An Hybrid Interior Point Algorithm for Performing Alternating Minimizations

Technical Report of the ISIS Group
at the University of Notre Dame
ISIS-95-003
Apr, 95

Peter T. Szymanski and Michael Lemmon
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556

Interdisciplinary Studies of Intelligent Systems

An Hybrid Interior Point Algorithm for Performing Alternating Minimizations

Peter T. Szymanski and Michael Lemmon*

Dept. of Electrical Engineering

University of Notre Dame

Notre Dame, IN 46556

April 28, 1995

Abstract

Optimization using alternating minimization (AM) alternately minimizes an objective function with respect to two disjoint subsets of a function's parameters. AM algorithms are useful for optimization problems which can be decomposed into convex subproblems. AM algorithms have recently appeared in the neural computing community. This paper examines the use of recently developed *Interior Point* (IP) techniques to implement an AM algorithm which solves an important class of optimization problems. The IP techniques studied in this paper solve optimization problems that can be decomposed into linear and quadratic subproblems. Such problems appear frequently in multiple model parameter identification. The proposed algorithm is a version of the generalized *Expectation-Maximization* algorithm and is proven to converge to locally optimal solutions in $O(\sqrt{n}L)$ iterations with an associated computational cost of $\max(O(n^{3.5}L), O(n^{1.5}R^2L), O(\sqrt{n}MR^3L))$ where n is the dimension of an associated linear programming problem, M is the number of models, R is the dimension of the parameter vectors being identified, and L is a measure of the solution's accuracy.

*The authors would like to acknowledge the partial financial support of the National Science Foundation (MSS92-16559) and the Electric Power Research Institute (RP8030-06)

1 Introduction

Consider the problem of minimizing an objective function, $h(\Phi)$, with respect to its parameters $\Phi = \cup_{i=1}^{n}\Phi_i$, where the Φ_i are mutually disjoint subsets of parameters $\Phi_i \cap \Phi_j = \emptyset$ for $i \neq j$.

Problem 1

$$\begin{aligned} \textit{minimize:} &\quad h(\Phi) \\ \textit{with respect to:} &\quad \Phi \end{aligned}$$

Alternating minimization (AM) algorithms decompose this problem into n subproblems. The i^{th} subproblem has the form

Problem 2

$$\begin{aligned} \textit{minimize:} &\quad h(\Phi) \\ \textit{with respect to:} &\quad \Phi_i \\ \textit{subject to:} &\quad \Phi_j \textit{ constant, for all } j \neq i \end{aligned}$$

for $i = 1, \ldots, n$. AM algorithms solve Problem 1 by minimizing $h(\Phi)$ with respect to each of the disjoint sets, Φ_i, in sequence from $i = 1$ to n and iterating over this sequence of minimizations until an optimal solution is found.

AM algorithms are useful when $h(\Phi)$ is a nonconvex function which is convex in some or all of its subsets of parameters, Φ_i $(i = 1, \ldots, n)$. Problem 1 may be difficult to solve due to the nonconvex nature of $h(\Phi)$. The n subproblems, however, can be readily solved using convex optimization techniques. The resulting solutions may be only locally optimal, but since the original problem is assumed to be nonconvex, that is the best that can be expected using conventional techniques.

Training or learning algorithms which use alternating minimization have recently appeared in the neural computing community. Soft competitive learning [1] can be viewed as an AM training algorithm where the likelihood of generating vectors from a number of classes is maximized by first computing estimates for the means of a set of radial basis functions and then by computing estimates for mixing probabilities based upon the estimates of the means. Training in the mixtures of experts [2] approach uses an alternating minimization where the mean square error of a network of experts is minimized by first computing estimates of a set of gating probabilities associated with the network and then by computing estimates for the parameters of the experts using the new values of the gating probabilities. Boltzmann machine training in [3] also uses an AM where locally optimal Boltzmann machines are found by minimizing the information divergence between two distributions that characterize the performance of the machine. The divergence is minimized first with respect to the connectivities or weights of the machine and then with respect to the probabilities associated with the Boltzmann machine.

The methods discussed above are all versions of the generalized Expectation-Maximization (EM) algorithm [4] used to compute maximum likelihood parameter estimates from incomplete data. They all optimize a performance measure based upon a stochastic formulation of the problem. In fact, the soft competition methods from the neural community are closely related to Blahut's algorithm [5], an EM algorithm used in the source coding community which alternately minimizes the average mutual information (AMI) of a coding subject to constraints on the coding's distortion. Blahut's algorithm minimizes AMI with respect to the marginal and conditional distributions associated with the coding. It was shown in [6] that the updates of the conditional probabilities performed by Blahut's algorithm are identical to the updates of the mixing or gating probabilities in soft competitive learning [1] or mixtures training [2].

This paper examines the use of hybrid *interior point* (IP) techniques for performing alternating minimizations. IP techniques generate a sequence of solutions to an optimization problem which lie in the *interior* of the set of feasible solutions to the problem. IP techniques are designed so that the sequence of solutions converges to the optimal solution in a controlled fashion. The proposed hybrid IP algorithm combines fast, efficient linear program solvers with standard Newton-Raphson (NR) methods. The hybrid algorithm is a modification of a small step, path-following IP algorithm for solving linear programming (LP) problems [7]. These path following techniques are fast, requiring $O(\sqrt{n}L)$ iterations to converge to an optimal solution with a computational cost of $O(n^{3.5}L)$ operations where n refers to the dimension of the LP problem and L is a measure of the accuracy of the solution. The hybrid IP algorithm retains the excellent computational properties of its LP counterpart by restricting how aggressively the component linear and quadratic optimizations may be pursued. The resulting algorithm using an alternating minimization finds locally optimal solutions with a computational complexity that is polynomial in the size of the problem which means that the algorithm will scale well for large problems.

The hybrid algorithm provides an efficient means of training linear regressor and radial basis function networks. Such networks have recently attracted a great deal of attention due to their interpretation as neural or fuzzy networks. The performance of these networks can be quantified by an error measure which is parameterized by the coefficients of the linear regressors, a set of means for the radial basis functions and a set of conditional training probabilities. The error measure is a nonconvex function of these two parameter sets. The measure is, however, convex in each set separately; being linear in the set of conditional probabilities and quadratic in the parameters of the network. Training of these networks consists of using the hybrid IP algorithm to alternately adjust the parameters of the network and training probabilities to minimize the error measure. The types of networks under consideration are those that may arise in problems in the control of hybrid systems [8, 9], time series prediction and system identification problems [10], and in supervisory and hierarchical control problems [11, 12].

The remainder of the paper examines the hybrid IP algorithm for performing alternating minimizations using a network of linear regressors as an example. Section 2 describes the linear regressor network and presents a constrained optimization problem that is solved in identifying the network's parameters. Section 3 describes the hybrid IP algorithm and the linear and quadratic optimization algorithms upon which it is based. Section 4 provides a theoretical analysis of the hybrid IP algorithm's convergence. Examples of the algorithm's use appear in Section 5, and are followed by a discussion in Section 6.

2 Problem Statement

This section describes a constrained optimization problem which is solved using an alternating minimization implemented using the hybrid IP algorithm presented in Section 3. Section 2.1 describes a network of linear regressors used to form local linear approximations of a nonlinear function. Section 2.2 presents the constrained optimization problem solved to determine optimal network parameters.

2.1 Network Formulation

Consider a problem to train a linear regressor network (which is described below) to approximate a bounded, continuous nonlinear function, $f : \mathbb{R}^R \to \mathbb{R}$. Assume that there is a set of observations of the function available to train the network, $\mathcal{T} = \{(y_j, \mathbf{z}_j) : y_j = f(\mathbf{z}_j)\}$ for $\mathbf{z}_j \in \mathbb{R}^R$ and $j = 1, \ldots, N$. Assume that the network produces a prediction, $\hat{y} \in \mathbb{R}$, in response to an input, $\mathbf{z} \in \mathbb{R}^R$. The problem is to determine a parametrization for the network such that the likelihood of producing a prediction, $\hat{y} \in \mathbb{R}$, which is close to the actual output, $y = f(\mathbf{z}) \in \mathbb{R}$, for any input, $\mathbf{z} \in \mathbb{R}^R$, in terms of the mean square output error, $E_{\mathbf{z}\in\mathbb{R}^R}[(\hat{y} - y)^2]$, is maximized.

The linear regressor network, pictured in Figure 1(a), consists of $M+1$ agents, M model agents which are moderated by a single gating agent. The model agents generate local approximations of f around different set points. The gating agent computes the approximating output for the network using the outputs of the model agents and a set of activation levels generated for each model agent. The specifics of the model agents and the gating agent follow.

The model agents are linear regressors with associated radial basis function (RBF) activation units as shown in Figure 1(b). The m^{th} model agent produces an output, y_m, in response to a given input, $\mathbf{z} \in \mathbb{R}^R$, as

$$y_m = \phi_m^T \mathbf{z} \tag{1}$$

where $\phi_m \in \mathbb{R}^R$ is the set of weights associated with the m^{th} agent ($m = 1, \ldots, M$). Associated

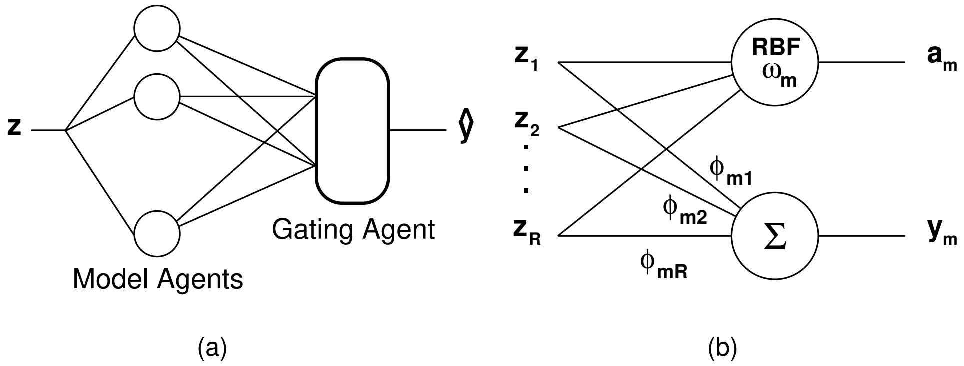

Figure 1: Linear regressor network

with the m^{th} regressor is an RBF parameterized by mean, $\omega_m \in \mathbb{R}^R$, and variance, σ_a^2

$$a_m(\mathbf{z}) = k_{m_a} \exp(\frac{-1}{\sigma_a^2}||\mathbf{z} - \omega_m||^2) \tag{2}$$

where k_{m_a} is a normalizing constant. The RBF associated with the m^{th} agent determines the activation level for the agent and serves as a mechanism to partition the input space among the M agents. The m^{th} RBF generates higher activation levels for those inputs which are nearer to its mean, ω_m. Higher activation levels indicate a greater likelihood that the input is best approximated by the m^{th} agent. The parametrization for the m^{th} model agent is $\theta_m = \langle \phi_m^T, \omega_m^T \rangle^T \in \mathbb{R}^{2R}$. The parametrization for all the agents is $\Theta = \langle \theta_1^T, \theta_2^T, \ldots, \theta_M^T \rangle^T$.

The gating agent computes the network output, $\hat{y}$, in response to an input $\mathbf{z} \in \mathbb{R}^R$, using the model agents' outputs, y_m $(m = 1, \ldots, M)$, and the activation levels, $a_m(\mathbf{z})$ $(m = 1, \ldots, M)$. It computes the output in one of two fashions. First, the gating agent can compute the output as a strict or *hard* choice where $\hat{y}$ is the output of the m^{th} agent

$$\hat{y} = y_m; \quad m = \arg\max_{m=1}^{M}(q(m)a_m(\mathbf{z})) \tag{3}$$

where $q(m)$ is the probability of using the m^{th} model. Second, the gating agent can compute the output as a *soft* choice where $\hat{y}$ is a weighted sum of the M model agents

$$\hat{y} = \frac{\sum_{m=1}^{M} q(m)a_m(\mathbf{z})y_m}{\sum_{m=1}^{M} q(m)a_m(\mathbf{z})}. \tag{4}$$

2.2 Problem formulation

The problem is to determine the parameters of the network that minimize both the square error between $\hat{y}$ and $f(\mathbf{z})$ and the error associated with classifying an input to one of the M models for any $\mathbf{z} \in \mathbb{R}^R$. A two-part measure is required that describes the performance of the network in terms of its parameters, $\boldsymbol{\theta}_m = \langle \phi_m^T, \omega_m^T \rangle^T$ $(m = 1, \ldots, M)$.

Let $(\phi_m^T \mathbf{z} - f(\mathbf{z}))^2$ be the *square output error* between the outputs of the m^{th} model agent and the function in response to $\mathbf{z}$. The functional form of $f(\mathbf{z})$ is unknown and cannot be used directly in the square error term. A finite design set, $\mathcal{T} = \{(y_j, \mathbf{z}_j) : y_j = f(\mathbf{z}_j), j = 1, \ldots, N\}$, is available, however, that describes $f(\mathbf{z})$. The square output error between the output of the m^{th} model agent and the function for the j^{th} design sample is $(\phi_m^T \mathbf{z}_j - y_j)^2$. Let $p(j)$ be the probability that the j^{th} design sample occurs in $\mathcal{T}$ and let $Q(m|j)$ be the conditional probability that the j^{th} design sample is assigned to the m^{th} model. Let the set of all $Q(m|j)$ be denoted by Q. The mean square output error of the network is then

$$e(Q, \Theta, \mathcal{T}) = \sum_{j=1}^{N} \sum_{m=1}^{M} p(j) Q(m|j)(y_j - \phi_m^T \mathbf{z}_j)^2 \tag{5}$$

The second part of the performance measure is in terms of ω_m and quantifies the *classification error* of the network. Recall that the m^{th} model agent has an associated RBF parameterized by ω_m. The M RBF's serve as a pattern classification mechanism for associating an input $\mathbf{z}$ with one of the M model agents. Let $\|\mathbf{z}_j - \omega_m\|^2$ be the square classification error associated with assigning the j^{th} sample point to the m^{th} model agent. Let $p(j)$ and $Q(m|j)$ be defined as above. The mean square classification error of the network in assigning the N design samples to the M model agents is

$$c(Q, \Theta, \mathcal{T}) = \sum_{j=1}^{N} \sum_{m=1}^{M} p(j) Q(m|j) \|\mathbf{z}_j - \omega_m\|^2 \tag{6}$$

A measure that combines the square output error and square classification error quantifies the performance of the network. Let $1/\sigma_e^2$ be a constant that weights the contribution of the output error to the performance measure, and let $1/\sigma_a^2$ be a constant that weights the contribution of the classification error to the performance measure. The complete performance of the network is

$$\bar{d}(Q, \Theta, \mathcal{T}) = \sum_{j=1}^{N} \sum_{m=1}^{M} p(j) Q(m|j) \left[\frac{1}{\sigma_e^2} (y_j - \phi_m^T \mathbf{z}_j)^2 + \frac{1}{\sigma_a^2} \|\mathbf{z}_j - \omega_m\|^2 \right] \tag{7}$$

Parameter identification is accomplished by minimizing $\bar{d}(Q, \Theta, \mathcal{T})$ with respect to both Q and Θ. Minimizing $\bar{d}(Q, \Theta, \mathcal{T})$ with respect to Q determines an optimal assignment of the N design samples to the M models. Minimizing $\bar{d}(Q, \Theta, \mathcal{T})$ with respect to Θ determines "optimal" regressor vector estimates, ϕ_m $(m = 1, \ldots, M)$, and RBF mean estimates, ω_m $(m = 1, \ldots, M)$. These optimal parameter estimates are based upon the partitioning of the samples as determined by the minimization with respect to Q. This leads to the following constrained optimization problem. The constraints force Q to be a set of valid probability mass functions.

Problem 3

$$\begin{aligned} \textit{minimize:} \quad & \textstyle\sum_{j=1}^{N} \sum_{m=1}^{M} p(j) Q(m|j) \left[\frac{1}{\sigma_e^2} (y_j - \phi_m^T \mathbf{z}_j)^2 + \frac{1}{\sigma_a^2} \|\mathbf{z}_j - \omega_m\|^2 \right] \\ \textit{with respect to:} \quad & Q, \Theta \\ \textit{subject to:} \quad & \textstyle\sum_{m=1}^{M} Q(m|j) = 1, \ \textit{for } j = 1, \ldots, N \\ & Q(m|j) \geq 0 \end{aligned}$$

Problem 3 was derived assuming that one wanted to minimize the mean square error of the outputs of the network. The problem can also be derived from a *maximum likelihood* framework. In that approach, the network is assumed to be a stochastic generator of outputs whose parameters are tuned to maximize the likelihood of producing the outputs from a set of design samples that describes the function that is to be approximated. The maximum likelihood parameter estimates are then produced using an Expectation-Maximization (EM) [4] algorithm. This perspective makes a closer connection with the work in the neural community and other areas which make use of EM algorithms.

The performance measure in Problem 3 does not necessarily have to be restricted to the mean square output error. The results which follow in Sections 3 and 4 are derived using this measure. However, any suitable seminorm (as described in [13]) could be used to measure performance as the results in those sections depend upon the convexity of the measure and an efficient solver for minimizing the objective function with respect to the parameters of the function.

3 AM Training

Problem 3 can be rewritten to emphasize the dependence of the mean square error measure upon the two disjoint sets of parameters and to motivate the use of an AM. Let $\mathbf{x}, \mathbf{c}(\Theta) \in \mathbb{R}^n$ where

$$\mathbf{x} = \langle Q(1|1), \ldots, Q(1|N), Q(2|1), \ldots, Q(m|j), \ldots \rangle^T \tag{8}$$

and

$$\mathbf{c}(\Theta) = \langle p(1)d_1(1), \ldots, p(N)d_1(N), p(2)d_2(2), \ldots, p(j)d_m(j), \ldots \rangle^T \tag{9}$$

with $d_m(j) = \left[\frac{1}{\sigma_e^2}(y_j - \phi_m^T \mathbf{z}_j)^2 + \frac{1}{\sigma_a^2}||\mathbf{z}_j - \omega_m||^2\right]$, and $n = MN$. $\bar{d}(Q, \Theta, \mathcal{T})$ can be written as

$$\bar{d}(Q, \Theta, \mathcal{T}) = \mathbf{c}(\Theta)^T \mathbf{x} \tag{10}$$

which leads to the following revised form for Problem 3
Problem 4

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

where $A = [I_{N \times N} \ldots I_{N \times N}] \in \mathbb{R}^{N \times MN}$, $\mathbf{b} = \langle 1, 1, \ldots, 1 \rangle^T \in \mathbb{R}^N$, and $\mathbf{x} \geq 0$ implies that $x_i \geq 0$ for all $i = 1, \ldots, n$. The choice of $\mathbf{x}$ and $\mathbf{c}(\Theta)$ to denote the terms associated with Q and the network parameter vectors, respectively, in $\bar{d}(Q, \Theta, \mathcal{T})$ is made to maintain consistency with the notation used in standard linear programming problem definitions in [7, 14, 15]. The cost function consists of the inner product of two vectors, one which depends solely upon the training probabilities, Q in the form of $\mathbf{x}$, and one which depends solely upon the network parameters, $\mathbf{c}(\Theta)$. Problem 4 can be decomposed into two subproblems, one which is an LP problem in terms of $\mathbf{x}$ and the other which is a quadratic optimization in term of Θ. It is precisely this decomposition into two convex optimization problems that recommends the use of alternating minimizations.

3.1 Linear Interior Point methods

The minimization of $\mathbf{c}(\Theta)^T \mathbf{x}$ with respect to Q can be solved using standard LP solution techniques. The most widely accepted approach to solving LP problems is the Simplex Algorithm [14], but this method has poor theoretical computational cost in spite of its normally fast empirical performance. An alternative approach is the use of recently developed interior point techniques [7]. These are barrier methods which use logarithmic barrier functions to keep solutions in the feasible region of an LP problem. The methods follow a path in the interior of the polyhedral set of feasible solutions to the optimal solution which lies at a vertex of the set. The methods are fast, taking $O(\sqrt{n}L)$ iterations to converge with a computational cost of $O(n^{3.5}L)$ where n is the dimension of the LP problem and L is a measure of the accuracy of the final solution [7].

The IP method is a path following algorithm [7] which solves the following sequence of optimization problems based on a primal parameterization of Problem 4 when minimized solely with respect to $\mathbf{x}$

$$\begin{aligned} \text{minimize:} \quad & \alpha^{(k)}\mathbf{c}^T\mathbf{x} - \textstyle\sum_i \log x_i^{(k)} \\ \text{with respect to:} \quad & \mathbf{x} \\ \text{subject to:} \quad & A\mathbf{x} = \mathbf{b} \end{aligned} \tag{11}$$

where $\alpha^{(k)} \geq 0$ $(k = 1, \cdots, K)$ is a finite length, monotone increasing sequence of real numbers. $\mathbf{x}^*(\alpha^{(k)})$ denotes the solution for the k^{th} optimization problem in the sequence and is referred to as a *central point*. The locus of all points $\mathbf{x}^*(\alpha^{(k)})$ 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 point 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 solution in a controlled manner.

Path following (IP) methods solve the LP problem by approximately solving the sequence of augmented problems shown in (11). The parameter sequence, $\alpha^{(0)}, \alpha^{(1)}, \cdots, \alpha^{(K)}$, is chosen to be a monotone increasing sequence so that the central points, $\mathbf{x}^*(\alpha^{(k)})$, of each augmented optimization approach the LP solution in a monotone manner. It has been shown that for specific choices of the α sequence, that the sequence of approximate central points will converge to an ϵ-neighborhood of the LP solution after a finite number of iterations. For primal IP algorithms, the required condition is that successive approximations of the central points lie within the region of quadratic convergence

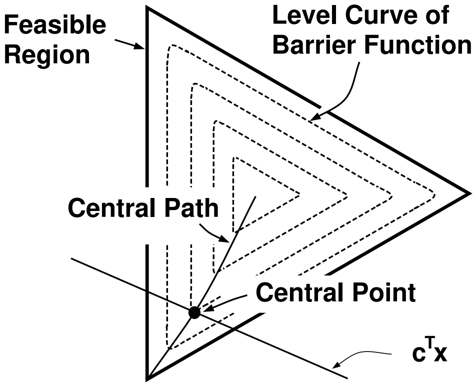

Figure 2: Interior Point Concepts

for a scaling steepest descent (SSD) algorithm [7]. In particular, it has been shown that if the k^{th} approximating solution is sufficiently close to the k^{th} central point and if $\alpha^{(k+1)} = \alpha^{(k)}(1+\nu/\sqrt{n})$ where $\nu \in (0, 0.1]$ controls the distance between successive central points, then the "closeness" to the $(k+1)^{st}$ central point is guaranteed and the resulting algorithm will converge in $O(\sqrt{n}L)$ iterations. If the algorithm takes small steps, then it is guaranteed to converge efficiently.

A brief description of a basic path following algorithm follows and a pictorial description appears in Figure 3. The IP method begins with an initial feasible solution, $\mathbf{x}^{(k)}$, for some $\alpha^{(k)}$ which is close to the k^{th} central point. $\alpha^{(k)}$ is increased as $\alpha^{(k+1)} = \alpha^{(k)}(1+\nu/\sqrt{n})$ to decrease the influence of the barrier functions, where $\nu \in \mathbb{R}$ and $\nu \in [0, 0.1]$ controls the distance to the next central point. This moves the $(k+1)^{st}$ central point closer to the boundaries of the feasible region. $\mathbf{x}^{(k+1)}$ is found using an SSD update [7]

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - XP_{AX}X(\alpha^{(k+1)}\mathbf{c} - (\mathbf{x}^{(k)})^{-1}) \tag{12}$$

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 matrix, and $\mathbf{x}^{-1} = \langle x_1^{-1}, x_2^{-1}, \ldots, x_n^{-1}\rangle^T$. This process is repeated until the k^{th} central point is moved arbitrarily close to the optimal solution at the boundaries of the set. During this process, increases to α must be kept small (via a small ν) to ensure that $\mathbf{x}^{(k)}$ is within the region of quadratic convergence for the SSD update with respect to $\mathbf{x}^{(k+1),*}$ [7].

Nearness of points to the central path plays an important role in standard IP algorithms, where best path following in terms of good computational cost and convergence rate is achieved when

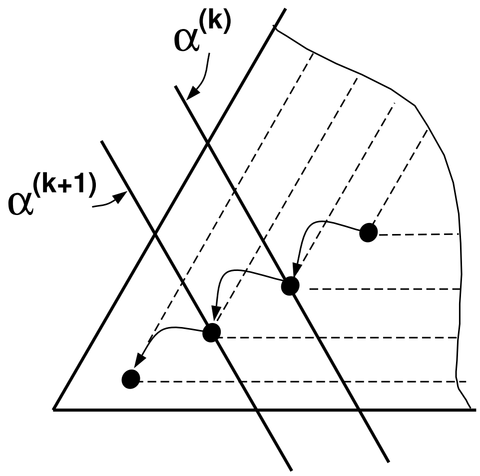

Figure 3: Path Following Example

solutions are close to the central path. The measure of closeness is the size of the scaled step taken in the SSD update, $\delta(\mathbf{x}, \alpha) = ||P_{AX}X(\alpha\mathbf{c} - (\mathbf{x}^{(k)})^{-1})||$. $\delta(\mathbf{x}, \alpha) < 1$ places $\mathbf{x}^{(k)}$ in the region of quadratic convergence for the SSD algorithm. Once a point is near the central path ($\delta(\mathbf{x}, \alpha) \leq 0.5$), the IP algorithm maintains the nearness of successive points through its choice of ν to update α. The process of ensuring that $\delta(\mathbf{x}, \alpha) \leq 0.5$ for all $k \geq 0$ is called *centralization.* When successive solutions are centralized and the distance between successive central points is small, as controlled via ν, the IP algorithm will converge in $O(\sqrt{n}L)$ iterations with an associated computational cost of $O(n^{3.5}L)$ [7].

The standard IP algorithm uses as a stopping criterion the distance between the current and optimal solutions. The value of the optimal solution, however, is usually unknown. Fortunately, mathematical programming offers the *duality gap* as a bound on the distance to an optimal solution[15]. The duality gap, $\Delta^{(k)}$, is the difference between the solution for the primal parameterization of the problem and the solution to its dual parameterization for the k^{th} linear optimization problem. In feasible linear problems, the optimal primal and dual solutions are equal implying that $\Delta^{(k)} \geq 0$. For IP methods, the duality gap can be expressed as [7]

$$\Delta^{(k)} = \frac{n + 0.5\sqrt{n}}{\alpha^{(k)}}. \tag{13}$$

3.2 Quadratic problem solver

The preceding IP method serves as the basis for determining successive values of the conditional training probabilities for the training problem and is termed the *Q-update*. Determination of successive values of the network parameters is accomplished by solving the alternate minimization of Problem 4, namely minimizing $\mathbf{c}(\Theta)^T\mathbf{x}$ with respect to Θ. The determination of successive values of the network parameters is termed the Θ*-update*.

Minimization of $\mathbf{c}(\Theta)^T\mathbf{x}$ with respect to Θ is a quadratic optimization problem and can be solved using standard Newton-Raphson (NR) techniques. The constrained optimization problem can be rewritten as

Problem 5

$$\begin{aligned} \textit{minimize:} &\quad \textstyle\sum_{m=1}^{M}\sum_{j=1}^{N} P(j,m)\left[\frac{1}{\sigma_e^2}(y_j - \phi_m^T\mathbf{z}_j)^2 + \frac{1}{\sigma_a^2}||\mathbf{z}_j - \omega_m||^2\right] \\ \textit{with respect to:} &\quad \Theta \\ \textit{subject to:} &\quad Q \ \textit{constant} \end{aligned}$$

where $\Theta = (\theta_1, \theta_2, \ldots, \theta_M)$ and $\theta_m = \langle \phi_m^T, \omega_m^T \rangle^T$. The minimizing solution to Problem 5 is $\theta_m^* = \langle \phi_m^{*T}, \omega_m^{*T} \rangle^T$ $(m = 1, \ldots, M)$ where

$$\phi_m^* = \left[\sum_{j=1}^{N} p(j)Q(m|j)\mathbf{z}_j\mathbf{z}_j^T\right]^{-1} \sum_{j=1}^{N} p(j)Q(m|j)\mathbf{z}_j y_j \tag{14}$$

and

$$\omega_m^* = \frac{\sum_{j=1}^{N} p(j)Q(m|j)\mathbf{z}_j}{\sum_{j=1}^{N} p(j)Q(m|j)}. \tag{15}$$

These minimizing solutions assume that that $\sum_{j=1}^{N} p(j)Q(m|j) \neq 0$ and $\left[\sum_{j=1}^{N} p(j)Q(m|j)\mathbf{z}_j\mathbf{z}_j^T\right]$ is nonsingular.

3.3 Hybrid IP AM algorithm

The hybrid IP algorithm used in training the network parameters combines the linear IP solver with the quadratic NR minimizer. The quadratic minimizer is embedded within the steps of a standard, small-step path following algorithm. Performing alternating minimizations of $\mathbf{c}(\Theta)^T\mathbf{x}$ with respect to Q and Θ, however, requires more than simply placing the NR minimizer in the steps of the

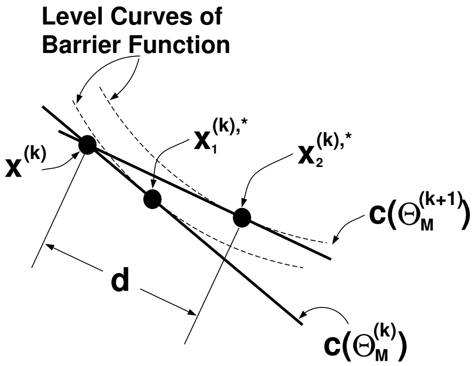

Figure 4: Effect of cost plane rotation

standard IP algorithm. The preceding discussion of the LP path following methods reveals that a key component to the path following method's computational efficiency lies in controlling the SSD iteration and the sequence $\{\alpha^{(k)}\}$ so that successive central points lie within the SSD algorithm's region of quadratic convergence. Any minimization of $\mathbf{c}(\Theta)^T\mathbf{x}$ with respect to Θ within the steps of an IP path following algorithm must maintain the "nearness" between successive solutions in order to retain the good computational properties of the original IP method. Minimizing $\mathbf{c}(\Theta)^T\mathbf{x}$ with respect to Θ within the steps of the IP algorithm will change the cost vector, $\mathbf{c}(\Theta)$. It is quite possible that the cost vector will be rotated in such a way that the current solution, $\mathbf{x}^{(k)}$, no longer lies in the region of quadratic convergence of the SSD algorithm with respect to the $(k+1)^{st}$ central point as shown in Figure 4. In order to ensure the nearness between successive solutions, it is necessary to be less "aggressive" in the update of the parameters. By constraining the parameter update to be the convex combination of the current and minimizing sets of parameter vectors as

$$\Theta^{(k+1)} = (1-\gamma^{(k)})\Theta^{(k)} + \gamma^{(k)}\Theta^{(k+1),*} \tag{16}$$

where $\gamma^{(k)} \in (0, \gamma_{bound}]$ (where γ_{bound} is the bound presented by Theorem 4 in Section 4), the nearness between successive points is maintained and the LP portion of the hybrid algorithm converges to a fixed point.

While constraining the updates of the parameters guarantees the nearness between successive LP solutions and the subsequent convergence of the LP portion of the hybrid IP algorithm to a fixed point, it does not guarantee the convergence of the parameter vectors to a fixed point.

Guaranteeing the convergence of the parameter estimates requires that $\gamma^{(k)} \to 1$ with increasing k. It will be shown in Section 4 that by controlling the distance between successive LP points, $\mathbf{x}^{(k)}$ and $\mathbf{x}^{(k+1)}$, through a bound on ν, the parameter estimates will converge to a fixed point. Constraining ν (as indicated by the bound on ν from Theorem 9 in Section 4) corresponds to being less "aggressive" in updating the LP points in the hybrid algorithm than is done in the standard IP algorithm. Thus, the hybrid algorithm requires that both the linear and quadratic portions of the algorithm be constrained to guarantee convergence. With these modifications to the standard IP algorithm, the modified IP algorithm for performing alternating minimizations appears below.

Algorithm 1 (AM Training Algorithm)

Initialize
  $k = 0$.
  Choose $x_i^{(k)}$, $\theta_m^{(k)}$, and $\alpha^{(k)}$ for $i = 1, \cdots, (MN)$, and for $m = 1, \cdots, M$.
repeat
  $\alpha^{(k+1)} = \alpha^{(k)}(1 + \nu/\sqrt{n})$, where $\nu \in (0, \nu_{bound}]$.
  Q-update:
    $\mathbf{x}_0 = \mathbf{x}^{(k)}$
    for $i = 0, 1$
      $\mathbf{x}_{i+1} = \mathbf{x}_i - X_i P_{AX_i} X(\alpha^{(k+1)}\mathbf{c}(\Theta^{(k)}) - \mathbf{x}_i^{-1})$
    $\mathbf{x}^{(k+1)} = \mathbf{x}_2$
  Θ-update: For $m = 1, \ldots, M$, where $\gamma^{(k)} \in (0, \gamma_{bound}]$

$$\theta_m^{(k+1)} = (1 - \gamma^{(k)})\theta_m^{(k)} + \gamma^{(k)}\theta_m^{(k+1),*}$$

  $k = k + 1$
until($\Delta^{(k)} < \epsilon$)

4 Theoretical Analysis

This section addresses the choices for the values of the parameters of the algorithm, the convergence of the component optimizations and the overall algorithm as well as the nature of the final solutions, and the computational properties of the hybrid IP algorithm.

4.1 Convergence properties

The hybrid IP algorithm combines a standard IP path-following algorithm with standard NR techniques. The hybrid algorithm produces a sequence of solutions, $(\mathbf{x}^{(k)}, \Theta^{(k)})$, $k \geq 0$. Convergence of the hybrid algorithm to an optimal solution requires that the component steps, the Q-update and the Θ-update, converge to fixed, optimal solutions, $(\mathbf{x}^*, \Theta^*)$. This section examines the selection of parameters that guarantees the convergence of the Q-update, Θ-update and of the hybrid algorithm to locally optimal solutions.

4.1.1 Q-update parameter selection

Convergence of the Q-update requires that the nearness between successive LP solutions in the hybrid algorithm is maintained. Nearness between successive LP solutions is controlled through proper application of the SSD steps in the Q-update and the controlled application of the Θ-update. Lemma 1 and its corollary describe how applications of the SSD step affect the nearness between LP solutions, while Lemma 3 provides a bound on $\|P_{AX}X\|$ from the SSD update. Theorem 4 provides a bound on $\gamma^{(k)}$ for use in the Θ-update to maintain the nearness of LP solutions after the Θ-update.

Lemma 1 (See [7]) *Let $\delta(\mathbf{x}^{(k)}, \alpha^{(k)}, \Theta) = \|P_{AX}X(\alpha^{(k)}\mathbf{c}(\Theta) - (\mathbf{x}^{(k)})^{-1})\|$ be the size of the SSD step in the scaled space where $\mathbf{x}^{-1} = \langle x_1^{-1}, x_2^{-1}, \ldots, x_n^{-1}\rangle^T$. If $\delta(\mathbf{x}^{(k)}, \alpha^{(k)}, \Theta) \leq 0.5$, $\alpha^{(k+1)} = \alpha^{(k)}(1 + \nu/\sqrt{n})$ where $\nu \in (0, 0.1]$, then one SSD step finds a new point, $(\mathbf{x}^{(k+1)}, \Theta)$, such that $\delta(\mathbf{x}^{(k+1)}, \alpha^{(k+1)}, \Theta) \leq 0.5$.*

Corollary 2 *Assuming the same conditions as in Lemma 1, a number, J, of SSD steps produces a point $(\mathbf{x}_J^{(k+1)}, \Theta)$ such that $\delta(\mathbf{x}_J^{(k+1)}, \alpha^{(k+1)}, \Theta) \leq \delta_1 = (0.7)^{2^J}$.*

Proof: From [7], if $\delta(\mathbf{x}^{(k)}, \alpha^{(k)}, \Theta) \leq 0.5$ and $\alpha^{(k+1)} = \alpha^{(k)}(1 + \nu/\sqrt{n})$ with $\nu \in (0, 0.1]$, then $\delta(\mathbf{x}^{(k)}, \alpha^{(k+1)}, \Theta) \leq 0.7$. When $\delta(\mathbf{x}^{(k)}, \alpha, \Theta) < 1$, $\mathbf{x}^{(k)}$ is in the SSD step's region of quadratic convergence with respect to $\mathbf{x}^*(\alpha)$ (Lemma 5.4 in [7]), so one SSD step produces $\mathbf{x}_1^{(k+1)}$ from $\mathbf{x}^{(k)}$ where

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

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

$$\delta(\mathbf{x}_2^{(k+1)}, \alpha^{(k+1)}, \Theta) \leq \delta(\mathbf{x}_1^{(k+1)}, \alpha^{(k+1)}, \Theta)^2 \leq (0.49)^2.$$

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

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

thus proving the corollary. □.

Lemma 3 *Let $\mathbf{c}^T\mathbf{x}$ be a cost function for an LP problem as described in Problem 4, and let the problem be solved using IP techniques as described in Section 3. Let $P_A = I - A^T(AA^T)^{-1}A \in \mathbb{R}^{n\times n}$ be an orthogonal projection matrix and let $X = diag(\mathbf{x})$ where $\mathbf{x} = \langle x_1, x_2, \cdots, x_n\rangle^T \in \mathbb{R}^n$. Let $\|P_{AX}X(\alpha\mathbf{c} - \mathbf{x}^{-1})\|$ be the size of the step in the scaled SSD space. Given these assumptions, then, for a feasible solution, $\mathbf{x}^{(k)}$, to the LP problem*

$$\|P_{AX^{(k)}}X^{(k)}\| \leq \Delta^{(k)} \tag{17}$$

where $\Delta^{(k)}$ is the duality gap associated with $\mathbf{x}^{(k)}$ and a feasible dual slack, $s^{(k)} \in \mathbb{R}^n$.

Proof: From [7], $P_{AX^{(k)}}X^{(k)}\mathbf{c} = P_{AX^{(k)}}X^{(k)}\mathbf{s}^{(k)}$. Let $X^{(k)}\mathbf{s}^{(k)} = X^{(k)}S^{(k)}e$ where $S^{(k)} = \text{diag}(\mathbf{s}^{(k)})$ and $e = \langle 1, 1, \ldots, 1\rangle^T \in \mathbb{R}^n$. Taking the norm of $X^{(k)}S^{(k)}e$

$$\begin{aligned} \|X^{(k)}S^{(k)}e\| &= (x_1s_1)^2 + (x_2s_x)^2 + \ldots + (x_ns_n)^2 \\ &\leq \left((x^{(k)})^Ts^{(k)}\right)^2 \end{aligned} \tag{18}$$

But from [7], $(\mathbf{x}^{(k)})^T\mathbf{s}^{(k)} = \Delta^{(k)}$. Since $\|P_{AX}\| \leq 1$ for all AX, this implies that

$$\|P_{AX^{(k)}}X^{(k)}s^{(k)}\| \leq \|P_{AX^{(k)}}\|\|X^{(k)}s^{(k)}\| \leq \|X^{(k)}s^{(k)}\| \leq \Delta^{(k)} \tag{19}$$

But $P_{AX^{(k)}}X^{(k)}\mathbf{c} = P_{AX^{(k)}}X^{(k)}\mathbf{s}^{(k)}$, so $\|P_{AX^{(k)}}X^{(k)}\mathbf{c}\|^2 \leq (\Delta^{(k)})^2$. The norm, $\|P_{AX^{(k)}}X^{(k)}\mathbf{c}\|^2 = \mathbf{c}^T(P_{AX^{(k)}}X^{(k)})^TP_{AX^{(k)}}X^{(k)}\mathbf{c}$, so

$$\|P_{AX^{(k)}}X^{(k)}\mathbf{c}\|^2 = \mathbf{c}^T(P_{AX^{(k)}}X^{(k)})^2\mathbf{c} \leq (\Delta^{(k)})^2 \tag{20}$$

Without loss of generality, let $\|\mathbf{c}\| = 1$. Then,

$$\frac{\mathbf{c}^T(P_{AX^{(k)}}X^{(k)})^2\mathbf{c}}{\mathbf{c}^T\mathbf{c}} \leq \frac{(\Delta^{(k)})^2}{\mathbf{c}^T\mathbf{c}} \tag{21}$$

Thus, $\|P_{AX^{(k)}}X^{(k)}\|^2 \leq (\Delta^{(k)})^2$ and $\|P_{AX^{(k)}}X^{(k)}\| \leq \Delta^{(k)}$. □.

Theorem 4 *Let $\Theta^{(k)}$ and $\Theta^{(k),*}$ be the current and minimizing parameter vectors at time k. Let $\mathbf{c}^{(k)} = \mathbf{c}(\Theta^{(k)})$ and $\mathbf{c}^{(k),*} = \mathbf{c}(\Theta^{(k),*})$ and assume $\mathbf{c}(\Theta)$ is a convex function of Θ. Let $\delta(\mathbf{x}, \alpha, \Theta) = \|P_{AX} X\left(\alpha \mathbf{c}(\Theta) - \mathbf{x}^{-1}\right)\|$ where $\mathbf{x}^{-1} = \langle x_1^{-1}, x_2^{-1}, \cdots, x_n^{-1}\rangle^T$. Assume $\delta(\mathbf{x}^{(k+1)}, \alpha^{(k+1)}, \Theta^{(k)}) = \delta_1 < 0.5$ and let $\Theta^{(k+1)} = (1 - \gamma^{(k)})\Theta^{(k)} + \gamma^{(k)}\Theta^{(k+1),*}$. If $\gamma^{(k)}$ is chosen as*

$$\gamma^{(k)} \leq \frac{\delta_2 - \delta_1}{2n\|\mathbf{c}^{(k+1),*} - \mathbf{c}^{(k)}\|} \tag{22}$$

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

Proof: The proof must demonstrate that using $\gamma^{(k)}$ in (22) maintains the nearness of $\mathbf{x}^{(k+1)}$ to the central path after updating Θ. Let $\mathbf{h}(\mathbf{x}, \alpha, \Theta) = P_{AX} X\left(\alpha \mathbf{c}(\Theta) - \mathbf{x}^{-1}\right)$ and define $\mathbf{h}_1$ and $\mathbf{h}_2$ as $\mathbf{h}_1 = \mathbf{h}(\mathbf{x}^{(k+1)}, \alpha^{(k+1)}, \Theta^{(k)})$ and $\mathbf{h}_2 = \mathbf{h}(\mathbf{x}^{(k+1)}, \alpha^{(k+1)}, \Theta^{(k+1)})$. Using the triangle inequality, $\mathbf{h}_1$ and $\mathbf{h}_2$ are related as

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

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

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

Applying the Θ-update and using the convexity assumption for $\mathbf{c}(\Theta)$ as $\mathbf{c}^{(k+1)} = (1 - \gamma^{(k)})\mathbf{c}^{(k)} + \gamma^{(k)}\mathbf{c}^{(k+1),*}$ results in

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

Using Lemma 3 $\|P_{AX^{(k)}} X^{(k)}\| \leq \Delta^{(k)} \leq 2n/\alpha^{(k)}$, then

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

Plugging in the value of γ from (22) yields

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

Simplifying produces the desired result that $\|\mathbf{h}_2\| \leq \delta_2 = 0.5$, thereby guaranteeing the nearness of $\mathbf{x}^{(k+1)}$ to the central path after updating Θ. $\square$

The following result demonstrates that one iteration of the hybrid IP algorithm maintains the "nearness" of successive training probabilities and network weights $(\mathbf{x}^{(k+1)}, \Theta^{(k+1)})$ to the central path.

Theorem 5 *Let* $\delta(\mathbf{x}^{(k)}, \alpha^{(k)}, \Theta^{(k)}) \leq 0.5$ *and let* $\alpha^{(k+1)} = \alpha^{(k)}(1 + \nu/\sqrt{n})$ *where* $\nu \in (0, 0.1]$*. Let* $\Theta^{(k+1)} = (1 - \gamma^{(k)})\Theta^{(k)} + \gamma^{(k)}\Theta^{(k+1),*}$ *where* $\gamma^{(k)}$ *is from (22). Then, one iteration of Algorithm 1 with two SSD steps produces* $(\mathbf{x}^{(k+1)}, \Theta^{(k+1)})$ *such that* $\delta(\mathbf{x}^{(k+1)}, \alpha^{(k+1)}, \Theta^{(k+1)}) \leq 0.5$*.*

Proof: By assumption, $\delta(\mathbf{x}^{(k)}, \alpha^{(k)}, \Theta^{(k)}) \leq 0.5$. From Corollary 2, two SSD steps produce $\mathbf{x}^{(k+1)}$ where $\delta(\mathbf{x}^{(k+1)}, \alpha^{(k+1)}, \Theta^{(k)}) \leq (0.7)^{2^2} = 0.2401$. A valid $\gamma^{(k)}$ for the Θ-update can be computed using (22) where $\delta_1 = 0.2401$ and $\delta_2 = 0.5$, and the Θ-update results in $\Theta^{(k+1)}$ such that

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

$\square$

4.1.2 Θ-update parameter selection

Proving the convergence of the Θ-update requires proving that $\gamma^{(k)} \to 1$ with increasing k. The analysis establishes a series of relations between $\gamma^{(k)}$ and $\|\Theta^{(k+1),*} - \Theta^{(k)}\|$, $\|\Theta^{(k+1),*} - \Theta^{(k)}\|$ and $\|\triangle Q\| = \|Q^{(k+1)}(m|j) - Q^{(k)}(m|j)\|$, and $\|\triangle Q\|$ and ν. The final result is a bound on ν which ensures that $\gamma^{(k)}$ will tend to unity with increasing k.

The parameter, $\gamma^{(k)}$, is defined by the bound from (22)

$$\gamma^{(k)} \leq \frac{\delta_2 - \delta_1}{2n\|\mathbf{c}^{(k+1),*} - \mathbf{c}^{(k)}\|}.$$

$\gamma^{(k)}$ will be defined to be equal to unity whenever $\|\mathbf{c}^{(k+1),*} - \mathbf{c}^{(k)}\| \leq (\delta_2 - \delta_1)/(2n)$. Since δ_2, δ_1, and n are constants, there is a largest $\|\mathbf{c}^{(k+1),*} - \mathbf{c}^{(k)}\|$ such that $\gamma^{(k)} = 1$. Since $\mathbf{c}^{(k)} = \mathbf{c}(\Theta^{(k)})$ is a continuous function of $\Theta^{(k)}$, there is a largest $\|\Theta^{(k+1),*} - \Theta^{(k)}\|$ such that $\gamma^{(k)} = 1$. Let ϵ_θ be the value of the largest $\|\Theta^{(k+1),*} - \Theta^{(k)}\|$ such that $\gamma^{(k)} = 1$ or $\|\mathbf{c}^{(k+1),*} - \mathbf{c}^{(k)}\| \leq (\delta_2 - \delta_1)/(2n)$.

Convergence of $\gamma^{(k)}$ to unity requires that $\|\Theta^{(k+1),*} - \Theta^{(k)}\| \leq \epsilon_\theta$ as k increases. Theorem 6 below specifies conditions on $\|\Theta^{(k+1),*} - \Theta^{(k),*}\|$ that guarantee that $\|\Theta^{(k+1),*} - \Theta^{(k)}\|$ will be a strictly decreasing sequence. Since the sequence will be strictly decreasing and bounded below by zero, $\|\Theta^{(k+1),*} - \Theta^{(k)}\| \rightarrow 0$ and as k increases, $\|\Theta^{(k+1),*} - \Theta^{(k)}\| \leq \epsilon_\theta$.

Theorem 6 *Let $\Theta^{(k)}$ and $\Theta^{(k),*}$ be the current and minimizing sets of parameter vectors at iteration k. Let $\gamma^{(k)} \in (0,1]$ be defined as in (22) and let the parameter vectors be updated as*

$$\Theta^{(k+1)} = (1-\gamma^{(k)})\Theta^{(k)} + \gamma^{(k)}\Theta^{(k+1),*}. \tag{23}$$

If

$$\|\Theta^{(k+2),*} - \Theta^{(k+1),*}\| < \gamma^{(k)}\|\Theta^{(k+1),*} - \Theta^{(k)}\|, \tag{24}$$

then

$$\|\Theta^{(k+2),*} - \Theta^{(k+1)}\| < \|\Theta^{(k+1),*} - \Theta^{(k)}\|. \tag{25}$$

Proof: What must be shown is that $\|\Theta^{(k+1),*} - \Theta^{(k)}\|$ is a strictly decreasing sequence, that is

$$\|\Theta^{(k+2),*} - \Theta^{(k+1)}\| < \|\Theta^{(k+1),*} - \Theta^{(k)}\|. \tag{26}$$

Begin by adding and subtracting $\Theta^{(k+1),*}$ to the left hand side of (26) and then break apart the norm

$$\|\Theta^{(k+2),*} - \Theta^{(k+1)}\| \leq \|\Theta^{(k+2),*} - \Theta^{(k+1),*}\| + \|\Theta^{(k+1),*} - \Theta^{(k+1)}\|. \tag{27}$$

Using the expression for $\Theta^{(k+1)}$ in (23)

$$\|\Theta^{(k+2),*} - \Theta^{(k+1)}\| \leq \|\Theta^{(k+2),*} - \Theta^{(k+1),*}\| + (1-\gamma^{(k)})\|\Theta^{(k+1),*} - \Theta^{(k)}\|. \tag{28}$$

Substituting (24) into (28)

$$\begin{aligned} \|\Theta^{(k+2),*} - \Theta^{(k+1)}\| &< \gamma^{(k)}\|\Theta^{(k+1),*} - \Theta^{(k)}\| + (1-\gamma^{(k)})\|\Theta^{(k+1),*} - \Theta^{(k)}\| \\ &< \|\Theta^{(k+1),*} - \Theta^{(k)}\|. \end{aligned} \tag{29}$$

Thus, when $\|\Theta^{(k+2),*} - \Theta^{(k+1),*}\| < \gamma^{(k)}\|\Theta^{(k+1),*} - \Theta^{(k)}\|$, then

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

and $\|\Theta^{(k+1),*} - \Theta^{(k)}\|$ is a strictly decreasing sequence bounded below by zero. Therefore, $\|\Theta^{(k+1),*} - \Theta^{(k)}\| \rightarrow 0$ with increasing k. □

Theorem 6 provides conditions on the distance between successive minimizing solutions to the Q problem for the convergence of the Θ-update. The next set of results derive bounds on ν that guarantee the conditions required by Theorem 6.

Lemma 7 *Let $Q^{(k)}$ and $\Delta^{(k)}$ be the set of probabilities and the duality gap at iteration k, respectively. Let $\mathbf{c}^{(k)} = \mathbf{c}(\Theta^{(k)})$ and*

$$\delta(\mathbf{x}^{(k)}, \alpha^{(k)}, \Theta^{(k)}) = \|P_{AX} X \left(\alpha^{(k)}\mathbf{c}^{(k)} - (\mathbf{x}^{(k)})^{-1}\right)\| \tag{30}$$

where $\mathbf{x}$ is the vector of $Q(m|j)$'s as defined above. Let $\alpha^{(k+1)} = \alpha^{(k)}(1+\nu/\sqrt{n})$ where $\nu \in (0, 0.1]$. Assume that $\mathbf{x}^{(k)}$ is near the central path, i.e. $\delta(\mathbf{x}^{(k)}, \alpha^{(k)}, \Theta^{(k)}) \leq 0.5$. For two SSD updates, the change in Q, $\|\triangle Q\| = \|Q^{(k+1)} - Q^{(k)}\|$ is

$$\|\triangle Q\| \leq 2(\delta(1+\nu) + \nu)\min(\Delta^{(k)}, 1). \tag{31}$$

Proof: Recall that the Q-update consists of two successive SSD steps

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{x}^{(k)} - X^{(k)} P_{AX^{(k)}} X^{(k)} (\alpha^{(k+1)}\mathbf{c}^{(k)} - (\mathbf{x}^{(k)})^{-1}) \\ \mathbf{x}^{(k+1)} &= \mathbf{x}_1 - X_1 P_{AX_1} X_1 (\alpha^{(k+1)}\mathbf{c}^{(k)} - \mathbf{x}_1^{-1}) \end{aligned} \tag{32}$$

where $X^{(k)} = \text{diag}(\mathbf{x}^{(k)}) \in \mathbb{R}^{MN \times MN}$ and $X_1 = \text{diag}(\mathbf{x}_1) \in \mathbb{R}^{MN \times MN}$. Let

$$\begin{aligned} \|\triangle Q\| &= \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| \\ &\leq \|\mathbf{x}^{(k+1)} - \mathbf{x}_1\| + \|\mathbf{x}_1 - \mathbf{x}^{(k)}\|. \end{aligned} \tag{33}$$

Examine the distance moved in the first update.

$$\|\mathbf{x}_1 - \mathbf{x}^{(k)}\| = \left\| X^{(k)} P_{AX^{(k)}} X^{(k)} (\alpha^{(k+1)}\mathbf{c}^{(k)} - (\mathbf{x}^{(k)})^{-1}) \right\| \tag{34}$$

Using the fact that $\alpha^{(k+1)} = \alpha^{(k)}(1+\nu/\sqrt{n})$ and $P_{AX} = (P_{AX})^2$ as P_{AX} is an orthogonal projection

$$\begin{aligned} \|\mathbf{x}_1 - \mathbf{x}^{(k)}\| &= \|X^{(k)} P_{AX^{(k)}} X^{(k)} (\alpha^{(k+1)}\mathbf{c}^{(k)} - (\mathbf{x}^{(k)})^{-1})\| \\ &\leq \|X^{(k)} (P_{AX^{(k)}})^2 X^{(k)} (\alpha^{(k)}\mathbf{c}^{(k)} - (\mathbf{x}^{(k)})^{-1}) + X^{(k)} (P_{AX^{(k)}})^2 X^{(k)} \alpha^{(k)} \nu \mathbf{c}^{(k)}/\sqrt{n}\| \\ &\leq \|X^{(k)} P_{AX^{(k)}}\| \|P_{AX^{(k)}} X^{(k)} (\alpha^{(k)}\mathbf{c}^{(k)} - (\mathbf{x}^{(k)})^{-1}) + P_{AX^{(k)}} X^{(k)} \alpha^{(k)} \nu \mathbf{c}^{(k)}/\sqrt{n}\| \end{aligned} \tag{35}$$

But $XP_{AX} = P_{AX}X$ and it can be shown (Lemma 3) that $\|P_{AX^{(k)}}X^{(k)}\| \leq \Delta^{(k)}$ and $\|P_{AX}X\| \leq \|P_{AX}\|\|X\| \leq 1$, so

$$\|\mathbf{x}_1 - \mathbf{x}^{(k)}\| \leq \|P_{AX^{(k)}}X^{(k)}(\alpha^{(k)}\mathbf{c}^{(k)} - (\mathbf{x}^{(k)})^{-1}) + \|P_{AX^{(k)}}X^{(k)}\alpha^{(k)}\nu\mathbf{c}^{(k)}/\sqrt{n}\| \min(\Delta^{(k)}, 1) \quad (36)$$

Now using $\delta(\mathbf{x}^{(k)}, \alpha^{(k)}, \Theta^{(k)}) = \|P_{AX^{(k)}}X^{(k)}(\alpha^{(k)}\mathbf{c}^{(k)} - (\mathbf{x}^{(k)})^{-1})\|$, and $(\alpha^{(k)}\nu/\sqrt{n})\|P_{AX^{(k)}}X^{(k)}\mathbf{c}^{(k)}\| \leq \nu(\delta+1)$ [7], then

$$\begin{aligned}
\|\mathbf{x}_1 - \mathbf{x}^{(k)}\| &\leq \left(\|P_{AX^{(k)}}X^{(k)}(\alpha^{(k)}\mathbf{c}^{(k)} - (\mathbf{x}^{(k)})^{-1})\| + \|(\alpha^{(k)}\nu/\sqrt{n})\|P_{AX^{(k)}}X^{(k)}\mathbf{c}^{(k)}\|\right) \\
&\quad \min(\Delta^{(k)}, 1) \\
&\leq (\delta + \nu(\delta+1)) \min(\Delta^{(k)}, 1) \\
&\leq (\delta(1+\nu) + \nu) \min(\Delta^{(k)}, 1)
\end{aligned} \quad (37)$$

Now examine the distance moved in the second step, using similar bounding arguments as above produces

$$\begin{aligned}
\|\mathbf{x}^{(k+1)} - \mathbf{x}_1\| &\leq \|X_1 P_{AX_1}\|\|P_{AX_1}X_1(\alpha^{(k+1)}\mathbf{c}^{(k)} - \mathbf{x}_1^{-1})\| \\
&\leq \|P_{AX_1}X_1(\alpha^{(k+1)}\mathbf{c}^{(k)} - \mathbf{x}_1^{-1})\| \min(\Delta^{(k)}, 1)
\end{aligned} \quad (38)$$

The term $\|P_{AX_1}X_1(\alpha^{(k+1)}\mathbf{c}^{(k)} - \mathbf{x}_1^{-1})\| = \delta(\mathbf{x}_1, \alpha^{(k+1)}, \Theta^{(k)})$. Using the result of Lemma 5.4 in [7], $\delta(\mathbf{w}^{(k+1)}, \alpha^{(k+1)}, \Theta^{(k)}) \leq \delta(\mathbf{w}^{(k)}, \alpha^{(k+1)}, \Theta^{(k)})^2$. Since $\delta(\mathbf{x}^{(k)}, \alpha^{(k+1)}, \Theta^{(k)}) = (\delta(1+\nu)+\nu)$, $\delta(\mathbf{x}_1, \alpha^{(k+1)}, \Theta^{(k)}) \leq (\delta(1+\nu)+\nu)^2$. Thus,

$$\begin{aligned}
\|\mathbf{x}^{(k+1)} - \mathbf{x}_1\| &\leq \delta(\mathbf{x}_1, \alpha^{(k+1)}, \Theta^{(k)}) \min(\Delta^{(k)}, 1) \\
&\leq (\delta(1+\nu)+\nu)^2 \min(\Delta^{(k)}, 1)
\end{aligned} \quad (39)$$

However, $(\delta(1+\nu)+\nu) < 1$, so $(\delta(1+\nu)+\nu)^2 < (\delta(1+\nu)+\nu)$ and

$$\begin{aligned}
\|\triangle Q\| &\leq \|\mathbf{x}^{(k+1)} - \mathbf{x}_1\| + \|\mathbf{x}_1 - \mathbf{x}^{(k)}\| \\
&\leq ((\delta(1+\nu)+\nu)^2 + (\delta(1+\nu)+\nu)) \min(\Delta^{(k)}, 1) \\
&\leq 2(\delta(1+\nu)+\nu) \min(\Delta^{(k)}, 1)
\end{aligned} \quad (40)$$

□

Lemma 8 establishes a bound on the distance between successive optimal parameter estimates, $\Theta^{(k+1),*}$ and $\Theta^{(k),*}$. The distance bound depends upon the smallest eigenvalue of the autocorrelation matrix associated with each model agent and on the probabilities associated with each model agent.

Lemma 8 *Let $Q(m|j) = Q^{(k)}(m|j)$ and $\triangle Q(m|j) = Q^{(k+1)}(m|j) - Q^{(k)}(m|j)$. Define $\theta^*_m = \langle(\phi^*_m)^T, (\omega^*_m)^T\rangle^T \in \mathbb{R}^{2R}$ where ϕ^*_m and ω^*_m are defined as in (14) and (15). Let Q_{CP} be the set of all Q's on or near the central path. Define*

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

where $A_m, E_m \in \mathbb{R}^{R\times R}$. Let

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

where $\mathbf{w}, \triangle\mathbf{w} \in \mathbb{R}^R$. Assume bounded inputs and outputs, $\|\mathbf{z}\| \leq \zeta$, $|y| \leq Y_{max}$. 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 $\beta_{max} = \sup_{Q\in Q_{CP}}(\sum_{j=1}^N p(j)Q(m|j))^{-1}$. Then,

$$\|\Theta^{(k+1),*} - \Theta^{(k),*}\| \leq M\mathcal{K}\|\triangle Q\| \tag{41}$$

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 $r_2 = (\sum_{j=1}^N p(j)\triangle Q(m|j))/(\sum_{j=1}^N p(j)Q(m|j)) < 1$.

Proof: Let $\Theta = \langle\theta_1^T, \theta_2^T, \ldots, \theta_M^T\rangle^T$, then

$$\|\Theta^{(k+1),*} - \Theta^{(k),*}\| \leq \sum_{m=1}^M \|\theta_m^{(k+1),*} - \theta_m^{(k),*}\|.$$

The terms in the summation can be further divided using $\theta_m = \langle\phi_m^T, \omega_m^T\rangle^T$ ($m = 1, \ldots, M$) as

$$\|\theta_m^{(k+1),*} - \theta_m^{(k),*}\| \leq \sum_{m=1}^M \|\phi_m^{(k+1),*} - \phi_m^{(k),*}\| + \|\omega_m^{(k+1),*} + \omega_m^{(k),*}\|. \tag{42}$$

Using the definitions for $Q^{(k)}(m|j)$, $Q^{(k+1)}(m|j)$, ϕ^*_m, and ω^*_m, one finds that

$$\phi_m^{(k+1),*} - \phi_m^{(k),*} = (A_m + E_m)^{-1}(\mathbf{w} + \triangle\mathbf{w}) - A_m^{-1}\mathbf{w}_m \tag{43}$$

$$\omega_m^{(k+1),*} - \omega_m^{(k),*} = \frac{\sum_{j=1}^N p(j)(Q(m|j) + \triangle Q(m|j))\mathbf{z}_j}{\sum_{j=1}^N p(j)(Q(m|j) + \triangle Q(m|j))} - \frac{\sum_{j=1}^N p(j)Q(m|j)\mathbf{z}_j}{\sum_{j=1}^N p(j)Q(m|j)} \tag{44}$$

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

$$\|\phi_m^{(k+1),*} - \phi_m^{(k),*}\| \leq \|(A_m + E_m)^{-1} - A_m^{-1}\|\ \|\mathbf{w}_m\| + \|(A_m + E_m)^{-1}\|\ \|\triangle\mathbf{w}_m\| \tag{45}$$

$$\begin{aligned}\|\omega_m^{(k+1),*} - \omega_m^{(k),*}\| &\le \left\| \left[\frac{\sum_{j=1}^N p(j)Q(m|j)\mathbf{z}_j}{\sum_{j=1}^N p(j)(Q(m|j) + \triangle Q(m|j))} - \frac{\sum_{j=1}^N p(j)Q(m|j)\mathbf{z}_j}{\sum_{j=1}^N p(j)Q(m|j)} \right] \right\| \\ &\quad + \left\| \frac{\sum_{j=1}^N p(j)\triangle Q(m|j)\mathbf{z}_j}{\sum_{j=1}^N p(j)(Q(m|j) + \triangle Q(m|j))} \right\| \end{aligned} \tag{46}$$

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

$$\|\phi_m^{(k+1),*} - \phi_m^{(k),*}\| \le \zeta Y_{max} \|(A_m + E_m)^{-1} - A_m^{-1}\| + \zeta Y_{max} \|\triangle Q\| \|(A_m + E_m)^{-1}\| \tag{47}$$

$$\begin{aligned}\|\omega_m^{(k+1),*} - \omega_m^{(k),*}\| &\le \zeta \left\| \left[\frac{1}{\sum_{j=1}^N p(j)(Q(m|j) + \triangle Q(m|j))} - \frac{1}{\sum_{j=1}^N p(j)Q(m|j)} \right] \right\| \\ &\quad + \frac{\zeta \|\triangle Q\|}{\| \sum_{j=1}^N p(j)(Q(m|j) + \triangle Q(m|j))\|} \end{aligned} \tag{48}$$

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

$$\|\phi_m^{(k+1),*} - \phi_m^{(k),*}\| \le \zeta Y_{max} \|(A_m + E_m)^{-1} - A_m^{-1}\| + \mu_{max} \zeta Y_{max} \|\triangle Q\| \tag{49}$$

$$\begin{aligned}\|\omega_m^{(k+1),*} - \omega_m^{(k),*}\| &\le \zeta \left\| \frac{1}{\sum_{j=1}^N p(j)(Q(m|j) + \triangle Q(m|j))} - \frac{1}{\sum_{j=1}^N p(j)Q(m|j)} \right\| \\ &\quad + \beta_{max} \zeta \|\triangle Q\| \end{aligned} \tag{50}$$

Using Theorem 2.3.4 from [16],

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

and

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

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

$$\|\phi_m^{(k+1),*} - \phi_m^{(k),*}\| \le \mu_{max}^2 X_{max}^3 Y_{max} \|\triangle Q\| / (1 - r_1) + \mu_{max} \zeta Y_{max} \|\triangle Q\| \tag{51}$$

$$\|\omega_m^{(k+1),*} - \omega_m^{(k),*}\| \le \beta_{max}^2 \zeta \|\triangle Q\| / (1 - r_2) + \beta_{max} \zeta \|\triangle Q\| \tag{52}$$

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

$$\|\Theta^{(k+1),*} - \Theta^{(k),*}\| \leq M\mathcal{K}\|\triangle Q\| \tag{53}$$

□

Lemmas 8 and 7 establish the relationship between $\|\Theta^{(k+1),*} - \Theta^{(k),*}\|$ and ν. The following theorem provides a bound on ν such that $\gamma^{(k)} \rightarrow 1$, and the Θ-update converges.

Theorem 9 *Let* $Q(m|j) = Q^{(k)}(m|j)$ *and* $\triangle Q(m|j) = Q^{(k+1)}(m|j) - Q^{(k)}(m|j)$. *Let* $\theta_m^* = \langle(\phi^*)^T, (\omega^*)^T\rangle^T$ *where*

$$\phi_m^* = \left[\sum_{j=1}^{N} p(j)Q(m|j)\mathbf{z}_j\mathbf{z}_j^T\right]^{-1} \sum_{j=1}^{N} p(j)Q(m|j)y_j\mathbf{z}_j \tag{54}$$

$$\omega_m^* = \frac{\sum_{j=1}^{N} p(j)Q(m|j)\mathbf{z}_j}{\sum_{j=1}^{N} p(j)Q(m|j)}. \tag{55}$$

Let Q_{CP} *be the set of all Q's on or near the central path. Define* A_m, E_m, $\mathbf{w}$, *and* $\triangle\mathbf{w}$ *as in Lemma 8. Assume that* $\|\mathbf{z}\| \leq \zeta$ *and* $|y_j| \leq Y_{max}$. *Assume* A_m *is full rank for all* $Q \in Q_{CP}$. *Let* μ_{max} *and* β_{max} *be defined as in Lemma 8. If*

$$\begin{aligned} \nu \quad \leq \quad & \min\{-1 + \sqrt{1 + r_1/(2\mu_{max}\zeta^2)}, \\ & -1 + \sqrt{1 + r_2/(2\beta_{max})}, \\ & -1 + \sqrt{1 + \gamma_{min}\epsilon_\theta/(2M\mathcal{K})}\} \end{aligned} \tag{56}$$

where $\gamma_{min} = (\delta_2 - \delta_1)/(2n\|\mathbf{c}^{(1),*} - \mathbf{c}^{(0)}\|)$ *is the bound from (22) for the first iteration of the hybrid IP algorithm, and* $\mathcal{K} = (\mu_{max}\zeta Y_{max}(1 + \mu_{max}\zeta^2/(1 - r_1))) + (\beta_{max}\zeta(1 + \beta_{max}/(1 - r_2)))$ *then the* $\|\Theta^{(k+2),*} - \Theta^{(k+1),*}\| \leq \gamma^{(k)}\|\Theta^{(k+1),*} - \Theta^{(k)}\|$ *and the* Θ*-update will converge.*

Proof: The bound comes from three conditions on the Θ-update. The first two are conditions resulting from bounding conditions required from [16]. Let $r_1 < 1$ and $r_2 < 1$ where r_1 and r_2 are arbitrary but E_m and $\sum_{j=1}^{N} p(j)\triangle Q(m|j)$ depend upon $\|\triangle Q\|$ and ν as

$$\|E_m\| \leq \zeta^2\|\triangle Q\| \leq 2(\delta(1 + \nu) + \nu)\min(\Delta^{(k)}, 1)\zeta^2 \tag{57}$$

$$\|\sum_{j=1}^{N} p(j)\triangle Q(m|j)\| \leq \|\triangle Q\| \leq 2(\delta(1+\nu)+\nu)\min(\Delta^{(k)},1) \tag{58}$$

Examine the conditions that $\|A_m^{-1}\|\|E_m\| \leq r_1 < 1$ and $\|\sum_{j=1}^{N} p(j)Q(m|j)\|^{-1}\|\sum_{j=1}^{N} p(j)\triangle Q(m|j)\| \leq r_2 < 1$ as

$$2(\delta(1+\nu)+\nu)\min(\Delta^{(k)},1)\mu_{max}\zeta^2 \leq r_1 \tag{59}$$

$$2(\delta(1+\nu)+\nu)\min(\Delta^{(k)},1)\beta_{max} \leq r_2. \tag{60}$$

δ can be chosen as a function of ν provided that $\delta \leq 0.5$. So let $\delta = \nu \leq 0.1$, then

$$\begin{aligned}\nu^2 + 2\nu &\leq r_1/(2\mu_{max}\zeta^2\min(\Delta^{(k)},1)) \\ \nu^2 + 2\nu &\leq r_2/(2\beta_{max}\min(\Delta^{(k)},1))\end{aligned} \tag{61}$$

Solving the quadratic equations for ν results in

$$\nu \leq -1 + \sqrt{1 + r_1/(2\zeta^2\mu_{max}\min(\Delta^{(k)},1))} \tag{62}$$

$$\nu \leq -1 + \sqrt{1 + r_2/(2\beta_{max}\min(\Delta^{(k)},1))} \tag{63}$$

The bounds on ν are smallest when $\min(\Delta^{(k)},1) = 1$ leading to

$$\nu \leq -1 + \sqrt{1 + r_1/(2\zeta^2\mu_{max})} \tag{64}$$

$$\nu \leq -1 + \sqrt{1 + r_2/(2\beta_{max})} \tag{65}$$

The third term in the bound results from the requirement that $\|\Theta^{(k+1),*} - \Theta^{(k)}\|$ be a strictly decreasing sequence so that $\gamma^{(k)} \rightarrow 1$. Begin by examining

$$\|\Theta^{(k+2),*} - \Theta^{(k+1),*}\| < \gamma^{(k)}\|\Theta^{(k+1),*} - \Theta^{(k)}\|. \tag{66}$$

This relationship must hold for all terms in the sequence. If ν can be chosen so that it holds for the smallest term in the sequence, then it will hold for all the terms in the sequence. If $\|\Theta^{(k+1),*} - \Theta^{(k)}\|$ is a strictly decreasing sequence, then $\gamma^{(k)}$ is a strictly increasing sequence. The smallest γ, then, will be the initial γ,

$$\gamma_{min} = \frac{\delta_2 - \delta_2}{2n\|\mathbf{c}^{(1),*} - \mathbf{c}^{(0)}\|}. \tag{67}$$

The term, ϵ_θ, is the largest $\|\Theta^{(k+1),*} - \Theta^{(k)}\|$ such that $\gamma^{(k)} = 1$. So if

$$\|\Theta^{(k+2),*} - \Theta^{(k+1),*}\| \leq \gamma_{min}\epsilon_\theta \tag{68}$$

then $\|\Theta^{(k+2),*} - \Theta^{(k+1)}\| < \|\Theta^{(k+1),*} - \Theta^{(k)}\|$ for all terms in the sequence. Setting the right-hand side of (68) to be less than $\gamma_{min}\epsilon_\theta$ and setting $\delta = \nu$ as above, then

$$\|\Theta^{(k+2),*} - \Theta^{(k+1),*}\| \leq 2(\nu^2 + 2\nu)\min(\Delta^{(k)}, 1)M\mathcal{K} < \gamma_{min}\epsilon_\theta. \tag{69}$$

Solving this quadratic equation results in

$$\nu < -1 + \sqrt{1 + \gamma_{min}\epsilon_\theta/(2M\mathcal{K}\min(\Delta^{(k)}, 1))} \tag{70}$$

and ν is smallest when $\min(\Delta^{(k)}, 1) = 1$, thus

$$\nu < -1 + \sqrt{1 + \gamma_{min}\epsilon_\theta/(2M\mathcal{K})} \tag{71}$$

Thus, when ν satisfies all three bounds, a valid ν exists such that $\|\Theta^{(k+1),*} - \Theta^{(k)}\| \to \epsilon_\theta$ and $\gamma^{(k)} \to 1$ and the Θ-update converges. $\square$

Theorem 9 provides a bound on ν that guarantees that the Θ-update will converge. The bound, however, is very conservative and practice has shown that $\nu = 0.1$ is usually satisfactory for convergence assuming appropriately scaled cost vectors.

4.1.3 Hybrid IP convergence

Sections 4.1.1 and 4.1.2 proved the convergence properties of the component optimizations. Now attention turns to proving the convergence of the hybrid IP algorithm and the nature of the solutions. Convergence of the algorithm to a fixed point is proved by showing that the sequence of solutions produced by the hybrid algorithm is a Cauchy sequence. The resulting fixed points are shown to be local minima by using analysis based upon the geometry of the problem.

The first result derives a closed form which can be used to bound the sequence $\|\Theta^{(k+1),*} - \Theta^{(k)}\|$. This bound will be used later to show that the sequence converges and to derive sufficient conditions on how large $\|\Theta^{(1),*} - \Theta^{(0)}\|$ can be to ensure convergence of the algorithm.

Lemma 10 *Let $\Delta^{(0)} = 1$ and let ν be chosen so that $\|\Theta^{(k+1),*} - \Theta^{(k),*}\| \leq \gamma_{min}\epsilon_\theta \min(\Delta^{(k)}, 1)$ for all $k \geq 0$. Let the Θ-update be*

$$\Theta^{(k+1)} = (1 - \gamma^{(k)})\Theta^{(k)} + \gamma^{(k)}\Theta^{(k+1),*}. \tag{72}$$

Let $\alpha^{(k+1)} = \alpha^{(k)}(1 + \nu/\sqrt{n})$. Define $\gamma_{min} = \gamma^{(0)} = (1 - \lambda)$ and define $A \in \mathbb{R}, A \geq 1$ such that $\|\Theta^{(1),*} - \Theta^{(0)}\| = A\epsilon_\theta$. Let $\beta = 1/(1 + \nu/\sqrt{n})$, then

$$\|\Theta^{(k+1),*} - \Theta^{(k)}\| \leq A\epsilon_\theta\lambda^k + \gamma_{min}\epsilon_\theta\lambda^{k-1}\frac{1 - (\beta/\lambda)^k}{1 - (\beta/\lambda)} \tag{73}$$

Proof: Assume $\|\Theta^{(1),*} - \Theta^{(0)}\| = A\epsilon_\theta$ and examine $\|\Theta^{(k+1)} - \Theta^{(k)}\|$ for $k = 0$

$$\begin{aligned} \|\Theta^{(2),*} - \Theta^{(1)}\| &< (1 - \gamma^{(0)})\|\Theta^{(1),*} - \Theta^{(0)}\| + \gamma_{min}\epsilon_\theta\Delta^{(0)} \\ &< (1 - \gamma^{(0)})A\epsilon_\theta + \gamma_{min}\epsilon_\theta \end{aligned} \tag{74}$$

But $\gamma^{(0)} = \gamma_{min} = (1 - \lambda)$ so

$$\|\Theta^{(2),*} - \Theta^{(1)}\| < \lambda A\epsilon_\theta + \gamma_{min}\epsilon_\theta \tag{75}$$

Examine $\|\Theta^{(k+1)} - \Theta^{(k)}\|$ for $k = 1$.

$$\|\Theta^{(3),*} - \Theta^{(2)}\| < (1 - \gamma^{(1)}\|\Theta^{(2),*} - \Theta^{(1)}\| + \gamma_{min}\epsilon_\theta\beta \tag{76}$$

But $\gamma^{(1)} > \gamma^{(0)}$ so $(1 - \gamma^{(1)}) < (1 - \gamma^{(0)}) = \lambda$ so

$$\|\Theta^{(3),*} - \Theta^{(2)}\| < \lambda^2 A\epsilon_\theta + \lambda\gamma_{min}\epsilon_\theta + \gamma_{min}\epsilon_\theta\beta \tag{77}$$

The norm for any k is

$$\|\Theta^{(k+1),*} - \Theta^{(k)}\| < \lambda^k A\epsilon_\theta + \gamma_{min}\epsilon_\theta(\lambda^{k-1} + \lambda^{k-2}\beta + \cdots + \lambda\beta^{k-2} + \beta^{k-1}) \tag{78}$$

But $\sum_{i=0}^{k-1} \lambda^{k-1+i}\beta^i = \lambda^{k-1}\frac{1-(\beta/\lambda)^k}{1-(\beta/\lambda)}$ so

$$\|\Theta^{(k+1),*} - \Theta^{(k)}\| < \lambda^k A\epsilon_\theta + \gamma_{min}\epsilon_\theta\lambda^{k-1}\frac{1 - (\beta/\lambda)^k}{1 - (\beta/\lambda)} \tag{79}$$

□

The next two results show that the sequence generated by the hybrid IP algorithm converges to a fixed point, $\mathbf{w}^* = \langle(\mathbf{x}^*)^T, (\Theta^*)^T\rangle^T$ and that the fixed point is actually a locally minimum solution to Problem 4.

Theorem 11 *Let* $\mathbf{w} = \langle \mathbf{x}^T, \Theta^T \rangle^T \in \mathbb{R}^{M(N+R)}$ *be a solution to Problem 4. Let the sequence of iterates from the hybrid IP algorithm be* $\mathbf{w}^{(k)} = \langle (\mathbf{x}^{(k)})^T, (\Theta^{(k)})^T \rangle^T$ $(k \geq 0)$. *Assume that* $\mathbf{x}^{(0)}$ *is chosen so that* $\delta(\mathbf{x}^{(0)}, \alpha^{(0)}, \Theta^{(0)}) \leq 0.5$ *and* $\Delta^{(0)} = 1$. *Let* $\alpha^{(k+1)} = \alpha^{(k)}(1 + \nu/\sqrt{n})$ *where* $\nu \in (0, \nu_{bound}]$ *where* ν_{bound} *is the bound in (56), and let* $\Theta^{(k+1)} = (1 - \gamma^{(k)})\Theta^{(k)} + \gamma^{(k)}\Theta^{(k+1),*}$ *where* $\gamma^{(k)}$ *is from (22). Then, the sequence* $\mathbf{w}^{(k)}$ *converges to within an* ϵ *neighborhood of a fixed point* $\mathbf{w}^* = \langle (\mathbf{x}^*)^T, (\Theta^*)^T \rangle^T$.

Proof: The proof of convergence shows that $\|\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)}\|$ is a Cauchy sequence that converges to a fixed point. Examine the following sum

$$\begin{aligned} \textstyle\sum_{k=\bar{N}}^{\infty} \|\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)}\| &= \|\textstyle\sum_{k=\bar{N}}^{\infty} \langle (\mathbf{x}^{(k+1)})^T, (\Theta^{(k+1)})^T \rangle^T - \langle (\mathbf{x}^{(k)})^T, (\Theta^{(k)})^T \rangle^T \| \\ &\leq \textstyle\sum_{k=\bar{N}}^{\infty} \|\langle (\mathbf{x}^{(k+1)})^T, (\Theta^{(k+1)})^T \rangle^T - \langle (\mathbf{x}^{(k)})^T, (\Theta^{(k)})^T \rangle^T \| \end{aligned} \tag{80}$$

The sum can be rewritten as

$$\sum_{k=\bar{N}}^{\infty} \|\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)}\| \leq \sum_{k=\bar{N}}^{\infty} \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| + \sum_{k=\bar{N}}^{\infty} \|\Theta^{(k+1)} - \Theta^{(k)}\| \tag{81}$$

The first term can be expressed as $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| = \|\triangle Q\|$. But from (31), $\|\triangle Q\| \leq 2(\delta(1+\nu) + \nu)\min(\Delta^{(k)}, 1)$ and $\Delta^{(k)} \leq \Delta^{(0)}\beta^k$ where $\beta = 1/(1 + \nu/\sqrt{n}) < 1$. The first term in (81) can be bounded as

$$\begin{aligned} \textstyle\sum_{k=\bar{N}}^{\infty} \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| &\leq \textstyle\sum_{k=\bar{N}}^{\infty} 2(\delta(1+\nu) + \nu)\Delta^{(0)}\beta^k \\ &\leq 2(\delta(1+\nu) + \nu)\Delta^{(0)}\frac{\beta^{\bar{N}+1}}{1-\beta} \end{aligned} \tag{82}$$

The second term of (81) can be rewritten as $\|\Theta^{(k+1)} - \Theta^{(k)}\| = \gamma^{(k)}\|\Theta^{(k+1),*} - \Theta^{(k)}\|$, and since $\gamma^{(k)} \leq 1$, then $\|\Theta^{(k+1)} - \Theta^{(k)}\| \leq \|\Theta^{(k+1),*} - \Theta^{(k)}\|$. Using the expression for $\|\Theta^{(k+1),*} - \Theta^{(k)}\|$ from (73), $\|\Theta^{(k+1)} - \Theta^{(k)}\|$ can be bounded as

$$\begin{aligned} \textstyle\sum_{k=\bar{N}}^{\infty} \|\Theta^{(k+1)} - \Theta^{(k)}\| &\leq \textstyle\sum_{k=\bar{N}}^{\infty} \|\Theta^{(k+1),*} - \Theta^{(k)}\| \\ &\leq \textstyle\sum_{k=\bar{N}}^{\infty} \lambda^k A\epsilon_\theta + \gamma_{min}\epsilon_\theta\lambda^{k-1}\frac{1-(\beta/\lambda)^k}{1-(\beta/\lambda)} \\ &\leq A\epsilon_\theta\frac{\lambda^{\bar{N}+1}}{1-\lambda} + \gamma_{min}\epsilon_\theta\frac{\lambda^{\bar{N}+1}}{\lambda-\beta} - \gamma_{min}\epsilon_\theta\frac{\beta^{\bar{N}+1}}{\lambda-\beta} \end{aligned} \tag{83}$$

where $A\epsilon_\theta = \|\Theta^{(1),*} - \Theta^{(0)}\|$ and $\gamma_{min} = \gamma^{(0)} = (1-\lambda)$. Now, for any $\epsilon > 0$, $\bar{N}$ can be chosen large enough so that $2(\delta(1+\nu)+\nu)\Delta^{(0)}\frac{\beta^{\bar{N}+1}}{1-\beta} < \epsilon/2$ and $A\epsilon_\theta\frac{\lambda^{\bar{N}+1}}{1-\lambda} + \gamma_{min}\epsilon_\theta\frac{\lambda^{\bar{N}+1}}{\lambda-\beta} - \gamma_{min}\epsilon_\theta\frac{\beta^{\bar{N}+1}}{\lambda-\beta} < \epsilon/2$ or

$$\sum_{k=\bar{N}}^{\infty} \|\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)}\| < \epsilon/2 + \epsilon/2 < \epsilon \tag{84}$$

Since $\sum_{k=\bar{N}}^{\infty} \|\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)}\| < \epsilon$ for $k = \bar{N}$ to ∞, then for all $\bar{m}, \bar{n} > \bar{N}$, $\|\mathbf{w}^{(\bar{m})} - \mathbf{w}^{(\bar{n})}\| < \epsilon$ and the sequence is Cauchy. Finally, the sequence lies in $\mathbb{R}^{M(N+R)}$, which is a Banach space. Therefore, $\mathbf{w}^{(k)}$ converges to a within an ϵ-neighborhood of a fixed point $\mathbf{w}^* = \langle (\mathbf{x}^*)^T, (\Theta^*)^T \rangle^T$. □

Theorem 12 *Let $\mathbf{w}^* = \langle (\mathbf{x}^*)^T, (\Theta^*)^T \rangle^T$ be the limit of the sequence $\mathbf{w}^{(k)} = \langle (\mathbf{x}^{(k)})^T, (\Theta^{(k)})^T \rangle^T$ produced by the hybrid IP algorithm. Let the minimization problem be*

Problem 6

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

Then, $\mathbf{w}^* = \langle (\mathbf{x}^*)^T, (\Theta^*)^T \rangle^T$ is a locally minimum solution to Problem 6.

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

$$f(\mathbf{w}^* + \lambda\triangle\mathbf{w}) < f(\mathbf{w}^*) \tag{85}$$

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

$$f(\mathbf{w}^* + \lambda\triangle\mathbf{w}) < f(\mathbf{w}^*) \tag{86}$$

for $\sigma \in [0, \lambda)$ where $f(\mathbf{w})$ is the cost function. Let W be the set of all feasible directions at $\mathbf{w}^*$. Let $\triangle\mathbf{w} \in W$ where $\triangle\mathbf{w} = \langle \triangle\mathbf{x}^T, \triangle\Theta^T \rangle^T$ and let $\mathbf{w}_1 = \mathbf{w}^* + \lambda\triangle\mathbf{w}$. Two mutually exclusive cases can occur: $\mathbf{x}^*$ can be a unique optimal solution to Problem 6 when Θ is constant, or $\mathbf{x}^*$ can be a non-unique optimal solution. The first case occurs when $\mathbf{c}(\Theta)$ supports the polyhedral set of feasible solutions only at the vertex, $\mathbf{x}^*$. The second occurs when $\mathbf{c}(\Theta)$ supports the set at vertex, $\mathbf{x}$ and is also parallel to one of the faces of the polyhedral set which also contains $\mathbf{x}^*$.

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

   $\mathbf{x}^*$ being optimal implies that for all feasible $\triangle\mathbf{x}$ from $\mathbf{x}^*$ [15]

$$\mathbf{c}(\Theta^*)^T \triangle\mathbf{x} > 0. \tag{87}$$

Moving in direction $\triangle\mathbf{w} = \langle\triangle\mathbf{x}^T, \triangle\Theta^T\rangle^T$ also changes Θ^* as $\Theta_1 = \Theta^* + \lambda\triangle\Theta$. This rotates the cost vector as $\mathbf{c}(\Theta_1) = \mathbf{c}(\Theta^*) + \lambda\hat{\mathbf{c}}(\triangle\Theta)$ where $\hat{\mathbf{c}}(\triangle\Theta)$ is a term that denotes the changes in the cost vector due to $\triangle\Theta$. If λ is small enough, then $\lambda\hat{\mathbf{c}}(\triangle\Theta)^T\triangle\mathbf{x} < \mathbf{c}(\Theta^*)^T\triangle\mathbf{x}$ and $\mathbf{c}(\Theta_1)^T\triangle\mathbf{x} > 0$ for all feasible $\triangle\mathbf{x}$ from $\mathbf{x}^*$. Thus, $\mathbf{x}^*$ is still the optimal solution to the LP problem associated with Problem 6. If $\mathbf{x}^*$ is the optimal solution, then

$$\mathbf{c}(\Theta_1)^T\mathbf{x}^* < \mathbf{c}(\Theta_1)^T\mathbf{x}_1. \tag{88}$$

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

$$\mathbf{c}(\Theta^*)^T\mathbf{x}^* < \mathbf{c}(\Theta_1)^T\mathbf{x}^* < \mathbf{c}(\Theta_1)^T\mathbf{x}_1 \tag{89}$$

For feasible direction $\triangle\mathbf{w} = \langle\triangle\mathbf{x}^T, \triangle\Theta^T\rangle^T$, the cost of the solution $\mathbf{w}^* = \langle(\mathbf{x}^*)^T, (\Theta^*)^T\rangle^T$ is less than the cost of the solution $\mathbf{w}_1 = \langle\mathbf{x}_1^T, \Theta_1^T\rangle^T$. Since $\triangle\mathbf{w}$ was any arbitrary feasible direction from $\mathbf{w}^*$, there is no feasible direction that satisfies (85) and (86), which is a contradiction. Thus the assumption is incorrect and $\mathbf{w}^* = \langle(\mathbf{x}^*)^T, (\Theta^*)^T\rangle^T$ is a local minimum.

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

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

$$\mathbf{c}(\Theta^*)^T\triangle\mathbf{x} \geq 0 \tag{90}$$

   This can be broken down into two cases, those for which $\mathbf{c}(\Theta^*)^T\triangle\mathbf{x} > 0$ and those for which $\mathbf{c}(\Theta^*)^T\triangle\mathbf{x} = 0$. 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 $\mathbf{c}(\Theta^*)^T\triangle\mathbf{x} > 0$.

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

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

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

$$\lambda\mathbf{c}(\triangle\Theta)^T\triangle\mathbf{x}_1 > 0 \tag{91}$$

which implies that

$$\mathbf{c}(\Theta_1)^T\triangle\mathbf{x}_1 = \mathbf{c}(\Theta^*)^T\triangle\mathbf{x}_1 + \lambda\hat{\mathbf{c}}(\triangle\Theta)^T\triangle\mathbf{x}_1 > 0 \tag{92}$$

With this small rotation of the cost plane, $\mathbf{x}^*$ is the unique optimal solution to the LP problem. Thus, $\mathbf{c}(\Theta_1)^T\mathbf{x}^* < \mathbf{c}(\Theta_1)^T\mathbf{x}_1$. Since Θ^* is the optimal quadratic minimizer for Problem 6 subject to constant $\mathbf{x} = \mathbf{x}^*$, then $\mathbf{c}(\Theta^*)^T\mathbf{x}^* < \mathbf{c}(\Theta_1)^T\mathbf{x}^* < \mathbf{c}(\Theta_1)^T\mathbf{x}_1$. This again leads to a contradiction which implies that the solution, $\mathbf{w}^*$ is a local minimum.

□

The last result in this section presents conditions on $\|\Theta^{(k+1),*} - \Theta^{(k),*}\|$ that guarantee that the algorithm will converge. This is a proximity condition similar to that required by the Newton-Raphson iteration for convergence to a local minimum. This is a sufficient condition that guarantees convergence. The proximity condition is not restrictive, however, as the cost vector in the LP formulation can be scaled by a constant factor to ensure that this condition is always met. Thus, the algorithm will converge for any problem assuming an appropriately scaled cost vector.

Theorem 13 *Let $\Delta^{(0)} = 1$ and let ν be chosen so that $\|\Theta^{(k+1),*} - \Theta^{(k),*}\| \leq \gamma_{min}\epsilon_\theta \min(\Delta^{(k)}, 1)$ for all $k \geq 0$. Let the Θ-update be*

$$\Theta^{(k+1)} = (1-\gamma^{(k)})\Theta^{(k)} + \gamma^{(k)}\Theta^{(k+1),*}. \tag{93}$$

Let $\alpha^{(k+1)} = \alpha^{(k)}(1+\nu/\sqrt{n})$. *Define* $\gamma_{min} = \gamma^{(0)} = (1-\lambda)$ *and define* $A \in \mathbb{R}, A \geq 1$ *such that* $\|\Theta^{(1),*} - \Theta^{(0)}\| = A\epsilon_\theta$. *Let* $\beta = 1/(1+\nu/\sqrt{n})$. *If*

$$A \leq \lambda^{-K}\left(1 - \gamma_{min}\lambda^{K-1}\frac{1-(\beta/\lambda)^K}{1-(\beta/\lambda)}\right) \tag{94}$$

then $\gamma^{(k)} \to 1$ *in* K *steps or less.*

Proof: In order for $\gamma^{(k)} = 1$, $\|\Theta^{(k+1),*} - \Theta^{(k)}\| \leq \epsilon_\theta$. From Lemma 10, the representation for the k^{th} term is

$$\|\Theta^{(k+1),*} - \Theta^{(k)}\| \leq A\epsilon_\theta\lambda^k + \gamma_{min}\epsilon_\theta\lambda^{k-1}\frac{1-(\beta/\lambda)^k}{1-(\beta/\lambda)} \tag{95}$$

Substitute (94) for A, then

$$\begin{aligned} \|\Theta^{(k+1),*} - \Theta^{(k)}\| &< \lambda^{-k}(1 - \gamma_{min}\lambda^{k-1}\tfrac{1-(\beta/\lambda)^k}{1-(\beta/\lambda)})\epsilon_\theta\lambda^k + \gamma_{min}\lambda^{k-1}\tfrac{1-(\beta/\lambda)^k}{1-(\beta/\lambda)} \\ &< \epsilon_\theta - \gamma_{min}\lambda^{k-1}\tfrac{1-(\beta/\lambda)^k}{1-(\beta/\lambda)} + \gamma_{min}\lambda^{k-1}\tfrac{1-(\beta/\lambda)^k}{1-(\beta/\lambda)} \\ &< \epsilon_\theta \end{aligned} \tag{96}$$

Thus, $\|\Theta^{(k+1),*} - \Theta^{(k)}\| < \epsilon$ and $\gamma^{(k)} = 1$. □

4.2 Complexity

The following results characterize the proposed algorithm's complexity. Complexity results are stated in terms of L, which again is some measure of the accuracy of the final solution. Establishing convergence and complexity properties of the algorithm uses the duality gap as a bound on the distance between the current solution and the optimal solution. For the LP problems examined here, the duality gap can be expressed as [7]

$$\Delta(\alpha) = \frac{n + \xi\sqrt{n}}{\alpha}. \tag{97}$$

where $\xi = 0.5$ The following lemma states the order of the number of iterations required to terminate the algorithm.

Lemma 14 (See Lemma 4.8 in [7]) *Let* $\Delta^{(0)} = \Delta(\alpha^{(0)}) < 2^L$. *The number of iterations required such that* $\Delta^{(k)} \leq 2^{-L}$ *is* $O(\sqrt{n}L)$.

The overall complexity of the algorithm is computed by determining the computational complexity of the individual steps and then multiplying by the complexity of the number of iterations of the algorithm. The Q-update executes two SSD steps each employing a matrix inversion with a worst case complexity of $O(n^3)$ where $n = MN$. The Θ-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 Θ-update is done on M models so the Θ-update has a computational complexity of $O(nR^2)$ or $O(MR^3)$. These two complexity estimates multiplied by the estimate of the number of iterations establish the following theorem:

Theorem 15 *The computational cost of the hybrid interior point algorithm is* $\max(O(n^{3.5}L), O(n^{1.5}R^2L), O(\sqrt{n}MR^3L))$.

5 Examples

A suite of experiments examines the theoretical and computational properties of the algorithm and its ability to identify parameters for linear regressors to form local linear approximations to a nonlinear function. The experiments use the output of the following nonlinear dynamical system for the test signal in a time series prediction problem

$$\begin{aligned} \mathbf{z}[n+1] &= A[n]\mathbf{z}[n] + \mathbf{b}u[n] \\ y[n] &= \mathbf{c}[n]^T \mathbf{z}[n] \end{aligned} \tag{98}$$

where

$$A[n] = \begin{bmatrix} 2e^{-(\alpha_0+\alpha)T}\cos(2\pi fT) & -e^{-2(\alpha_0+\alpha)T} \\ 1 & 0 \end{bmatrix}$$
$$\mathbf{b} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} c_1 e^{-(\alpha_0+\alpha)T}\sin(2\pi fT) \\ 0 \end{bmatrix} \tag{99}$$

with $\alpha_0 = 1$, $f = 5$, $T = 0.01$ seconds, $c_1 = 0.3249$, and $\alpha = 10|y[n]|$. This corresponds to a structured perturbation of a linear system with impulse response

$$h(nT) = e^{-\alpha nT} \sin(2\pi f nT) \tag{100}$$

where the system gets faster (α increases) as the magnitude of the output increases. The specific signal used in the experiments, which appears in Figure 5, is the output of the dynamical system in response to

$$u[n] = 10\sin(\pi nT) + 10\sin(3\pi nT) + 10\sin(5\pi nT). \tag{101}$$

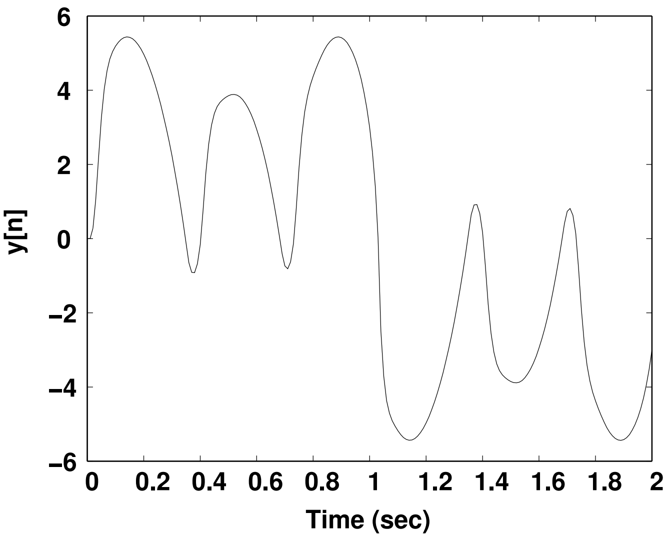

Figure 5: Test signal

The goal of the time series prediction problem is to generate a prediction, $\hat{y}[n]$ of a function, $y[n]$, that has a small mean square error, $(\hat{y}[n] - y[n])^2$, using a window of R past outputs, $\mathbf{z}[n] = \langle y[n-1], y[n-2], \ldots, y[n-R]\rangle^T$. A network of linear regressors as discussed in Section 2 is assumed available to do the prediction. The m^{th} regressor is an FIR filter parameterized by the vector $\phi_m \in \mathbb{R}^R$ and produces an output as $y_m[n] = \phi_m^T \mathbf{z}[n]$. The problem assumes a training set, $\mathcal{T} = \{(y_i, \mathbf{z}_i)\}$, of 100 random samples of the output of (98) over the range, $nT \in [0, 2]$ seconds. The samples are used in conjunction with the hybrid IP algorithm to identify the regressor parameter vectors. The problems below assume an additional 100 randomly sampled data points for use in testing. Both soft and hard approximation schemes, as described in Section 2, are used.

The first set of results examines parameter identification for the signal with no noise. In this case, a window of past outputs of size $R = 8$ is used. This choice was made as an IIR filter of order 8 with appropriate initial conditions can accurately reproduce the response of the linear dynamical system to the input in (101). Thus, the regressor size is chosen to match the size of the optimal IIR filter. Figure 6 displays the convergence rate of the hybrid IP algorithm for $M = 2$ to 9 models. The empirical results (open circles) agree very well with the predicted theoretical convergence rate of $O(\sqrt{n}L)$ iterations (solid line). Figure 7 depicts the computational cost of the algorithm versus the number of models. In this case, the empirical cost (open circles) agrees more closely with a computational cost of $O(n^2L)$ floating point operations (FLOPS) (dashed line) than with the predicted theoretical rate of $O(n^{3.5}L)$ FLOPS (solid line). This discrepancy is due to the facts that sparse matrix techniques were used to implement the SSD update and that R was small relative to n, which means that the cost of the scaling steepest descent algorithm dominates. Figure 8 shows the mean square error of the resulting networks of linear regressors versus the number of models for both soft and hard approximating networks. The plots demonstrate a trend that the mean square error first decreases until a minimum is found for $M = 8$ models and then begins to increase. The initial decrease is most likely attributable to mismodelling due to an insufficient number of models, while the increase at the end is due to overfitting the training data.

The second set of results examines parameter identification for the signal with zero mean Gaussian measurement noise with a variance of $\sigma^2 = 0.0625$ added to the signal. In this case, a window of $R = 7$ past outputs is used which corresponds to the optimal regressor size for this signal and noise level. Figure 9 displays the convergence rate of the hybrid IP algorithm for $M = 2$ to 9 models. As the number of training sample points was the same as for the previous example ($N = 100$), the number of iterations to converge per model was identical. This demonstrates that the number of iterations to converge depends only on $n = MN$ where N is the size of the training set and M is the number of models. Figure 10 depicts the computational cost of the algorithm versus the number of models. Again, the empirical cost is better than the predicted, although in this case,

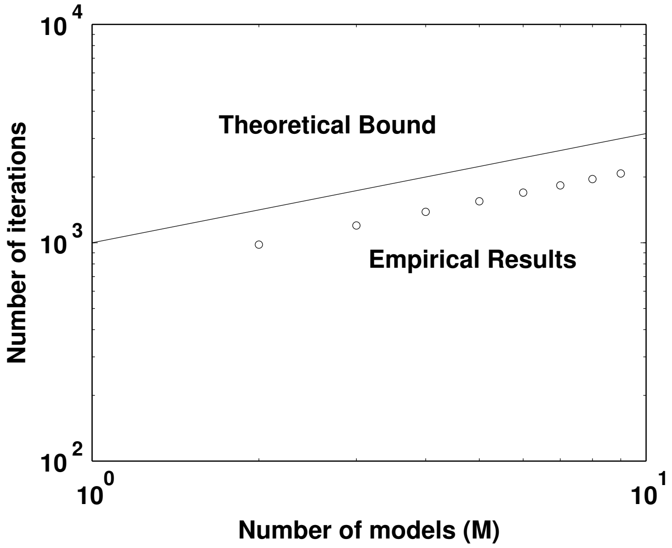

Figure 6: Convergence rate example 1

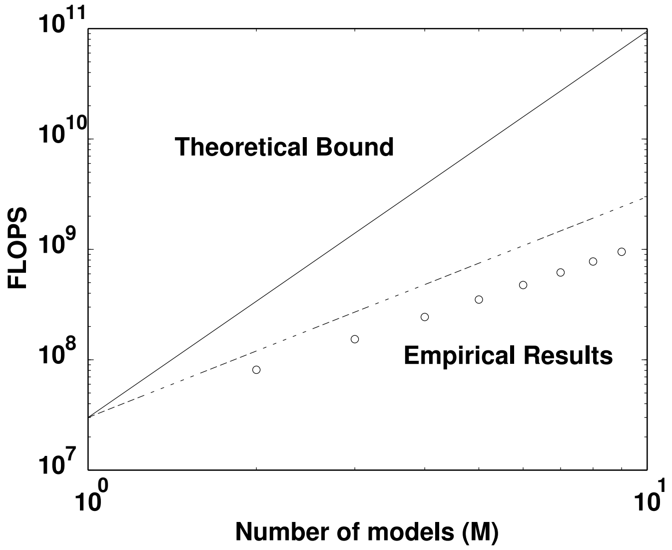

Figure 7: Computational cost example 1

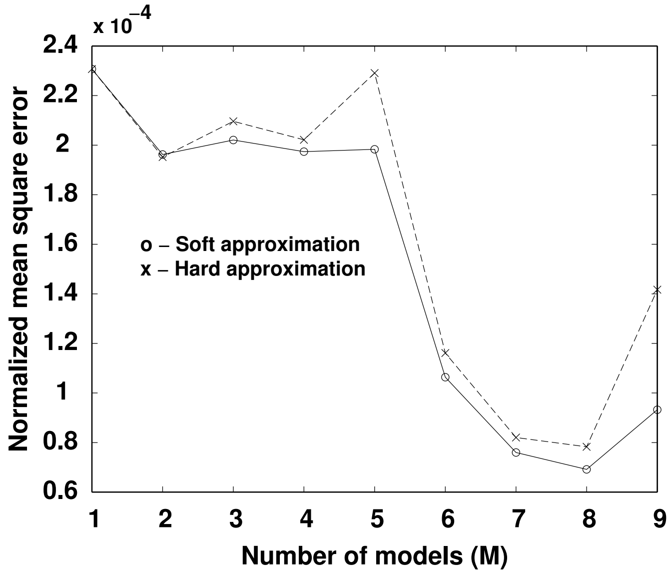

Figure 8: Mean square error example 1

the number of FLOPS is slightly less than that of the first example. This is due to the fact that at each iteration matrix inversions of size 7 were done instead of size 8 as for the first case. Figure 11 shows the mean square error of the resulting networks of linear regressors versus the number of models. Again, the mean square error initially decreases to a minimum at $M = 4$ models and then has a general increasing trend with larger numbers of models.

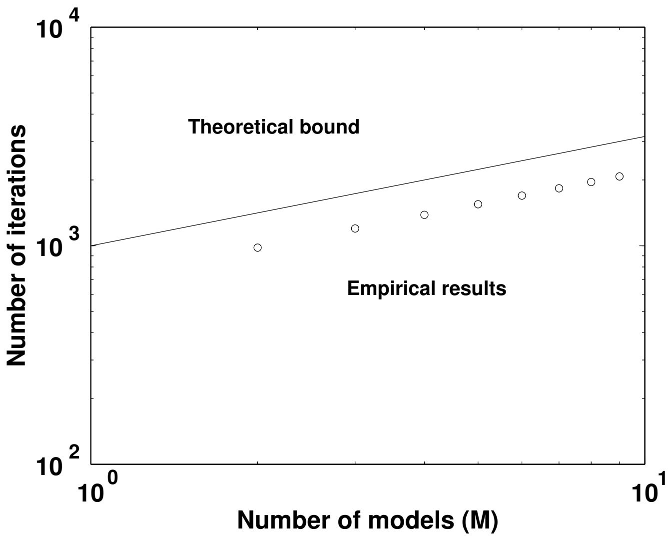

Figure 9: Convergence rate example 2

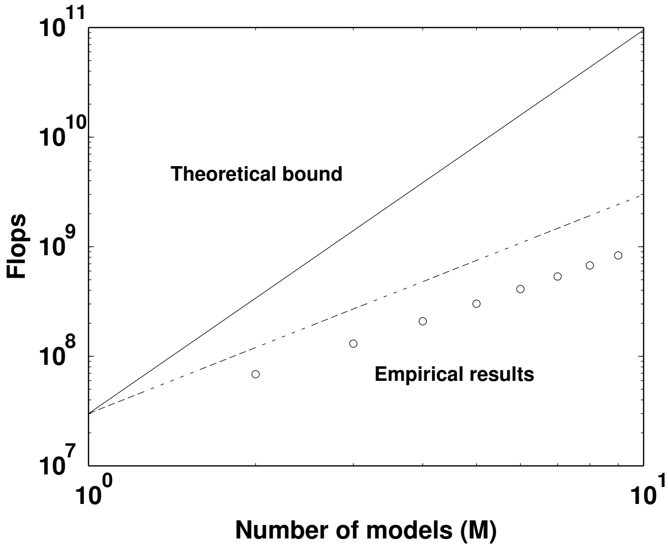

Figure 10: Computational cost example 2

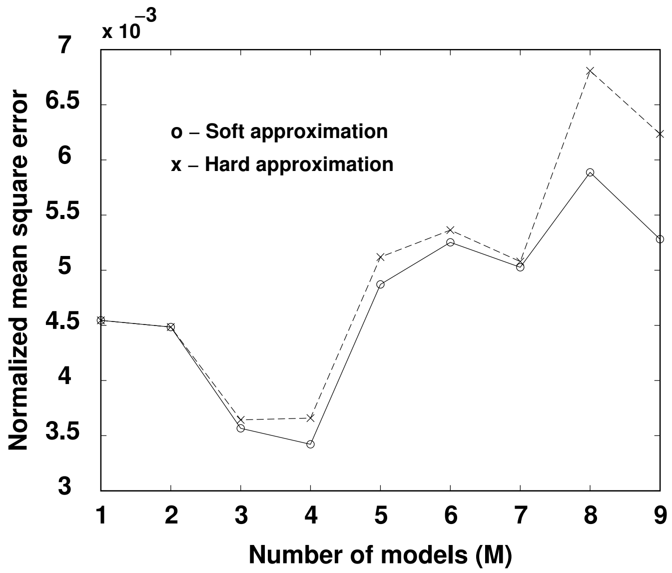

Figure 11: Mean square error example 2

6 Summary

The modified IP algorithm and its analysis offer insights into the efficient use of the generalized EM algorithm. The basic generalized EM algorithm performs "complete" maximizations of the expected log likelihood function with respect to the parameters in each iteration. The analysis in this paper demonstrates that by being less aggressive in the parameter updates, by taking controlled steps in the update of the conditional training probabilities via α and ν and by constraining the update of the parameters via $\gamma^{(k)}$, the modified IP version of the EM algorithm can be shown to have a polynomial convergence rate and computational cost.

The polynomial nature of the computational properties of the algorithm extend to the number of models and training data points. The theoretical results were expressed in terms of the LP subproblem dimension, n. n, however, is the product of the number of models and the size of the training data set, $n = MN$. The convergence rate and computational cost are thus also polynomial in both M and N, implying that the algorithm scales polynomially in both M and N. Thus, the algorithm can be used effectively on problems with large numbers of models and/or large numbers of training samples.

The use of the IP algorithm is not limited to the identification of parameters for sets of linear regressors. The form of the cost function, $\bar{d}(Q, \Theta, \mathcal{T})$, may be modified to address different problems. For example, by removing the dependence of $\bar{d}$ on the square output error, $(y_j - \phi_m^T \mathbf{z}_j)^2$, the algorithm can be used to efficiently determine parameter estimates for Bayesian classifiers in pattern classification [17] or detection tasks [18]. Other tasks might be addressed by replacing the square output error in $\bar{d}(Q, \Theta, \mathcal{T})$ with other suitable convex error measures.

The AM algorithm provides an efficient tool for determining neural and fuzzy network parameters. In neural networks, radial basis functions serve as a classification mechanism to partition the input space. Such a use of RBF's appears in [2, 19]. Membership functions in fuzzy networks can also be specified as Gaussian functions [20]. The AM algorithm can be used to place the RBF's and membership functions' means in an efficient manner using training data. Since the algorithm has a polynomial complexity, it can be used for large scale neural or fuzzy networks.

References

[1] S. J. Nowlan, "Maximum likelihood competitive learning," in *Advances in Neural Information Processing Systems 2*, pp. 574–582, San Mateo, California: Morgan Kaufmann Publishers, Inc., 1990.

[2] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, pp. 79–87, 1991.

[3] W. Byrne, "Alternating minimization and Boltzmann machine learning," *IEEE Transactions on Neural Networks*, vol. 3, pp. 612–620, July 1992.

[4] A. Dempster, N. Laird, and D.B.Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 1977.

[5] R. E. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE Transactions on Information Theory*, vol. IT-18, pp. 460–473, July 1972.

[6] P. T. Szymanski and M. D. Lemmon, "Adaptive mixtures of local experts are source coding solutions," in *Proceedings of the International Conference on Neural Networks*, vol. 3, (San Francisco, California), pp. 1391–1396, 1993.

[7] C. C. Gonzaga, "Path-following methods for linear programming," *SIAM Review*, vol. 34, pp. 167–224, June 1992.

[8] M. Lemmon, P. Szymanski, and C. Bett, "Interior point competitive learning of control agents in colony-style systems." To appear in *Proceedings of the SPIE.*

[9] M. Lemmon, C. Bett, P. Szymanski, and P. Antsaklis, "Constructing hybrid control systems from robust linear control agents." To appear in *Proceedings of the 1994 Cornell Workshop.*

[10] M. Lemmon and P. T. Szymanski, "Interior point implementations of alternating minimization training." To appear in *Advances in Neural Information Processing Systems 7*, 1994.

[11] P. T. Szymanski and M. Lemmon, "Interior point methods in the design of hierarchical controllers," in *Proceedings of the 9th International Symposium on Intelligent Control*, (Columbus, Ohio), pp. 33–38, Aug. 1994.

[12] P. T. Szymanski and M. Lemmon, "A modified interior point method for supervisory controller design," in *Proceedings of the 33rd IEEE Conference on Decision and Control*, (Orlando, Florida), pp. 1381–1386, Dec. 1994.

[13] R. M. Gray, *Source Coding Theory.* Boston, Massachusetts: Kluwer Academic Publishers, 1990.

[14] S.-C. Fang and S. Puthenpura, *Linear Optimization and Extensions: Theory and Algorithms.* Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1993.

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

[16] G. H. Golub and C. F. Van Loan, *Matrix Computations.* Baltimore, Maryland: The Johns Hopkins University Press, 1989.

[17] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* San Diego, California: Academic Press, Inc., second ed., 1990.

[18] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association.* Boston: Academic Press, Inc., 1988.

[19] J. Moody and C. Darken, "Learning with localized receptive fields," in *Proceedings of the 1988 Connectionist Models Summer School*, (San Mateo, California), pp. 133–143, 1988.

[20] L.-X. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning," *IEEE Transactions on Neural Networks*, vol. 3, pp. 807–814, Sept. 1992.