\gamma}(\theta_c - \theta_b)$ for TJEM and TJpEM.

7.2 Models and Data for Experiments

The experiments were designed to compare different algorithms under the impact of different models, data sets, and initial values. We synthesized 100 data sets with randomly generated initial values from each of the following models:

- Hidden Markov Models (HMM): we considered five-state, 20-symbol HMMs with randomly generated parameters to synthesize training data sets. Each data set contains 500 sequences with an alphabet of 100 symbols.
- Bayesian networks (BN): we used the ALARM model (Cooper and Herskovits, 1992), a large real world Bayesian Network with 37 multinomial nodes. We randomly assigned conditional probabilities and synthesized 2,000 examples for each experimental data set. In addition, we removed values from each data set with a different missing rate to make data set sparse.
- Mixture of Gaussians (MoG): we also investigated the speedup for MoG with Gaussian components that overlapped with one another. In particular, we sampled 2,000 cases for each experimental data set using five equally-weighted Gaussians with means at $\{(0,0), (0,1), (1,0), (0,-1), (-1,0)\}$ and variances 0.8.
- Semisupervised Bayesian classifier (SB): We used a Bayesian classifier that classifies instances with 100 10-valued discrete features into 5 categories. 3,000 training cases were generated with 90% unknown labels and missing feature values.

7.3 Accelerating EM by TJEM

Here we empirically show that TJEM outperforms EM. Note that Hesterberg (2005) has run experiments for similar algorithms, but his experiments are not as comprehensive as ours because he only used a quite simple probabilistic model with a two-dimensional parameter vector.

Figure 3 shows the results of our performance comparison. For the data sets of the same type of models, there is one scattered plot to show the required iterations for convergence for each algorithm. The coordination of each data point is the iterations of TJEM (the X-axis) and EM (the Y-axis) for the same data set. There are 100 data

points in each plot, representing the results of 100 trials. A data point lays in the upper triangle if TJEM converges faster, and in the lower triangle if EM is faster. We can see that in the 400 trials of all four models, TJEM converges faster for 392 times, and slower only for 8 times. Moreover, in the cases where TJEM is faster, it can run about 7-fold faster than EM. When TJEM is slower, it is only slightly slower than EM. The overall speedup for TJEM over EM is about two folds.

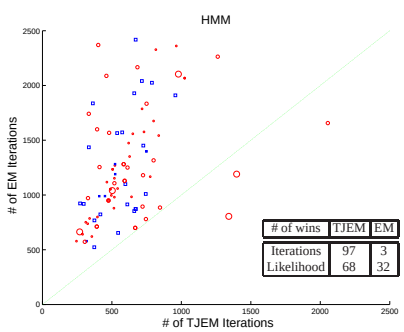
We also compare the likelihood of the output parameter vector from TJEM and EM. An acceleration method is desirable if it converges faster and stops at a local optimum with a higher likelihood. When TJEM and EM converge to different local optima, the final likelihood is determined by local optima. When they converge to the same local optimum, TJEM converges with higher likelihood most of the time. The reason is that EM may prematurely satisfy the termination condition because the improvement of the likelihood made by EM is limited in the neighborhood of θ^*. In contrast, the extrapolation made by TJEM allows it to obtain a parameter vector closer to θ^* and thus converge with a higher likelihood. This applies to other TJEM variants and explains why TJEM yields better likelihoods in most trials.

In Figure 3, a circle means that TJEM (the X-axis algorithm) converges with a higher likelihood, while a box indicates that EM (the Y-axis algorithm) stops with a higher likelihood score. The size of a data point shows the difference between their likelihood scores. A small point means that the difference is less than 10^{-5}, a medium one between 10^{-3} and 10^{-5}, and a large one more than 10^{-3}. We found that TJEM converges with a higher likelihood score in 60 trials for HMM, 90 for ALARM, 83 for MoG, and 60 for SB. Therefore, we can conclude that TJEM can actually accelerate the EM algorithm.

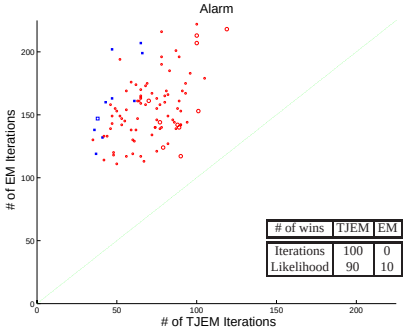
7.4 Accelerating TJEM by TJpEM

Next, we empirically compare the acceleration performance of the TJpEM and TJEM algorithms. We have shown analytically that TJpEM can further accelerate TJEM with a small learning rate η in Section 4.1. Figure 4 shows the results of our experimental comparison. The scattered plots show that, in 100 trials of HMM training, TJpEM with $\eta = 1.2$ (TJpEM12) and $\eta = 1.4$ (TJpEM14) converge faster than TJEM in 94 and 81 trials, respectively. Also, TJpEM12 is superior to TJEM in terms of the likelihood for 84 times, and TJpEM14 for 73 times. We obtained similar results in the experiments for other probabilistic models with the same settings (not shown here).

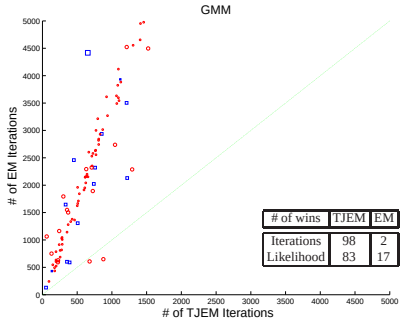
Proposition 4 implies that TJpEM may converge slower than TJEM with a large η, depending on the eigenvalues of the Jacobian of the original EM mapping of the problem at hand. We assigned $\eta = 1.6$ and 1.8 for TJpEM to compare its acceleration performance with TJEM for HMM training. Figure 4 shows the comparison results confirming that in most trials, TJpEM converges slower than TJEM with a large η, as predicted by our theorem.



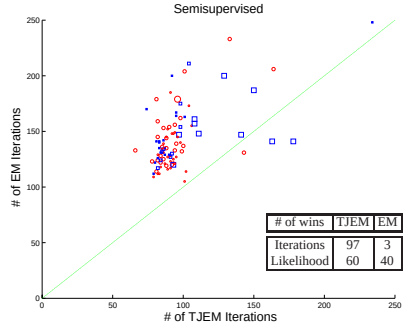
(a) Training HMM with TJEM and EM



(b) Training ALARM with TJEM and EM



(c) Training MoG with TJEM and EM



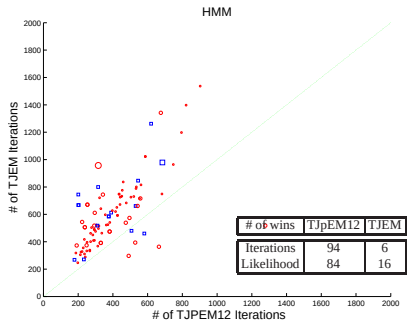
(d) Training SB with TJEM and EM

Fig. 3 Scattered plots that compare the TJEM and EM algorithms. TJEM converges faster in almost all trials.

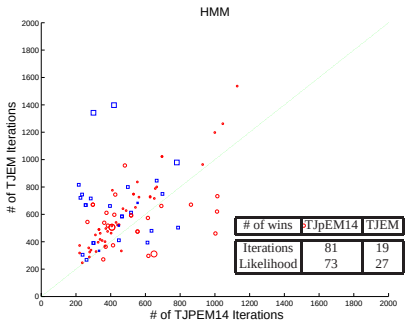
7.5 Comparison of TJpEM and TJ^{2}pEM

In Section 4.2, we concluded that because of negative eigenvalues, TJpEM may converge slower with a large learning rate or a small eigenvalue. We proposed another extrapolation method called TJ^{2}pEM to address the issue. Here we empirically demonstrate that TJ^{2}pEM can actually alleviate the impact of negative eigenvalues against TJpEM.

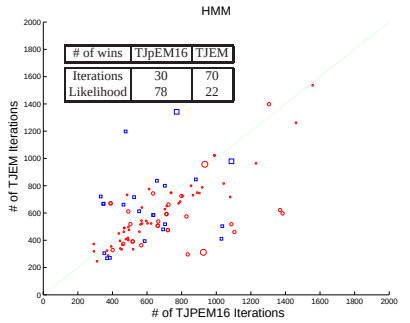
For our empirical demonstration, we considered a toy MoG model with two weighted one-dimensional Gaussians whose variances are a fixed known constant (1.0 in this case). Therefore, the parameter vector of our model has three dimensions,



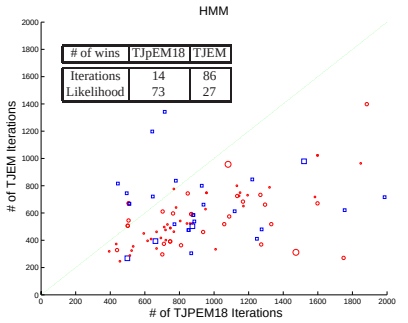
(a) Training HMM with TJpEM12 and TJEM



(b) Training HMM with TJpEM14 and TJEM



(c) Training HMM with TJpEM16 and TJEM

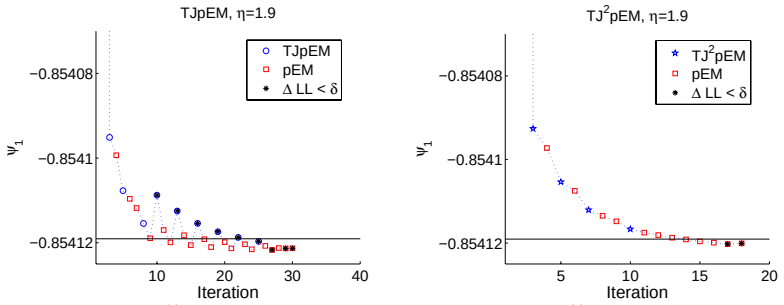


(d) Training HMM with TJpEM18 and TJEM

Fig. 4 Scattered plots that compare the TJpEM and TJEM algorithms. TJpEM converges faster when the learning rate $\eta = 1.2$ and 1.4 but slower when $\eta = 1.6$ and 1.8, as predicted.

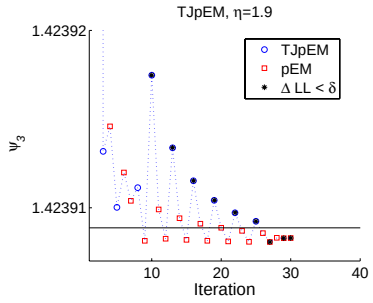
two for the means and one for the weight. We chose $[1.0, -1.0, 0.5]$ as the parameters of our model and sampled 500 data points for our experiment. This model is selected because it has been studied by Louis (1982), who also derived a general form of its Jacobian. We used his general form to obtain the Jacobian of our model and its eigenvalues $[0.7812, 0.3089, 0.2569]$. Then we chose a large learning rate $\eta = 1.9$ and the eigenvalues of the Jacobian of the resulting pEM mapping became $[0.5843, -0.3131, -0.4119]$. In this way, we have created a test case with negative eigenvalues.

Then we applied TJpEM and TJ^{2}pEM and plotted ψ_1 and ψ_3 as a function of iterations in Figure 5. ψ_1 and ψ_3 are the transformed parameters along the directions

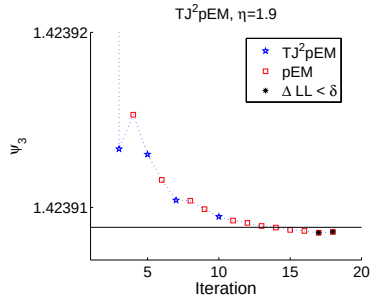


(a) $\psi_1^{(t)}$ in TJpEM

(b) $\psi_1^{(t)}$ in TJ^{2}pEM



(c) $\psi_3^{(t)}$ in TJpEM

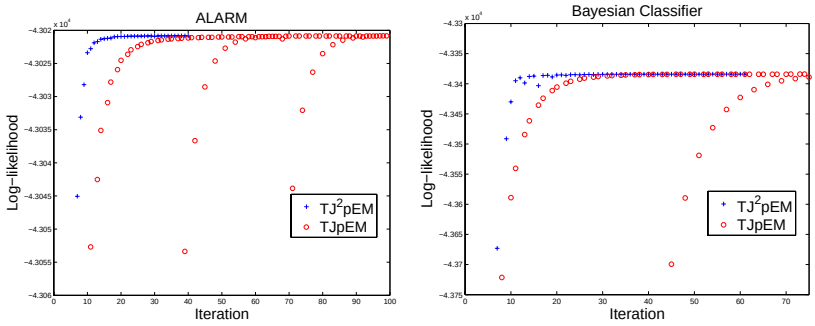


(d) $\psi_3^{(t)}$ in TJ^{2}pEM

Fig. 5 The change of the transformed parameters ψ by the TJpEM and TJ^{2}pEM algorithms during the training of a toy MoG model. pEM extrapolations are marked by a square, TJpEM by a circle, and TJ^{2}pEM by a pentagram. An asterisk mark on a data point indicates that the improvement of likelihood is less than a threshold. TJ^{2}pEM is more stable with a large η because its extrapolations never fail even for $\psi_3^{(t)}$, which is along the direction of $\lambda_{\eta min}$.

of $\lambda_{\eta max}$ and $\lambda_{\eta min}$, respectively. In Figure 5, the horizontal line in each chart is the position of the optimal parameter. A square data point is the parameter generated by pEM or EM. A circle point is generated by TJpEM, and a pentagram point by a extrapolation in TJ^{2}pEM. An asterisk mark on a data point indicates that the data point fails to improve the likelihood more than a given threshold and thus will be discarded at the next iteration. We can see that the extrapolation of TJpEM failed six times before it reached the optimal, while all the extrapolations by TJ^{2}pEM were successful. The failure of TJpEM was mainly due to the errant direction along $\lambda_{\eta min}$. More specifically, its extrapolation jumped farther away from θ_3^* than the extrapolation along the direction of $\lambda_{\eta max}$.

We also illustrate the impact on likelihood improvement with two large models. We generated sparse data sets with the ALARM and Bayesian classifier models and used a learning rate $\eta = 1.9$ for both algorithms to train these models. The learning rate was chosen so that the Jacobian of pEM will contain negative eigenvalues. We used sparse data sets because the eigenvalues of the Jacobian of EM for a sparse data set tend to cover a wide range between 0.0 and 1.0, hereby increase the chance



(a) ALARM (b) Bayesian classifier

Fig. 6 Two cases of negative eigenvalues. TJpEM makes consecutive errant jumps to worse estimates, as shown by two circles followed by a circle with a small likelihood. In contrast, TJ^{2}pEM makes no errant jumps and therefore converges faster than TJpEM.

that we have negative eigenvalues for the Jacobian of pEM. Figure 6(a) and 6(b) show two example runs of both algorithms for both models and the data sets. The results show that TJpEM usually jumped to estimates with a much worse likelihood score, while TJ^{2}pEM moved steadily toward local optima and converged faster.

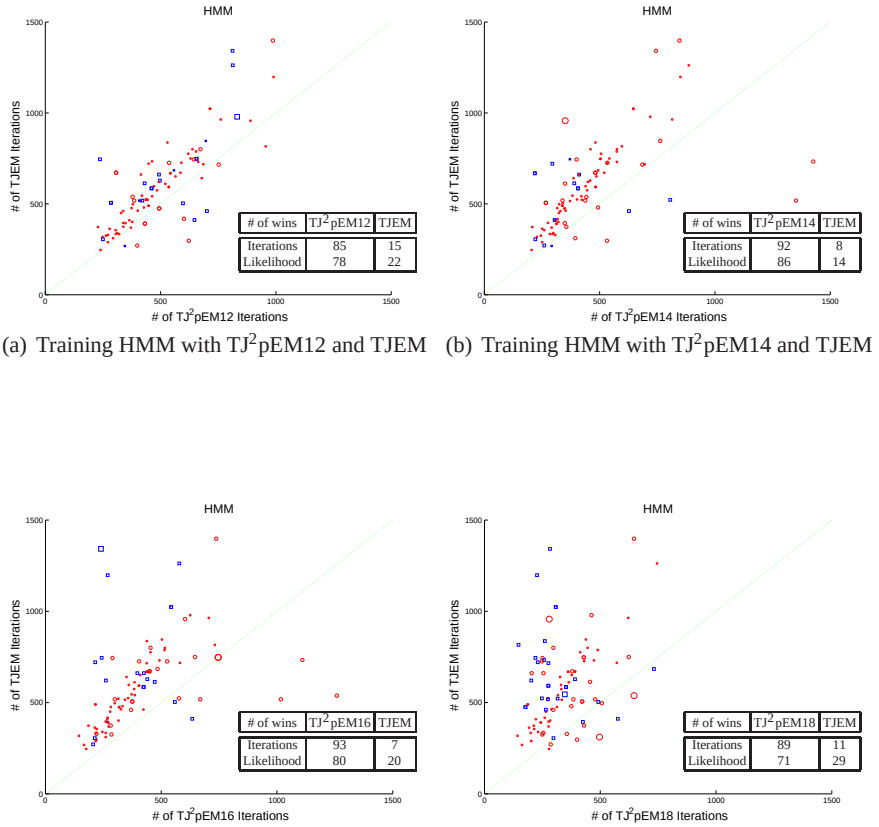
At last, we compare the acceleration performance of TJ^{2}pEM and TJEM for training HMM models. Figure 7 shows that TJ^{2}pEM consistently outperforms TJEM, regardless of the learning rates. Compared with the results of TJpEM in Figure 4, TJ^{2}pEM is less sensitive to the change of η. Clearly, the results above confirm that TJ^{2}pEM can alleviate the impact of negative eigenvalues.

7.6 Comparison of TJ^{2}aEM and aEM

In this subsection, we empirically demonstrate the advantage of dynamic learning rates and compare the acceleration performance of aEM and TJ^{2}aEM.

We compared the acceleration performance of pEM with an optimal learning rate and two algorithms that dynamically adjust their learning rates, aEM and TJ^{2}aEM. We empirically determined the optimal learning rate η^* for a large MoG model as follows. First, we ran EM with a tiny threshold ($1.0e-11$) and kept track of the parameter vectors searched and their likelihood. It took the EM algorithm 4,885 iterations to converge. We chose $\theta^{(501)}$ obtained by EM as the initial value because it is near the local optimum θ^*. Then, we tried pEM with various learning rates η and found that $\eta^* = 1.96$ is the optimal learning rate.

After that, we ran both aEM and TJ^{2}aEM from $\theta^{(501)}$. At each iteration, they dynamically adjust their learning rates. For aEM, its learning rate is adjusted by $\eta^{(t+1)} = 1.1\eta^{(t)}$, while for TJ^{2}aEM, η is dynamically assigned to 1.2, 1.4, 1.6, or 1.8 in a zigzag manner. With a different η, TJ^{2}aEM will come up with a different

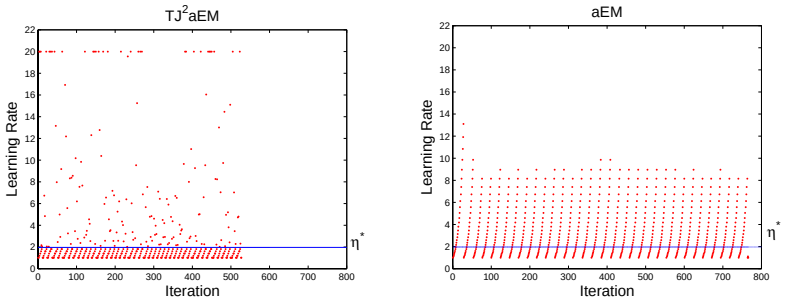


(a) Training HMM with TJ^{2}pEM12 and TJEM (b) Training HMM with TJ^{2}pEM14 and TJEM

(c) Training HMM with TJ^{2}pEM16 and TJEM (d) Training HMM with TJ^{2}pEM18 and TJEM

Fig. 7 Scattered plots that compare the TJ^{2}pEM and TJEM algorithms. TJ^{2}pEM converges faster in almost all cases regardless of the learning rate.

estimate γ_η at each iteration, and use the *effective learning rate* $\frac{1}{1-\gamma_\eta}$ to perform double extrapolation (see Lemma 3). We compared the effective learning rate of TJ^{2}aEM and the learning rate of aEM at each iteration, as shown in Fig 8. We can see that aEM increases its learning rate linearly until it reaches a point where it cannot satisfactorily improve the likelihood, while TJ^{2}aEM adjusts its effective learning rate irregularly and much aggressively. TJ^{2}aEM may adjust its learning rate to up to our predefined upper bound many times while aEM only reaches as high as 14 once and usually stops at 9. In the end, the elapsed iterations for TJ^{2}aEM and aEM are 527 and 766, respectively. Both outperform pEM with a fixed optimal learning rate, which required 1,327 iterations to converge.



(a) Learning rates used by TJ^{2}aEM (b) Learning rates used by aEM

Fig. 8 Trace of learning rates used by TJ^{2}aEM and aEM as a function of iterations. The Figure also shows that the number of iterations required to converge for TJ^{2}aEM is less than aEM.

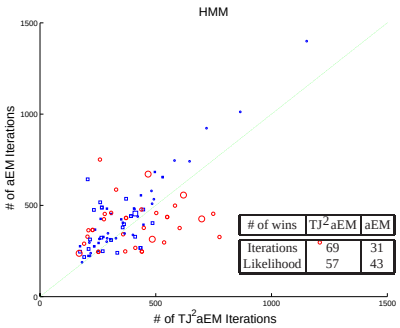
Finally, we perform a more comprehensive comparison of TJ^{2}aEM and aEM with large models and data sets. Figure 9 shows the comparison between the two algorithms on the four models. In 100 trials, TJ^{2}aEM converges faster for 69 trials for HMM, 72 for ALARM, 66 for MoG, and 70 for SB. Besides, TJ^{2}aEM reaches estimates with higher likelihood scores in 57 trials for HMM, 88 for ALARM, 57 for MoG, and 58 for SB. The results show that TJ^{2}aEM converges faster and yields higher likelihood scores more often than aEM in our experiments. In many cases, two- to three-fold or even higher speedup can be achieved. The results also show that the performance of TJ^{2}aEM is insensitive to different data distributions and initial conditions.

8 Conclusions

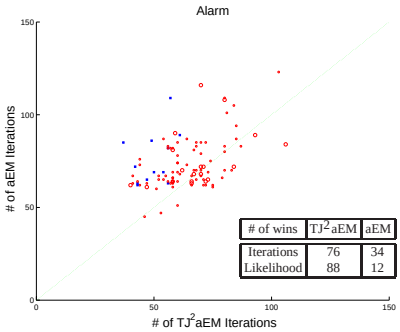
We have presented TJ^{2}aEM, a fast yet computationally efficient method to accelerate the EM algorithm. TJ^{2}aEM uses targeted aggressive extrapolation along with dynamic learning rate to outperform previous works like aEM that only use one of the techniques. Moreover, the average time of a TJ^{2}aEM iteration is almost the same as the average time of an EM iteration, which is extremely useful for accelerating machine learning problems with probabilistic models.

We constructed TJ^{2}aEM step by step, from the baseline TJEM algorithm, TJpEM, TJ^{2}pEM, and finally to TJ^{2}aEM. We provided theoretical analysis and experimental verification of the improvements made at each step:

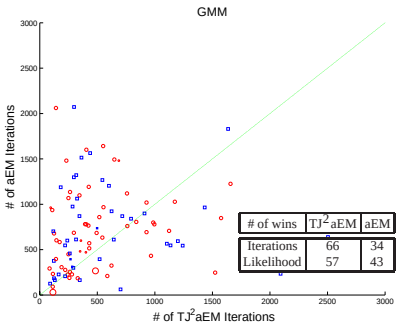
- we use TJpEM to speed up TJEM, and induce M_η with η for dynamic adjustment;
- we propose TJ^{2}pEM to stabilize TJpEM to reduce the impact of negative eigenvalues when η is large;
- we adopt dynamic learning rates as in aEM to obtain TJ^{2}aEM.



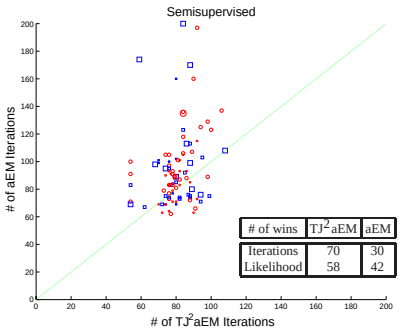
(a) Training HMM with TJ^{2}aEM and aEM



(b) Training ALARM with TJ^{2}aEM and aEM



(c) Training MoG with TJ^{2}aEM and aEM



(d) Training SB with TJ^{2}aEM and aEM

Fig. 9 Comparison of TJ^{2}aEM and aEM. The results show that TJ^{2}aEM usually converges faster than aEM.

Compared with previous works, we contribute many new ideas to explore further acceleration. The first is that a mapping whose Jacobian contains negative eigenvalues, like pEM, can still achieve speedup. Traditionally, only mappings with semi-positive definite Jacobians are considered. The second is that negative eigenvalues can be handled by double extrapolation like TJ^{2}pEM. Finally, integrating several non-optimal mappings may converge faster than sticking to a fixed optimal mapping.

References

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

Berlinet, A. and Roland, C. (2007). Acceleration schemes with application to the EM algorithm. *Computational Statistics and Data Analysis*, 51:3689–3702.

Burden, R. L. and Faires, D. (1988). *Numerical Analysis*. PWS-KENT Pub Co.

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

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

Hammerlin, G. and Hoffmann, K.-H. (1991). *Numerical Mathematics*. Springer-Verlag: New York.

Hesterberg, T. (2005). Staggered Aitken acceleration for EM. In *Proceedings of the Statistical Computing Section of the American Statistical Association*, Minneapolis, Minnesota, USA.

Huang, H.-S., Yang, B.-H., and Hsu, C.-N. (2005). Triple-jump acceleration for the EM algorithm. In *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM-2005)*, pages 649–652, Houston, TX, USA.

Jamshidian, M. and Jennrich, R. I. (1997). Acceleration of the EM algorithm by using quasi-newton methods. *Journal of the Royal Statistical Society, , Series B*, 59(3):569–587.

Kuroda, M. and Sakakihara, M. (2006). Accelerating the convergence of the EM algorithm using the vector ε algorithm. *Computational Statistics and Data Analysis*, 51:1549–1561.

Louis, T. A. (1982). Finding the observed information matrix when using the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 44:226–233.

Luis, O. and Leslie, K. (1999). Accelerating EM: an empirical study. In *Proc. of the 15th Conference on Uncertainty in Artificial Intelligence*, pages 512–521.

McLachlan, G. J. and Krishnan, T. (1997). *The EM Algorithm and Extensions*. Wiley Series in Probability and Statistics. Wiley-Interscience.

Salakhutdinov, R. and Roweis, S. (2003). Adaptive overrelaxed bound optimization methods. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 664–671.

Schafer, J. L. (1997). *Analysis of Incomplete Multivariate Data*. London: Chapman and Hall / CRC Press.

Varadhan, R. and Roland, C. (2004). Squared extrapolation methods (SQUAREM): a newclass of simple and efficient numerical schemes for accelerating the convergence of the EM algorithm. Department of Biostatistics Working Paper, Johns Hopkins University, Paper 63.

Wynn, P. (1962). Acceleration techniques for iterated vector and matrix problems. *Mathematics of Computation*, 16.

9 Notation

	Notation for Parameter Vectors
θ	Parameter vector variable
θ^*	A parameter vector which is a local maximum of likelihood
$\theta^{(t)}$	The output of a search algorithm at iteration t
$\theta_{EM}^{(t)}$	The output of one iteration of EM given $\theta^{(t)}$
$\theta_{\eta}^{(t)}$	The output of a pEM extrapolation with learning rate η given $\theta^{(t)}$
$\theta_{\gamma}^{(t)}$	The output of a TJEM extrapolation given $\theta^{(t-1)}$, $\theta^{(t)}$, and $\theta_{EM}^{(t)}$
$\theta_{\gamma_\eta}^{(t)}$	The output of a extrapolation in TJpEM with learning rate η given $\theta^{(t-1)}$, $\theta^{(t)}$, and $\theta_{\eta}^{(t)}$
$\theta_{\gamma_\eta^2}^{(t)}$	The output of a extrapolation in TJ^{2}pEM with learning rate η given $\theta^{(t-1)}$, $\theta^{(t)}$, and $\theta_{\eta}^{(t)}$
ψ, ψ^*, $\psi^{(t)}$, and $\psi_{EM}^{(t)}$	The transformed parameter vectors on the eigenspace corresponding to θ, θ^*, $\theta^{(t)}$, and $\theta_{EM}^{(t)}$, respectively
θ_i, θ_i^*, $\cdots$	The i-th element of θ, θ^*, $\cdots$ (the same convention for any parameter vector symbol)

	Notation for Mappings
M	The EM mapping
$M'(\theta^*)$ and J	The Jacobian of M at θ^*
Q	The eigenvectors of J
Λ	A diagonal matrix of the eigenvalues of J in the descendent order
λ_i	The i-th eigenvalue of J, which is at position (i,i) of Λ
ρ	The spectral radius of J, which is the maximal absolute eigenvalue in Λ
$\gamma^{(t)}$	Estimation of ρ at iteration t
M_η, M_γ, M_{γ_η}, and $M_{\gamma_\eta^2}$	The pEM, TJEM, TJpEM, and TJ^{2}pEM mapping, respectively
J_η, J_γ, J_{γ_η}, and $J_{\gamma_\eta^2}$	The Jacobians of M_η, M_γ, M_{γ_η}, and $M_{\gamma_\eta^2}$, respectively
Λ_η, Λ_γ, Λ_{γ_η}, and $\Lambda_{\gamma_\eta^2}$	The eigenmatrices of J_η, J_γ, J_{γ_η}, and $J_{\gamma_\eta^2}$, respectively
$\lambda_{\eta i}$, $\lambda_{\gamma i}$, $\lambda_{\gamma_\eta i}$, and $\lambda_{\gamma_\eta^2 i}$	The i-th eigenvalues of J_η, J_γ, J_{γ_η}, and $J_{\gamma_\eta^2}$, respectively
ρ_η, ρ_γ, ρ_{γ_η}, and $\rho_{\gamma_\eta^2}$	The spectral radii of J_η, J_γ, J_{γ_η}, and $J_{\gamma_\eta^2}$, respectively
$\gamma_\eta^{(t)}$, $\gamma_\gamma^{(t)}$, $\gamma_{\gamma_\eta}^{(t)}$, and $\gamma_{\gamma_\eta^2}^{(t)}$	Estimation of ρ_η, ρ_γ, ρ_{γ_η}, and $\rho_{\gamma_\eta^2}$, respectively
$\rho_{\circ\gamma}$ and $\rho_{\circ\gamma_\eta}$	The spectral radii of JJ_γ and $J_\eta J_{\gamma_\eta}$, which are the eigenvalues in $(\Lambda\Lambda_\gamma)$ and $(\Lambda_\eta\Lambda_{\gamma_\eta})$, respectively

	Others
η^*	The optimal learning rate for pEM
Symbols with subscript η^*	The same as those subscripted with η^* except that pEM with η^* is used

10 Proofs

Proof. (Proposition 1)

To ensure convergence, it is required that all the eigenvalues lie within the range (-1,1). The condition holds if $\lambda_{\eta max} < 1$ and $\lambda_{\eta min} > -1$. By Lemma 1, we substitute $\lambda_{\eta max}$ with $(1-\eta)+\eta\lambda_{max}$ and $\lambda_{\eta min}$ with $(1-\eta)+\eta\lambda_{min}$. Then, $\lambda_{\eta max} < 1$ implies that $\eta > 0$, and $\lambda_{\eta min} > -1$ implies that $\eta < \frac{2}{1-\lambda_{min}}$. □

Proof. (Proposition 2)

Based on Assumption 1, when $\lambda_{min} \geq 0$, the upper bound $\frac{2}{1-\lambda_{min}}$ in Proposition 1 becomes 2. Therefore, pEM must converge if $0 < \eta < 2$. □

Proof. (Lemma 2)

Let $\eta' = \frac{1}{1-\gamma_\eta^{(t)}}$. The i-th eigenvalue is $\lambda_{\eta i}\lambda_{\gamma_\eta i} = \lambda_{\eta i}(1-\eta'+\eta'\lambda_{\eta i}) = \lambda_{\eta i}\frac{\lambda_{\eta i}-\gamma_\eta^{(t)}}{1-\gamma_\eta^{(t)}}$. □

Proof. (Proposition 3)

The convergence rate of pEM is determined by $\rho_\eta = \max\{|\lambda_{\eta max}|, |\lambda_{\eta min}|\}$. ρ_η is minimized when $\lambda_{\eta max} = -\lambda_{\eta min}$. Substituting $\lambda_{\eta max}$ and $\lambda_{\eta min}$ with Lemma 1, we obtain that $\eta^* = \frac{2}{2-\lambda_{max}-\lambda_{min}}$. □

Proof. (Proposition 4)

Let $f_{\gamma_\eta^{(t)}}(\lambda)$ be a function such that $f_{\gamma_\eta^{(t)}}(\lambda_{\eta i}) = \lambda_{\circ\gamma_\eta i}$. Then,

$$f_{\gamma_\eta^{(t)}}(\lambda) = \lambda\frac{\lambda-\gamma_\eta^{(t)}}{1-\gamma_\eta^{(t)}} = \frac{1}{1-\gamma_\eta^{(t)}}\left[(\lambda-\frac{\gamma_\eta^{(t)}}{2})^2 - \frac{(\gamma_\eta^{(t)})^2}{4}\right]. \tag{22}$$

Let $\rho_{\circ\gamma_\eta}$ denote the spectral radius of $J_{\gamma_\eta}J_\eta$, then its upper bound $\sup\rho_{\circ\gamma_\eta}$ is the maximum of $|f_{\gamma_\eta^{(t)}}(\lambda)|$ with $\lambda_{\eta min} \leq \lambda \leq \lambda_{\eta max}$. Since $f_{\gamma_\eta^{(t)}}(\lambda)$ is quadratic, the upper bound is determined by either $f_{\gamma_\eta^{(t)}}(\lambda_{\eta min})$, $f_{\gamma_\eta^{(t)}}(\frac{\gamma_\eta^{(t)}}{2})$, or $f_{\gamma_\eta^{(t)}}(\lambda_{\eta max})$:

$$\sup\rho_{\circ\gamma_\eta i} \leq \max_\lambda |f_{\gamma_\eta^{(t)}}(\lambda)| = \max\{|f_{\gamma_\eta^{(t)}}(\lambda_{\eta min})|, |f_{\gamma_\eta^{(t)}}(\frac{\gamma_\eta^{(t)}}{2})|, |f_{\gamma_\eta^{(t)}}(\lambda_{\eta max})|\}. \tag{23}$$

After simple rearrangement, we can obtain specific expressions for the upper bound according to the ranges that $\lambda_{\eta max}$ and $\lambda_{\eta min}$ lie within.

$$\max_\lambda |f_{\gamma_\eta^{(t)}}(\lambda)| = \begin{cases} |f_{\gamma_\eta^{(t)}}(\frac{\gamma_\eta^{(t)}}{2})| & \text{if } \frac{1-\sqrt{2}}{2}\gamma_\eta^{(t)} \leq \lambda_{\eta min}, \lambda_{\eta max} \leq \frac{1+\sqrt{2}}{2}\gamma_\eta^{(t)} \\ |f_{\gamma_\eta^{(t)}}(\lambda_{\eta max})| & \text{if } \frac{1-\sqrt{2}}{2}\gamma_\eta^{(t)} \leq \lambda_{\eta min}, \frac{1+\sqrt{2}}{2}\gamma_\eta^{(t)} \leq \lambda_{\eta max} \\ |f_{\gamma_\eta^{(t)}}(\lambda_{\eta min})| & \text{if } \lambda_{\eta min} \leq \frac{1-\sqrt{2}}{2}\gamma_\eta^{(t)}, \lambda_{\eta max} \leq \frac{1+\sqrt{2}}{2}\gamma_\eta^{(t)} \\ \max\{|f_{\gamma_\eta^{(t)}}(\lambda_{\eta min})|, |f_{\gamma_\eta^{(t)}}(\lambda_{\eta max})|\} & \text{otherwise.} \end{cases} \tag{24}$$

Let $\lambda_{\circ\gamma i}$ denote the i-th eigenvalue and $\rho_{\circ\gamma}$ the spectral radius of the Jacobian of the TJEM mapping. We check the sufficient conditions of η for each possible upper bounds in Equation (24) to prove that $\sup\rho_{\circ\gamma_\eta} < \sup\rho_{\circ\gamma}$ with a proper η:

1. When $f_{\gamma_\eta^{(t)}}(\lambda_{\eta max}) \geq \frac{1+\sqrt{2}}{2}\gamma_\eta^{(t)}$, we have $\sup\rho_{\circ\gamma_\eta} = |f_{\gamma_\eta^{(t)}}(\lambda_{\eta max})| < |f_{\gamma^{(t)}}(\lambda_{max})| = \sup\rho_{\circ\gamma}$.
2. $|f_{\gamma_\eta^{(t)}}(\frac{\gamma_\eta^{(t)}}{2})| < |f_{\gamma^{(t)}}(\frac{\gamma^{(t)}}{2})|$ always holds.
3. When $\eta < \frac{1+\frac{\gamma}{4}}{1-\lambda_{min}}$, we have $|f_{\gamma_\eta^{(t)}}(\lambda_{\eta min})| < |f_{\gamma^{(t)}}(\frac{\gamma^{(t)}}{2})| \leq \sup\rho_{\circ\gamma}$.

First, we show that $|f_{\gamma_\eta^{(t)}}(\lambda_{\eta max})| < |f_{\gamma^{(t)}}(\lambda_{max})|$ when $f_{\gamma_\eta^{(t)}}(\lambda_{\eta max}) \geq \frac{1+\sqrt{2}}{2}\gamma_\eta^{(t)}$. We substitute $f_{\gamma_\eta^{(t)}}(.)$ and $f_{\gamma^{(t)}}(.)$ with Equation (22):

$$|f_{\gamma_\eta^{(t)}}(\lambda_{\eta max})| = \lambda_{\eta max}\frac{\lambda_{\eta max}-\gamma_\eta^{(t)}}{1-\gamma_\eta^{(t)}} = \lambda_{\eta max}\frac{\eta(\lambda_{max}-\gamma^{(t)})}{\eta(1-\gamma^{(t)})} = \lambda_{\eta max}\frac{\lambda_{max}-\gamma^{(t)}}{1-\gamma^{(t)}}.$$

Since $\lambda_{\eta max} < \lambda_{max}$ when $1 < \eta < 2$, the first inequality must hold:

$$|f_{\gamma_\eta^{(t)}}(\lambda_{\eta max})| = \lambda_{\eta max}\frac{\lambda_{max}-\gamma^{(t)}}{1-\gamma^{(t)}} < \lambda_{max}\frac{\lambda_{max}-\gamma^{(t)}}{1-\gamma^{(t)}} = |f_{\gamma^{(t)}}(\lambda_{max})|.$$

Therefore, when $|f_{\gamma_\eta^{(t)}}(\lambda_{\eta max})|$ is the upper bound, $\sup\rho_{\circ\gamma_\eta}$ must be smaller than $\sup\rho_{\circ\gamma}$.

Then, we show that $|f_{\gamma_\eta^{(t)}}(\frac{\gamma_\eta^{(t)}}{2})| < |f_{\gamma^{(t)}}(\frac{\gamma^{(t)}}{2})|$. From Equation (22), we have:

$$|f_{\gamma_\eta^{(t)}}(\frac{\gamma_\eta^{(t)}}{2})| = \frac{(\gamma_\eta^{(t)})^2}{4(1-\gamma_\eta^{(t)})} = \frac{(\gamma_\eta^{(t)})^2}{4\eta(1-\gamma^{(t)})}.$$

Similarly, we have $|f_{\gamma^{(t)}}(\frac{\gamma^{(t)}}{2})| = \frac{(\gamma^{(t)})^2}{4(1-\gamma^{(t)})}$. In addition, the relation $\gamma_\eta^{(t)} > 0$ also holds, which implies that $\gamma^{(t)} > \gamma_\eta^{(t)}$ here. Otherwise, $\sup\rho_{\circ\gamma_\eta}$ is determined by $\max\{|f_{\gamma_\eta^{(t)}}(\lambda_{\eta min})|, |f_{\gamma_\eta^{(t)}}(\lambda_{\eta max})|\}$. Since $1 < \eta$ and $\gamma_\eta^{(t)} \leq \gamma^{(t)}$, $|f_{\gamma_\eta^{(t)}}(\frac{\gamma_\eta^{(t)}}{2})|$ must be smaller than $|f_{\gamma^{(t)}}(\frac{\gamma^{(t)}}{2})|$:

$$|f_{\gamma_\eta^{(t)}}(\frac{\gamma_\eta^{(t)}}{2})| = \frac{(\gamma_\eta^{(t)})^2}{4\eta(1-\gamma^{(t)})} < \frac{(\gamma^{(t)})^2}{4(1-\gamma^{(t)})} = |f_{\gamma^{(t)}}(\frac{\gamma^{(t)}}{2})|.$$

Therefore, when $|f_{\gamma_\eta^{(t)}}(\frac{\gamma_\eta^{(t)}}{2})|$ is the upper bound, $\sup\rho_{\circ\gamma_\eta}$ must be smaller than $\sup\rho_{\circ\gamma}$.

At last, we show the sufficient condition of η such that $|f_{\gamma_\eta^{(t)}}(\lambda_{\eta min})| < |f_{\gamma^{(t)}}(\frac{\gamma^{(t)}}{2})|$ to guarantee that $\sup\rho_{\circ\gamma_\eta} < \sup\rho_{\circ\gamma}$. We solve the following inequality:

$$\begin{aligned}
|f_{\gamma_\eta^{(t)}}(\lambda_{\eta min})| &= \lambda_{\eta min}\frac{\lambda_{min}-\gamma^{(t)}}{1-\gamma^{(t)}} \\
&< -\lambda_{\eta min}\frac{\gamma^{(t)}}{1-\gamma^{(t)}} \\
&< \frac{(\gamma^{(t)})^2}{4(1-\gamma^{(t)})} \\
&= |f_{\gamma^{(t)}}(\frac{\gamma^{(t)}}{2})|.
\end{aligned}$$

We can obtain the sufficient condition of η:

$$\eta < \frac{1+\frac{\gamma}{4}}{1-\lambda_{min}},$$

which is greater than 1. □

Proof. (Lemma 3)

TJ^{2}pEM in Equation (21) is of the same form as pEM in Equation (2) by substituting the EM mapping with two consecutive pEM mappings, in which the i-th eigenvalue is $(\lambda_{\eta i})^2$. Then, note that $\frac{1}{1-(\gamma_\eta^{(t)})^2}$ corresponds to η in Equation (21). Therefore, we can apply Lemma 1 to compute the eigenvalue:

$$\lambda_{\gamma_\eta^2 i} = 1 - \frac{1}{1-(\gamma_\eta^{(t)})^2} + \frac{1}{1-(\gamma_\eta^{(t)})^2}(\lambda_{\eta i})^2 = \frac{(\lambda_{\eta i})^2-(\gamma_\eta^{(t)})^2}{1-(\gamma_\eta^{(t)})^2}. \tag{25}$$

□

Proof. (Proposition 5)

As in our discussion in Section 4.1, $\lambda_{\gamma_\eta^2 i}$ can be considered as a quadratic function of $\lambda_{\eta i}$ and the upper bound of its spectral radius $\rho_{\gamma_\eta^2}$ is:

$$\sup\rho_{\gamma_\eta^2} \le \max\{\frac{(\gamma_\eta^{(t)})^2}{1-(\gamma_\eta^{(t)})^2}, \frac{\lambda_{\eta max}^2-(\gamma_\eta^{(t)})^2}{1-(\gamma_\eta^{(t)})^2}, \frac{\lambda_{\eta min}^2-(\gamma_\eta^{(t)})^2}{1-(\gamma_\eta^{(t)})^2}\}$$

We can obtain specific expressions for $\sup\rho_{\gamma_\eta^2}$ under different conditions of $\lambda_{\eta max}$ and $\lambda_{\eta min}$:

$$\sup \rho_{\gamma_\eta^2} \leq \begin{cases} \frac{(\gamma_\eta^{(t)})^2}{1-(\gamma_\eta^{(t)})^2} & \text{if } -\sqrt{2}\gamma_\eta^{(t)} < \lambda_{\eta min}, \lambda_{\eta max} < \sqrt{2}\gamma_\eta^{(t)} \\ \frac{\lambda_{\eta max}^2-(\gamma_\eta^{(t)})^2}{1-(\gamma_\eta^{(t)})^2} & \text{if } -\sqrt{2}\gamma_\eta^{(t)} < \lambda_{\eta min} < \sqrt{2}\gamma_\eta^{(t)} < \lambda_{\eta max} \\ \frac{\lambda_{\eta min}^2-(\gamma_\eta^{(t)})^2}{1-(\gamma_\eta^{(t)})^2} & \text{if } \lambda_{\eta min} < -\sqrt{2}\gamma_\eta^{(t)} < \lambda_{\eta max} < \sqrt{2}\gamma_\eta^{(t)} \\ \max\{\frac{\lambda_{\eta max}^2-(\gamma_\eta^{(t)})^2}{1-(\gamma_\eta^{(t)})^2}, \frac{\lambda_{\eta min}^2-(\gamma_\eta^{(t)})^2}{1-(\gamma_\eta^{(t)})^2}\} & \text{otherwise.} \end{cases} \tag{26}$$

When $\lambda_{\eta min} \leq \frac{1-\sqrt{2}}{2}\gamma_\eta^{(t)}$ and $\lambda_{\eta max} \leq \frac{1+\sqrt{2}}{2}\gamma_\eta^{(t)}$, $\sup \rho_{\circ\gamma_\eta}$ is $\lambda_{\eta min}\frac{\lambda_{\eta min}-\gamma_\eta^{(t)}}{1-\gamma_\eta^{(t)}}$. In the same range of $\lambda_{\eta max}$, $\sup \rho_{\gamma_\eta^2}$ is:

$$\sup \rho_{\gamma_\eta^2} = \begin{cases} \frac{(\gamma_\eta^{(t)})^2}{1-(\gamma_\eta^{(t)})^2} & \text{if } -\sqrt{2}\gamma_\eta^{(t)} < \lambda_{\eta min} < \frac{1-\sqrt{2}}{2}\gamma_\eta^{(t)} \\ \frac{\lambda_{\eta min}^2-(\gamma_\eta^{(t)})^2}{1-(\gamma_\eta^{(t)})^2} & \text{if } \lambda_{\eta min} < -\sqrt{2}\gamma_\eta^{(t)}. \end{cases}$$

Therefore, $\sup \rho_{\gamma_\eta^2}$ is less than $\sup \rho_{\circ\gamma_\eta}$ if the following situations hold:

1. $\frac{(\gamma_\eta^{(t)})^2}{1-(\gamma_\eta^{(t)})^2} < \lambda_{\eta min}\frac{\lambda_{\eta min}-\gamma_\eta^{(t)}}{1-\gamma_\eta^{(t)}}$ when $\lambda_{\eta min} \leq \frac{1-\sqrt{2}}{2}\gamma_\eta^{(t)}$, and
2. $\frac{\lambda_{\eta min}^2-(\gamma_\eta^{(t)})^2}{1-(\gamma_\eta^{(t)})^2} < \lambda_{\eta min}\frac{\lambda_{\eta min}-\gamma_\eta^{(t)}}{1-\gamma_\eta^{(t)}}$ when $\lambda_{\eta min} < -\sqrt{2}\gamma_\eta^{(t)}$.

For situation 1, we solve the last inequality below:

$$\begin{aligned} \text{RHS} &= \lambda_{\eta min}\frac{\lambda_{\eta min}-\gamma_\eta^{(t)}}{1-\gamma_\eta^{(t)}} \\ &= |\lambda_{\eta min}|\frac{|\lambda_{\eta min}|+\gamma_\eta^{(t)}}{1-\gamma_\eta^{(t)}} \\ &> |\lambda_{\eta min}|\frac{\gamma_\eta^{(t)}}{1-\gamma_\eta^{(t)}} \\ &> \frac{(\gamma_\eta^{(t)})^2}{1-(\gamma_\eta^{(t)})^2} \\ &= \text{LHS}. \end{aligned}$$

The last inequality can be simplified as:

$$|\lambda_{\eta min}| > \frac{(\gamma_\eta^{(t)})}{1+(\gamma_\eta^{(t)})} > \frac{1}{2}.$$

Since $\lambda_{\eta min} < 0$, we have that $\lambda_{\eta min} < -\frac{1}{2} < \frac{1-\sqrt{2}}{2}\gamma_\eta^{(t)}$ is a sufficient condition such that $\frac{(\gamma_\eta^{(t)})^2}{1-(\gamma_\eta^{(t)})^2} < \lambda_{\eta min}\frac{\lambda_{\eta min}-\gamma_\eta^{(t)}}{1-\gamma_\eta^{(t)}}$.

For situation 2, we start from the LHS. Note that in this case, $|\lambda_{\eta min}| > \gamma_\eta^{(t)}$:

$$\begin{aligned}
\text{LHS} &= \frac{\lambda_{\eta min}^2 - (\gamma_\eta^{(t)})^2}{1-(\gamma_\eta^{(t)})^2} \\
&= \frac{\lambda_{\eta min}+\gamma_\eta^{(t)}}{1+\gamma_\eta^{(t)}} \cdot \frac{\lambda_{\eta min}-\gamma_\eta^{(t)}}{1-\gamma_\eta^{(t)}} \\
&= \frac{|\lambda_{\eta min}|-\gamma_\eta^{(t)}}{1+\gamma_\eta^{(t)}} \left| \frac{\lambda_{\eta min}-\gamma_\eta^{(t)}}{1-\gamma_\eta^{(t)}} \right| \\
&> |\lambda_{\eta min}| \left| \frac{\lambda_{\eta min}-\gamma_\eta^{(t)}}{1-\gamma_\eta^{(t)}} \right| \\
&= \text{RHS}.
\end{aligned}$$

Therefore, situation 2 always holds. Combining the two situations, we obtain the result that $\sup \rho_{\gamma_\eta^2} < \sup \rho_{\circ\gamma_\eta}$ if $\lambda_{\eta min} < -\frac{1}{2}$ and $\lambda_{\eta max} \leq \frac{1+\sqrt{2}}{2}\gamma_\eta^{(t)}$. $\square$