Adaptive Overrelaxed Bound Optimization Methods

Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Sam Roweis ROWEIS@CS.TORONTO.EDU
Department of Computer Science, University of Toronto
6 King's College Rd, M5S 3G4, Canada

Abstract

We study a class of *overrelaxed* bound optimization algorithms, and their relationship to standard bound optimizers, such as Expectation-Maximization, Iterative Scaling, CCCP and Non-Negative Matrix Factorization. We provide a theoretical analysis of the convergence properties of these optimizers and identify analytic conditions under which they are expected to outperform the standard versions. Based on this analysis, we propose a novel, simple adaptive overrelaxed scheme for practical optimization and report empirical results on several synthetic and real-world data sets showing that these new adaptive methods exhibit superior performance (in certain cases by several orders of magnitude) compared to their traditional counterparts. Our "drop-in" extensions are simple to implement, apply to a wide variety of algorithms, almost always give a substantial speedup, and do not require any theoretical analysis of the underlying algorithm.

1. Introduction

Many problems in machine learning and pattern recognition ultimately reduce to the optimization of a scalar valued function $L(\Theta|\text{data})$ of a free parameter vector Θ. For example, in (supervised or) unsupervised probabilistic modeling the objective function may be the (conditional) data likelihood or the posterior over parameters. In discriminative learning we may use a classification or regression score; in reinforcement learning we may use average discounted reward. Optimization may also arise during inference; for example we may want to reduce the cross entropy between two distributions or minimize a function such as the Bethe free energy.

Bound optimization takes advantage of the fact that many objective functions arising in practice have a special structure. We can often exploit this structure to obtain a bound on the objective function and proceed by optimizing this bound. Ideally, we seek a bound that is valid everywhere in parameter space, easily optimized, and equal to the true objective function at one (or more) point(s). A general form of a bound maximizer which iteratively lower bounds the objective function is given below:

General Bound Optimizer for maximizing $L(\Theta)$:
• **Assume**: $\exists$ function $G(\Theta, \Psi)$ such that for any Θ' and Ψ':

1. $G(\Theta', \Theta') = L(\Theta')$ & $L(\Theta) \geq G(\Theta, \Psi') \; \forall \; \Psi' \neq \Theta$
2. $\arg\max_\Theta G(\Theta, \Psi')$ can be found easily for any Ψ'.

• **Iterate**: $\Theta^{t+1} = \arg\max_\Theta G(\Theta, \Theta^t)$
• **Guarantee**: $L(\Theta^{t+1}) = G(\Theta^{t+1}, \Theta^{t+1}) \geq G(\Theta^{t+1}, \Theta^t) \geq G(\Theta^t, \Theta^t) = L(\Theta^t)$

Many popular iterative algorithms are bound optimizers, including the EM algorithm for maximum likelihood learning in latent variable models[3], iterative scaling (IS) algorithms for parameter estimation in maximum entropy models[2] and the recent CCCP algorithm for minimizing the Bethe free energy in approximate inference problems[17]. Bound optimization algorithms enjoy a strong guarantee; they never worsen the objective function.

2. Overrelaxed Bound Optimization: BO(η)

To guarantee an increase in the objective function at each iteration, BO methods must sometimes construct very conservative bounds, resulting in extremely slow convergence behavior. Below, we analyze a family of *overrelaxed* BO algorithms called BO(η) algorithms with η denoting the overrelaxation learning rate:

Naive Overrelaxed BO(η) algorithm for max $L(\Theta)$:
• **Iterate**: $\Theta^{t+1} = \Theta^t + \eta\big(\arg\max_\Theta G(\Theta, \Theta^t) - \Theta^t\big)$
• **No convergence guarantee.**
• **Difficult to set η in practice.**

Clearly, for $\eta = 1$ BO(η) algorithms become just regular bound optimizers. Several authors have studied a particular variant of this idea as applied to Expectation Maximization. In particular, Helmbold et al. (1995) [6] investigated the problem of estimating the component priors for a mixture of given densities, and discovered that an EM(η) update rule can be viewed as a first order approximation to the exponentiated gradient EG(η) update. Following this, Bauer et al. (1997) [1] presented an analysis of EM(η) similar

to [6] and derived the update rules for parameter estimation in discrete Bayesian networks. However, more general BO(η) methods have not been widely used for several reasons. First, only one particular variant of BO(η) is well studied: EM(η), and update rules have been published only in the special case of discrete Bayesian networks. Second, if a learning rate larger than optimal is used, BO(η) algorithms cannot guarantee convergence to even a local optimum of their objective function, in contrast to standard bound optimizers. Finally, it is computationally very difficult to obtain the optimal learning rate η^*.

In this paper, we analyze a broad class of BO(η) algorithms beyond EM(η) and show how one can design a simple adaptive algorithm that, in general, will possess superior convergence rates over standard BO(1) methods while at the same time guaranteeing convergence and avoiding the need to calculate an optimal learning rate η^*.

3. Convergence Properties of BO(1) & BO(η)

Standard bound optimization methods implicitly define a mapping: $M : \Theta \rightarrow \Theta'$ from parameter space to itself, such that $\Theta^{t+1} = M(\Theta^t)$. If the iterates Θ^t converge to Θ^* and $M(\Theta)$ is continuous, then $\Theta^* = M(\Theta^*)$, and in the neighborhood of Θ^*, by Taylor series expansion:

$$\Theta^{t+1} - \Theta^* = M'(\Theta^*)(\Theta^t - \Theta^*) \tag{1}$$

where $M'(\Theta^*) = \frac{\partial M}{\partial \Theta}|_{\Theta=\Theta^*}$. Since $M'(\Theta^*)$ is typically nonzero, then any bound optimizer is essentially a linear iteration algorithm with a convergence rate matrix $M'(\Theta^*)$.

For multidimensional vector Θ, a measure of the actual observed convergence rate is the "global" rate, defined as:

$$r = \lim_{t\rightarrow\infty} \frac{\|\Theta^{t+1}-\Theta^*\|}{\|\Theta^t-\Theta^*\|} \tag{2}$$

with $\| \cdot \|$ being Euclidean norm[11]. It is also well-known that under some regularity conditions $r = \lambda_{max}(M') \equiv$ the largest eigenvalue of $M'(\Theta^*)$. All of the eigenvalues of the convergence rate matrix $M'(\Theta^*)$ lie in the interval $[0, 1)$. Larger values of λ_{max} (as they approach unity) imply slower convergence.

BO(η) methods, just as standard bound optimizers, implicitly define a mapping: $\Phi : \Theta \rightarrow \Theta$ from parameter space to itself, such that $\Theta^{t+1} = \Phi(\Theta^t)$. In particular,

$$\begin{aligned} \Phi(\Theta^t) &= \Theta^t + \eta(\Theta_{BO}^{t+1} - \Theta^t) \\ &= \Theta^t + \eta(M(\Theta^t) - \Theta^t) \end{aligned} \tag{3}$$

We can now analyze convergence behavior of the BO(η) methods as well as their relationship to the standard BO(1) algorithms.

Lemma 1: If BO(η) iterates converge to Θ^* for any value of η, then $\Phi(\Theta^*) = \Theta^*$.

This follows from (3) due to the necessity of a fixpoint of the mapping M: $M(\Theta^*) = \Theta^*$.

Lemma 2: If BO(η) iterates converge to Θ^* and $\Phi(\Theta)$ and $M(\Theta)$ are differentiable in the parameter space Θ, then:

$$\Phi'(\Theta^t) = I - \eta(I - M'(\Theta^t)) \tag{4}$$

with I being identity matrix and $\Phi'_{ij}(\Theta^t) = \frac{\partial \Theta_i^{t+1}}{\partial \Theta_j^t}$ is the input-output derivative matrix for the BO(η) mapping. This can be shown by differentiating both sides of (3) with respect to Θ.

Equation (4) shows a very interesting relationship between convergence properties of BO(η) and its standard BO(1) counterparts. If the eigenvalues of $M'(\Theta^t)$ approach unity in the neighborhood of Θ^*, BO(1) algorithm will exhibit extremely slow convergence. In this case, larger values of η will in fact force the eigenvalues of $\Phi'(\Theta^t)$ to decrease, and thus result in faster global rate of convergence of the BO(η) algorithm.

Proposition 1: If BO(1) iterates converge to Θ^*, then within some neighborhood of Θ^*, BO(η) algorithm will converge to the local maximum of the objective function for any $0 < \eta < 2$.

Proposition 2: The optimal learning rate η^* is:

$$\eta^* = 2/(2 - \lambda_{max} - \lambda_{min}) \tag{5}$$

with λ_{max} and λ_{min} being the largest and smallest eigenvalues of $M'(\Theta^*)$. Moreover, $\eta^* \geq 1$.

We provide the proofs of both propositions in the appendix for the completeness. The important consequence of the above analysis is that for the typical real problems with $\lambda_{max} > 0$, the optimal learning rate is $\eta^* > 1$. Moreover, the global rate of convergence of BO(η) algorithms is upper bounded by the spectral radius ρ_η of $\Phi'(\Theta^*)$, which is defined as:

$$r \leq \rho_\eta = \max\{|1 - \eta(1 - \lambda_{max})|, |1 - \eta(1 - \lambda_{min})|\}$$

This implies that, within some neighborhood of Θ^*, BO(η) methods can significantly outperform standard BO(1) algorithms in terms of convergence. Indeed, after M iterations BO(η) will shrink the distance $\| \Theta - \Theta^* \|$ by a factor of ρ_η^M, whereas standard BO(1) will shrink it by ρ_1^M. This clearly constitutes exponential gain of $(\rho_\eta/\rho_1)^M$ in the vicinity of the local optimum.

The presented convergence results for the family of the BO(η) algorithms are only valid within some neighborhood of Θ^*, as opposed to BO(1) methods that are guaranteed to converge from any point in the parameter space. In the next section we show how we can easily overcome this problem by describing a simple adaptive BO algorithm.

4. Adaptive Overrelaxed Bound Optimization

Computing the optimal learning rate may be very expensive, since it requires knowledge of the minimum and maximum eigenvalues λ_{min} and λ_{max} of a particular mapping matrix that depends on the algorithm details, data set, and current parameters. Furthermore, this calculation is only valid in a very small neighborhood around a local optimum. Ideally, we would like to find the optimal learning rate in an adaptive fashion that is computationally inexpensive and valid everywhere. It is possible to perform a line search at each step to determine η^* [7]; however, this is quite expensive. We now describe a very simple adaptive overrelaxed bound optimization (ABO) algorithm that is guaranteed not to decrease the objective function at each iteration and requires only a very slight overhead in computation over regular BO(1) methods yet can often be many times faster.

Adaptive Overrelaxed Bound Optimization (ABO) for maximizing $L(\Theta)$:

- **Iterate** starting with $\eta = 1$:
  1. $\Theta_{BO}^{t+1} = \arg\max_{\Theta} G(\Theta, \Theta^t)$
  2. $\Theta^{t+1} = \Theta^t + \eta(\Theta_{BO}^{t+1} - \Theta^t)$
  3. If $L(\Theta^{t+1}) > L(\Theta^t)$ Then Increase η Else $\Theta^{t+1} = \Theta_{BO}^{t+1}$ and Decrease η
- **Guarantee**: $L(\Theta^{t+1}) = G(\Theta^{t+1}, \Theta^{t+1}) \geq G(\Theta^{t+1}, \Theta^t) \geq G(\Theta^t, \Theta^t) = L(\Theta^t)$

Note that for many objective functions, computing $\arg\max_{\Theta} G(\Theta, \Theta^t)$ also evaluates the function $L(\Theta^t)$ "for free", so that step 3 above can be efficiently interleaved between steps 1 and 2 with essentially no extra computation (except when the optimizer oversteps).

4.1. Reparameterization of Constrained Quantities

The description of the adaptive algorithm above assumes that the parameters being optimized are unconstrained. In many cases, parameters must remain non-negative (or positive definite), sum to unity, respect symmetries or other parameter tying constraints. In these situations, the appropriate update rules can be derived by first reparameterizing the optimization using unconstrained variables which are related to the original variables through a fixed (possibly nonlinear) mapping. As examples, we develop several cases that arise often in practice.

If parameter values Θ_j must be positive (e.g. variances), the overrelaxation step can be derived from the reparameterization $\Theta_j = \exp(\beta_j)$ and results in:

$$\Theta_j^{t+1} = \Theta_j^t \left(\frac{\Theta_{jBO}^{t+1}}{\Theta_j^t} \right)^{\eta} \tag{6}$$

For parameter values $\vec{\Theta}_j$ that represent a discrete distribution (e.g. mixing proportions in a mixture model, conditional probability tables for discrete quantities, or state transition probabilities in dynamic models), we reparameterize $\vec{\Theta}_j$ via softmax function $\vec{\Theta}_{ji} = \frac{\exp(\beta_{ji})}{\sum_{i=1} \exp(\beta_{ji})}$, and perform overrelaxation in the unconstrained $\vec{\beta}$ space. In the constrained space this corresponds to the update:

$$\vec{\Theta}_j^{t+1} = \frac{1}{Z} \vec{\Theta}_j^t \left(\frac{\vec{\Theta}_{jBO}^{t+1}}{\vec{\Theta}_j^t} \right)^{\eta} \tag{7}$$

using elementwise multiplication and division operations and with Z being an appropriate normalizing constant.

4.2. Adaptive Overrelaxed EM

We now consider a particular bound optimizer, the popular Expectation-Maximization (EM) algorithm and present its adaptive overrelaxed version. As an example, consider a probabilistic model of continuous observed data $\boldsymbol{x}$ which uses continuous latent variables $\boldsymbol{y}$. Then:

$$L(\Theta) = \ln p(\boldsymbol{x}|\Theta) = \int p(\boldsymbol{y}|\boldsymbol{x}, \Psi) \ln p(\boldsymbol{x}|\Theta) d\boldsymbol{y} =$$
$$\int p(\boldsymbol{y}|\boldsymbol{x}, \Psi) \ln p(\boldsymbol{x}, \boldsymbol{y}|\Theta) d\boldsymbol{y} - \int p(\boldsymbol{y}|\boldsymbol{x}, \Psi) \ln p(\boldsymbol{y}|\boldsymbol{x}, \Theta) d\boldsymbol{y}$$
$$= Q(\Theta, \Psi) - H(\Theta, \Psi)$$

Setting $\Psi = \Theta^t$, it can be easily verified that the following is the lower bound function:

$$L(\Theta) \geq Q(\Theta, \Psi) - H(\Psi, \Psi) = G(\Theta, \Psi) \tag{8}$$

The EM algorithm is nothing more than coordinate ascent in the functional $G(\Theta, \Psi)$, alternating between maximizing G with respect to Ψ for fixed Θ (E-step) and with respect to Θ for fixed Ψ (M-step). Our new adaptive overrelaxed version of EM is given below:

Adaptive Overrelaxed EM (AEM) algorithm:

- η=1; $L(\Theta^0)$ = -∞; δ=tol;
- While ($\delta \geq$tol and t<Tmax)
  - Perform E-step with Θ^t and get $L(\Theta^t)$
  - $\delta = (L(\Theta^t) - L(\Theta^{t-1}))/\text{abs}(L(\Theta^t))$
  - If $\delta <$tol /* We have gone too far */
    - $\eta = 1$; Perform E-step with Θ_{EM}^t
    - Get $L(\Theta_{EM}^t)$ and compute new δ;
    - /* Count this as an additional step */

    Else $\eta = \alpha * \eta$; EndIf
  - Perform M-step to get Θ_{EM}^{t+1}
  - $\Theta^{t+1} = \Theta^t + \eta(\Theta_{EM}^{t+1} - \Theta^t)$
- EndWhile

Dempster, Laird, and Rubin [3] showed that if EM iterates converge to Θ^*, then

$$\frac{\partial M(\Theta)}{\partial \Theta}|_{\Theta=\Theta^*} = \left[\frac{\partial^2 H(\Theta, \Theta^*)}{\partial \Theta^2}|_{\Theta=\Theta^*} \right] \left[\frac{\partial^2 Q(\Theta, \Theta^*)}{\partial \Theta^2}|_{\Theta=\Theta^*} \right]^{-1}$$

which can be interpreted as the ratio of missing information to the complete information near the local optimum. According to Lemma 2, the convergence rate matrix of EM(η) algorithm can be represented as follows: In the neighborhood of a solution (for sufficiently large t):

$$\Phi'(\Theta^t) = I - \eta \left[I - \left(\frac{\partial^2 H}{\partial \Theta^2} \right) \left(\frac{\partial^2 Q}{\partial \Theta^2} \right)^{-1} |_{\Theta=\Theta^t} \right] \tag{9}$$

This view of the EM(η) algorithm has a very interesting interpretation: *An increase in the proportion of missing information corresponds to higher values of the learning rate η.* If the fraction of missing information approaches unity, standard EM will be forced to take very small, conservative steps in parameter space, therefore higher and more aggressive values of η will result in much faster convergence. When the missing information is small compared to the complete information, the potential advantage of EM(η) over EM(1) becomes much less.

4.3. Generalized Iterative Scaling (GIS) Algorithm

The Generalized Iterative Scaling algorithm is widely used for parameter estimation in maximum entropy models [2]. Its goal is to determine the parameters Θ^* of an exponential family distribution $p(x|\Theta) = \frac{1}{Z(\Theta)} \exp(\Theta^T F(x))$ such that certain generalized marginal constraints are preserved: $\sum_x p(x|\Theta^*)F(x) = \sum_x \bar{p}(x)F(x)$, where $Z(\Theta)$ is the normalizing factor, $\bar{p}(x)$ is a given empirical distribution and $F(x) = [f_1(x), ..., f_n(x)]^T$ is a given feature vector. These types of problems can be expressed in a standard form, with $f_i(x) > 0 \; \forall i$, and $\sum_i f_i(x) = 1$ for each x. The log-likelihood function is:

$$L(\Theta) = \sum_x \bar{p}(x) \ln p(x|\Theta) = \sum_x \bar{p}(x)\Theta^T F(x) - \ln Z(\Theta)$$

By noting that $\ln Z(\Theta) \leq Z(\Theta)/Z(\Psi) + \ln Z(\Psi) - 1$ for all Ψ, and $\exp \sum_i \Theta_i f_i(x) \leq \sum_i f_i(x) \exp \Theta_i$ we can construct a lower bound on $L(\Theta)$:

$$L(\Theta) \geq \sum_x \bar{p}(x) \sum_i \Theta_i f_i(x) - \ln Z(\Psi) + 1 - \sum_x p(x|\Psi) \sum_i f_i(x) \exp(\Theta_i - \Psi_i) = G(\Theta, \Psi)$$

We can now derive an adaptive overrelaxed version of GIS:

Adaptive Overrelaxed GIS (AGIS) algorithm:

- η=1; $L(\Theta^0)$ = -∞; δ=tol;
- While ($\delta \geq$tol and t<Tmax)
  - $\Theta^t_{i_{GIS}} = \Theta^{t-1}_i + \ln \frac{\sum_x \bar{p}(x) f_i(x)}{\sum_x p(x|\Theta^t) f_i(x)}$
  - $\Theta^t = \Theta^{t-1} + \eta(\Theta^t_{GIS} - \Theta^{t-1})$ and get $L(\Theta^t)$
  - $\delta = (L(\Theta^t) - L(\Theta^{t-1}))/\text{abs}(L(\Theta^t))$
  - If $\delta <$tol /* We have gone too far */
    - $\eta = 1$; $\Theta^t = \Theta^t_{GIS}$
    - Get $L(\Theta^t_{GIS})$ and compute new δ;
    - /* Count this as an additional step */

    Else $\eta = \alpha * \eta$; EndIf
- EndWhile

It has been observed that the convergence of GIS algorithms depends on the correlation between features. In general, *an increase in the feature correlation corresponds to higher values of the learning rate η.* If feature vectors are highly correlated, GIS will take very small conservative steps in the parameter space. Thus higher and more aggressive values of η will result in much faster convergence.

4.4. Non-Negative Matrix Factorization (NMF)

Given non-negative matrix V, we are interested in finding non-negative matrices W and H, such that $V \approx WH$ [9]. Posed as an optimization problem, we are interested in maximizing a negative divergence $L(\Theta) = -D(V||WH)$, subject to $\Theta = (W, H) \geq 0$ elementwise, where:

$$L(\Theta) = -\sum_{ij} \left(V_{ij} \ln \frac{V_{ij}}{(WH)_{ij}} - V_{ij} + (WH)_{ij} \right) \quad (10)$$

We use $\ln \sum_c W_{ic}H_{cj} \leq \sum_c \alpha_{ij}(c,c) \ln \frac{W_{ic}H_{cj}}{\alpha_{ij}(c,c)}$ where $\alpha_{ij}(a,b) = W^t_{ia}H^t_{bj} / \sum_r W^t_{ir}H^t_{rj}$, so that $\alpha_{ij}(c,c)$ sum to one. Defining $\Theta = (W, H)$ and $\Psi = (W^t, H^t)$, we can construct a lower bound on the cost function:

$$L(\Theta) \geq -\sum_{ij} V_{ij} \ln V_{ij} + V_{ij} - \sum_{ijc} W_{ic}H_{cj} + \sum_{ijc} V_{ij}\alpha_{ij}(c,c) \left[\ln \frac{W_{ic}H_{cj}}{\alpha_{ij}(c,c)} \right] = G(\Theta, \Psi)$$

Adaptive overrelaxed NMF algorithm is then derived as:

Adaptive Overrelaxed NMF (ANMF) algorithm:

- η=1; $L(\Theta^0)$ = -∞; δ=tol;
- While ($\delta \geq$tol and t<Tmax)
  - $W^t_{icNMF} = W^{t-1}_{ic} \left[\frac{\sum_j H_{cj} V_{ic}/(WH)_{ic}}{\sum_v H_{cv}} \right]^\eta$

    $H^t_{cjNMF} = H^{t-1}_{cj} \left[\frac{\sum_i W_{ic} V_{ic}/(WH)_{ic}}{\sum_w W_{wc}} \right]^\eta$
  - $\Theta^t = \Theta^{t-1} + \eta(\Theta^t_{NMF} - \Theta^{t-1})$; and get $L(\Theta^t)$
  - $\delta = (L(\Theta^t) - L(\Theta^{t-1}))/\text{abs}(L(\Theta^t))$
  - If $\delta <$tol /* We have gone too far */
    - $\eta = 1$; Update W^t and H^t; get $L(\Theta^t_{NMF})$
    - Compute new δ;
    - /* Count this as an additional step */

    Else $\eta = \alpha * \eta$; EndIf
- EndWhile

For many models overrelaxation is straightforward to implements and does not require significant computational overhead. As we will see in the next section, it can substantially outperform standard bound optimization algorithms.

5. Experimental Results

We now present empirical results comparing the performance of adaptive overrelaxed bound optimizers to standard BO(1) algorithms for learning the model parameters. We begin by showing convergence results on synthetic data sets, since it makes it easier to interpret, display and analyze. We then proceed to reporting similar empirical results on the real world data sets, supporting our presented analysis. Though not reported, we confirmed that the convergence results presented below do not vary significantly for different initial starting points in the parameter space. For all of the experiments reported below, we used tol = 10^{-8} and $\alpha = 1.1$.

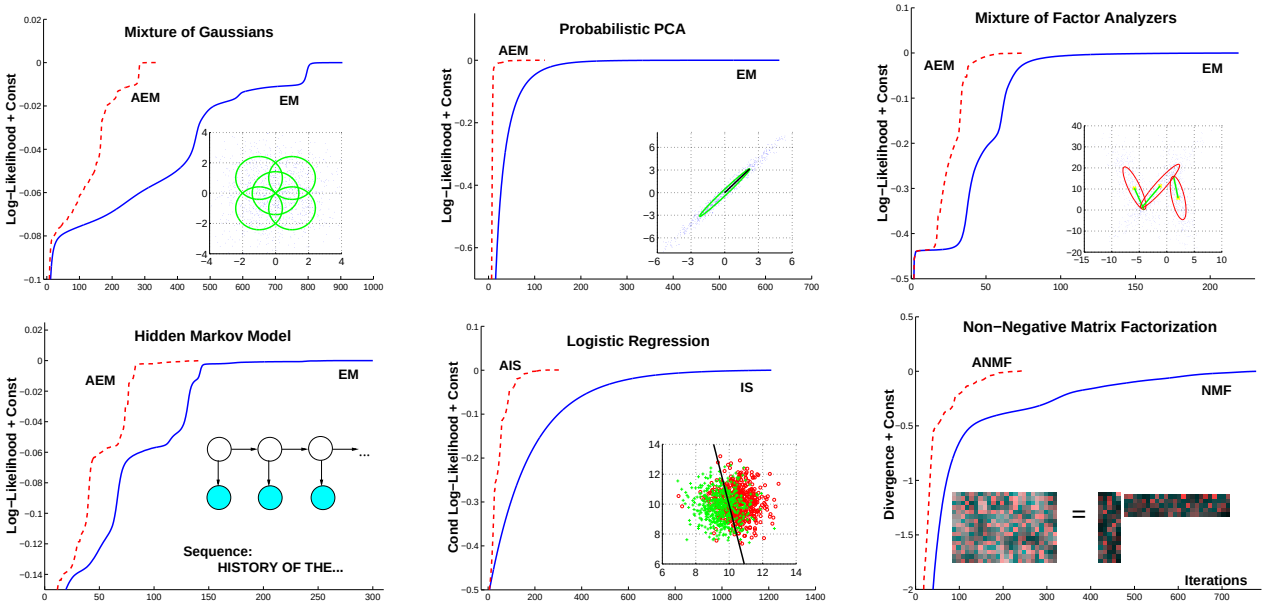

Figure 1. Learning curves for adaptive overrelaxed and standard bound optimization algorithms, showing convergence performance for different models. Upper panel displays MoG (left), PPCA (middle), MFA (right); and bottom panel displays HMM (left), logistic regression (middle), NMF (right). The iteration numbers are shown on the horizontal axis, and the value of the cost function is shown on the vertical axis, with zero-level corresponding to the converging point of the BO(1) algorithm.

5.1. Synthetic Data Sets

To compare AEM and EM algorithms, we considered several latent variable models: mixture of Gaussians (MoG), Hidden Markov Model (HMM), Probabilistic PCA (PPCA), and mixture of Factor Analyzers (MFA) models. As predicted by theory, high proportion of missing information in these models will result in slow convergence of EM, more aggressive learning rates η and thus superior performance of AEM.

First, consider a mixture of Gaussians (MoG) model. The data was generated from 5 Gaussian mixture components (see Fig 1). In this model the proportion of missing information corresponds to how "well" or "not-well" data is separated into distinct clusters. Note that in the considered data set, mixture components overlap in one contiguous region, which constitutes the large proportion of missing information. Figure 1 shows that AEM outperforms standard EM algorithm by almost a factor of three.

We then applied our algorithm to the training of Hidden Markov Model. Missing information in this model is high when the observed data do not well determine the underlying state sequence (given the parameters). A simple 5-state fully-connected model was trained on 41 character sequences from the book "Decline and Fall of the Roman Empire" by Gibbon, with an alphabet size of 30 characters (parameters were randomly initialized). We observe that even for the real, structured data AEM is superior to EM.

We also experimented with the Probabilistic Principal Component Analysis (PPCA) latent variable model[13, 15], which has continuous rather than discrete hidden variables. Here the concept of missing information is related to the ratios of the leading eigenvalues of the sample covariance, which corresponds to the ellipticity of the distribution. We observe that even for "nice" data, AEM outperforms EM by almost a factor of four. Similar results are displayed in figure 1 for the MFA [5] model.

As a confirmation to our analysis, in figure 3 we show the evolution of the adaptive learning rate η and the optimal learning rate η^* during fitting the means of the four mixture components in the MoG model, holding the mixing proportions and covariances fixed. The optimal learning rate was obtained by calculating λ_{min} and λ_{max} eigenvalues of $M'(\Theta^*)$ matrix and applying equation 5.

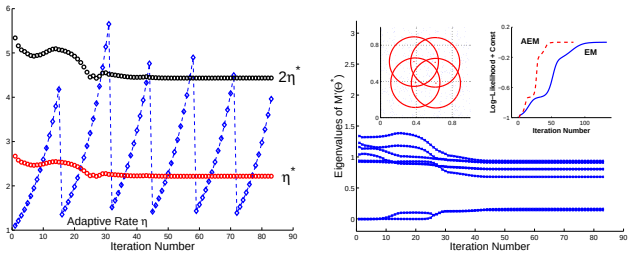

Figure 3. Left panel pictorially illustrates the adaptive learning rate η, the optimal rate η^*, and the approximate upper bound on the learning rate $2\eta^*$. The right panel shows the evolution of the eigenvalues of the convergence rate matrix $M'(\Theta^*)$ in eq.(1).

To compare IS and adaptive IS algorithms, we applied both methods to the simple 2-class logistic regression model: $p(y = \pm 1|x, w) = 1/(1 + \exp(-yw^T x))$ [12]. Feature

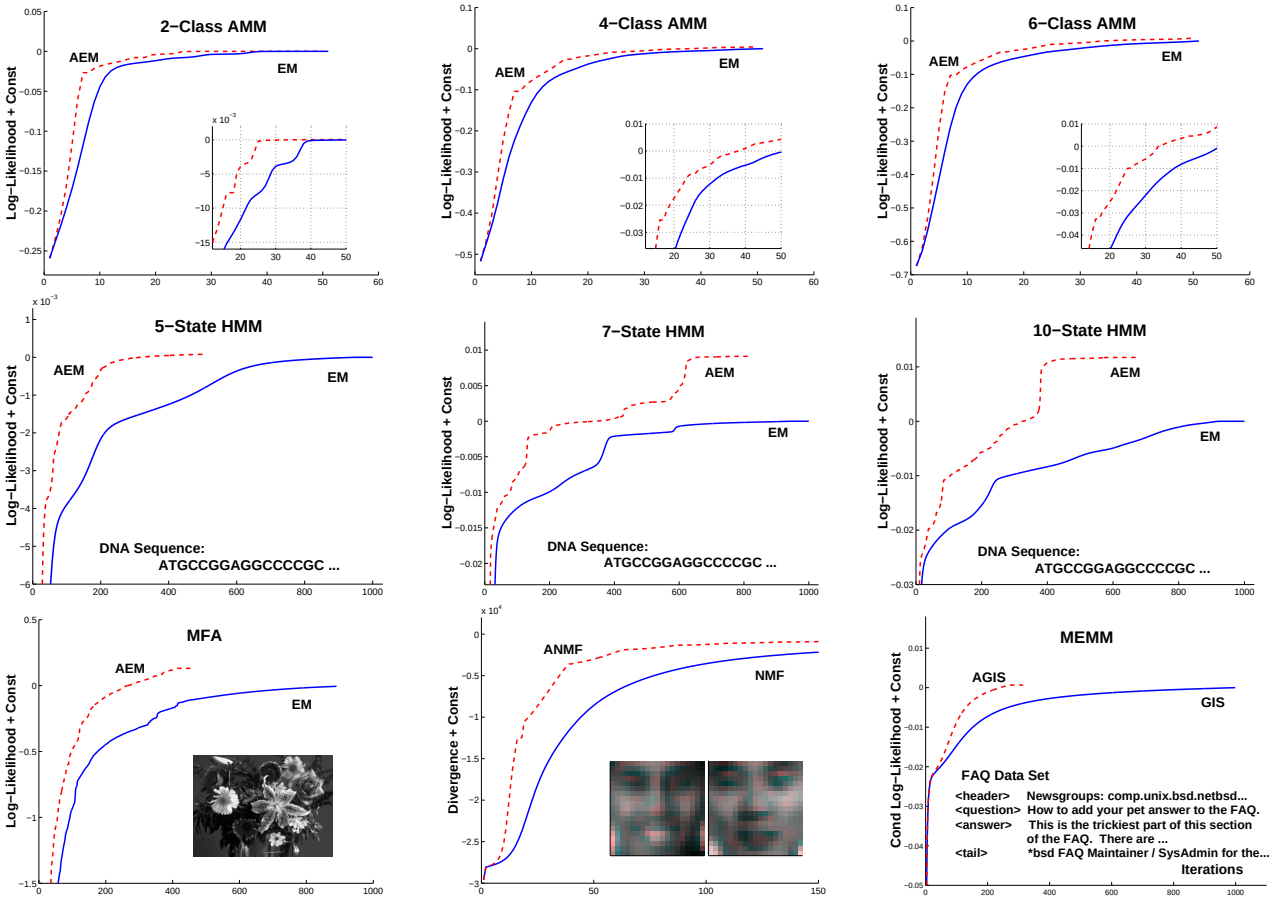

Figure 2. Learning curves for adaptive overrelaxed and standard bound optimization algorithms, showing convergence performance for different models. Upper panel displays AMM model with number of learned classes being: 2 (left), 4 (middle), and 6 (left). Middle panel shows HMM model with number of states being: 5 (left), 7 (middle), and 10 (right). Lower panel displays MFA (left), NMF (middle), and MEMM (right). The iteration numbers are shown on the horizontal axis, and the value of the cost function is shown on the vertical axis, with zero-level corresponding to the converging point of the BO(1) algorithm.

vectors of dimensionality d were drawn from standard normal: $x \sim \mathcal{N}(0, I_d)$, with true parameter vector w^* being randomly chosen on surface of the d-dimensional sphere with radius $\sqrt{2}$. To make features positive, the data set was modified by adding 10 to all feature values. This in term introduces significant correlation, and thus results in slow convergence of IS. To insure that $w^t x$ is unchanged, an extra feature was added. Figure 1 reveals that for d=2, AIS is superior to standard IS by at least a factor of 3. Similar results are obtained if dimensionality of the data is increased.

At last, to compare ANMF and NMF, we randomly initialized the non-negative matrix $V_{16\times24}$, and applied both algorithms to perform non-negative factorization: $V_{16\times24} \approx W_{16\times5} H_{5\times24}$. Once again, results confirm the fact that overrelaxed methods can give speedup over conventional bound optimizers by several orders of magnitude!

5.2. Real World Data Sets

To compare AEM and EM, our first experiment consisted of training Aggregate Markov models AMM [14] on the ARPA North American Business News (NAB) corpus, kindly provided to us by Lawrence Saul. AMMs are class-based bigram models in which the mapping from words to classes is probabilistic. The task of AMMs is to discover "soft" word classes. The experiment used a vocabulary of sixty-thousand words, including tokens for punctuation, sentence boundaries, etc. The training data consisting of approximately 78 million words (three million sentences), with all sentences drawn without replacement from the NAB corpus. The number of classes was set C=2,4,6 and all parameter values were randomly initialized.[1] Figure 2 (upper panels) reveals that AEM outperforms EM by at least a factor of 1.5. The considered data sets contains rather structured real data, suggesting relatively small fraction of the missing information. Nevertheless, to perform fair comparison, we have run both algorithms until the convergence criterion: $(L(\Theta^{t+1}) - L(\Theta^t))/\text{abs}(L(\Theta^{t+1})) \leq 10^{-8}$ is met. Setting the number of classes to 2, EM has converged in 164 iterations, whereas AEM has converged

[1] For the details of the model and the data set, refer to [14].

to exactly the same likelihood only after 72 iterations. This clearly constitutes a gain of a factor of over two.

Our second experiment consisted of training a fully connected HMM to model DNA sequences. For the training, we used publicly available "GENIE gene finding data set", provided by UCSC and LBNL [4], that contains 793 unrelated human genomic DNA sequences. We applied our AEM algorithm to 66 DNA sequences with length varying anywhere between 200 to 3000 genes per sequence. The number of states were set to 5,7, and 10 and all the parameter values were randomly initialized. Figure 2 shows superior convergence of AEM over EM algorithm. In this case the considered data set contains very little overall structure, which constitutes high proportion of the missing information.

We have also applied the MFA model to the block transform image coding problem. A data set of 360×496 pixel images (see fig 2 bottom left panel) were subdivided into nonoverlapping blocks of size 8×8 pixels. Each block was regarded as a d=8×8 dimensional vector. The blocks (total of 2,790) were then compressed down to five dimensions using 10 mixture components.[2] Once again, AEM beats EM by a factor of over two, converging to the better likelihood.

To present the comparison between GIS and AGIS, we trained Maximum Entropy Markov Model [10] on the Frequently Asked Questions (FAQ) data set. The data set consisted of 38 files belonging to 7 Usenet groups. Each file contains header, followed by a series of one or more question/answer pairs, and ends with tail. The goal is to automatically label each line according to whether it is header, question, answer, or tail by using 24 boolean features of lines, like *begin-with-number, contains-http*, etc.[3] We observe that AGIS outperforms GIS by several orders of magnitude. We have also obtained analogous results training Conditional Random Fields [8].

In our last experiment, we trained NMF and adaptive NMF on the data set of facial images to learn part-based representation of faces [9]. The data set consisted of m=2,429 facial images, each consisting of n=19×19 gray pixels, thus forming an $n \times m$ matrix V. In this experiment, the number of learned basis images were set to 49.[4] Once again, figure 2 reveals that ANMF substantially outperforms standard NMF algorithm. In particular, ANMF has converged in only about 3,500 iterations until the convergence criterion is met, whereas NMF converged in approximately 13,500 iterations to exactly the same value of the cost function, loosing to ANMF by a factor of almost four.

[2] This experiment is similar to the one described in [16].

[3] See [10] for the description of the model and the data set.

[4] See [9] for the detailed description of the experiment.

6. Discussion & Future Work

In this paper we have analyzed the convergence properties of a large family of overrelaxed bound optimization BO(η) algorithms. Based on this analysis, we introduced a novel class of simple, adaptive overrelaxed bound optimization (ABO) methods and provided empirical results on several synthetic as well as real-world data sets showing superior convergence of the ABO methods over standard bound optimizers.

We have also experimented with models where parameter values Θ represent symmetric positive definite matrices (e.g. covariance matrices in the MoG model). We can use the matrix exponential $\Theta = \exp \Lambda$ to perform overrelaxation in the unconstrained Λ space. In particular, we use a spectral decomposition: $\Theta = VDV^T$, with D being the diagonal matrix of the eigenvalues, and V being the orthogonal matrix of the corresponding eigenvectors. The matrix functions $\ln$ and $\exp$ are then defined: $\ln \Theta = V \ln(D) V^T$, and $\exp \Theta = V \exp(D) V^T$. When the matrix is diagonal, the overrelaxation corresponds to equation (6).

In all of our experiments with adaptive algorithms we found that the value of objective function at any iterate was better than the value at the same iterate of the standard bound optimizer: $L(\Theta^t_{ABO}) > L(\Theta^t_{BO}) : \forall t$. In other words, we have never found a disadvantage to using adaptive methods; and often there is a large advantage, particularly for complex models with large training data sets, where due to the time constraints one could only afford to perform a few number of the BO iterations.

Acknowledgments

We would like to thank Zoubin Ghahramani for many useful discussions and Lawrence Saul for providing us with ARPA North American Business News corpus.

References

[1] Eric Bauer, Daphne Koller, and Yoram Singer. Update rules for parameter estimation in bayesian networks. In *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence (UAI-97)*, pages 3–13, August 1–3 1997.

[2] Stephen Della Pietra, Vincent J. Della Pietra, and John D. Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, 1997.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J. of the RS Society series B*, 39:1–38, 1977.

[4] GENIE gene data set. LBNL and UC Santa Cruz, `http://www.fruitfly.org/sequence`.

[5] Zoubin Ghahramani and Geoffrey Hinton. The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, University of Toronto, May 1996.

[6] D. Helmbold, R. E. Schapire, Y. Singer, and M. K. Warmuth. A comparison of new and old algorithms for a mixture estimation problem. *Machine Learning*, pages 97–119, 1997.

[7] Mortaza Jamshidian and Robert I. Jennrich. Conjugate gradient acceleration of the EM algorithm. *Journal of the American Statistical Association*, 88(421):221–228, March 1993.

[8] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289, 2001.

[9] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Letters to Nature*, 401:788–791, 1999.

[10] Andrew McCallum, Dayne Freitag, and Fernando Pereira. Maximum entropy Markov models for information extraction and segmentation. In *Proc. 17th International Conf. on Machine Learning*, pages 591–598, 2000.

[11] X. L. Meng and D. B. Rubin. On the global and componentwise rates of convergence of the EM algorithm. *Linear Algebra and Its Applications*, 199:413–425, 1994.

[12] Tom Minka. Algorithms for maximum-likelihood logistic regression. Technical Report 758, Dept. of Statistics, Carnegie Melon University, 2001.

[13] S. T. Roweis. EM algorithms for PCA and SPCA. In *Advances in neural information processing systems*, volume 10, pages 626–632, Cambridge, MA, 1998. MIT Press.

[14] Lawrence Saul and Fernando Pereira. Aggregate and mixed-order Markov models for statistical language processing. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 81–89. Association for Computational Linguistics, 1997.

[15] M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analysers. *Neural Computation*, 11(2):443–482, 1999.

[16] Naonori Ueda, Ryohei Nakano, Zoubin Ghabramani, and Geoffrey E. Hinton. SMEM algorithm for mixture models. *Neural Computation*, 12(9):2109–2128, 2000.

[17] Alan Yuille and Anand Rangarajan. The convex-concave computational procedure (CCCP). In *Advances in Neural Information Processing Systems*, volume 13, 2001.

Appendix

Proof of Proposition 1:[5] First, by using Taylor series expansion of $\Phi(\Theta^t)$ around Θ^*, we have:

$$\Phi(\Theta^t) = \Phi(\Theta^*) + \Phi'(\Theta^*)(\Theta^t - \Theta^*) + ... \tag{11}$$

In the vicinity of Θ^* (for sufficiently large t), we have the following linear approximation:

$$\Theta^{t+1} - \Theta^* = \Phi(\Theta^t) - \Phi(\Theta^*) = \Phi'(\Theta^*)(\Theta^t - \Theta^*) \tag{12}$$

[5]Our proof is similar in spirit to [1].

For a fixed η, consider $\gamma_1, ..., \gamma_k$ being the non-zero eigenvalues of $\Phi'(\Theta^*)$. The corresponding eigenvalues $v_1, ..., v_k$ form an orthonormal basis for the real k-subspace Ω.[6] Assume that within some neighborhood of Θ^* (for sufficiently large t), $(\Theta^t - \Theta^*) \in \Omega$, in which case a vector $(\Theta^t - \Theta^*)$ can be represented uniquely as $(\Theta^t - \Theta^*) = \sum_{i=1}^k c_i v_i$. Moreover

$$\Phi'(\Theta^*)(\Theta^t - \Theta^*) = \textstyle\sum_{i=1}^k \gamma_i c_i v_i \tag{13}$$

The application of $\Phi'(\Theta^*)$ to $(\Theta^t - \Theta^*)$ results in linear coefficients c_i to be scaled by γ_i. Therefore the rate at which different components of $(\Theta^t - \Theta^*)$ shrink or stretch depends on the size of the eigenvalues γ_i. To guarantee the shrinkage of each component of $(\Theta^t - \Theta^*)$, we require $|\gamma_i| < 1$ for i=1,...k. In this case:

$$\| \Phi'(\Theta^*)(\Theta^t - \Theta^*) \| = \sqrt{\textstyle\sum_{i=1}^k \gamma_i^2 c_i^2 v_i^T v_i} \leq$$
$$\rho_\eta \sqrt{\sum_{i=1}^k c_i^2 v_i^T v_i} = \rho_\eta \| \Theta^t - \Theta^* \| \tag{14}$$

with $\| * \|$ denoting Euclidean norm.[7] and ρ_η being the spectral radius of $\Phi'(\Theta^*)$: ρ_η=max$|\gamma|$. Clearly, the spectral radius of $\Phi'(\Theta^*)$ is defined as:

$$\rho_\eta \quad = \quad \max\{|1 - \eta(1 - \lambda_{max})|, |1 - \eta(1 - \lambda_{min})|\}$$

with λ_{max} and λ_{min} being the largest and smallest eigenvalues of $M'(\Theta^*)$. And thus for any $0 < \eta < 2$ we have $\rho_\eta < 1$. We can now analyze the global rate of convergence of $\Phi'(\Theta^t)$ in the neighborhood of Θ^*. In particular, for sufficiently large t and any $0 < \eta < 2$:

$$\begin{aligned} r \quad &= \quad \frac{\| \Theta^{t+1} - \Theta^* \|}{\| \Theta^t - \Theta^* \|} = \frac{\| \Phi'(\Theta^*)(\Theta^t - \Theta^*) \|}{\| \Theta^t - \Theta^* \|} \\ &\leq \quad \frac{\rho_\eta \| \Theta^t - \Theta^* \|}{\| \Theta^t - \Theta^* \|} = \rho_\eta < 1 \end{aligned} \tag{15}$$

Therefore in the vicinity of Θ^* for any $0 < \eta < 2$, BO(η) algorithm is guaranteed to converge to the local optimum of the objective function.

Proof of Proposition 2: We have established that the global rate of convergence of the BO(η) algorithms is upper bounded by:

$$r \leq \max\{|1 - \eta(1 - \lambda_{max})|, |1 - \eta(1 - \lambda_{min})|\}$$

Clearly, the fastest uniform global rate of convergence is obtained when $|1 - \eta(1 - \lambda_{max})| = |1 - \eta(1 - \lambda_{min})|$. We can now easily derive that $\eta^* = 2/(2 - \lambda_{max} - \lambda_{min})$. Since $0 \leq \lambda_{min} \leq \lambda_{max} < 1$, we have

$$\eta^* = 2/(2 - \lambda_{max} - \lambda_{min}) \geq 1 \tag{16}$$

[6]We note that $M'(\Theta^*)$ has exactly the same eigenvectors with eigenvalues defined as $\lambda_i = 1 - (1 - \gamma_i)/\eta$ for $i = 1, ..., k$.

[7]This argument is valid for any norm defined on the k-dimensional Euclidean space. However, for the sake of simplicity of the proof, we reserve to the Euclidean norm.