Hybrid Interior Point Training of Modular Neural Networks

Technical Report of the ISIS Group

at the University of Notre Dame

ISIS-96-007

Month, Year

Peter T. Szymanski, Michael Lemmon, and Christopher J. Bett

Department of Electrical Engineering

University of Notre Dame

Notre Dame, IN 46556

Interdisciplinary Studies of Intelligent Systems

Hybrid Interior Point Training of Modular Neural Networks

Peter T. Szymanski, Michael Lemmon, and Christopher J. Bett*

Dept. of Electrical Engineering

University of Notre Dame

Notre Dame, IN 46556

December 20, 1996

Abstract

Modular neural networks use a single gating neuron to select the outputs of a collection of agent neurons. Expectation-maximization (EM) algorithms provide one way of training modular neural networks to approximate nonlinear functionals. This paper introduces a hybrid interior-point (HIP) algorithm for training modular networks. The HIP algorithm combines an interior-point linear programming (LP) agorithm with a Newton-Raphson iteration in such a way that the computational efficiency of the interior-point LP methods is preserved. The algorithm is formally proven to converge asypmtotically to locally optimal networks with a total computational cost that scales in a polynomial manner with problem size. Simulation experiments show that the HIP algorithm produces networks whose average approximation error is better than that of EM-trained networks. These results also demonstrate that the computational cost of the HIP algorithm scales at a slower rate than the EM-procedure; and that for small-size networks, the total computational costs of both methods are comparable.

Keyworks: modular neural networks, training, algorithms, interior-point methods, expectation-maximization methods

*The authors would like to acknowledge the partial financial support of the National Science Foundation (MSS92-16559), the Electric Power Research Institute (RP8030-06), and the Army Research Office (DAAH04-95-1-0600, DAAH04-96-0134)

1 Introduction

An artificial neural network is a distributed computing paradigm consisting of a set of self-similar processing units called *neurons*. In many neural network architectures, the network's output, $\hat{y}$, has the form

$$\hat{y}(\bar{\mathbf{z}}) = \sum_i f(\bar{\phi}_m^T \bar{\mathbf{z}}) \tag{1}$$

where $\bar{\phi}_m$ and $\bar{\mathbf{z}}$ are vectors in $\mathbb{R}^R$ and f is a mapping from $\mathbb{R}$ into $\mathbb{R}$. Given a *target mapping*, $y : \mathbb{R}^R \to R$, we are interested in designing a network whose outputs form an approximation, $\hat{y}$, of the target mapping. Neural networks are well-known to form good approximations of complex input-output mappings. This *universal approximation* [7] [2] [12] ability of neural networks is probably one of the most important reasons for their use. In the networks represented by equation (1), each neuron contributes to the network's output. In certain cases, however, it is reasonable to expect that the structure of the target mapping will be different in disjoint regions of the input space. In such cases, there is little benefit to be gained by additively mixing the outputs of the neurons. It might be better to use the output of a single neuron to exclusively compute the network approximation. Such networks can be viewed as a collection of *experts* where each neuron is an expert at approximating the target function over a region in $\mathbb{R}^R$. These *mixture of expert* or *modular* networks were first introduced in [8].

The training of modular networks can be carried out using *Expectation-Maximization* (EM) procedures [3] [9]. Results in [14] raised concerns about the applicability of EM algorithms due to their linear convergence rate. That work called for the use of more sophisticated optimization methods whose computational costs are well understood. This paper presents just such an algorithm for the training of modular networks using Gaussian activation functions to form piecewise linear approximations. The algorithm is based on a modification of a primal interior point linear programming (LP) algorithm [5] [10]. The resulting algorithm combines an interior-point LP algorithm with a Newton-Raphson (NR) iteration in a way that preserves the computational efficiency of the original interior-point LP algorithm. The resulting algorithm is referred to as the *hybrid interior point* or HIP algorithm. A small-step version of the HIP procedure was first presented in [11]. One principal contribution of this paper is a formal analysis of the HIP procedure's asymptotic behaviour and computational complexity. The second principal contribution of this paper is a detailed simulation which compares the performance of the HIP algorithm and Expectation-Maximization procedure.

The remainder of this paper is organized as follows. Modular neural networks are introduced in section 2. THe HIP algorithm is presented in section 3. Theoretical results on the HIP algorithm's asymptotic behaviour and computational costs are summarized in section 4. Formal proofs for the results in section 4 will be found in appendix A. Detailed simulation experiments comparing the HIP and EM algorithms will be found in section 5. This section also examines the use of the HIP algorithm in training a modular network to model the input/output behaviour of a fossil fuel electric power generating plant. A final discussion will be found in section 6.

2 Modular Neural Networks

This paper adopts a viewpoint in which neural network training is treated as a function approximation problem [13]. Let $y : \mathbb{R}^R \rightarrow \mathbb{R}$ be a continuous function from $\mathbb{R}^R$ into $\mathbb{R}$ which we call the *target mapping*. It is assumed that y is unknown, except at a finite set of input/output pairs. The set of N input/output pairs is denoted as

$$\mathcal{T} = \{(\bar{\mathbf{z}}_1, y(\bar{\mathbf{z}}_1)), (\bar{\mathbf{z}}_2, y(\bar{\mathbf{z}}_2)), \ldots, (\bar{\mathbf{z}}_N, y(\bar{\mathbf{z}}_N))\} \quad , \tag{2}$$

where $\bar{\mathbf{z}}_i \in \mathbb{R}^R$ $(i = 1, \ldots, N)$ is the *input* and $y(\bar{\mathbf{z}}_i) \in \mathbb{R}$ is the associated *output*. The set, $\mathcal{T}$, will usually be referred to as the *training set*. The objective of the *training problem* is to find a functional, $\hat{y} : \mathbb{R}^R \rightarrow \mathbb{R}$ from a known parameterized set of functionals, $\mathcal{Y}$, which minimizes the *approximation error's* size over the training set. In this paper, the approximation will be realized by a special class of modular neural network.

The structure of a *modular network* is shown in Figure 1. The network uses a *gating neuron* or *gate* to select between the approximations generated by a collection of *agent neurons* or *agents*. Let $\bar{\mathbf{z}} \in \mathbb{R}^R$ be the input to the network. In response to this input, each agent produces an ordered pair of outputs, $(a_m(\bar{\mathbf{z}}), \hat{y}_m(\bar{\mathbf{z}}))$. $a_m : \mathbb{R}^R \rightarrow \mathbb{R}$ is a functional from the input space, $\mathbb{R}^R$, into the non-negative Reals. The functional measures the *activity level* of the mth agent in response to an input $\bar{\mathbf{z}} \in \mathbb{R}^R$. $\hat{y}_m : \mathbb{R}^R \rightarrow \mathbb{R}$ is a functional from the input space into the Reals representing the mth agent's approximation of the target mapping at $\bar{\mathbf{z}}$. The gate chooses that agent with the largest activity level and the selected agent's approximation is then used as the network's output. Mathematically, the network's output in response to $\bar{\mathbf{z}}$ is written as

$$\hat{y}(\bar{\mathbf{z}}) = \hat{y}_m(\bar{\mathbf{z}}) \tag{3}$$

$$m = \arg \max_{1 \leq m \leq M} a_m(\bar{\mathbf{z}}) \tag{4}$$

Linear Gaussian (LG) modular networks arise when the approximation function is linear and the activation functions are Gaussian. In particular, we assume that the mth agent's activation function has the form,

$$a_m(\bar{\mathbf{z}}) = q(m) e^{s\|\bar{\mathbf{z}} - \bar{\omega}_m\|^2} \tag{5}$$

where $q(m) \in \mathbb{R}$, $\bar{\omega}_m \in \mathbb{R}^R$, s is a negative real number, and $\sum_m q(m) = 1$. The activation function is parameterized by the scalar $q(m)$ and the R-vector $\bar{\omega}_m$. The mth agent's approximation function has the form

$$\hat{y}_m(\bar{\mathbf{z}}) = \bar{\phi}_m^T \bar{\mathbf{z}} \tag{6}$$

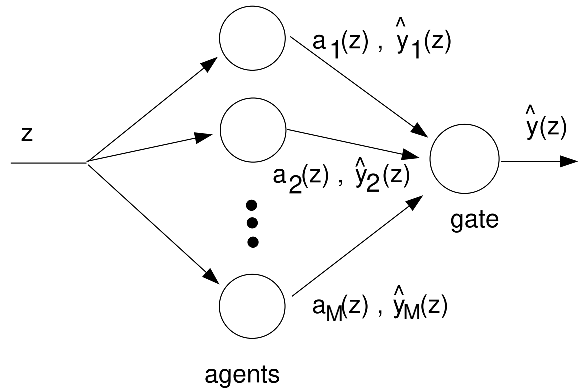

Figure 1: Modular Neural Network

where $\bar{\phi}_m \in \mathrm{IR}^R$. This approximation is parameterized by the R-vector $\bar{\phi}_m$. We therefore see that the mth agent is parameterized by the scalar $q(m)$ and the two R-vectors $\bar{\omega}_m$ and $\bar{\phi}_m$. In our later discussion, it will be convenient to aggregate these parameter vectors $\bar{\omega}_m$ and $\bar{\phi}_m$ into a single parameter vector $\bar{\theta}_m \in \mathrm{IR}^{2R}$. In particular, we will let $\bar{\theta}_m = \begin{bmatrix} \bar{\phi}_m^T & \bar{\omega}_m^T \end{bmatrix}^T$. The vectors,

$$\bar{\Theta} = \begin{bmatrix} \bar{\theta}_1^T & \bar{\theta}_2^T & \ldots & \bar{\theta}_M^T \end{bmatrix}^T \tag{7}$$

$$\bar{q} = \begin{bmatrix} q(1) & q(2) & \ldots & q(M) \end{bmatrix}^T \tag{8}$$

provide a parameterization $(\bar{q}, \bar{\Theta})$ of the entire network.

The squared *output* approximation error over training set $\mathcal{T} = \{(\bar{\mathbf{z}}_j, y(\bar{\mathbf{z}}_j)\}_{j=1,\ldots,N}$ is denoted as

$$d^2_{\text{out}}(\bar{q}, \bar{\Theta}) = \sum_{j=1}^{N} (y(\bar{\mathbf{z}}_j) - \hat{y}(\bar{\mathbf{z}}_j))^2 \tag{9}$$

$$= \sum_{j=1}^{N} \sum_{m=1}^{M} Q(m|j)(y(\bar{\mathbf{z}}_j) - \bar{\phi}_m^T \bar{\mathbf{z}}_j)^2 \tag{10}$$

The second equation arises by expanding $\hat{y}(\bar{\mathbf{z}}_j)$ and letting

$$Q(m|j) = \begin{cases} 1 & \text{if } a_m(\bar{\mathbf{z}}_j) \leq a_n(\bar{\mathbf{z}}_j) \text{ for all } 1 \leq n \leq M \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

The vector

$$\bar{Q} = \begin{bmatrix} Q(1|1) & Q(1|2) & \cdots & Q(m|j) & \cdots & Q(M|N) \end{bmatrix}^T \tag{12}$$

of $Q(m|j)$ introduced in equation 11 will be called the network's *selection strategy* since it denotes which agent is associated with which input vector. Note that for a given network parameterization,

$(\bar{q}, \bar{\Theta})$, that there will be a unique selection strategy, $\bar{Q}$. This means that the network can also be parameterized by $(\bar{Q}, \bar{\Theta})$.

A naive approach to the training problem would involve minimizing $d^2_{\mathrm{out}}(\bar{q}, \bar{\Theta})$ with respect $\bar{q}$ and $\bar{\Theta}$. This problem, however, is ill-posed [6] in the sense that solutions may not be continuous with respect to the data given in the training set. To obtain a well-posed training problem, it is essential that the original output-error criterion be augmented with a regularization term. One common regularization term presents the error of the mth agent on the jth input as

$$d^2_m(j) = \left(y(\bar{\mathbf{z}}_j) - \bar{\phi}^T_m \bar{\mathbf{z}}_j\right)^2 + \|\bar{\omega}_m - \bar{\mathbf{z}}_j\|^2 \tag{13}$$

The total network error then has the form,

$$d^2(\bar{q}, \bar{\Theta}) = \frac{1}{N} \sum_{j=1}^{N} \sum_{m=1}^{M} Q(m|j) d^2_m(j) \tag{14}$$

where $Q(m|j)$ is as defined in equation 11.

The training problem involves minimizing $d^2(\bar{q}, \bar{\Theta})$ with respect to $\bar{q}$ and $\bar{\Theta}$. With respect to parameterization, $(\bar{q}, \bar{\Theta})$, this is a nonlinear optimization problem which can be very difficult to solve. By reparameterizing the network in terms of the selection strategies (i.e. the network is parameterized by $(\bar{Q}, \bar{\Theta})$), we obtain a problem which is linear in $\bar{Q}$ and quadratic in $\bar{\Theta}$. The resulting minimization problem can now be stated as follows. It is first assumed that there exists a design set consisting of N training points, $\mathcal{T} = \{(y_j, \bar{\mathbf{z}}_j) \in \mathbb{R} \times \mathbb{R}^R, j = 1, \ldots, N\}$ where y_j represents a desired output in response to an input, $\bar{\mathbf{z}}_j$. The training problem is

Problem 1

$$\begin{aligned} \textit{minimize:} \quad & \textstyle\sum_{j=1}^{N} \sum_{m=1}^{M} \frac{1}{N} Q(m|j) \left[\|\bar{\mathbf{z}}_j - \bar{\omega}_m\|^2 + (y_j - \bar{\phi}^T_m \bar{\mathbf{z}}_j)^2\right] \\ \textit{with respect to:} \quad & Q(m|j), \bar{\omega}_m, \bar{\phi}_m (m = 1, \ldots, M; j = 1, \ldots, N) \\ \textit{subject to:} \quad & Q(m|j) \geq 0 \\ & \textstyle\sum_{m=1}^{M} Q(m|j) = 1, (j = 1, \ldots, N) \end{aligned}$$

where $Q(m|j) \in \mathbb{R}$, $\bar{\omega}_m \in \mathbb{R}^R$, and $\bar{\phi}_m \in \mathbb{R}^R$ $(m = 1, \ldots, M; j = 1, \ldots, N)$. The problem is to determine the set of parameters, $\left[\bar{Q}^T, \bar{\Theta}^T\right]^T \in \mathbb{R}^{MN} \times \mathbb{R}^{2MR}$, which minimize the objective where

$$\bar{Q}^T = [Q(1|1), Q(1|2), \ldots, Q(m|j), \ldots] \tag{15}$$

and

$$\bar{\Theta}^T = \left[\theta^T_1, \theta^T_2, \ldots, \theta^T_M\right] \tag{16}$$

with $\theta_m^T = \left[\bar{\phi}_m^T, \bar{\omega}_m^T\right] \in \mathrm{I\!R}^R \times \mathrm{I\!R}^R$. Note that this problem is convex with respect to $\bar{Q}$ and $\bar{\Theta}$, separately. The special structure of this problem means that it can be solved by using Expectation-Maximization (EM) algorithms or by using the HIP algorithm presented in the next section.

Once the network error has been minimized with respect to selection strategies, $\bar{Q}$, and network parameters, $\bar{\Theta}$, then the other network parameters, $\bar{q}$, need to be determined. This can be done as follows. Recall that $Q(m|l)$ is unity if the activity level

$$a_m(\bar{\mathbf{z}}) = q(m)e^{s\|\bar{\omega}_m - \bar{\mathbf{z}}_j\|^2} \tag{17}$$

for the mth agent (assuming input $\bar{\mathbf{z}}_j$) is larger than any other agent's activity level. Let $e_m(j) = \|\bar{\omega}_m - \bar{\mathbf{z}}_j\|^2$. Note that $Q(m|j)$ can be rewritten as

$$Q(m|j) = \lim_{s \to -\infty} \frac{q(m)e^{se_m(j)}}{\sum_l q(l)e^{se_l(j)}} \tag{18}$$

If $e_m(j)$ is smaller than the other $e_l(j)$, then we can easily show that the above limit is unity. If this is not the case, then the limit is zero. We now sum over all j to obtain

$$\begin{aligned}
\sum_j Q(m|j) &= \lim_{s \to -\infty} \sum_j \frac{q(m)e^{se_m(j)}}{\sum_l q(l)e^{se_l(j)}} && (19)\\
&= \sum_j q(m) \lim_{s \to -\infty} \frac{e^{se_m(j)}}{\sum_l q(l)e^{se_l(j)}} && (20)\\
&= \sum_j q(m) && (21)\\
&= Nq(m) && (22)
\end{aligned}$$

This last equation suggests that we can determine $q(m)$ from Q by setting

$$q(m) = \frac{1}{N} \sum_j Q(m|j) \tag{23}$$

This section has introduced a set of approximations that can be realized as modular linear Gaussian neural networks. The modular LG networks were characterized by the parameters, $(\bar{q}, \bar{\Theta})$. The regularized training problem was stated with respect to this parameterization. It was noted that solving this problem is more easily accomplished if we reparameterize the network with respect to the selection strategies, $\bar{Q}$. This observation led to a training problem statement whose performance measure is linear with respect to the selection strategies $\bar{Q}$ and is quadratic with respect to the other network parameters $\bar{\Theta}$. Problems of this form can be solved using expectation-maximization (EM) or hybrid interior-point (HIP) algorithms. The next section introduces the HIP algorithm.

3 Hybrid Interior Point (HIP) Training Algorithm

The training problems under consideration consist of those which may be decomposed into linear and quadratic subproblems. These problems can be written in an alternate form which emphasizes their dependence upon disjoint subsets of their parameters. Letting $\bar{\mathbf{x}} = \bar{Q}$ and letting

$$\bar{\mathbf{c}}(\bar{\Theta}) = \left[\begin{array}{cccccc} \frac{1}{N}d_1(1) & \frac{1}{N}d_1(2) & \ldots & \frac{1}{N}d_m(j) & \ldots \end{array} \right]^T \tag{24}$$

with $d_m(j) = \|\bar{\mathbf{z}}_j - \bar{\omega}_m\|^2 + (y_j - \bar{\phi}_m^T \bar{\mathbf{z}}_j)^2$, Problem 1 can be rewritten as

Problem 2

$$\begin{aligned} \textit{minimize:} \quad & \bar{\mathbf{c}}(\bar{\Theta})^T \bar{\mathbf{x}} \\ \textit{with respect to:} \quad & \bar{\mathbf{x}}, \bar{\Theta} \\ \textit{subject to:} \quad & A\bar{\mathbf{x}} = \bar{\mathbf{b}}, \bar{\mathbf{x}} \geq 0 \end{aligned}$$

where $A = [I_{N \times N} \ldots I_{N \times N}] \in \mathrm{I\!R}^{N \times MN}$ and $\bar{\mathbf{b}} = [1, \ldots, 1]^T \in \mathrm{I\!R}^N$. The rewritten objective in Problem 2 is linear in the parameters, $\bar{\mathbf{x}}$, and is quadratic in the parameters, $\bar{\Theta}$.

The dependence upon the two distinct parameter sets suggests that an *alternating minimization* (AM) strategy can be employed to solve Problem 1. In an AM procedure, the original optimization problem is decomposed into several subproblems which are iteratively solved by component minimizers for each subproblem. Subsection 3.1 briefly describes the notion of an alternating minimization algorithm. This subsection introduces the hybrid interior-point (HIP) algorithm as an example of an AM procedure. Subsection 3.2 discusses the component minimizers that are used in the HIP algorithm. Subsection 3.3 shows how the two component minimizers can be efficiently combined to produce a computationally efficient algorithm.

3.1 Alternating Minimization

Let $h : \mathrm{I\!R}^R \to \mathrm{I\!R}$ be a positive semi-definite functional over $\mathrm{I\!R}^R$. Consider the problem of minimizing this functional $h(\bar{\Phi})$ with respect to the parameter vector, $\bar{\Phi}$. It is assumed that the parameter vector can be decomposed as

$$\bar{\Phi}^T = \left[\begin{array}{cccc} \bar{\Phi}_1^T & \bar{\Phi}_2^T & \cdots & \bar{\Phi}_K^T \end{array} \right] \tag{25}$$

where $\bar{\Phi}_i \in \mathrm{I\!R}^{R_i}$ $(i = 1, \ldots, K)$ and $\bar{\Phi} \in \mathrm{I\!R}^R$ where $R = \sum_i R_i$. An *alternating minimization* (AM) procedure attempts to solve this problem by iteratively minimizing the functional with respect to $\bar{\Phi}_i$ for $i = 1, \ldots, K$ while holding the other subvectors, $\bar{\Phi}_j$ $(j \neq i)$, constant. AM procedures can be implemented by two nested loops. The inner loop performs the K component minimizations of

$h(\bar{\Phi})$ with respect to $\bar{\Phi}_i$ $(i = 1, \ldots, K)$ while holding $\bar{\Phi}_j$ constant for $j \neq i$. The outer loop encloses the inner loop and controls the algorithm's iterations until some stopping criterion is satisfied. Algorithm 3.1 gives a pseudo-code description of a general AM algorithm.

Algorithm 3.1 (General AM algorithm)

Initialize
  $j = 0$.
  Choose initial feasible $\bar{\Phi}^{(0)} = \langle \bar{\Phi}_1^{(0)}, \bar{\Phi}_2^{(0)}, \ldots, \bar{\Phi}_k^{(0)} \rangle$.
repeat
  for $i = 1$ to K
    minimize $f(\bar{\Phi}^{(j)})$
    with respect to $\bar{\Phi}_i^{(j)}$
    subject to $\bar{\Phi}_k^{(j)}$ held constant for $k \neq i$
  end
  $j = j + 1$
until(stopping criterion is satisfied)

Expectation-Maxmization (EM) algorithms [14] provide well-known examples of AM procedure. The HIP procedure described below and first introduced in [11] provides another example of an AM procedure. The HIP algorithm consists of a parameterized outer loop, whose iteration produces a sequence that converges to a local optimum of the training problem. Each element within that sequence is generated by the inner loop. This inner loop consists of a component linear and quadratic minimizer. A pseudo-code listing for the HIP algorithm is shown in algorithm 3.2 . The behaviour of this algorithm is controlled by parameters β and γ.

The outer loop in the HIP algorithm generates a sequence of solutions $\{\bar{\mathbf{x}}^{(k)}, \bar{\Theta}^{(k)}\}$. Associated with the kth iteration in the loop is a parameter $\alpha^{(k)} \geq 0$ which is a term in a monotone increasing sequence of the form

$$\alpha^{(k+1)} = \beta \alpha^{(k)} \tag{26}$$

where $\beta > 1$. $\alpha^{(k)}$ is a barrier coefficient used by the component linear minimizer. The parameter β is a free parameter of the algorithm controlling the number of iterations in the outer loop of the HIP algorithm.

The inner loops of the HIP algorithm consist of the component linear and quadratic minimizers. The linear minimizer solves the linear programming (LP) problem associated with minimizing the network error with respect to the selection strategies, $\bar{Q}$, subject to $\bar{\Theta}$ being constant. The quadratic minimizer uses a Newton-Raphson (NR) descent method to minimize the network error with respect to $\bar{\Theta}$ subject to the selection strategies, $\bar{Q}$, being constant. This minimizer is controlled by the parameter γ. The combination of linear IP solvers with Newton-Raphson (NR) descent methods forms the *hybrid* interior point or HIP algorithm. Detailed discussion of each component minimizer will be found in subsection 3.2. Subsection 3.3 shows how to efficienctly combine both minimizers.

Algorithm 3.2 (Hybrid IP AM algorithm)

Initialize
  $k = 0$.
  Choose $\alpha^{(k)} > 0$, $\bar{\Theta}^{(k)}$, and initial $\bar{\mathbf{x}}^{(k)}$.
repeat
  $\alpha^{(k+1)} = \beta\alpha^{(k)}$, $\beta > 1$.
  Linear update:
    $\bar{\mathbf{x}}_0 = \bar{\mathbf{x}}^{(k)}$
    $i = 0$
    while$\left(\|P_{AX_i}X_i(\alpha^{(k+1)}\bar{\mathbf{c}}(\bar{\Theta}) - \bar{\mathbf{x}}_i^{-1})\| > 0.2401\right)$
      $\bar{\mathbf{x}}_{i+1} = \bar{\mathbf{x}}_i - X_i P_{AX_i} X_i(\alpha^{(k+1)}\bar{\mathbf{c}}(\bar{\Theta}^{(k)}) - \bar{\mathbf{x}}_i^{-1})$
      $i = i + 1$
    end
    $\bar{\mathbf{x}}^{(k+1)} = \bar{\mathbf{x}}_i$
  Quadratic update:
    For $m = 1, \ldots, M$
      $\bar{\theta}^{(k+1)} = (1 - \gamma^{(k)})\bar{\theta}^{(k)} + \gamma^{(k)}\bar{\theta}^{(k+1),*}$
    end
  $k = k + 1$
until$(\|(\bar{\mathbf{x}}^{(k)}, \bar{\Theta}^{(k)}) - (\bar{\mathbf{x}}^*, \bar{\Theta}^*)\| < \epsilon)$

3.2 Component Minimizers

The HIP algorithm combines an interior-point linear programming algorithm with a quadratic minimization procedure. Both of these component minimizers are discussed in detail below.

Linear Minimizer

The interior-point (IP) method for solving the LP subproblems is a barrier method that uses logarithmic barriers to keep solutions inside the feasible set. The method generates a sequence of points that follow a path in the interior of the polyhedral set of feasible solutions. The resulting central path guides the approximate solutions to the optimal solution on the feasible set's boundary. The path following algorithm is fast, taking $O(\sqrt{n}L)$ iterations to converge with a computational cost of $O(n^{3.5}L)$ floating point operations. n is the dimension of the LP problem; and L denotes the "size" of the LP problem, being the number of bits required to represent the coefficients of A, $\bar{\mathbf{b}}$, and $\bar{\mathbf{c}}$. (Note: in our applications the LP problems will be of size $n = NM$).

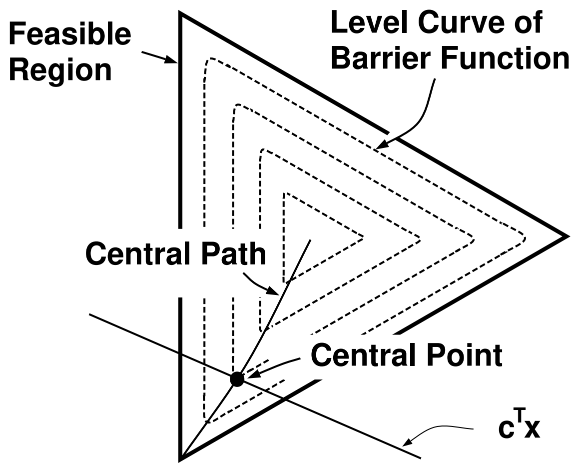

Figure 2: Interior Point concepts

The IP method is a path following algorithm [5] which solves the following sequence of optimization problems. Each problem is based upon a primal parameterization of Problem 2 when minimized solely with respect to $\bar{\mathbf{x}}$. Let the kth solution to this problem be denoted as $\bar{\mathbf{x}}^{(k)}$ with ith component $x_i^{(k)}$. The kth problem has the form,

$$\begin{array}{rl} \text{minimize:} & \alpha^{(k)}\bar{\mathbf{c}}^T\bar{\mathbf{x}}^{(k)} - \sum_i \log x_i^{(k)} \\ \text{with respect to:} & \bar{\mathbf{x}}^{(k)} \\ \text{subject to:} & A\bar{\mathbf{x}}^{(k)} = \bar{\mathbf{b}}, \bar{\mathbf{x}}^{(k)} \geq 0 \end{array} \tag{27}$$

where $\alpha^{(k)} \geq 0$ $(k = 1, \cdots, K)$ is a monotone increasing sequence of real numbers generated by the iteration $\alpha^{(k+1)} = \beta\alpha^{(k)}$ where $\beta > 1$. $\bar{\mathbf{x}}^*(\alpha^{(k)})$ denotes the optimal solution for the k^{th} optimization problem in the sequence and is referred to as a *central point.* The locus of all points, $\bar{\mathbf{x}}^*(\alpha^{(k)})$ where $\alpha^{(k)} \geq 0$, is called the *central path.* The augmented problem takes the original LP cost function and adds a logarithmic barrier which keeps the central points away from the boundaries of the feasible set. As α increases, the effect of the barrier is decreased, thereby allowing the k^{th} central point to approach the LP problem's optimal solution in a controlled manner. Figure 2 depicts the feasible set for a sample LP problem along with the central path, central points, and a set of level curves corresponding to contours of the barrier function.

Interior point algorithms solve the LP problem by approximately solving the sequence of augmented problems shown in (27). The approximate central points are computed using one or more scaling steepest descent (SSD) updates of the form

$$\bar{\mathbf{x}}^{(k+1)} = \bar{\mathbf{x}}^{(k)} - XP_{AX}X(\alpha^{(k+1)}\mathbf{c} - (\bar{\mathbf{x}}^{(k)})^{-1}) \tag{28}$$

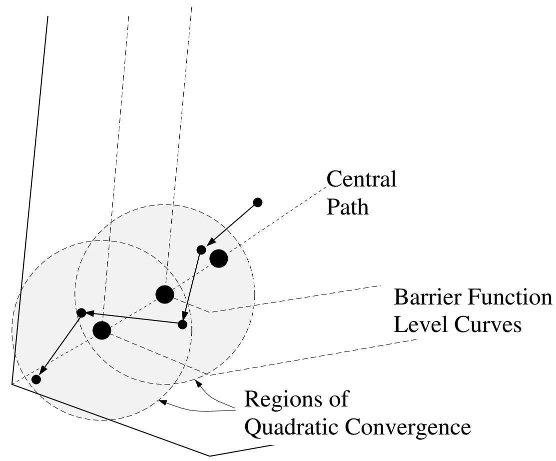

Figure 3: Path Following Example

where $X = \text{diag}(x_1, x_2, \ldots, x_n) \in \mathbb{R}^{n\times n}$ is a scaling transformation, $P_A = I - A^T(AA^T)^{-1}A \in \mathbb{R}^{n\times n}$ is an orthogonal projection, and $\bar{\mathbf{x}}^{-1} = \left[x_1^{-1}, x_2^{-1}, \ldots, x_n^{-1}\right]^T$. Figure 3 illustrates this sequence of approximate central points.

The interior-point algorithm can be proven to converge to an ϵ-neighborhood of the optimal solution to the LP problem provided these approximate solutions are close to the central path. The following *proximity measure* is essential for characterizing this notion of "nearness" to the central path,

$$\delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}) = ||P_{AX^{(k)}} X^{(k)}(\alpha^{(k)}\bar{\mathbf{c}} - (\bar{\mathbf{x}}^{(k)})^{-1})|| \tag{29}$$

$\delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}) < 1$ places $\bar{\mathbf{x}}^{(k)}$ in the region of quadratic convergence for the SSD algorithm. Once a point is near the central path and satisfies the *proximity condition*,

$$\delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}) \leq 0.5 \ , \tag{30}$$

the IP algorithm must maintains the proximity of successive points to the central path. This is generally done by selecting a β and then using a sufficient number of SSD steps to enforce the proximity condition. If β is chosen so that only a single SSD step is required, then the IP algorithm is said to be a *small-step* procedure. Small-step algorithms have been shown to converge in $O(\sqrt{n}L)$ iterations with an associated computational cost of $O(n^{3.5}L)$ [5]. If a larger value of β is used, then several SSD steps may be required to enforce the proximity condition. Such algorithms are often referred to as *large-step* IP algorithms. In general, large-step algorithms can exhibit considerably lower total computational cost than their small-step counterparts.

Quadratic Minimizer

The quadratic minimizer minimizes $\bar{\mathbf{c}}(\bar{\Theta})^T\bar{\mathbf{x}}$ with respect to $\bar{\Theta}$. The determination of successive quadratic parameters is accomplished using standard descent techniques to optimize $\bar{\mathbf{c}}(\bar{\Theta})^T\bar{\mathbf{x}}$ with respect to $\bar{\Theta}$. The determination of successive quadratic parameter estimates is termed the Θ-update.

The Θ-update uses a Newton-Raphson (NR) approach to solve the quadratic optimization with respect to $\bar{\Theta}$. The basic form of an NR update is

$$\bar{\Theta}^{(k+1)} = \bar{\Theta}^{(k)} - \left[H(\bar{\Theta}^{(k)})\right]^{-1} \nabla_{\bar{\Theta}}(\bar{\Theta}^{(k)})$$

where $H(\bar{\Theta})$ is the Hessian of the objective functional with respect to $\bar{\Theta}$ and $\nabla_{\bar{\Theta}}(\bar{\Theta})$ is the gradient with respect to $\bar{\Theta}$ evaluated at point $\bar{\Theta}^{(k)}$. Typically, the update is performed iteratively for general problems. The optimization is quadratic, however, so the following closed form solution results: $\bar{\Theta}^* = [\bar{\theta}_1^*, \bar{\theta}_2^*, \ldots, \bar{\theta}_M^*]$ where $\bar{\theta}_m^* = [\bar{\phi}_m^*, \bar{\omega}_m^*]$,

$$\bar{\phi}_m^* = \left[\sum_{j=1}^{N} \frac{1}{N} Q(m|j)\bar{\mathbf{z}}_j\bar{\mathbf{z}}_j^T\right]^{-1} \sum_{j=1}^{N} \frac{1}{N} Q(m|j)\bar{\mathbf{z}}_j y_j \tag{31}$$

and

$$\bar{\omega}_m^* = \frac{\sum_{j=1}^{N} \frac{1}{N} Q(m|j)\bar{\mathbf{z}}_j}{\sum_{j=1}^{N} \frac{1}{N} Q(m|j)}. \tag{32}$$

These minimizing solutions assume that $\sum_{j=1}^{N} \frac{1}{N} Q(m|j) \neq 0$ and that $\left[\sum_{j=1}^{N} \frac{1}{N} Q(m|j)\bar{\mathbf{z}}_j\bar{\mathbf{z}}_j^T\right]$ is nonsingular. The first condition will always be satisfied as at least one sample point will always be assigned to a model ($\forall m, \exists j$ such that $Q(m|j) > 0$). The second condition can be violated for insufficient amounts of data or when the data is linearly dependent. The problem can be corrected in such a case by reducing the number of models or increasing the number of samples to ensure the linear independence of the data.

3.3 Combining Component Minimizers

The behaviour of the HIP algorithm is controlled by the parameters, β and γ. Choices for β are dictated by the desired number of iterations in the outer loop of the HIP algorithm. If a small enough β is used, then we have a *small-step* HIP algorithm. For larger β, a *large-step* HIP procedure results. The primary difference between these methods lies in the number of SSD iterations required in the linear component minimzer. β is therefore chosen to control the total number of iterations in the outer loop of the HIP algorithm. The effect of specific β choices on the algorithm's total computational cost is discussed in section 4. These results suggest that choices

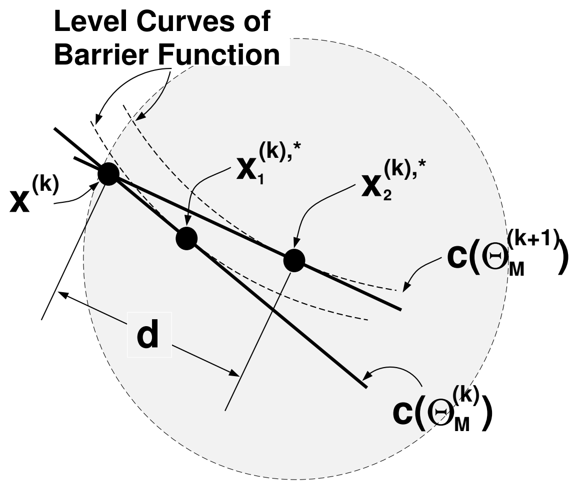

Figure 4: Effect of cost plane rotation

for β resulting in large-step HIP algorithms should be fastest. The simulation results in section 5 supports this assertion. Choices for γ are important for ensuring that the component minimzers work well together. The following subsection summarizes the desirable choices for γ. Formal proofs of justifying the desirability of these choices will be found in section 4 and appendix A.

A key concern in the HIP algorithm is the proximity of intermediate solutions in the sequence, $\{\bar{\mathbf{x}}^{(k)}, \bar{\Theta}^{(k)}\}$, to the central path. The proximity of successive solutions to the central path (see eq. (30)) results in the attractive convergence properties for the basic IP algorithm. This condition must be maintained by the HIP algorithm if the attractive computational properties of the interior-point method is to be preserved. Note, however, that performing the quadratic minimization changes the parameter vector, $\bar{\Theta}$, and hence changes the cost vector, $\bar{\mathbf{c}}(\bar{\Theta})$ (See Figure 4). It is possible that changes to the cost vector will rotate the cost vector about the current solution $\bar{\mathbf{x}}^{(k)}$ in such a way that $\bar{\mathbf{x}}^{(k)}$ no longer satisfies the proximity condition in (30). To make sure that the proximity condition is not violated, it is necessary to constrain the step size of the $\bar{\Theta}$-update by choosing the γ parameter appropriately. In section 4 and appendix A it is shown that the appropriate choice is

$$\gamma^{(k)} < \frac{0.12995}{n\|\bar{\mathbf{c}}(\bar{\Theta}^{(k+1),*}) - \bar{\mathbf{c}}(\bar{\Theta}^{(k)})\|}. \tag{33}$$

For this choice of γ, the HIP algorithm preserves the computational efficiency of the underyling interior-point linear programming algorithm.

4 Theoretical results

The utility of HIP algorithms rests in their convergence and computational properties. This section summarizes the key technical results supporting the algorithm's asserted computational efficiency. The results pertain to small-step versions of the HIP algorithm, but related results to the large-step version can be obtained using similar arguments. Proofs for the following results appear in [15], [5], and in appendix A. The results deal with proximity, convergence, and computational properties.

4.1 Proximity

As discussed earlier, the HIP algorithm must maintain the proximity condition (30). This proximity condition is enforced by using the appropriate number of SSD iterations for a given β. The following classical results [5] reparameterize β with respect to ν using the following equation,

$$\beta = 1 + \nu/\sqrt{n} \tag{34}$$

When $\nu \in (0, 0.1]$, then we say the algorithm is a *small-step* procedure. Theorem 1 and Corollary 2 specify how SSD steps affect proximity for a small-step HIP algorithm. Theorem 1 indicates that only a single SSD step is required to maintain proximity in moving from $\bar{\mathbf{x}}^{(k)}$ to $\bar{\mathbf{x}}^{(k+1)}$, while Corollary 2 details the "extra" proximity which is gained by taking multiple SSD steps. Theorem 3 specifies the effect of SSD steps on proximity for large-step HIP algorithms. It states that a linear number of steps ($O(n\mu^2/(1+\mu^2))$ steps with $\mu = \nu/\sqrt{n}$) is required to find $\bar{\mathbf{x}}^{(k+1)}$ satisfying (30) given an initial $\bar{\mathbf{x}}^{(k)}$ which satisfies (30).

Theorem 1 ([5]) *Let $\delta(\bar{\mathbf{x}}, \alpha, \bar{\Theta}) = \|P_{AX}X(\alpha\bar{\mathbf{c}}(\bar{\Theta}) - \bar{\mathbf{x}}^{-1})\|$ be the proximity measure where $\bar{\mathbf{x}}^{-1} = \langle x_1^{-1}, x_2^{-1}, \ldots, x_n^{-1}\rangle^T$ and $\alpha > 0$. If $\delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}, \bar{\Theta}^{(k)}) \leq 0.5$ and $\alpha^{(k+1)} = \alpha^{(k)}(1 + \nu/\sqrt{n})$ where $\nu \in (0, 0.1]$, then one SSD step finds $(\bar{\mathbf{x}}^{(k+1)}, \bar{\Theta}^{(k)})$, such that $\delta(\bar{\mathbf{x}}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq 0.5$.*

Corollary 2 (Appendix A) *Assuming the same conditions as in Theorem 1, then J SSD steps produce a point $(\bar{\mathbf{x}}_J^{(k+1)}, \bar{\Theta}^{(k)})$ such that $\delta(\bar{\mathbf{x}}_J^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq \delta_1 = (0.7)^{2^J}$.*

Theorem 3 ([5]) *Let $\delta(\bar{\mathbf{x}}, \alpha, \bar{\Theta})$ be the proximity measure where $\alpha > 0$. If $\delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}, \bar{\Theta}^{(k)}) \leq 0.5$ and $\alpha^{(k+1)} = \alpha^{(k)}(1 + \mu)$ where $\mu \geq 0.1/\sqrt{n}$, then $O(n\mu^2/(1+\mu^2))$ SSD updates find a new point, $(\mathbf{x}^{(k+1)}, \bar{\Theta}^{(k)})$, such that $\delta(\bar{\mathbf{x}}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq 0.5$.*

The preceding results are used in determining the choice of γ introduced in section 3, equation (33). Theorem 4 describes a bound on $\gamma^{(k)}$ that maintains the proximity condition. Note that this bound applies to either the small-step or large-step HIP algorithms.

Theorem 4 (Appendix A) *Let $\bar{\Theta}^{(k)}$ and $\bar{\Theta}^{(k),*}$ be the current and minimizing parameter vectors at time k. Let $\bar{\mathbf{c}}^{(k)} = \bar{\mathbf{c}}(\bar{\Theta}^{(k)})$ and $\bar{\mathbf{c}}^{(k),*} = \bar{\mathbf{c}}(\bar{\Theta}^{(k),*})$ and assume $\bar{\mathbf{c}}(\bar{\Theta})$ is a non-negative, convex function of $\bar{\Theta}$. Let $\delta(\bar{\mathbf{x}}, \alpha, \bar{\Theta})$ be the proximity measure. Assume $\delta(\bar{\mathbf{x}}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) = \delta_1 < 0.5$ and let $\bar{\Theta}^{(k+1)} = (1 - \gamma^{(k)})\bar{\Theta}^{(k)} + \gamma^{(k)}\bar{\Theta}^{(k+1),*}$. If $\gamma^{(k)}$ is chosen as*

$$\gamma^{(k)} < \frac{\delta_2 - \delta_1}{2n\|\bar{\mathbf{c}}^{(k+1),*} - \bar{\mathbf{c}}^{(k)}\|} \tag{35}$$

where $\delta_1 < \delta_2 = 0.5$, then $\delta(\bar{\mathbf{x}}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k+1)}) < \delta_2 = 0.5$.

The component minimizers together form an iteration that computes $(\bar{\mathbf{x}}^{(k+1)}, \bar{\Theta}^{(k+1)})$ in proximity to $\bar{\mathbf{x}}^*(\alpha^{(k+1)}, \bar{\Theta}^{(k+1)})$ given $(\bar{\mathbf{x}}^{(k)}, \bar{\Theta}^{(k)})$ which is already in proximity to $\bar{\mathbf{x}}^*(\alpha^{(k)}, \bar{\Theta}^{(k)})$. Theorem 5 quantifies the result for both the small-step and large-step cases that is found in equation (33).

Theorem 5 (Appendix A) *Let $\delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}, \bar{\Theta}^{(k)}) \leq 0.5$ and let $\alpha^{(k+1)} = \alpha^{(k)}(1 + \mu)$ where $\mu < 0.1/\sqrt{n}$ for the small-step algorithm and $\mu \geq 0.1/\sqrt{n}$ for the large-step case. Let $\bar{\Theta}^{(k+1)} = (1 - \gamma^{(k)})\bar{\Theta}^{(k)} + \gamma^{(k)}\bar{\Theta}^{(k+1),*}$ where*

$$\gamma^{(k)} < \frac{0.12995}{n\|\bar{\mathbf{c}}^{(k+1),*} - \bar{\mathbf{c}}^{(k)}\|}.$$

Then, one iteration of the small-step variant of Algorithm 3.2 with two SSD steps or one iteration of the large-step variant with $O(n\mu^2/(1+\mu^2))$ SSD steps produces $(\bar{\mathbf{x}}^{(k+1)}, \bar{\Theta}^{(k+1)})$ from $(\bar{\mathbf{x}}^{(k)}, \bar{\Theta}^{(k)})$ such that $\delta(\bar{\mathbf{x}}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k+1)}) \leq 0.5$.

4.2 Asymptotic convergence

The component minimizers' combination as an hybrid IP algorithm is only useful if it converges to an optimum for Problem 1. The preceding theorems only imply that the HIP algorithm will converge to a fixed point, $(\bar{\mathbf{x}}^*, \bar{\Theta}^*)$, as $\alpha^{(k)}$ increases. This point corresponds to an optimum of the LP subproblem and an optimum for the quadratic subproblem. Proximity and central paths do not, however, guarantee that $(\bar{\mathbf{x}}^*, \bar{\Theta}^*)$ is a local optimum for Problem 1. This section summarizes results establishing the aymptotic convergence of the small-step HIP algorithm to a local optimum. These results first quantify the changes in the linear and quadratic solutions as functions of the LP subproblem's duality gap. Those results then imply that the solution sequence is Cauchy and converges to a fixed point. It is then shown that this fixed point is a local optimum for the regularized training problem.

Convergence results for the HIP algorithm depend heavily on the LP subproblem's duality gap, as do similar results for the LP solvers. The duality gap, $\Delta^{(k)}$, is a non-negative quantity that denotes an upper bound on the distance to an optimal solution to the LP subproblem in terms of problem cost. An expression for the bound on the duality gap ,$\delta(\bar{\mathbf{x}}, \alpha)$, will be found in [5]. This expression extends to the HIP algorithm where $\delta(\bar{\mathbf{x}}, \alpha)$ becomes $\delta(\bar{\mathbf{x}}, \alpha, \bar{\Theta})$. With this extension, a bound on the duality gap at the k^{th} iteration for the HIP procedure is

$$\Delta^{(k)} \leq \frac{n + \delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}, \bar{\Theta}^{(k)})\sqrt{n}}{\alpha^{(k)}}. \tag{36}$$

$\Delta^{(k)}$ is a function of $\alpha^{(k)}$ since $\delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}, \bar{\Theta}^{(k)})$ can be bounded by 0.5 from the previous proximity conditions. Thus, as $\alpha^{(k)}$ increases, the distance to an optimal LP solution decreases. This property is used in the succeeding results where $\Delta^{(k)}$ takes on a role similar to that of a descent function.

The component linear and quadratic minimizers make incremental changes to $\bar{\mathbf{x}}^{(k)}$ (or $\bar{Q}$) and $\bar{\Theta}^{(k)}$ where the changes can be bounded by a function of the duality gap. These changes rely on satisfaction of proximity conditions and on the forms of the updates, themselves. The changes in $\bar{\mathbf{x}}^{(k)}$ can be related directly to the duality gap. Theorem 6 describes the change in $\bar{\mathbf{x}}$ and provides a bound on $\|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}^{(k)}\|$ in terms of the duality gap. The changes in $\bar{\Theta}$ depend directly on the changes in $\bar{\mathbf{x}}$ due to the updates in (31) and (32). Theorem 7 details the changes in $\bar{\Theta}^*$ and provides a bound on $\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k),*}\|$ in terms of $\|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}^{(k)}\|$. Both norms have upper bounds which are functions of the duality gap. Both are therefore strictly decreasing as the duality gap is strictly decreasing, implying that $\|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}^{(k)}\| \to 0$ and $\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k),*}\| \to 0$ as $\Delta^{(k)} \to 0$.

Theorem 6 (Appendix A) *Let $\bar{Q}^{(k)}$ and $\Delta^{(k)}$ be the set of linear parameters (probabilities) and the duality gap at iteration k, respectively. Let $\bar{\mathbf{c}}^{(k)} = \bar{\mathbf{c}}(\bar{\Theta}^{(k)})$ and $\delta(\bar{\mathbf{x}}, \alpha, \bar{\Theta})$ be the proximity measure where $\bar{\mathbf{x}} = \bar{Q}$. Let $\alpha^{(k+1)} = \alpha^{(k)}(1 + \nu/\sqrt{n})$ where $\nu \in (0, 0.1]$. Assume that $\bar{\mathbf{x}}^{(k)}$ satisfies the proximity condition $\delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}, \bar{\Theta}^{(k)}) \leq 0.5$. For two SSD updates, the change in $\bar{Q}$, $\|\triangle Q\| = \|Q^{(k+1)} - Q^{(k)}\|$ is*

$$\|\triangle Q\| < 2(\delta(1 + \nu) + \nu)\Delta^{(k)}. \tag{37}$$

where $\delta = \delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}, \bar{\Theta}^{(k)})$.

Theorem 7 (Appendix A) *Let $Q(m|j) = Q^{(k)}(m|j)$ and $\triangle Q(m|j) = Q^{(k+1)}(m|j) - Q^{(k)}(m|j)$. Define $\bar{\theta}_m^* = \left[(\bar{\phi}_m^*)^T, (\bar{\omega}_m^*)^T\right]^T \in \mathbb{R}^{2R}$ where $\bar{\phi}_m^*$ and $\bar{\omega}_m^*$ are defined as in (31) and (32). Let Q_{CP} be the set of all Q's on or near the central path. Define*

$$\begin{aligned} A_m &= \textstyle\sum_{j=1}^N \frac{1}{N} Q(m|j)\bar{\mathbf{z}}_j\bar{\mathbf{z}}_j^T \\ E_m &= \textstyle\sum_{j=1}^N \frac{1}{N} \triangle Q(m|j)\bar{\mathbf{z}}_j\bar{\mathbf{z}}_j^T \end{aligned}$$

where $A_m, E_m \in \mathbb{R}^{R\times R}$. *Assume bounded inputs and outputs for the design set* $\mathcal{T}$, $\|\bar{\mathbf{z}}\| \leq \zeta$, $|y| \leq Y_{max}$. *Assume that* A_m *is of full rank for* $Q \in Q_{CP}$. *Let* $\mu_{max} = \sup_{Q\in Q_{CP}} \|A_m^{-1}\|$, $\|E_m\| < 1/\mu_{max}$, *and* $\beta_{max} = \sup_{Q\in Q_{CP}} (\sum_{j=1}^{N} \frac{1}{N} Q(m|j))^{-1}$. *Then,*

$$\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k),*}\| \leq M\mathcal{K}\|\triangle Q\| \tag{38}$$

where $\mathcal{K} = \mu_{max}\zeta Y_{max}(1 + \mu_{max}\zeta^2/(1 - r_1)) + \beta_{max}\zeta(1 + \beta_{max}/(1 - r_2))$, $r_1 = \|A_m^{-1}\|\ \|E_m\| < 1$, *and* $r_2 = (\sum_{j=1}^{N} \frac{1}{N}\triangle Q(m|j))/(\sum_{j=1}^{N} \frac{1}{N} Q(m|j)) < 1$.

The significance of Theorems 6 and 7 is that the norms of the changes in the parameters have bounds which are functions of the duality gap. The results indicate that the changes in the parameters are guaranteed to decrease as the duality gap decreases. It is initially surprising that both parameter changes can be bounded by the duality gap. However, considering that the problem is expressed as a linear subproblem, it is not surprising that the linear parameters' changes are functions of $\Delta^{(k)}$, which is intrinsic to LP problems. It is also not surprising that the quadratic parameters' changes depend upon $\Delta^{(k)}$ as the closed form solutions for the minimizing parameter vectors depend directly upon the linear parameters which themselves depend upon $\Delta^{(k)}$.

Asymptotic convergence of the small-step variant requires that $\{(\bar{\mathbf{x}}^{(k)}, \bar{\Theta}^{(k)})\} \rightarrow (\bar{\mathbf{x}}^*, \bar{\Theta}^*)$ as k increases. The preceding two results do not imply this condition. Rather, they imply that the linear parameters converge and the *optimal estimates* of the quadratic parameters converge. In order to prove convergence, both the linear and quadratic parameters must converge simultaneously. For this to occur, the convex combination coefficient, $\gamma^{(k)}$, must tend to and remain fixed at unity as k increases. This implies that the quadratic parameters are always updated exactly as their optimal estimates and that $\|\bar{\Theta}^{(k+1)} - \bar{\Theta}^{(k)}\| = \|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k),*}\|$. When this condition holds, then both $\|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}^{(k)}\| \rightarrow 0$ and $\|\bar{\Theta}^{(k+1)} - \bar{\Theta}^{(k)}\| \rightarrow 0$. Theorem 8 states that there is an iteration number, K, for which all iterations greater than K satisfy the condition on $\gamma^{(k)}$. Theorem 9 provides the result that $\{(\bar{\mathbf{x}}^{(k)}, \bar{\Theta}^{(k)})\} \rightarrow (\bar{\mathbf{x}}^*, \bar{\Theta}^*)$. The proof of Theorem 9 shows that the sequence is Cauchy and converges to a fixed point.

Theorem 8 (see Appendix A) *For the HIP algorithm described above, there exists a* $K \geq 0$ *such that* $\gamma^{(k)} = 1$ *for all* $k \geq K$.

Theorem 9 (Appendix A) *For the HIP algorithm described above, if* $\gamma^{(k)} = 1$ *for all* $k \geq K \geq 0$, *then the sequence of solutions,* $\{(\bar{\mathbf{x}}^{(k)}, \bar{\Theta}^{(k)})\}$, *converges to a fixed point,* $(\bar{\mathbf{x}}^*, \bar{\Theta}^*)$.

The optimality of the fixed point determined by the HIP algorithm is established in the following theorem 10.

Theorem 10 (Appendix A) *Let* $\bar{\mathbf{w}}^* = \left[(\bar{\mathbf{x}}^*)^T, (\bar{\Theta}^*)^T\right]^T$ *be the limit of the sequence,* $\{\bar{\mathbf{w}}^{(k)}\} = \{\left[(\bar{\mathbf{x}}^{(k)})^T, (\bar{\Theta}^{(k)})^T\right]^T\}$*, produced by the HIP algorithm. Let the minimization problem be*

$$\begin{aligned} \textit{minimize:} \quad & \bar{\mathbf{c}}(\bar{\Theta})^T \bar{\mathbf{x}} \\ \textit{with respect to:} \quad & \bar{\mathbf{x}}, \bar{\Theta} \\ \textit{subject to:} \quad & A\bar{\mathbf{x}} = \mathbf{b}, \bar{\mathbf{x}} \geq 0. \end{aligned} \tag{39}$$

$\bar{\mathbf{w}}^* = \left[(\bar{\mathbf{x}}^*)^T, (\bar{\Theta}^*)^T\right]^T$ *is a locally minimum solution to (39).*

4.3 Computational properties

The final two results refer to the small-step variant's computational properties. Small-step HIP variants are direct extensions of small-step interior point LP solvers. The latter algorithms enjoy excellent computational properties, $O(\sqrt{n}\log_2(1/\epsilon))$ iterations convergence rate and $O(n^{3.5}\log_2(1/\epsilon))$ flops cost, to converge to an ϵ-neighborhood of an optimal solution. Under appropriate problem scaling assumptions, the HIP procedure inherits the computational efficiency of its linear counterpart. The extension requires $O(\sqrt{n}\log_2(\sqrt{n}/\epsilon))$ iterations to converge with an associated computational cost of $O((n^{3.5} + n^{1.5}R^2 + \sqrt{n}MR^3)\log_2(\sqrt{n}/\epsilon))$ flops to converge to an ϵ-neighborhood of locally optimal solutions. In this case $n = NM$. The two results are summarized below. Together, they indicate that the small-step variant is computationally efficient and that it will scale well as problem size increases.

Theorem 11 (Appendix A) *Let* $\bar{\mathbf{w}}^{(k)} = \left[(\bar{\mathbf{x}}^{(k)})^T, (\bar{\Theta}^{(k)})^T\right]^T$ *be the* k^{th} *approximate solution generated by the HIP algorithm. Assume that* $\Delta^{(0)} < 1/\epsilon$*. Assume that the cost vector is scaled by* S *such that* $\gamma^{(k)} = 1$ *for all* $k \geq K$ *and that* $2(\delta(1+\nu)+\nu)(1+M\mathcal{K})\Delta^{(K)} > \epsilon$*. Given these assumptions, the HIP algorithm will converge to an* ϵ*-neighborhood of a locally optimal solution* ($\|\bar{\mathbf{w}}^{(k)} - \bar{\mathbf{w}}^*\| < \epsilon$) *in* $O(\sqrt{n}\log_2(\sqrt{n}/\epsilon))$ *iterations.*

The HIP algorithm's overall computational cost is computed by determining the computational complexity of the individual steps and then multiplying by the algorithm's convergence rate. The linear update executes two SSD steps each employing a matrix inversion with a worst case complexity of $O(n^3)$ where $n = MN$. The quadratic update requires NR^2 multiplications resulting from the vector outer product and employs a matrix inversion with a computational complexity of

$O(R^3)$. Also, the quadratic update is done on M models so the quadratic update has a computational complexity of $O(nR^2)$ or $O(MR^3)$. These complexity estimates multiplied by the estimate of the number of iterations establish the following theorem:

Theorem 12 (Appendix A) *The computational cost of the small-step, HIP algorithm is* $O((n^{3.5} + n^{1.5}R^2 + \sqrt{n}MR^3)\log_2(\sqrt{n}/\epsilon))$

5 Examples

Two collections of experiments examine the hybrid IP algorithms' computational properties and use. The first set performs comparisons between the two hybrid IP algorithms and the EM algorithm. Comparisons in Section 5.1 address the relative computational costs and error characteristics of the three algorithms. The second set of experiments demonstrates the two IP algorithms' use in training networks to determine multiple, linear, set point models of a nonlinear plant. Those experiments use data from a fossil fuel burning, electric power generating plant.

5.1 Mixture density parameter estimation

The first collection of experiments compares the IP algorithms with the EM algorithm when used on mixture density parameter estimation problems. The three algorithms identify the means of a set of normal densities using sample vectors generated from the densities. The densities have means which correspond to the vertices of a simplex in an R dimensional space. Similar to experiments in [14], the means are equispaced. These experiments assume $M = 2$ to 15 candidate densities. Samples sets ranging in size from $N = 300$ to $N = 750$ vectors are the training sets used. All the experiments assume known and equal covariances, $\Sigma = I$.

The specific EM algorithm used in these experiments is a two step procedure modeled after the EM algorithms described in [14] for solving mixture problems. It consists of implementing equations (4.0) and (4.8) from [14]. These equations are

$$q(m)^{(k+1)} = \frac{1}{N}\sum_{j=1}^{N} \frac{q(m)^{(k)} P(\bar{\mathbf{z}}_j | m, \bar{\Theta}^{(k)})}{P(\bar{\mathbf{z}}_j | \bar{\Theta}^{(k)})} \tag{40}$$

and

$$\bar{\omega}_m^{(k+1)} = \left[\sum_{j=1}^{N} \bar{\mathbf{z}}_j \frac{q(m)^{(k+1)} P(\bar{\mathbf{z}}_j | m, \bar{\Theta}^{(k)})}{P(\bar{\mathbf{z}}_j | \bar{\Theta}^{(k)})}\right] \left[\sum_{j=1}^{N} \frac{q(m)^{(k+1)} P(\bar{\mathbf{z}}_j | m, \bar{\Theta}^{(k)})}{P(\bar{\mathbf{z}}_j | \bar{\Theta}^{(k)})}\right]^{-1} \tag{41}$$

for $m = 1, \ldots, M$, where

$$P(\bar{\mathbf{z}}_j | \bar{\Theta}^{(k)}) = \sum_{m=1}^{M} q(m) k_1 \exp\left(-\kappa_1 \|\bar{\mathbf{z}}_j - \bar{\omega}_m\|^2\right)$$

with k_1 normalizing constant and $q(m)$ being the marginal probability of observing a sample from the m^{th} class.

The first set of experiments makes a qualitative analysis of the three methods. This set assumed problem parameter values of $R = 9$, $N = 300$, $M = 2$ to 9, and a spacing of four units between the means. This spacing between the means corresponds to a well separated set of densities whose parameters should be easy to identify. The algorithms were initialized with random initial conditions. In the runs, the algorithms executed until the solutions were within 10^{-5} of an optimal solution. Results from this set examine the convergence rate in iterations and the computational cost in flops.

Results for the first set of experiments appear in Figures 5 and 6. These results serve as a means to rank the algorithms' performance relative to one another. Mean empirical iteration results, along with a "best-fit" characteristic, in Figure 5 exhibit $O(n^{1.53})$, $O(n^{0.45})$, and constant iteration rates for the EM, small-step, and large-step algorithms, respectively, where n is the dimension of the linear subproblem. Mean computational cost results in Figure 6 exhibit $O(n^{2.42})$, $O(n^{1.78})$, and $O(n^{1.53})$ flops cost rate for the EM, small-step, and large-step algorithms, respectively. While the IP variants exhibit lower convergence rates and cost rates than does EM, the plots demonstrate that the EM algorithm incurs lower cost for these well separated problems. As the number of models increases, however, EM performance begins to approach the large-step algorithm's performance in terms of cost. The same cannot be said with respect to the small-step IP variant, which still has a much greater cost. These results allow us to eliminate the small-step variant from further consideration in our comparisons, as it always has much greater cost.

The next sets of experiments perform a more quantitative analysis of the large-step IP variant and the EM algorithm. These experiments assume $R = 19$, $N = 300, 500, 750$ samples, $M = 2$ to 15 mixtures, and a spacing of two units between the densities' means. This spacing corresponds to more closely spaced, harder to identify means. The algorithms terminate in these runs when the solutions are within 5×10^{-5} of an optimal solution.

Computational cost results for the three cases appear in Figures 7-9. These graphs display the mean computational costs for the two algorithms along with bounds indicating the maximum and minimum costs incurred for the various sized mixtures. The cost rates for the EM and large-step algorithms are $O(n^{1.7})$ and $O(n^{1.45})$ flops, respectively. The two algorithms are comparable, exhibiting very similar mean costs for all three cases. The EM algorithm generally exhibits slightly better cost, but the IP algorithm outperforms the EM algorithm at some points.

Analysis of the costs' standard deviations reveals two observations. First, Table 1 tabulates the percentages for the two algorithms of the number of runs resulting in costs greater than the mean cost and the number of runs with costs within a standard deviation of the mean. The EM algorithm again performs slightly better in both accounts, but the results are comparable. Thus,

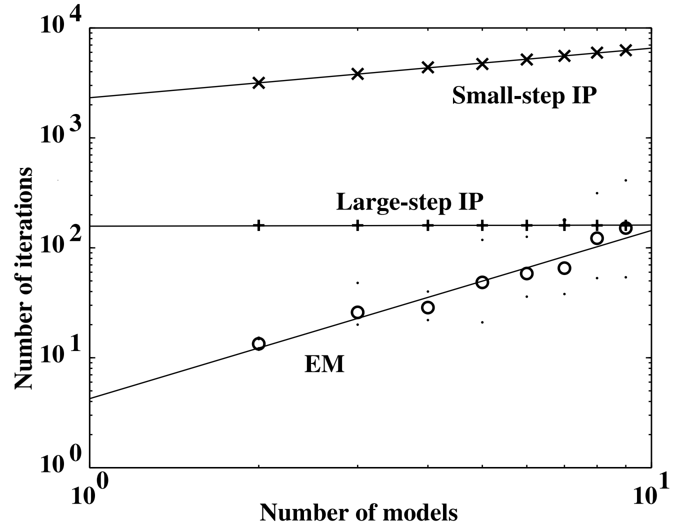

Figure 5: Algorithm iteration rates

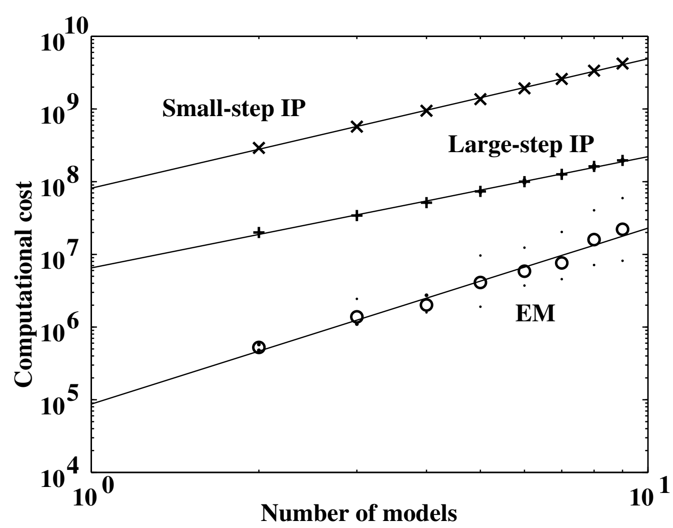

Figure 6: Algorithm computational costs

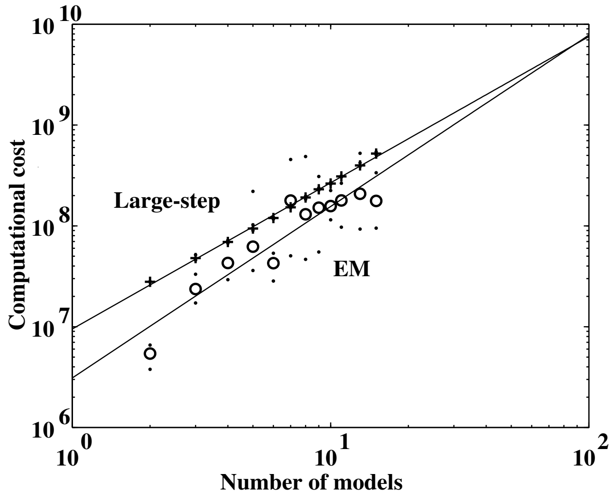

Figure 7: Computational cost ($N = 300$)

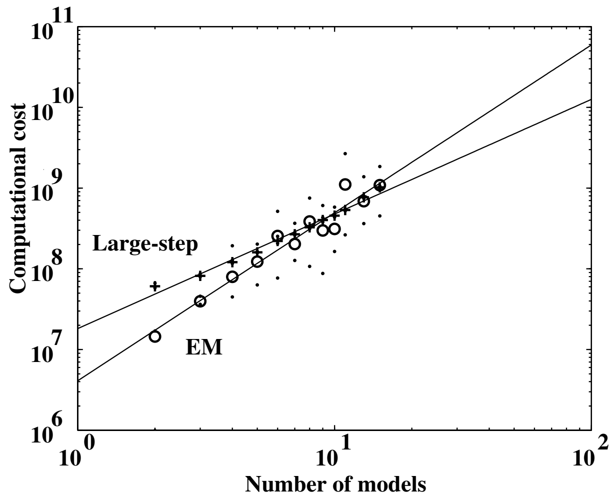

Figure 8: Computational cost ($N = 500$)

Table 1: Cost standard deviation results

Algorithm	N	% of costs > mean	% of costs within one standard deviation
EM	300	0.367	0.767
Large-step IP	300	0.417	0.717
EM	500	0.433	0.742
Large-step IP	500	0.467	0.700
EM	750	0.425	0.750
Large-step IP	750	0.400	0.700

the algorithms' statistical performances are comparable.

The second observation deals with the standard deviations, themselves. Figures 10-12 plot the standard deviations of the costs normalized by the mean cost for each number of mixtures for the three experiments. The results show that the IP algorithm consistently exhibits a smaller, normalized standard deviation than does EM. This implies that the IP algorithm requires a number of computations that vary less from run to run than does EM. In terms of algorithm operation, it appears that the IP algorithm is less susceptible to the effects of initial conditions in the number of required computations than is EM.

Final cost results examine the worst case costs incurred by the two algorithms. Figures 13-15 plot the maximum costs incurred by the two algorithms for the three test cases. In all three cases, the IP algorithm's worst case cost is generally better than the EM algorithm's worst case performance. Considering that EM has a cost with greater standard deviation than does the large-step method, this suggests that for these closely spaced problems the IP algorithm may be better in terms of cost incurred.

Error analysis, similar to the preceding cost analysis, follows here. Mean square parameter error results appear in Figures 16-18. The plots depict the mean square parameter error between the computed estimates and the actual values for each number of models. The errors are averaged over the number of runs at each point. The plots demonstrate that the EM algorithm always results in worse error than does the IP algorithm.

Examination of the errors' standard deviation yields additional observations. Table 2 contains information similar to that contained in Table 1. It lists the percentages for the two algorithms of the number of runs resulting in errors greater than the mean error and the number of runs with

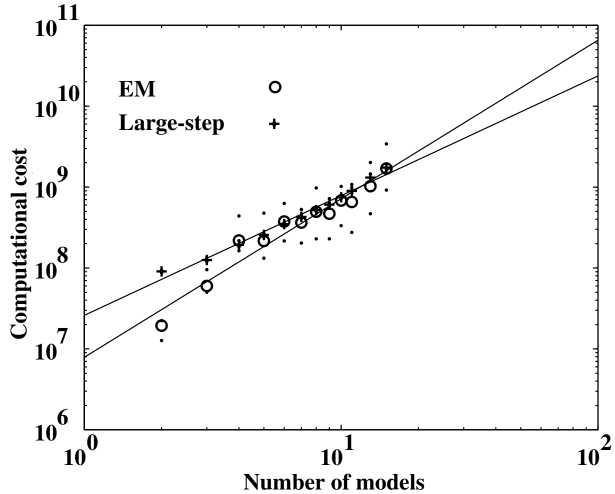

Figure 9: Computational cost ($N = 750$)

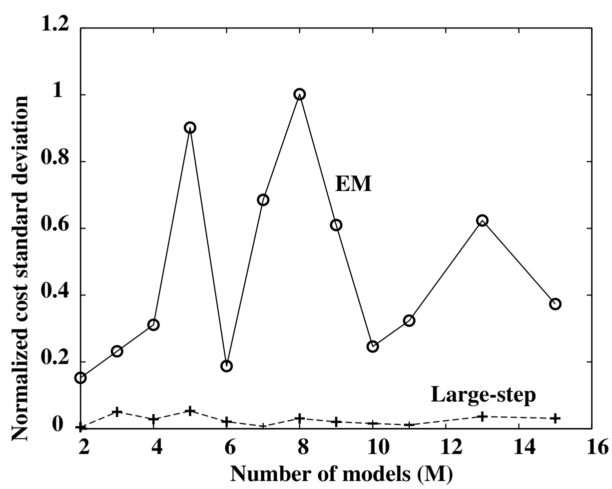

Figure 10: Normalized cost standard deviation ($N = 300$)

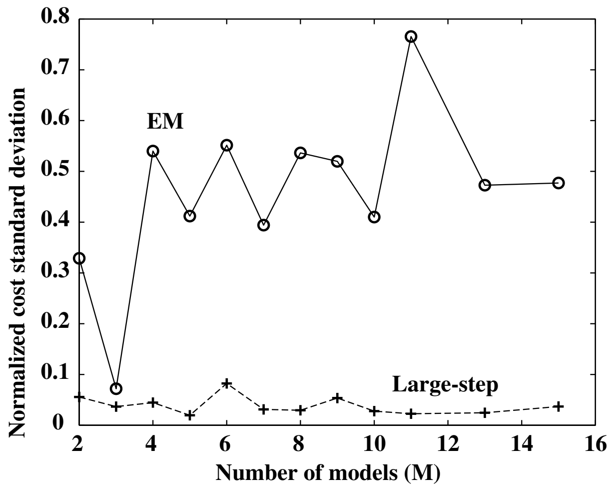

Figure 11: Normalized cost standard deviation ($N = 500$)

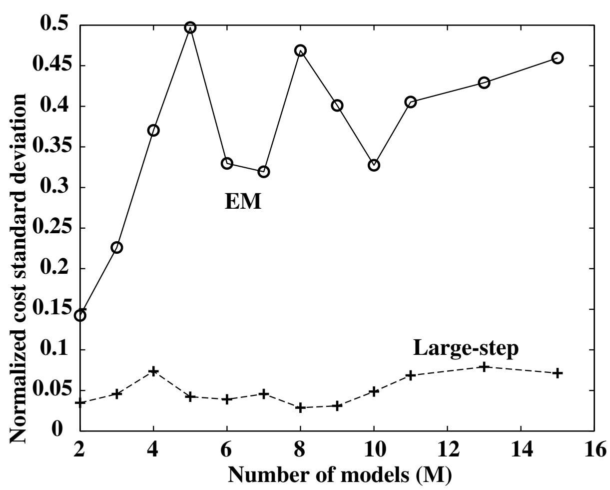

Figure 12: Normalized cost standard deviation ($N = 750$)

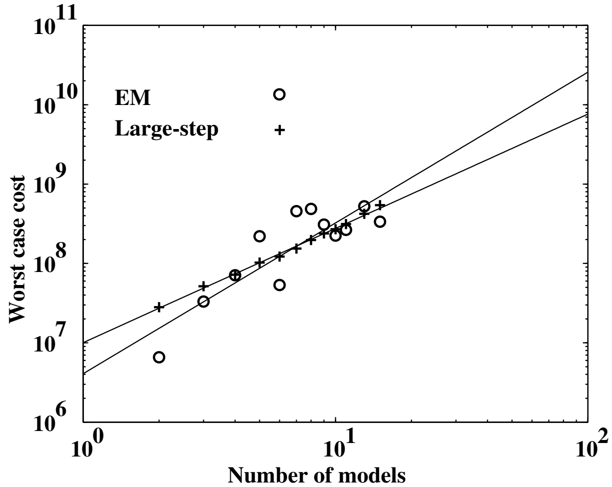

Figure 13: Worst case computational cost ($N = 300$)

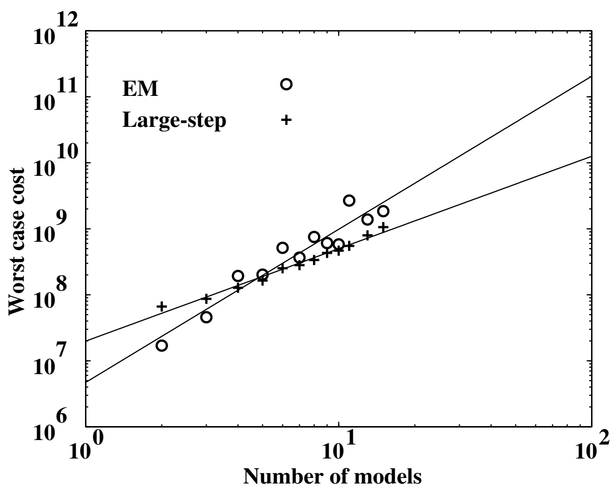

Figure 14: Worst case computational cost ($N = 500$)

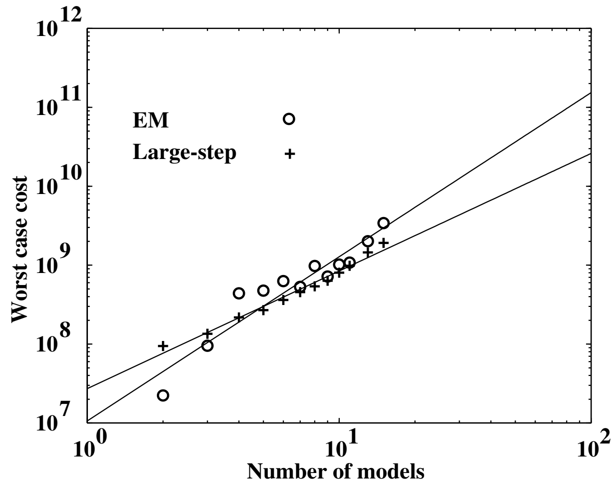

Figure 15: Worst case computational cost ($N = 750$)

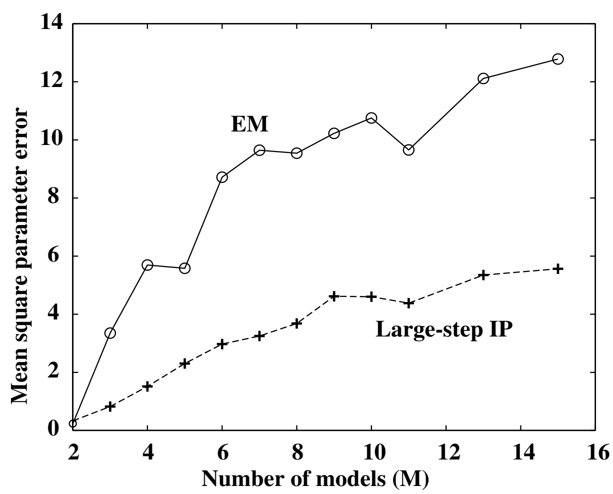

Figure 16: Parameter error($N = 300$)

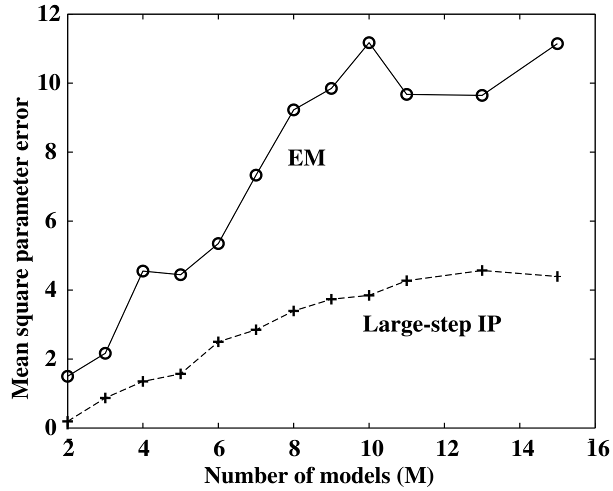

Figure 17: Parameter error ($N = 500$)

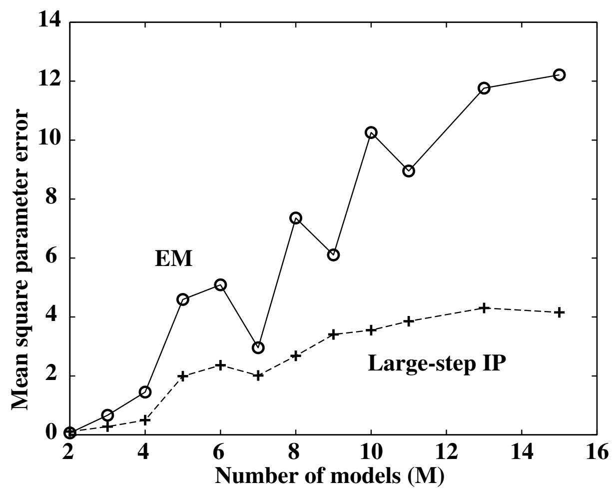

Figure 18: Parameter error ($N = 750$)

Table 2: Error standard deviation results

Algorithm	N	% errors > mean	% costs within one standard deviation
EM	300	0.392	0.658
Large-step IP	300	0.517	0.667
EM	500	0.383	0.725
Large-step IP	500	0.483	0.667
EM	750	0.467	0.692
Large-step IP	750	0.467	0.683

errors that fall within a standard deviation of the mean error. This table suggests that the error distributions are comparable between both methods

Additional results in Figures 19-21 plot the standard deviations of the errors normalized by the mean error for each case. The plots demonstrate that the EM algorithm has a greater variance in the errors it incurs in producing its estimates for all three trials. The larger errors occur because the EM algorithm has more difficulty in separating the samples from run to run. The IP method, however, separates the samples with a much smaller variance in its estimates.

Final error results examine the worst case errors incurred by the two algorithms. Figures 22-24 plot the maximum errors incurred by the two algorithms. In all the cases, the IP algorithm's worst case error is significantly better than the EM algorithm's worst case error. This again supports the observation that the IP algorithm generalizes better than the EM algorithm on the test cases.

5.2 Model parameter identification

A final experiment examines the small-step HIP algorithm's use on a multiple model system identification problem. Here, a modular network is trained to identify multiple linear, set point models of a nonlinear system. The problems uses input/output data for the fuel flow process from a fossil fuel burning electric power generating plant (see Figure 25). Two sets of data consisting of input/output pairs of the fuel flow process are available to train the network: an $N = 200$ sample design set and an $N = 200$ sample validation set. The experiment used the small-step HIP algorithm to train modular network and recorded the computational properties of the HIP algorithm for $M = 2$ to 9 model units. Also recorded was the network's performance in approximating the fuel flow process.

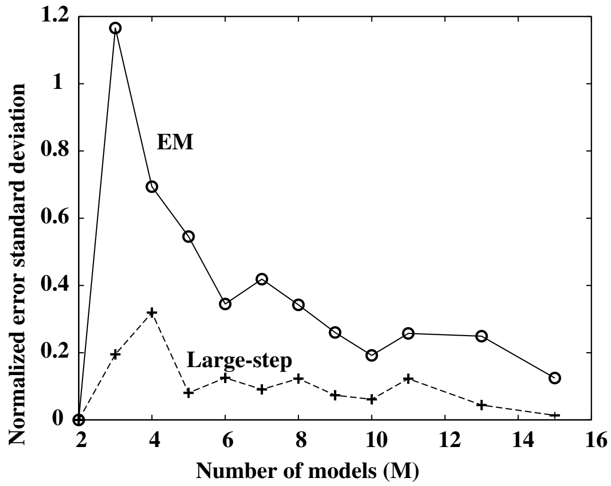

Figure 19: Normalized error standard deviation ($N = 300$)

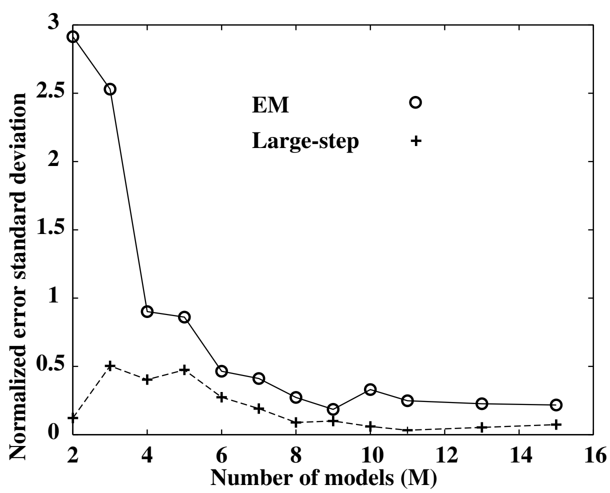

Figure 20: Normalized error standard deviation ($N = 500$)

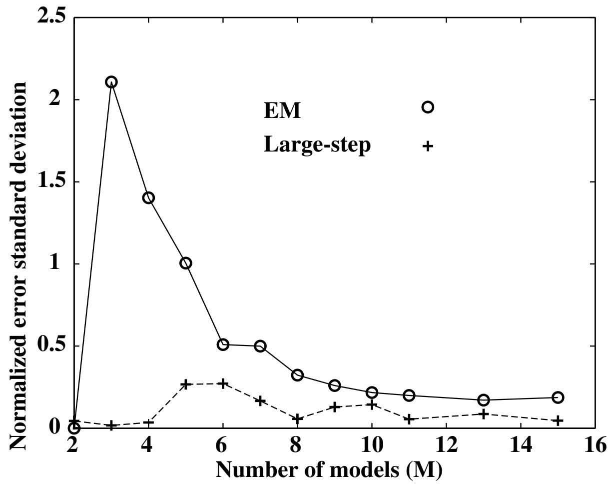

Figure 21: Normalized error standard deviation ($N = 750$)

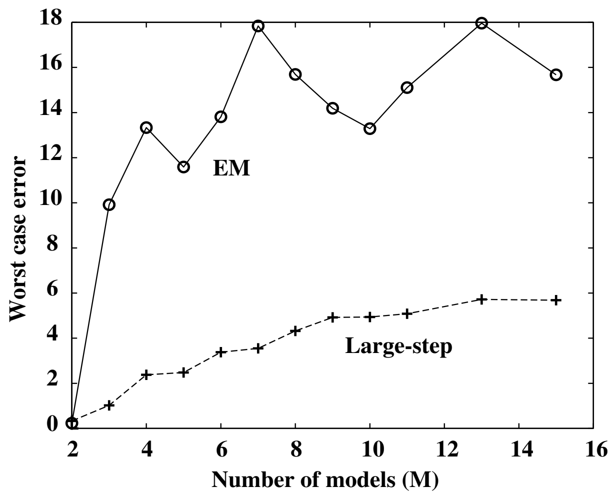

Figure 22: Worst case error ($N = 300$)

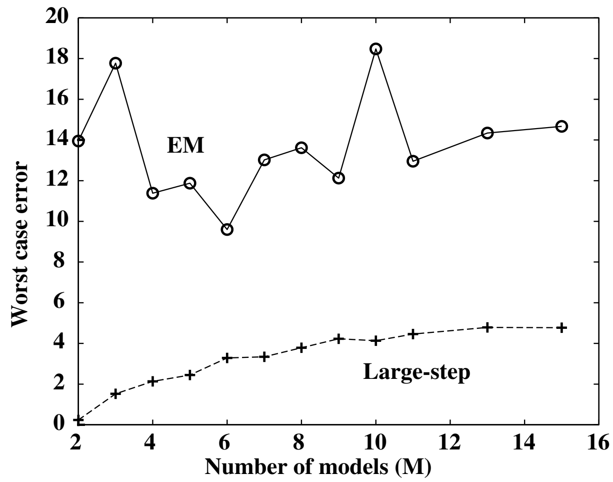

Figure 23: Worst case error ($N = 500$)

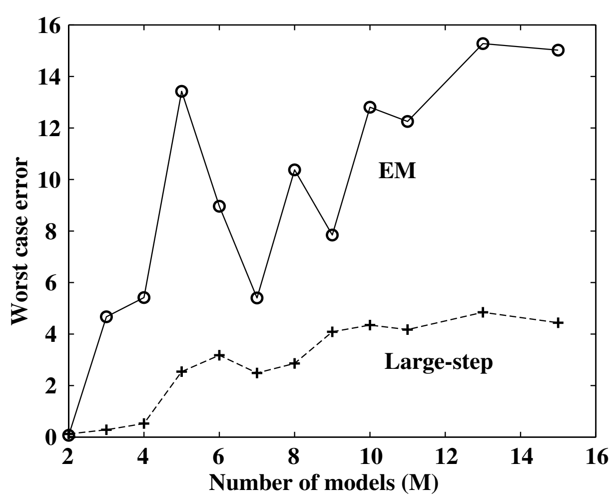

Figure 24: Worst case error ($N = 750$)

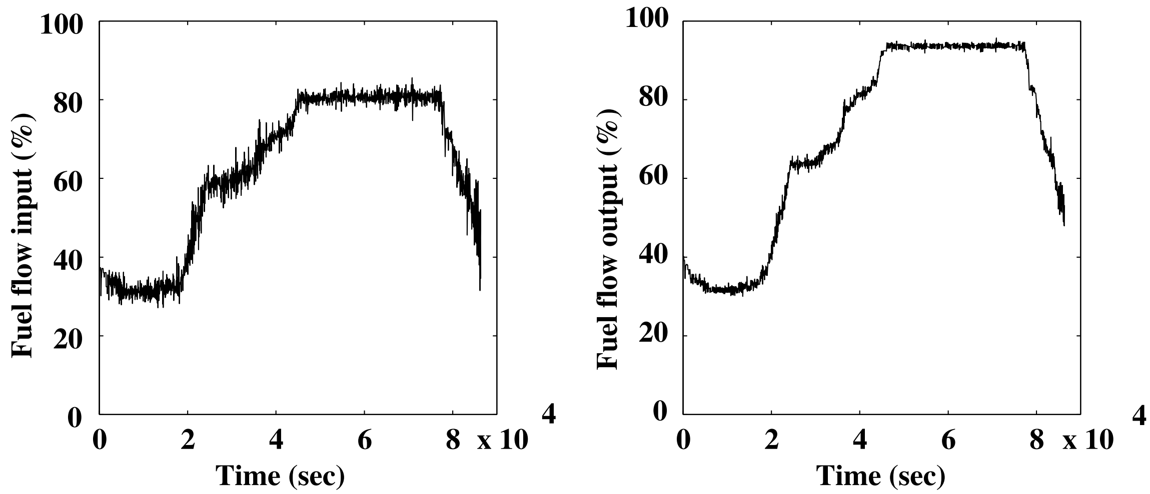

Figure 25: Fuel flow input/output data

The experiment's results follow here. Figure 26 shows that the empirical computational properties meet or exceed the theoretical properties. The convergence rate in Figure 26(a) shows that the empirical rate (open circles) agrees with the theoretical rate (solid line). Figure 26(b) shows that the $O(n^{1.7})$ flop empirical cost (open circles/solid line) exceeds the $O(n^{3.5})$ flop theoretical cost rate (dashed line). The difference between the theoretical and empirical rates in this and the previous experiments is attributed to the fact that sparse matrix techniques were used to implement the IP algorithms. Finally, Figure 27 gives an indication of the resulting networks' performance in approximating the fuel flow process. Figure 27(a) plots the mean square error of the approximation for hard (dashed line) and soft (solid line) switching. The results show an initially decreasing and then increasing mean square error trend where the network's approximation capabilities initially increase as more models are added and then decrease due to overfitting of the training data. Finally, Figure 27(b) compares the network's performance for $M = 3$ model units with the target data. It shows that the resulting network approximates the fuel flow process extremely well, with some larger errors being recorded in the transient regions.

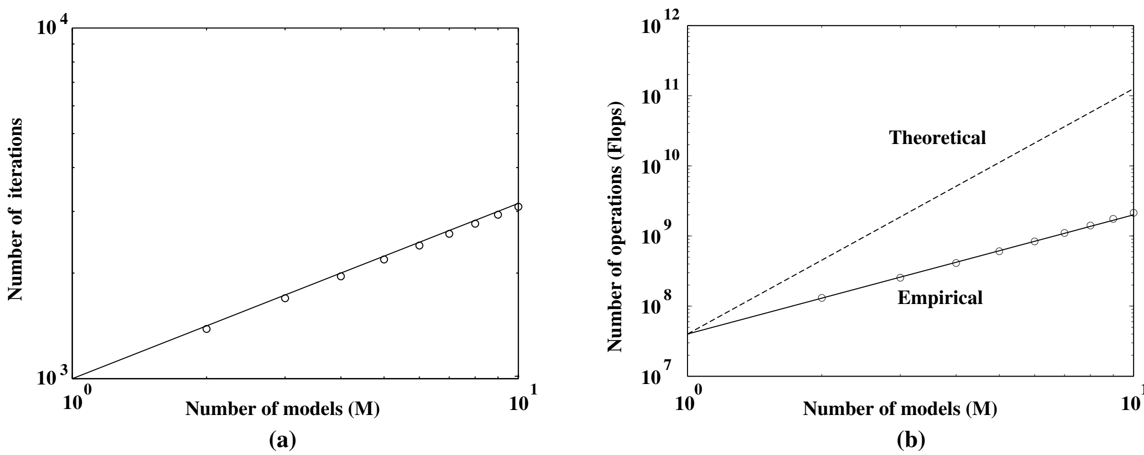

Figure 26: Fuel flow computational properties

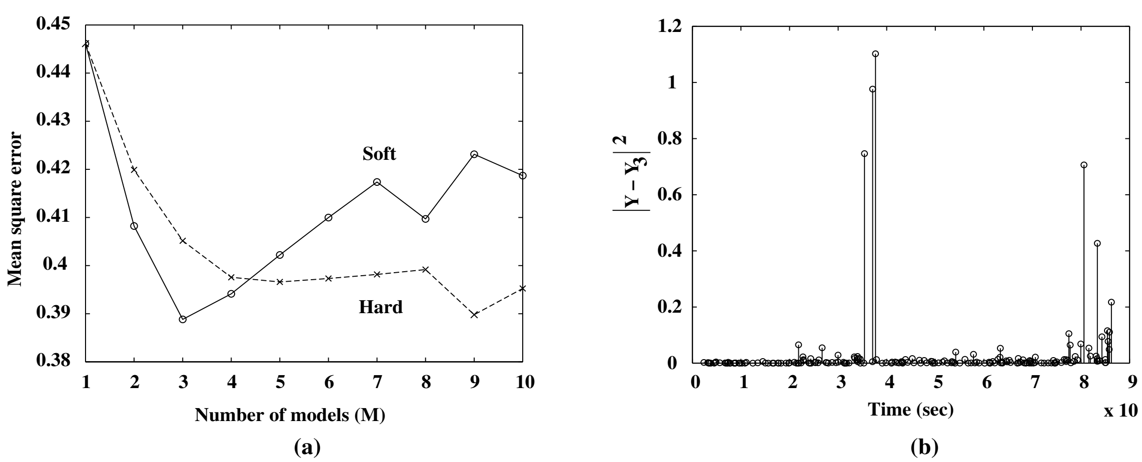

Figure 27: Fuel flow results

6 Discussion

This paper used the hybrid interior point (HIP) algorithm to train modular neural networks. The HIP procedure combines an interior-point linear programming algorithm with a Newton-Raphson iteration to produce an algorithm whose computational cost scales in a polynomial manner with problem size. In particular, this paper formally proves that the HIP algorithm converges to a fixed point that is a local optimum of the regularized training problem. This paper also shows how to configure the algorithm so that the number of iterations and total computational cost scale at the same rate as fast interior-point linear programming algorithms. The formal results obtained in this paper were validated through simulation experiments which compared large-step and small-step versions of the HIP algorithm against Expectation-Maxmimization (EM) procedures. These results showed that for the test suite of problems large-step HIP algorithms exhibited a total computational cost that was comparable to that of the EM procedures, but that the resulting solutions obtained by the HIP algorithm had lower errors than the EM procedure's solutions. The results also showed that the HIP algorithm's total computational costs were relatively insensitive to a problem's initial conditions; something which was definitely not the case with the EM methods. Further experiments demonstrated the use of the HIP algorithm in identifying multiple predictors for nonlinear dynamical plants.

A Appendix

A.1 Proximity proofs

Proof of Corollary 2: From [5], if $\delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}, \bar{\Theta}^{(k)}) \leq 0.5$ and $\alpha^{(k+1)} = \alpha^{(k)}(1 + \nu/\sqrt{n})$ with $\nu \in (0, 0.1]$, then $\delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq 0.7$. If $\delta(\bar{\mathbf{x}}^{(k)}, \alpha, \bar{\Theta}^{(k)}) < 1$, then one SSD step produces $\bar{\mathbf{x}}_1^{(k+1)}$ from $\bar{\mathbf{x}}^{(k)}$ where (Lemma 5.4 in [5]),

$$\delta(\bar{\mathbf{x}}_1^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq \delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)})^2 \leq (0.7)^2.$$

Application of another SSD step produces $\bar{\mathbf{x}}_2^{(k+1)}$ from $\bar{\mathbf{x}}_1^{(k+1)}$ with

$$\delta(\bar{\mathbf{x}}_2^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq \delta(\bar{\mathbf{x}}_1^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)})^2 \leq (0.7)^{2^2}.$$

Proceeding inductively, assuming that $\delta(\bar{\mathbf{x}}_J^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq (0.7)^{2^J}$, then

$$\delta(\bar{\mathbf{x}}_{J+1}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq \delta(\bar{\mathbf{x}}_J^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)})^2 \leq ((0.7)^{2^J})^2 = (0.7)^{2^{J+1}}$$

□

Proof of Theorem 4: The proof must demonstrate that using $\gamma^{(k)}$ in (33) maintains the nearness of $\bar{\mathbf{x}}^{(k+1)}$ to the central path after updating $\bar{\Theta}^{(k)}$. Let $\bar{\mathbf{h}}(\bar{\mathbf{x}}, \alpha, \bar{\Theta}) = P_{AX}X\left(\alpha\bar{\mathbf{c}}(\bar{\Theta}) - \bar{\mathbf{x}}^{-1}\right)$ and define $\bar{\mathbf{h}}_1$ and $\bar{\mathbf{h}}_2$ as $\bar{\mathbf{h}}_1 = \bar{\mathbf{h}}(\bar{\mathbf{x}}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)})$ and $\bar{\mathbf{h}}_2 = \bar{\mathbf{h}}(\bar{\mathbf{x}}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k+1)})$. Using the triangle inequality, $\bar{\mathbf{h}}_1$ and $\bar{\mathbf{h}}_2$ are related as

$$\|\bar{\mathbf{h}}_2\| \leq \|\bar{\mathbf{h}}_2 - \bar{\mathbf{h}}_1\| + \|\bar{\mathbf{h}}_1\|.$$

Substituting $\bar{\mathbf{h}} = P_{AX}X\left(\alpha\bar{\mathbf{c}}(\bar{\Theta}) - \bar{\mathbf{x}}^{-1}\right)$ and δ_1 for $\|\bar{\mathbf{h}}_1\|$ produces

$$\|\bar{\mathbf{h}}_2\| \leq \|\alpha^{(k+1)}P_{AX^{(k+1)}}X^{(k+1)}(\bar{\mathbf{c}}^{(k+1)} - \bar{\mathbf{c}}^{(k)})\| + \delta_1.$$

Applying the quadratic update and using the non-negativity and convexity of $\bar{\mathbf{c}}(\bar{\Theta})$ ($\bar{\mathbf{c}}^{(k+1)} \leq (1 - \gamma^{(k)})\bar{\mathbf{c}}^{(k)} + \gamma^{(k)}\bar{\mathbf{c}}^{(k+1),*}$) results in

$$\|\bar{\mathbf{h}}_2\| \leq \|\alpha^{(k+1)}\gamma^{(k)}P_{AX^{(k+1)}}X^{(k+1)}(\bar{\mathbf{c}}^{(k+1),*} - \bar{\mathbf{c}}^{(k)})\| + \delta_1.$$

Using $\|P_{AX^{(k)}}X^{(k)}\| \leq \Delta^{(k)}$ from [15] and $\Delta^{(k)} \leq (n + 0.5\sqrt{n})/\alpha^{(k)} < 2n/\alpha^{(k)}$, then

$$\begin{aligned}\|\bar{\mathbf{h}}_2\| &\leq \gamma^{(k)}\alpha^{(k+1)}\|P_{AX^{(k+1)}}X^{(k+1)}\|\ \|\bar{\mathbf{c}}^{(k+1),*} - \bar{\mathbf{c}}^{(k)}\| + \delta_1. \\ &< \gamma^{(k)}2n\|\bar{\mathbf{c}}^{(k+1),*} - \bar{\mathbf{c}}^{(k)}\| + \delta_1.\end{aligned}$$

Plugging in the value of γ from (33) yields

$$\|\bar{\mathbf{h}}_2\| < \left[\frac{(\delta_2 - \delta_1)}{2n\|\bar{\mathbf{c}}^{(k+1),*} - \bar{\mathbf{c}}^{(k)}\|}\right] 2n\|\bar{\mathbf{c}}^{(k+1),*} - \bar{\mathbf{c}}^{(k)}\| + \delta_1 = \delta_2.$$

□

Proof of Theorem 5: By assumption, $\delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}, \bar{\Theta}^{(k)}) \leq 0.5$. For the small-step algorithm, Corollary 2 states that two SSD steps produce $\bar{\mathbf{x}}^{(k+1)}$ where $\delta(\bar{\mathbf{x}}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq (0.7)^{2^2} = 0.2401$. For the large-step algorithm, Theorem 3 states that $O(n\mu^2/(1+\mu^2))$ steps produce $\bar{\mathbf{x}}^{(k+1)}$ such that $\delta(\bar{\mathbf{x}}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq 0.5$. Taking a constant two additional steps reduces $\delta(\bar{\mathbf{x}}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)})$ below 0.2401. Thus, $O(n\mu^2/(1+\mu^2))$ SSD steps produce $\bar{\mathbf{x}}^{(k+1)}$ such that $\delta(\bar{\mathbf{x}}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq 0.2401$. A valid $\gamma^{(k)}$ for the quadratic update can be computed using (33) where $\delta_1 = 0.2401$ and $\delta_2 = 0.5$, and the $\bar{\Theta}$-update results in $\bar{\Theta}^{(k+1)}$ such that

$$\delta(\bar{\mathbf{x}}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k+1)}) \leq 0.5.$$

□

A.2 Asymptotic convergence proofs

The convergence analysis begins with a set of definitions. Let $\Omega = (\bar{Q}, \bar{\Theta})$ be the set of solutions produced by the algorithm. Let $\Omega(\bar{Q}) \subset \Omega$ be the set of all $\bar{Q}$'s which are solutions to the LP problem. Let $\Omega(\bar{\Theta}) \subset \Omega$ be the set of all solutions to the quadratic problem. Define $\mathrm{co}(\Omega)$ to be the convex hull of the set of points in Ω. Let γ be defined as

$$\gamma = \min\left(\frac{0.12995}{n\|\bar{\mathbf{c}}(\bar{\Theta}_1) - \bar{\mathbf{c}}(\bar{\Theta}_2)\|}, 1\right).$$

Lemma 13 *Let $\bar{\Theta}^{(k)}, \bar{\Theta}^{(k),*} \in co(\Omega(\bar{\Theta}))$ be the current and minimizing parameter vectors at iteration k. Let $\gamma^{(k)} \in (0,1]$ be defined as in (33) and let the parameter vectors be updated as*

$$\bar{\Theta}^{(k+1)} = (1-\gamma^{(k)})\bar{\Theta}^{(k)} + \gamma^{(k)}\bar{\Theta}^{(k+1),*}. \tag{42}$$

If

$$\|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1),*}\| < \gamma^{(k)}\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\|, \tag{43}$$

then

$$\|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1)}\| < \|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\|. \tag{44}$$

Proof: Begin by adding and subtracting $\bar{\Theta}^{(k+1),*}$ to $\|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1)}\|$ and then using the triangle inequality to break apart the norm

$$\|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1)}\| \leq \|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1),*}\| + \|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k+1)}\|.$$

Using the expression for $\bar{\Theta}^{(k+1)}$ in (42) and then substituting (43) into the result produces the desired result

$$\begin{aligned}
\|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1)}\| &\leq \|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1),*}\| + (1-\gamma^{(k)})\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\| \\
&< \gamma^{(k)}\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\| + (1-\gamma^{(k)})\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\| \\
&< \|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\|
\end{aligned}$$

□

Proof of Theorem 6: Starting with $(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)})$, updating $\alpha^{(k)}$, and performing two SSD steps as

$$\begin{aligned}
\bar{\mathbf{x}}_1 &= \bar{\mathbf{x}}^{(k)} - X^{(k)} P_{AX^{(k)}} X^{(k)} (\alpha^{(k+1)} \bar{\mathbf{c}}^{(k)} - (\bar{\mathbf{x}}^{(k)})^{-1}) \\
\bar{\mathbf{x}}^{(k+1)} &= \bar{\mathbf{x}}_1 - X_1 P_{AX_1} X_1 (\alpha^{(k+1)} \bar{\mathbf{c}}^{(k)} - \bar{\mathbf{x}}_1^{-1})
\end{aligned}$$

where $X^{(k)} = \mathrm{diag}(\bar{\mathbf{x}}^{(k)}) \in \mathbb{R}^{n\times n}$ and $X_1 = \mathrm{diag}(\bar{\mathbf{x}}_1) \in \mathbb{R}^{n\times n}$ results in $\bar{\mathbf{x}}^{(k+1)}$. Let

$$\begin{aligned}
\|\triangle Q\| &= \|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}^{(k)}\| \\
&\leq \|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}_1\| + \|\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}^{(k)}\|.
\end{aligned}$$

Examine the distance moved in the first update, $\|\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}^{(k)}\|$. Use the facts that $\alpha^{(k+1)} = \alpha(1+\nu/\sqrt{n})$ and that P_{AX} is an orthogonal projection ($P_{AX} = P_{AX}^2$).

$$\begin{aligned}
\|\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}^{(k)}\| &= \left\| X^{(k)} P_{AX^{(k)}} X^{(k)} (\alpha^{(k+1)} \bar{\mathbf{c}}^{(k)} - (\bar{\mathbf{x}}^{(k)})^{-1}) \right\| \\
&= \|X^{(k)} P^2_{AX^{(k)}} X^{(k)} \left[(\alpha^{(k)} \bar{\mathbf{c}}^{(k)} - (\bar{\mathbf{x}}^{(k)})^{-1}) + \alpha^{(k)} \nu \bar{\mathbf{c}}^{(k)}/\sqrt{n} \right] \| \\
&\leq \left\| X^{(k)} P_{AX^{(k)}} \right\| \left\| P_{AX^{(k)}} X^{(k)} \left[(\alpha^{(k)} \bar{\mathbf{c}}^{(k)} - (\bar{\mathbf{x}}^{(k)})^{-1}) + \alpha^{(k)} \nu \bar{\mathbf{c}}^{(k)}/\sqrt{n} \right] \right\|
\end{aligned}$$

But $XP_{AX} = P_{AX}X$ and $\|P_{AX^{(k)}} X^{(k)}\| \leq \Delta^{(k)}$ [15] so

$$\|\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}^{(k)}\| \leq \left\| P_{AX^{(k)}} X^{(k)} \left[(\alpha^{(k)} \bar{\mathbf{c}}^{(k)} - (\bar{\mathbf{x}}^{(k)})^{-1}) + \alpha^{(k)} \nu \bar{\mathbf{c}}^{(k)}/\sqrt{n} \right] \right\| \Delta^{(k)}$$

Using (29), $\alpha^{(k)} \frac{\nu}{\sqrt{n}} \|P_{AX^{(k)}} X^{(k)} \bar{\mathbf{c}}^{(k)}\| \leq \nu(\delta + 1)$ [5], and letting $\delta = \delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}, \bar{\Theta}^{(k)})$, the result becomes

$$\begin{aligned}
\|\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}^{(k)}\| &\leq [\delta + \nu(\delta+1)]\, \Delta^{(k)} \\
&\leq (\delta(1+\nu) + \nu)\, \Delta^{(k)}
\end{aligned}$$

Now examine the distance moved in the second update, $\|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}_1\|$

$$\begin{aligned}
\|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}_1\| &= \left\|X_1 P_{AX_1} X_1 (\alpha^{(k+1)}\bar{\mathbf{c}}^{(k)} - \bar{\mathbf{x}}_1^{-1})\right\| \\
&= \|X_1 P_{AX_1}^2 X_1 (\alpha^{(k+1)}\bar{\mathbf{c}}^{(k)} - \bar{\mathbf{x}}_1^{-1})\| \\
&\leq \|P_{AX_1} X_1 (\alpha^{(k+1)}\bar{\mathbf{c}}^{(k)} - \bar{\mathbf{x}}_1^{-1})\|\Delta^{(k)}
\end{aligned}$$

However, $\|P_{AX_1} X_1(\alpha^{(k+1)}\bar{\mathbf{c}}^{(k)} - \bar{\mathbf{x}}_1^{-1})\| = \delta(\bar{\mathbf{x}}_1, \alpha^{(k+1)}, \bar{\Theta}^{(k)})$. Using Lemma 5.4 in [5], it can be shown that $\delta(\bar{\mathbf{w}}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq \delta(\bar{\mathbf{w}}^{(k)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)})^2$. Since $\delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) = \|P_{AX^{(k)}} X^{(k)}(\alpha^{(k+1)}\bar{\mathbf{c}}^{(k)} - (\bar{\mathbf{x}}^{(k)})^{-1})\| = (\delta(1+\nu)+\nu)$ from above, then $\delta(\bar{\mathbf{x}}_1, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq (\delta(1+\nu)+\nu)^2$. Thus,

$$\begin{aligned}
\|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}_1\| &\leq \delta(\bar{\mathbf{x}}_1, \alpha^{(k+1)}, \bar{\Theta}^{(k)})\Delta^{(k)} \\
&\leq (\delta(1+\nu)+\nu)^2 \Delta^{(k)}
\end{aligned}$$

But $(\delta(1+\nu)+\nu) < 1$ and $(\delta(1+\nu)+\nu)^2 < (\delta(1+\nu)+\nu)$, therefore

$$\begin{aligned}
\|\triangle Q\| &\leq \|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}_1\| + \|\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}^{(k)}\| \\
&\leq (\delta(1+\nu)+\nu)^2\Delta^{(k)} + (\delta(1+\nu)+\nu)\Delta^{(k)} \\
&< 2(\delta(1+\nu)+\nu)\Delta^{(k)}
\end{aligned}$$

□

Proof of Theorem 7: Let $\bar{\Theta} = \langle \bar{\theta}_1^T, \bar{\theta}a_2^T, \ldots, \bar{\theta}a_M^T\rangle^T$ and let $\bar{\theta}a_m = \langle \bar{\phi}_m^T, \bar{\omega}_m^T\rangle^T$, then

$$\begin{aligned}
\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k),*}\| &\leq \textstyle\sum_{m=1}^M \|\bar{\theta}_m^{(k+1),*} - \bar{\theta}_m^{(k),*}\| \\
&\leq \textstyle\sum_{m=1}^M \|\bar{\phi}_m^{(k+1),*} - \bar{\phi}_m^{(k),*}\| + \|\bar{\omega}_m^{(k+1),*} + \bar{\omega}_m^{(k),*}\|.
\end{aligned}$$

Let

$$\begin{aligned}
\bar{\mathbf{w}} &= \textstyle\sum_{j=1}^N \frac{1}{N} Q(m|j) y_j \bar{\mathbf{z}}_j \\
\triangle\bar{\mathbf{w}} &= \textstyle\sum_{j=1}^N \frac{1}{N} \triangle Q(m|j) y_j \bar{\mathbf{z}}_j
\end{aligned}$$

where $\bar{\mathbf{w}}, \triangle\bar{\mathbf{w}} \in \mathbb{R}^R$. Using the definitions for $Q^{(k)}(m|j)$, $Q^{(k+1)}(m|j)$, $\bar{\phi}_m^*$, and $\bar{\omega}_m^*$, one finds that

$$\bar{\phi}_m^{(k+1),*} - \bar{\phi}_m^{(k),*} = (A_m + E_m)^{-1}(\bar{\mathbf{w}} + \triangle\bar{\mathbf{w}}) - A_m^{-1}\bar{\mathbf{w}}_m$$

$$\bar{\omega}_m^{(k+1),*} - \bar{\omega}_m^{(k),*} = \frac{\sum_{j=1}^N \frac{1}{N}(Q(m|j) + \triangle Q(m|j))\bar{\mathbf{z}}_j}{\sum_{j=1}^N \frac{1}{N}(Q(m|j) + \triangle Q(m|j))} - \frac{\sum_{j=1}^N \frac{1}{N} Q(m|j)\bar{\mathbf{z}}_j}{\sum_{j=1}^N \frac{1}{N} Q(m|j)}$$

Grouping like terms, taking the norms of the differences, and using the triangle inequality

$$\|\bar{\phi}_m^{(k+1),*} - \bar{\phi}_m^{(k),*}\| \leq \|(A_m + E_m)^{-1} - A_m^{-1}\|\ \|\bar{\mathbf{w}}_m\| + \|(A_m + E_m)^{-1}\|\ \|\triangle\bar{\mathbf{w}}_m\|$$

$$\begin{aligned}
\|\bar{\omega}_m^{(k+1),*} - \bar{\omega}_m^{(k),*}\| \leq\ & \left\|\left[\frac{\sum_{j=1}^N \frac{1}{N} Q(m|j)\bar{\mathbf{z}}_j}{\sum_{j=1}^N \frac{1}{N}(Q(m|j)+\triangle Q(m|j))} - \frac{\sum_{j=1}^N \frac{1}{N} Q(m|j)\bar{\mathbf{z}}_j}{\sum_{j=1}^N \frac{1}{N} Q(m|j)}\right]\right\| \\
&+ \left\|\frac{\sum_{j=1}^N \frac{1}{N}\triangle Q(m|j)\bar{\mathbf{z}}_j}{\sum_{j=1}^N \frac{1}{N}(Q(m|j)+\triangle Q(m|j))}\right\|
\end{aligned}$$

The triangle inequality and the properties of norms can be used to show that $\|\triangle \bar{\mathbf{w}}_m\| \leq \zeta Y_{max}\|\triangle Q\|$, $\|\bar{\mathbf{w}}\| \leq \zeta Y_{max}$, and $\|\sum_{j=1}^{N}\frac{1}{N}Q(m|j)\bar{\mathbf{z}}_j\| \leq \zeta$. Thus

$$\|\bar{\phi}_m^{(k+1),*} - \bar{\phi}_m^{(k),*}\| \leq \zeta Y_{max}\|(A_m+E_m)^{-1} - A_m^{-1}\| + \zeta Y_{max}\|\triangle Q\|\|(A_m+E_m)^{-1}\|$$

$$\begin{aligned}\|\bar{\omega}_m^{(k+1),*} - \bar{\omega}_m^{(k),*}\| &\leq \zeta \left\| \left[\frac{1}{\sum_{j=1}^{N}\frac{1}{N}(Q(m|j)+\triangle Q(m|j))} - \frac{1}{\sum_{j=1}^{N}\frac{1}{N}Q(m|j)} \right] \right\| \\ &+ \frac{\zeta\|\triangle Q\|}{\|\sum_{j=1}^{N}\frac{1}{N}(Q(m|j)+\triangle Q(m|j))\|}\end{aligned}$$

$[A_m + E_m] = \sum_{j=1}^{N}\frac{1}{N}Q^{(k+1)}(m|j)\bar{\mathbf{z}}_j\bar{\mathbf{z}}_j^T$ is full rank by assumption and $\|(A_m+E_m)^{-1}\| \leq \mu_{max}$ and $(\sum_{j=1}^{N}\frac{1}{N}Q^{(k+1)}(m|j))^{-1} \leq \beta_{max}$, so

$$\|\bar{\phi}_m^{(k+1),*} - \bar{\phi}_m^{(k),*}\| \leq \zeta Y_{max}\|(A_m+E_m)^{-1} - A_m^{-1}\| + \mu_{max}\zeta Y_{max}\|\triangle Q\|$$

$$\begin{aligned}\|\bar{\omega}_m^{(k+1),*} - \bar{\omega}_m^{(k),*}\| &\leq \zeta \left\| \frac{1}{\sum_{j=1}^{N}\frac{1}{N}(Q(m|j)+\triangle Q(m|j))} - \frac{1}{\sum_{j=1}^{N}\frac{1}{N}Q(m|j)} \right\| \\ &+\beta_{max}\zeta\|\triangle Q\|\end{aligned}$$

Using Theorem 2.3.4 from [4],

$$\|(A_m+E_m)^{-1} - A_m^{-1}\| \leq \|E_m\|\mu_{max}^2/(1-r_1)$$

and

$$\left\| \frac{1}{\sum_{j=1}^{N}\frac{1}{N}(Q(m|j)+\triangle Q(m|j))} - \frac{1}{\sum_{j=1}^{N}\frac{1}{N}Q(m|j)} \right\| \leq \beta_{max}^2\|\triangle Q\|/(1-r_2)$$

where $\|E_m\| \leq \zeta^2\|\triangle Q\|$ the bound becomes

$$\|\bar{\phi}_m^{(k+1),*} - \bar{\phi}_m^{(k),*}\| \leq \mu_{max}^2\zeta^3 Y_{max}\|\triangle Q\|/(1-r_1) + \mu_{max}\zeta Y_{max}\|\triangle Q\|$$

$$\|\bar{\omega}_m^{(k+1),*} - \bar{\omega}_m^{(k),*}\| \leq \beta_{max}^2\zeta\|\triangle Q\|/(1-r_2) + \beta_{max}\zeta\|\triangle Q\|$$

Substituting these results into the expression for $\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k),*}\|$ produces

$$\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k),*}\| \leq M\mathcal{K}\|\triangle Q\|$$

$\square$

Proof of Theorem 8: Let $\mathcal{A}$ be the set

$$\mathcal{A} = \left\{ \mathrm{co}(\Omega(\bar{\Theta})) \times \mathrm{co}(\Omega(\bar{\Theta})) : \|\bar{\mathbf{c}}(\bar{\Theta}_1) - \bar{\mathbf{c}}(\bar{\Theta}_2)\| \geq \frac{0.12995}{n} \right\}.$$

$\mathcal{A}$ corresponds to all pairs of $\bar{\Theta}$'s in $\mathrm{co}(\Omega(\bar{\Theta})) \times \mathrm{co}(\Omega(\bar{\Theta}))$ that result in γ being less than or exactly equal to unity. Further define $\epsilon_{\bar{\Theta}}$ as

$$\epsilon_{\bar{\Theta}} = \min_{(\bar{\Theta}_1,\bar{\Theta}_2)\in\mathcal{A}} \gamma\|\bar{\Theta}_1 - \bar{\Theta}_2\|. \tag{45}$$

$\epsilon_{\bar{\Theta}}$ corresponds to the minimum value of $\gamma\|\bar{\Theta}_1 - \bar{\Theta}_2\|$ for all pairs of $\bar{\Theta}$'s which result in γ being less than or exactly equal to unity. If the set, $\mathcal{A}$, is empty, then $\epsilon_{\bar{\Theta}}$ is defined to be ∞ as all values of $(\bar{\Theta}_1, \bar{\Theta}_2) \in \mathrm{co}(\bar{\Theta}) \times \mathrm{co}(\bar{\Theta})$ for the problem are such that $\gamma^{(k)} = 1$. $\epsilon_{\bar{\Theta}}$ is bounded away from zero as $\mathrm{co}(\bar{\Theta})$ is a bounded, convex set which means that there is a maximum $\|\bar{\Theta}_1 - \bar{\Theta}_2\|$ and $\|\bar{\mathbf{c}}(\bar{\Theta}_1) - \bar{\mathbf{c}}(\bar{\Theta}_2)\|$ for the set. Let K_1 be the smallest k for which

$$\|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1),*}\| < 2(\delta(1+\nu)+\nu)M\mathcal{K}\Delta^{(k)} < \epsilon_{\bar{\Theta}}. \tag{46}$$

K_1 exists as $\epsilon_{\bar{\Theta}} > 0$ and $\Delta^{(k)} \to 0$ with increasing k. The inequality with $k = K_1$ signifies the iteration, K_1, at which $\|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1),*}\|$ is smaller than the minimum possible term, $\gamma\|\bar{\Theta}_1 - \bar{\Theta}_2\|$. It is at K_1 that the sequence, $\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\|$ becomes strictly decreasing.

Examine $\gamma^{(k)}$ for the following two cases where $k \geq K_1$.

$\gamma^{(k)} < 1$: Equation (46) implies that

$$2(\delta(1+\nu)+\nu)M\mathcal{K}\Delta^{(k)} < \gamma^{(k)}\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\|$$

as $2(\delta(1+\nu)+\nu)M\mathcal{K}\Delta^{(k)} < \epsilon_{\bar{\Theta}}$ and $\epsilon_{\bar{\Theta}} \leq \gamma^{(k)}\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\|$ by its definition. Since $\|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1),*}\| < \gamma^{(k)}\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\|$ and $\gamma^{(k)} < 1$, then by Lemma 13, $\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\|$ is a strictly decreasing sequence. Let

$$\bar{\mathcal{A}} = \left\{ \mathrm{co}(\Omega(\bar{\Theta})) \times \mathrm{co}(\Omega(\bar{\Theta})) : \|\bar{\mathbf{c}}(\bar{\Theta}_1) - \bar{\mathbf{c}}(\bar{\Theta}_2)\| = \frac{0.12995}{n} \right\}.$$

be the set of all pairs of $\bar{\Theta}$'s in $\mathrm{co}(\Omega(\bar{\Theta})) \times \mathrm{co}(\Omega(\bar{\Theta}))$ that result in $\gamma^{(k)}$ being exactly equal to unity. Define the term, a, as

$$a = \min_{(\bar{\Theta}_1, \bar{\Theta}_2) \in \bar{\mathcal{A}}} \|\bar{\Theta}_1 - \bar{\Theta}_2\|$$

a is the minimum separation between parameter vectors in $\mathrm{co}(\Omega(\bar{\Theta}))$ that results in $\gamma^{(k)}$ being exactly equal to one. a is greater than or equal to $\epsilon_{\bar{\Theta}}$. Since $\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\|$ is strictly decreasing to zero as it is bounded above by $\Delta^{(k)}$ which decreases to zero, it eventually decreases past a at which point $\gamma^{(k)} = 1$. Let the iteration at which $\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\| < a$ be $k = K_2$.

$\gamma^{(k)} = 1$: The first case showed that $\gamma^{(k)} \to 1$ for $k \geq K_1$. This second case demonstrates that $\gamma^{(k)} = 1$ for all $k \geq K_2 \geq K_1$. Let $k \geq K_2$ and $\gamma^{(k)} = 1$. Assume that $\gamma^{(k+1)} < 1$. Now, $\gamma^{(k)} = 1$ implies that $\bar{\Theta}^{(k+1)} = (1-\gamma^{(k)})\bar{\Theta}^{(k)} + \gamma^{(k)}\bar{\Theta}^{(k+1),*} = \bar{\Theta}^{(k+1),*}$. $\gamma^{(k+1)} < 1$ implies that

$\|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1)}\| = \|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1),*}\| > a$. But by assumption, $\|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1),*}\| < \epsilon_{\bar{\Theta}}$ and $\epsilon_{\bar{\Theta}} < a$ by definition. This is a contradiction, so the assumption is incorrect and $\gamma^{(k+1)} = 1$.

Thus, for $k \geq K = K_2 \geq K_1 \geq 0$, $\gamma^{(k)} = 1$. □

Proof of Theorem 9: Let the k^{th} solution be $\bar{\mathbf{w}}^{(k)} = \langle (\bar{\mathbf{x}}^{(k)})^T, (\bar{\Theta}^{(k)})^T \rangle^T$ and define the term, $\bar{\mathbf{w}}^* = \langle (\bar{\mathbf{x}}^*)^T, (\bar{\Theta}^*)^T \rangle^T$. Begin by examining $\|\bar{\mathbf{w}}^{(k+1)} - \bar{\mathbf{w}}^{(k)}\|$ for $k \geq K$.

$$\begin{aligned}
\|\bar{\mathbf{w}}^{(k+1)} - \bar{\mathbf{w}}^{(k)}\| &\leq \|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}^{(k)}\| + \|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\| \\
&\leq \|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}^{(k)}\| + \|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k),*}\| \\
&< 2(\delta(1+\nu)+\nu)\Delta^{(k)} + M\mathcal{K}2(\delta(1+\nu)+\nu)\Delta^{(k)} \\
&< 2(\delta(1+\nu)+\nu)(1+M\mathcal{K})\Delta^{(k)}
\end{aligned}$$

Examine the following sum for all $j \geq k \geq K$

$$\sum_{j=k}^{\infty} \|\bar{\mathbf{w}}^{(j+1)} - \bar{\mathbf{w}}^{(j)}\| < \sum_{j=k}^{\infty} 2(\delta(1+\nu)+\nu)(1+M\mathcal{K})\Delta^{(j)} \tag{47}$$

The duality gap, $\Delta^{(k)}$, can be expressed as $\Delta^{(k)} = \beta^k \Delta^{(0)}$ where $\beta = 1/(1+\nu/\sqrt{n})$ [5]. Using this fact, (47) can be simplified as

$$\begin{aligned}
\textstyle\sum_{j=k}^{\infty} \|\bar{\mathbf{w}}^{(j+1)} - \bar{\mathbf{w}}^{(j)}\| &< \textstyle\sum_{j=k}^{\infty} 2(\delta(1+\nu)+\nu)(1+M\mathcal{K})\Delta^{(j)} \\
&< \textstyle\sum_{j=k}^{\infty} 2(\delta(1+\nu)+\nu)(1+M\mathcal{K})\beta^j \Delta^{(0)} \\
&< 2(\delta(1+\nu)+\nu)(1+M\mathcal{K})\tfrac{\Delta^{(k)}}{1-\beta}
\end{aligned}$$

This result corresponds to a constant, $2(\delta(1+\nu)+\nu)(1+M\mathcal{K})/(1-\beta)$, multiplied by the strictly decreasing duality gap, $\Delta^{(k)}$. Let k be chosen so that $\sum_{j=k}^{\infty} \|\bar{\mathbf{w}}^{(j+1)} - \bar{\mathbf{w}}^{(j)}\| < \epsilon$ for $\epsilon > 0$ (i.e. let the duality gap be small enough so that the condition holds). If k is so chosen, then $\forall i, j \geq k$, $\|\bar{\mathbf{w}}^{(i)} - \bar{\mathbf{w}}^{(j)}\| < \epsilon$ for arbitrary ϵ. Thus, $\{\bar{\mathbf{w}}^{(k)}\}$ is a Cauchy sequence; and it converges to a fixed point, $\bar{\mathbf{w}}^* = \langle (\bar{\mathbf{x}}^*)^T, (\bar{\Theta}^*)^T \rangle^T$. □

Proof of Theorem 10: Begin by assuming that $\bar{\mathbf{w}}^* = \langle (\bar{\mathbf{x}}^*)^T, (\bar{\Theta}^*)^T \rangle^T$ is not a local minimum. Then, there exists a feasible direction $\triangle\bar{\mathbf{w}} = \langle \triangle\bar{\mathbf{x}}^T, \triangle\bar{\Theta}^T \rangle^T$ from $\bar{\mathbf{w}}^*$ in the solution space such that

$$f(\bar{\mathbf{w}}^* + \lambda\triangle\bar{\mathbf{w}}) \leq f(\bar{\mathbf{w}}^*) \tag{48}$$

for some $\lambda \in (0, \infty)$ and

$$f(\bar{\mathbf{w}}^* + \sigma\triangle\bar{\mathbf{w}}) < f(\bar{\mathbf{w}}^*) \tag{49}$$

for some $\sigma \in [0, \lambda)$ where $f(\bar{\mathbf{w}})$ is the cost function. Let W be the set of all feasible directions at $\bar{\mathbf{w}}^*$. Let $\triangle\bar{\mathbf{w}} \in W$ where $\triangle\bar{\mathbf{w}} = \langle \triangle\bar{\mathbf{x}}^T, \triangle\bar{\Theta}^T \rangle^T$ and let $\bar{\mathbf{w}}_1 = \bar{\mathbf{w}}^* + \lambda\triangle\bar{\mathbf{w}} = \langle \bar{\mathbf{x}}_1^T, \bar{\Theta}_1^T \rangle^T$. Two mutually exclusive cases can occur: $\bar{\mathbf{x}}^*$ can be a unique optimal solution to (39) when $\bar{\Theta}$ is held constant, or $\bar{\mathbf{x}}^*$ can be a non-unique optimal solution. The first case occurs when $\bar{\mathbf{c}}(\bar{\Theta})$ supports the polyhedral set of feasible solutions only at the vertex, $\bar{\mathbf{x}}^*$. The second occurs when $\bar{\mathbf{c}}(\bar{\Theta})$ supports the set at vertex, $\bar{\mathbf{x}}$, and is also parallel to one of the faces of the polyhedral set which also contains $\bar{\mathbf{x}}^*$.

1. $\bar{\mathbf{x}}^*$ is a unique optimal solution.

   $\bar{\mathbf{x}}^*$ being optimal implies that for all feasible directions, $\triangle\bar{\mathbf{x}}$, from $\bar{\mathbf{x}}^*$ [1]

   $$\bar{\mathbf{c}}(\bar{\Theta}^*)^T \triangle\bar{\mathbf{x}} > 0.$$

   Moving in direction $\triangle\bar{\mathbf{w}} = \langle \triangle\bar{\mathbf{x}}^T, \triangle\bar{\Theta}^T \rangle^T$ changes $\bar{\Theta}^*$ as $\bar{\Theta}_1 = \bar{\Theta}^* + \lambda\triangle\bar{\Theta}$. This rotates the cost vector as $\bar{\mathbf{c}}(\bar{\Theta}_1) = \bar{\mathbf{c}}(\bar{\Theta}^*) + \lambda\hat{\mathbf{c}}(\triangle\bar{\Theta})$ where $\hat{\mathbf{c}}(\triangle\bar{\Theta})$ is a term that denotes the change in the cost vector due to $\triangle\bar{\Theta}$. If λ is small enough, then $\lambda\hat{\mathbf{c}}(\triangle\bar{\Theta})^T\triangle\bar{\mathbf{x}} < \bar{\mathbf{c}}(\bar{\Theta}^*)^T\triangle\bar{\mathbf{x}}$ and $\bar{\mathbf{c}}(\bar{\Theta}_1)^T\triangle\bar{\mathbf{x}} > 0$ for all feasible $\triangle\bar{\mathbf{x}}$ from $\bar{\mathbf{x}}^*$. Thus, $\bar{\mathbf{x}}^*$ is still the optimal solution to the LP problem associated with (39). If $\bar{\mathbf{x}}^*$ is the optimal solution, then

   $$\bar{\mathbf{c}}(\bar{\Theta}_1)^T\bar{\mathbf{x}}^* < \bar{\mathbf{c}}(\bar{\Theta}_1)^T\bar{\mathbf{x}}_1.$$

   The minimizing solution to the quadratic optimization of $\bar{\mathbf{c}}(\bar{\Theta})^T\bar{\mathbf{x}}^*$ with respect to $\bar{\Theta}$ is $\bar{\Theta}^*$. Thus,

   $$\bar{\mathbf{c}}(\bar{\Theta}^*)^T\bar{\mathbf{x}}^* < \bar{\mathbf{c}}(\bar{\Theta}_1)^T\bar{\mathbf{x}}^* < \bar{\mathbf{c}}(\bar{\Theta}_1)^T\bar{\mathbf{x}}_1$$

   For feasible direction $\triangle\bar{\mathbf{w}} = \langle \triangle\bar{\mathbf{x}}^T, \triangle\bar{\Theta}^T \rangle^T$, the cost of the solution $\bar{\mathbf{w}}^* = \langle (\bar{\mathbf{x}}^*)^T, (\bar{\Theta}^*)^T \rangle^T$ is less than the cost of the solution $\bar{\mathbf{w}}_1 = \langle \bar{\mathbf{x}}_1^T, \bar{\Theta}_1^T \rangle^T$. Since $\triangle\bar{\mathbf{w}}$ was any arbitrary feasible direction from $\bar{\mathbf{w}}^*$, there is no feasible direction that satisfies (48) and (49), which is a contradiction. Thus, the assumption is incorrect and $\bar{\mathbf{w}}^* = \langle (\bar{\mathbf{x}}^*)^T, (\bar{\Theta}^*)^T \rangle^T$ is a local minimum.

2. $\bar{\mathbf{x}}^*$ is a non-unique optimal solution to the LP problem.

   $\bar{\mathbf{x}}^*$ being non-unique implies that for all feasible directions $\triangle\bar{\mathbf{x}}$ from $\bar{\mathbf{x}}^*$

   $$\bar{\mathbf{c}}(\bar{\Theta}^*)^T \triangle\bar{\mathbf{x}} \geq 0$$

   This can be broken down into two cases, those for which $\bar{\mathbf{c}}(\bar{\Theta}^*)^T\triangle\bar{\mathbf{x}} > 0$ and those for which $\bar{\mathbf{c}}(\bar{\Theta}^*)^T\triangle\bar{\mathbf{x}} = 0$. The first considers those directions in which movement causes a strict increase in the cost function, while the second considers those directions in which movement causes no change in the cost function.

(a) Directions for which $\bar{\mathbf{c}}(\bar{\Theta}^*)^T \triangle\bar{\mathbf{x}} > 0$.

For movement in these directions, λ can be chosen small enough that $\lambda\hat{\mathbf{c}}(\triangle\bar{\Theta})^T\triangle\bar{\mathbf{x}} < \bar{\mathbf{c}}(\bar{\Theta}^*)^T\triangle\bar{\mathbf{x}}$ and the reasoning of case 1 applies, which generates the contradiction.

(b) Directions for which $\bar{\mathbf{c}}(\bar{\Theta}^*)^T \triangle\bar{\mathbf{x}} = 0$.

Here, the cost hyperplane $\bar{\mathbf{c}}(\bar{\Theta}^*)$ is parallel to one of the constraint planes a_i where a_i is a row of A. This implies a set of optimal solutions $\{\bar{\mathbf{x}} : \bar{\mathbf{x}} = \beta\bar{\mathbf{x}}^* + (1-\beta)\bar{\mathbf{x}}_1^*, \beta \in [0,1]\}$ where $\bar{\mathbf{x}}^*$ and $\bar{\mathbf{x}}_1^*$ are extreme points of the polyhedral set of feasible solutions and $\triangle\bar{\mathbf{x}}_1 = (\bar{\mathbf{x}}_1^* - \bar{\mathbf{x}}^*)/\|\bar{\mathbf{x}}_1^* - \bar{\mathbf{x}}^*\|$. Thus, $\triangle\bar{\mathbf{x}}_1$ is a feasible direction from $\bar{\mathbf{x}}^*$ to $\bar{\mathbf{x}}_1^*$ and $-\triangle\bar{\mathbf{x}}_1$ is a feasible direction from $\bar{\mathbf{x}}_1^*$ to $\bar{\mathbf{x}}^*$. Since $\bar{\mathbf{x}}_1^*$ is also an optimal solution, $\bar{\mathbf{c}}(\bar{\Theta}^*)^T\triangle\bar{\mathbf{x}} \geq 0$ for all feasible directions from $\bar{\mathbf{x}}_1^*$ and $\bar{\mathbf{c}}(\bar{\Theta}^*)^T(-\triangle\bar{\mathbf{x}}_1) = 0$ for the feasible direction $-\triangle\bar{\mathbf{x}}_1$ leading from $\bar{\mathbf{x}}_1^*$ to $\bar{\mathbf{x}}^*$. Without loss of generality, assume $\triangle\bar{\mathbf{x}}_1$ is such that

$$\lambda\hat{\mathbf{c}}(\triangle\bar{\Theta})^T\triangle\bar{\mathbf{x}}_1 > 0$$

which implies that

$$\bar{\mathbf{c}}(\bar{\Theta}_1)^T\triangle\bar{\mathbf{x}}_1 = \bar{\mathbf{c}}(\bar{\Theta}^*)^T\triangle\bar{\mathbf{x}}_1 + \lambda\hat{\mathbf{c}}(\triangle\bar{\Theta})^T\triangle\bar{\mathbf{x}}_1 > 0$$

With this small rotation of the cost plane, $\bar{\mathbf{x}}^*$ is the unique optimal solution to the LP problem. Thus, $\bar{\mathbf{c}}(\bar{\Theta}_1)^T\bar{\mathbf{x}}^* < \bar{\mathbf{c}}(\bar{\Theta}_1)^T\bar{\mathbf{x}}_1$. Since $\bar{\Theta}^*$ is the optimal quadratic minimizer for (39) subject to constant $\bar{\mathbf{x}} = \bar{\mathbf{x}}^*$, then $\bar{\mathbf{c}}(\bar{\Theta}^*)^T\bar{\mathbf{x}}^* < \bar{\mathbf{c}}(\bar{\Theta}_1)^T\bar{\mathbf{x}}^* < \bar{\mathbf{c}}(\bar{\Theta}_1)^T\bar{\mathbf{x}}_1$. This again leads to a contradiction which implies that the solution, $\bar{\mathbf{w}}^*$ is a local minimum.

□

A.3 Computational property proofs

Proof of Theorem 11: First, the assumption that S is chosen as indicated guarantees that $\|\bar{\mathbf{w}}^{(k+1)} - \bar{\mathbf{w}}^{(k)}\|$ will be a Cauchy sequence for $k \geq K$. Since $\|\bar{\mathbf{w}}^{(k+1)} - \bar{\mathbf{w}}^{(k)}\|$ is Cauchy, the norm of the difference between the current and optimal solutions can be expressed in closed form as

$$\|\bar{\mathbf{w}}^{(k)} - \bar{\mathbf{w}}^*\| < 2(\delta(1+\nu)+\nu)(1+M\mathcal{K})\frac{\Delta^{(k)}}{1-\beta} \tag{50}$$

Using (50), examine how long it will take for $\|\bar{\mathbf{w}}^{(k)} - \bar{\mathbf{w}}^*\| < \epsilon$. The goal is to find the smallest k that violates

$$\epsilon < 2(\delta(1+\nu)+\nu)(1+M\mathcal{K})\frac{\Delta^{(k)}}{1-\beta}$$

By definition of the duality gap, $\Delta^{(k)} = \beta^k \Delta^{(0)} < \beta^k (1/\epsilon)$. So

$$\epsilon < 2(\delta(1+\nu)+\nu)(1+M\mathcal{K})\frac{\beta^k(1/\epsilon)}{1-\beta}.$$

Taking the base 2 logarithm,

$$\log_2 \epsilon < \log_2 2(\delta(1+\nu)+\nu)(1+M\mathcal{K}) + \log_2 1/\epsilon + \log_2 \frac{1}{1-\beta} + k\log_2 \beta$$

After some arithmetic, one finds that

$$k < \left(2\log_2 1/\epsilon + \log_2 2(\delta(1+\nu)+\nu)(1+M\mathcal{K}) + \log_2 \frac{1}{1-\beta}\right) / \log_2 \frac{1}{\beta}$$

But $\log_2(1+\mu) \geq \mu/(1+\mu)$ [5], and using this expression for $\log_2(1/\beta)$ results in

$$k < \left(2\log_2 1/\epsilon + \log_2 2(\delta(1+\nu)+\nu)(1+M\mathcal{K}) + \log_2 \frac{1}{1-\beta}\right)(\sqrt{n}/\nu+1)$$

Also, $\log_2(1/(1-\beta)) = \log_2(\sqrt{n}/\nu+1)$, so

$$k < (2\log_2 1/\epsilon + \log_2 2(\delta(1+\nu)+\nu)(1+M\mathcal{K}) + \log_2(\sqrt{n}/\nu+1))(\sqrt{n}/\nu+1)$$

Since $2(\delta(1+\nu)+\nu)(1+M\mathcal{K})$ and ν are constants, $k = O(\sqrt{n}\log_2(\sqrt{n}/\epsilon))$ iterations. $\square$

References

[1] Bazaraa, M.S., Sherali, H.D., and Shetty, C.M. (1993). *Nonlinear Programming: Theory and Algorithms.* John Wiley and Sons, Inc.

[2] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, **2**, 304-314

[3] Dempster, A.P., Laird, N.M., and Rubin, D.B., (1977). Maximum Likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society.* **39**, 1-38.

[4] Golub, G.H., Van Loan, C.F.(1989). *Matrix Computations.* The Johns Hopkins University Press, Baltimore, Maryland.

[5] Gonzaga, C.C. (1992). Path follwoing methods for linear programming. *SIAM Review.* **34**, 167-224.

[6] Hadamard, J. (1964). *La Theorie des Equations aux Derivees Partielles.* Edition Scientifiques, Beijing.

[7] Hornik, K. M. Stinchcombe, and H. White (1989). Multilayer feedfoward networks are universal approximators, *Neural Networks.* **2**, 359-366.

[8] Jacobs, R.A., Jordan, M.I., Nowlan, S.J., and Hinton, G.E. (1991). Adaptive mixtures of local experts. *Neural Computation.* **3**, 79-87.

[9] Jordan, M.I., Xu, L. (1993). Convergence results for the EM approach to mixtures of experts architectures. *Technical Report 9303.* MIT Computational Cognitive Science, September 1993.

[10] Karmarkar, N. (1984). a new polynomial-time algorithm for linear programming. *Combinatorics.* **4**, 373-395.

[11] Lemmon, M.D., Szymanski, P.T. (1994). Interior point implementations of alternating minimization training. In *Advances in Neural Information Processing Systems 7*,574-582, San Mateo California, MIT Press.

[12] Park, J. and Sandberg, I.W. (1991), Universal approximation using radial basis function networks. *Neural Computation.* **3**, 246-257.

[13] Poggio, T. , Girosi, F. (1990). Networks for Approximation and Learning. *Proceedings of the IEEE*, **78**, 1481-1497.

[14] Redner, R.A., Walker, H.F. (1984). mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, **26**, 195-239.

[15] Szymanski, P.T. (1995) *Alternating Minimization of Non-convex Constrained Optimizations via Interior Point Techniques.* Ph.D. thesis, Dept. of Electrical Engineering, University of Notre Dame, November 1995.