Breaking the gridlock in Mixture-of-Experts: Consistent and Efficient Algorithms

Ashok Vardhan Makkuva [1] Sewoong Oh [2] Sreeram Kannan [3] Pramod Viswanath [1]

Abstract

Mixture-of-Experts (MoE) is a widely popular model for ensemble learning and is a basic building block of highly successful modern neural networks as well as a component in Gated Recurrent Units (GRU) and Attention networks. However, present algorithms for learning MoE, including the EM algorithm and gradient descent, are known to get stuck in local optima. From a theoretical viewpoint, finding an efficient and provably consistent algorithm to learn the parameters remains a long standing open problem for more than two decades. In this paper, we introduce the first algorithm that learns the true parameters of a MoE model for a wide class of non-linearities with global consistency guarantees. While existing algorithms jointly or iteratively estimate the expert parameters and the gating parameters in the MoE, we propose a novel algorithm that breaks the deadlock and can directly estimate the expert parameters by sensing its echo in a carefully designed cross-moment tensor between the inputs and the output. Once the experts are known, the recovery of gating parameters still requires an EM algorithm; however, we show that the EM algorithm for this simplified problem, unlike the joint EM algorithm, converges to the true parameters. We empirically validate our algorithm on both the synthetic and real data sets in a variety of settings, and show superior performance to standard baselines.

1. Introduction

In this paper, we study a popular gated neural network architecture known as Mixture-of-Experts (MoE). MoE is a basic building block of highly successful modern neural networks like Gated Recurrent Units (GRU) and Attention networks. A key interesting feature of MoE is the presence of a gating mechanism that allows for specialization of experts in their respective domains. MoE allows for the underlying expert models to be simple while allowing to capture complex non-linear relations between the data. Ever since their inception more than two decades ago (Jacobs et al., 1991), they have been a subject of great research interest (Tresp, 2001; Collobert et al., 2002; Ng & Deisenroth, 2014; Theis & Bethge, 2015; Le et al., 2016; Gross et al., 2017; Sun et al., 2017; Wang et al., 2018) across multiple domains such as computer vision, natural language processing, speech recognition, finance, and forecasting.

The basic MoE model is the following: let $\boldsymbol{x} \in \mathbb{R}^d$ be the input feature vector and $y \in \mathbb{R}$ be the corresponding label. Then the discriminative model $P_{y|\boldsymbol{x}}$ for the k-mixture of experts (k-MoE) in the regression setting is:

$$\begin{aligned} P_{y|\boldsymbol{x}} &= \sum_{i=1}^{k} P_{i|\boldsymbol{x}} P_{y|\boldsymbol{x},i} \\ &= \sum_{i=1}^{k} \frac{e^{\langle \boldsymbol{w}_i^*, \boldsymbol{x} \rangle}}{\sum_j e^{\langle \boldsymbol{w}_j^*, \boldsymbol{x} \rangle}} \mathcal{N}(y | g(\langle \boldsymbol{a}_i^*, \boldsymbol{x} \rangle), \sigma^2). \qquad (1) \end{aligned}$$

Figure 1 details the architecture for k-MoE.

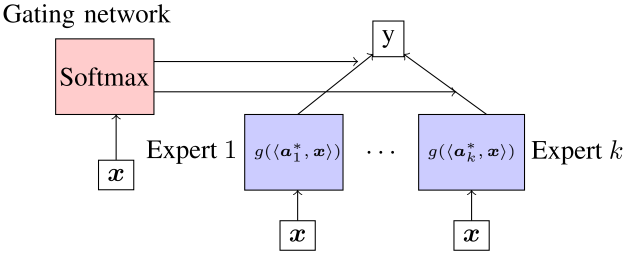

Gating network
Softmax
y
Expert 1
$g(\langle \boldsymbol{a}_1^*, \boldsymbol{x} \rangle)$
$\cdots$
$g(\langle \boldsymbol{a}_k^*, \boldsymbol{x} \rangle)$
Expert k
$\boldsymbol{x}$
$\boldsymbol{x}$
$\boldsymbol{x}$

Figure 1: Architecture for k-MoE

The interpretation behind (1) is that for each input $\boldsymbol{x}$, the gating network chooses an expert based on the outcome

[1] Department of Electrical and Computer Engineering, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, IL, USA [2] Allen School of Computer Science & Engineering, University of Washington, Seattle, USA [3] Department of Electrical Engineering, University of Washington, Seattle, USA. Correspondence to: Ashok Vardhan Makkuva <makkuva2@illinois.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

of a multinomial random variable $z \in [k]$, whose probability depends on $\boldsymbol{x}$ in a parametric way, i.e. $z|\boldsymbol{x} \sim \mathrm{softmax}(\langle \boldsymbol{w}_1^*, \boldsymbol{x} \rangle, \ldots, \langle \boldsymbol{w}_k^*, \boldsymbol{x} \rangle)$. The chosen expert then generates the output y from a Gaussian distribution centred at a non-linear activation of $\boldsymbol{x}$, i.e. $g(\langle \boldsymbol{a}_z^*, \boldsymbol{x} \rangle)$, with variance σ^2. We want to learn the expert parameters $\boldsymbol{a}_i^* \in \mathbb{R}^d$ (also referred to as the regressors) and the gating parameters $\boldsymbol{w}_i^* \in \mathbb{R}^d$, assuming we know the non-linear activation $g : \mathbb{R} \to \mathbb{R}$.

This problem of learning MoE has been a long standing open problem for more than two decades, even though it is a fundamental building block of several state-of-the-art gated neural network architectures. Gated neural networks such as GRUs and Sparsely-gated-MoEs have been widely successful in challenging tasks like machine translation (Chung et al., 2014; Shazeer et al., 2017; Vaswani et al., 2017). Parameters are typically learnt through (stochastic) gradient descent on a non-convex loss function. However, these methods do not possess any theoretical guarantees, even for the simplest gated neural network, which is the MoE.

On the other hand, existing guarantees for simpler models without gating units do not extend to MoEs. Consider the mixture of generalized linear models (M-GLMs) (Sedghi et al., 2014; Sun et al., 2014; Yi et al., 2016; Zhong et al., 2016), which is a strict simplification of the k-MoE model in (1), where $\boldsymbol{w}_i^* = 0$ for all $i \in \{1, \ldots, k\}$. The learning in M-GLMs is usually done through a combination of spectral methods and greedy methods such as EM. A major limitation of these methods is that they rely critically on the fact that the mixing probability is a constant and hence they do not generalize to MoEs (see Section 2). In addition, the EM algorithm, which is the workhorse for learning in parametric mixture models, is prone to bad local minima (Sedghi et al., 2014; Balakrishnan et al., 2017; Zhong et al., 2016) (we independently verify this for MoEs in Section 4). These theoretical shortcomings and practical relevance of the MoE models lead to the following fundamental question:

Can we find an efficient and a consistent algorithm (with global initializations) that recovers the true parameters of the model with theoretical guarantees?

In this paper, we address this question precisely and make the following contributions:

1) First theoretical guarantees: We provide the first (poly-time) efficient algorithm that recovers the true parameters of a MoE model with global initializations (Theorem 1 and Theorem 2). We allow for a wide class of non-linearities which includes the popular choices of identity, sigmoid, and ReLU. To the best of our knowledge, ours is the first work to give global convergence guarantees for MoE.

2) Algorithmic innovations: Existing algorithms jointly or iteratively estimate the expert parameters and the gating paramters in the MoE and can get stuck in local minima. In this paper, we propose a novel algorithm that breaks the gridlock and can directly estimate the expert parameters by sensing its echo in a cross-moment tensor between the inputs and the output (Algorithm 1 and Algorithm 2). Once the experts are known, the recovery of gating parameters still requires an EM algorithm; however, we show that the EM algorithm for this simplified problem, unlike the joint EM algorithm, converges to the true parameters. The proofs of global convergence of EM as well as the design of the cross-moment tensor are of independent mathematical interest.

3) Novel transformations: In this paper, we introduce the novel notion of "Cubic and Quadratic Transform (CQT)". These are polynomial transformations on the output labels tailored to specific non-linear activation functions and the noise variance. The key utility of these transforms is to equip MoEs with a supersymmetric tensor structure in a principled way (Theorem 1).

Related work. While there is a huge literature on MoEs ((Yuksel et al., 2012; Masoudnia & Ebrahimpour, 2014) are detailed surveys), there are relatively few works on its learning guarantees. (Jordan & Xu, 1995) is the first work to analyze the local convergence of joint-EM for both the gating and the expert parameters. As noted earlier, however, EM is prone to bad local minima. In contrast, our algorithms have *global convergence* guarantees. It is important to note that even for the simpler problem of mixtures of Gaussians, it is known that EM gets stuck in local minima, whenever number of mixtures, k, is at least 3 (Jin et al., 2016), whereas we can handle $2k - 1 < d$ with global convergence.

The simplified versions of MoE, M-GLMs, are widely studied in the literature. The key techiques for parameter inference in M-GLMs include EM algorithm, spectral methods, convex relaxations, and their variants. (Yi et al., 2014; Balakrishnan et al., 2017) prove convergence of EM for 2-mixtures of linear regressions; in contrast, we handle $k \geq 2$ mixtures for a wide class of non-linearities and provide global convergence. (Sedghi et al., 2014) construct a 3^{rd}-order supersymmetric tensor containing the regressors as its rank-1 components. However, this approach fails to generalize for MoE. (Zhong et al., 2016) use a similar tensor construction followed by EM to learn the parameters; however, they can only handle linear noiseless mixtures and no gating parameters. In contrast, our algorithms can handle non-linearities and the gating parameters. (Chen et al., 2014) use a convex objective to learn the regressors for a special setting of 2-mixtures of linear regressions. Similar to earlier approaches, this relaxation too does not generalize to $k > 2$.

Notation. In this paper, we denote Euclidean vectors by bold face lowercase letters $\boldsymbol{a}, \boldsymbol{b}$, etc., and scalars by plain lowercase letters y, z, etc. We use $\mathcal{N}(y|\mu, \sigma^2)$ either to denote the density or the distribution of a Gaussian random

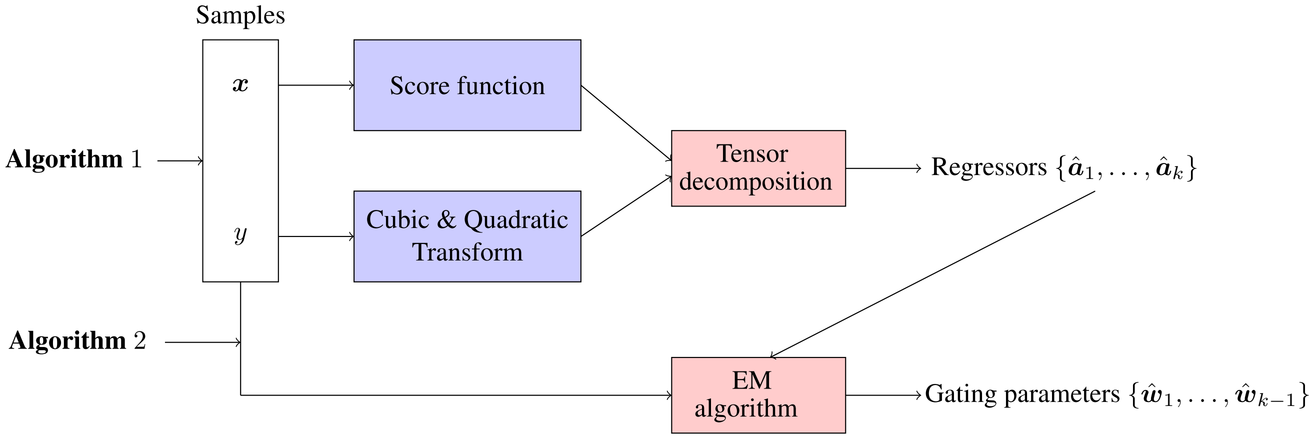

Samples
x
y
Algorithm 1
Algorithm 2
Score function
Cubic & Quadratic Transform
Tensor decomposition
Regressors $\{\hat{a}_1, \ldots, \hat{a}_k\}$
EM algorithm
Gating parameters $\{\hat{w}_1, \ldots, \hat{w}_{k-1}\}$

Figure 2: Algorithm to learn the MoE parameters. **Algorithm** 1: First we take non-linear transformations on the samples $(\boldsymbol{x}_i, y_i)$ to compute the tensors $\mathcal{T}_2, \mathcal{T}_3$. Spectral decomposition on $\mathcal{T}_2, \mathcal{T}_3$ recovers the regressors. **Algorithm** 2: EM uses the learnt regressors and samples to learn the gating parameters with random initializations

variable y with mean μ and variance σ^2, depending on the context. $[d] \triangleq \{1, \ldots, d\}$. $\mathrm{Perm}[d]$ denotes the set of all permutations on $[d]$. We use $\otimes$ to denote the tensor outer product of vectors in $\mathbb{R}^d$. $\boldsymbol{x}^{\otimes 3}$ denotes $\boldsymbol{x} \otimes \boldsymbol{x} \otimes \boldsymbol{x}$, where $(\boldsymbol{x} \otimes \boldsymbol{x} \otimes \boldsymbol{x})_{ijk} = x_i x_j x_k$. $\mathrm{sym}(\boldsymbol{x} \otimes \boldsymbol{y} \otimes \boldsymbol{z})$ denotes the symmetrized version of $\boldsymbol{x} \otimes \boldsymbol{y} \otimes \boldsymbol{z}$, i.e. $\mathrm{sym}(\boldsymbol{x} \otimes \boldsymbol{y} \otimes \boldsymbol{z})_{ijk} = \sum_{\sigma \in \mathrm{Perm}[d]} x_{\sigma(i)} y_{\sigma(j)} z_{\sigma(k)}$. $\boldsymbol{e}_i, i \in [d]$ denotes the standard basis vectors for $\mathbb{R}^d$. Through out the paper, we assume that $\boldsymbol{w}_k^* = 0$, without loss of generality.

2. Algorithms

In this section, we present our algorithms to learn the regression and gating parameters *separately*. Figure 2 summarizes our algorithm. First we take a moment to highlight the issues of the existing approaches.

For illustration purposes, we suppose that $k = 2$ in (1). We assume without loss of generality that $\boldsymbol{w}_k^* = \boldsymbol{w}_2^* = 0$ and denote $\boldsymbol{w}_1^* = \boldsymbol{w}^*$. Thus the 2-MoE model is given by $P_{y|\boldsymbol{x}}$:

$$\frac{e^{\langle \boldsymbol{w}^*, \boldsymbol{x} \rangle} \mathcal{N}(y | g(\langle \boldsymbol{a}_1^*, \boldsymbol{x} \rangle), \sigma^2)}{1 + e^{\langle \boldsymbol{w}^*, \boldsymbol{x} \rangle}} + \frac{\mathcal{N}(y | g(\langle \boldsymbol{a}_2^*, \boldsymbol{x} \rangle), \sigma^2)}{1 + e^{\langle \boldsymbol{w}^*, \boldsymbol{x} \rangle}} \quad (2)$$

Issues with traditional tensor methods. In the far simplified setting of the absence of the gating parameter, i.e. $\boldsymbol{w}^* = 0 \in \mathbb{R}^d$, we see that 2-MoE reduces to 2-uniform mixture of GLMs. In this case, for $\boldsymbol{x} \sim \mathcal{N}(0, I_d)$, the standard approach is to construct a 3^{rd}-order tensor $\mathcal{T}$ by regressing the output y on the score transformation $\mathcal{S}_3(\boldsymbol{x}) \triangleq \boldsymbol{x} \otimes \boldsymbol{x} \otimes \boldsymbol{x} - \sum_{i \in [d]} \mathrm{sym}\,(\boldsymbol{x} \otimes \boldsymbol{e}_i \otimes \boldsymbol{e}_i)$, i.e.

$$\mathcal{T} \triangleq \mathbb{E}[y \cdot \mathcal{S}_3(\boldsymbol{x})] = \frac{1}{2}\mathbb{E}[g'''(\langle \boldsymbol{a}_1^*, \boldsymbol{x} \rangle)] \cdot (\boldsymbol{a}_1^*)^{\otimes 3} + \frac{1}{2}\mathbb{E}[g'''(\langle \boldsymbol{a}_2^*, \boldsymbol{x} \rangle)] \cdot (\boldsymbol{a}_2^*)^{\otimes 3}. \quad (3)$$

Here the second equality follows from the generalized Stein's lemma that $\mathbb{E}[f(\boldsymbol{x}) \cdot \mathcal{S}_3(\boldsymbol{x})] = \mathbb{E}[\nabla_{\boldsymbol{x}}^{(3)} f(\boldsymbol{x})]$ under some regularity conditions on $f : \mathbb{R}^d \mapsto \mathbb{R}$ (see Lemma 2 in Appendix A). Then the regressors can be learned through spectral decomposition on $\mathcal{T}$, where the uniqueness of decomposition follows from (Kruskal, 1977). If we apply a similar technique for 2-MoE in (2), we obtain that

$$\mathbb{E}[y \cdot \mathcal{S}_3(\boldsymbol{x})] = \sum_{i=1,2} \alpha_i (\boldsymbol{a}_i^*)^{\otimes 3} + \beta_i \,\mathrm{sym}(\boldsymbol{a}_i^* \otimes \boldsymbol{a}_i^* \otimes \boldsymbol{w}^*) + \gamma_i \,\mathrm{sym}(\boldsymbol{a}_i^* \otimes \boldsymbol{w}^* \otimes \boldsymbol{w}^*) + \delta (\boldsymbol{w}^*)^{\otimes 3}, \quad (4)$$

where $\alpha_i, \beta_i, \gamma_i, \delta$ are some scalar constants depending on the parameters $\boldsymbol{a}_1^*, \boldsymbol{a}_2^*, \boldsymbol{w}^*$ and g (see Appendix D.1 for the proof). Thus (4) reveals that traditional spectral methods do not yield a supersymmetric tensor of the desired parameters for MoEs. In fact, (4) contains all the 3^{rd}-order rank-1 terms formed by $\boldsymbol{a}_1^*, \boldsymbol{a}_2^*$ and $\boldsymbol{w}^*$. Hence we cannot recover these parameters uniquely. Note that the inherent coupling between the regressors $\boldsymbol{a}_1^*, \boldsymbol{a}_2^*$ and the gating parameter $\boldsymbol{w}^*$ in (2) manifests as a cross tensor in (4). This coupling serves as a key limitation for the traditional methods which critically rely on the fact that the mixing probability $p = \frac{1}{2}$ in (4) is a constant. In fact, we recover (3) by letting $\boldsymbol{w}^* = 0$ in (4).

Issues with EM algorithm. EM algorithm is the workhorse for parameter learning in both the k-MoE and HME models (Jordan & Jacobs, 1994). However, it is well known that EM is prone to spurious minima and existing theoretical results only establish local convergence for the regressors and the gating parameters. Indeed, our numerical experiments in Section 4.3 verify this fact. Figure 3b and Figure 3c highlight that joint-EM often gets stuck in bad local minima.

2.1. The proposed algorithm for learning MoE

In order to tackle these challenges, we take a different route and propose to estimate the regressors and gating parameters *separately*. To gain intuition about our approach, let us consider 2-MoE model in (2) with $\sigma = 0$ and linear g. Then we have that y either equals $\langle \boldsymbol{a}_1^*, \boldsymbol{x} \rangle$ with probability $\sigma(\langle \boldsymbol{w}^*, \boldsymbol{x} \rangle)$ or equals $\langle \boldsymbol{a}_2^*, \boldsymbol{x} \rangle$ with probability $1 - \sigma(\langle \boldsymbol{w}^*, \boldsymbol{x} \rangle)$, where $\sigma(\cdot)$ is the sigmoid function. If we exactly know $\boldsymbol{w}^*$, we can recover $\boldsymbol{a}_1^*$ and $\boldsymbol{a}_2^*$ by solving a simple linear regression problem since we can recover the true latent variable $\boldsymbol{z} \in \{1, 2\}$ with high probability. Similarly, if we know $\boldsymbol{a}_1^*$ and $\boldsymbol{a}_2^*$, it is easy to see that we can recover $\boldsymbol{w}^*$ by solving a binary linear classification problem. Thus knowing either the regressors or the gating parameters makes the estimation of other parameters easier. However, how do we first obtain one set of parameters without any knowledge about the other?

Our approach precisely addresses this question and breaks the *grid lock*. We show that we can extract the regressors $\boldsymbol{a}_1^*$ and $\boldsymbol{a}_2^*$ without knowing $\boldsymbol{w}^*$ at all, just using the samples. Although we explain our approach with two mixtures, all claims are made precise for general k in Theorems 1 and 2, and the algorithms are written for general k as well in Algorithms 1 and 2.

Step 1: Estimation of regressors

To learn the regressors, we first pre-process $\boldsymbol{x} \sim \mathcal{N}(0, I_d)$ using the score transformations $\mathcal{S}_3$ and $\mathcal{S}_2$, i.e.

$$\mathcal{S}_3(\boldsymbol{x}) \triangleq \boldsymbol{x} \otimes \boldsymbol{x} \otimes \boldsymbol{x} - \sum_{i \in [d]} \operatorname{sym}\left(\boldsymbol{x} \otimes \boldsymbol{e}_i \otimes \boldsymbol{e}_i\right), \tag{5}$$

$$\mathcal{S}_2(\boldsymbol{x}) \triangleq \boldsymbol{x} \otimes \boldsymbol{x} - I. \tag{6}$$

These score functions can be viewed as higher-order feature extractors from the inputs. As we have seen in (3), these transformations suffice to learn the parameters in M-GLMs. However this approach fails in the context of MoE, as highlighted in (4). Can we still construct a supersymmetric tensor for MoE?

To answer this question in a principled way, we introduce the notion of "Cubic and Quadratic Transform (CQT)" for the labels, i.e.

$$\mathcal{P}_3(y) \triangleq y^3 + \alpha y^2 + \beta y, \quad \mathcal{P}_2(y) \triangleq y^2 + \gamma y.$$

The coefficients (α, β, γ) in these polynomial transforms are obtained by solving a linear system of equations (see Appendix C). For the special case of $g =$linear, we obtain $\mathcal{P}_3(y) = y^3 - 3(1 + \sigma^2)\, y$ and $\mathcal{P}_2(y) = y^2$. These special transformations are specific to the choice of non-linearity g and the noise variance σ. The key intuition behind the design of these transforms is that we can nullify the cross moments and obtain supersymmetric tensor in (3) if we regress $\mathcal{P}_3(y)$ instead of y, for properly chosen constants α and β. This is made mathematically precise in Theorem 1. A similar argument holds for $\mathcal{P}_2(y)$ too. In addition, the choice of these polynomials is unique in the sense that any other polynomial transformations fail to yield the desired tensor structure. Using these transforms, we construct two special tensors $\hat{\mathcal{T}}_3 \in (\mathbb{R}^d)^{\otimes 3}$ and $\hat{\mathcal{T}}_2 \in (\mathbb{R}^d)^{\otimes 2}$. Later we use the robust tensor power method (Anandkumar et al., 2014) on these tensors to learn the regressors. Algorithm 1 details our learning procedure. Theorem 1 establishes the theoretical justification for our algorithm.

Algorithm 1 Learning the regressors

```
Input: Samples $(\boldsymbol{x}_i, y_i), i \in [n]$
Compute $\hat{\mathcal{T}}_3 = \frac{1}{n}\sum_{i=1}^{n} \mathcal{P}_3(y_i) \cdot \mathcal{S}_3(\boldsymbol{x}_i)$ and $\hat{\mathcal{T}}_2 = \frac{1}{n}\sum_{i=1}^{n} \mathcal{P}_2(y_i) \cdot \mathcal{S}_2(\boldsymbol{x}_i)$
$\hat{\boldsymbol{a}}_1, \ldots, \hat{\boldsymbol{a}}_k$ = Rank-$k$ tensor decomposition on $\hat{\mathcal{T}}_3$ using $\hat{\mathcal{T}}_2$
```

Step 2: Estimation of gating parameters

To gain intuition for estimating the gating parameters, let $g = \text{linear}$ in (2) for simplicity. Moreover, assume that we know both $\boldsymbol{a}_1^*$ and $\boldsymbol{a}_2^*$. Then taking conditional expectation on y, we obtain from (2) that

$$\begin{aligned}\mathbb{E}[y|\boldsymbol{x}] &= f(\langle \boldsymbol{w}^*, \boldsymbol{x} \rangle) \cdot \langle \boldsymbol{a}_1^*, \boldsymbol{x} \rangle + (1 - f(\langle \boldsymbol{w}^*, \boldsymbol{x} \rangle)) \cdot \langle \boldsymbol{a}_2^*, \boldsymbol{x} \rangle, \\ &= \langle \boldsymbol{a}_2^*, \boldsymbol{x} \rangle + f(\langle \boldsymbol{w}^*, \boldsymbol{x} \rangle) \cdot \langle \boldsymbol{a}_1^* - \boldsymbol{a}_2^*, \boldsymbol{x} \rangle, \end{aligned} \tag{7}$$

where f is the sigmoid function. Thus,

$$\mathbb{E}\left[\frac{y - \langle \boldsymbol{a}_2^*, \boldsymbol{x} \rangle}{\langle \boldsymbol{a}_1^* - \boldsymbol{a}_2^*, \boldsymbol{x} \rangle} | \boldsymbol{x}\right] = \frac{\mathbb{E}[y|\boldsymbol{x}] - \langle \boldsymbol{a}_2^*, \boldsymbol{x} \rangle}{\langle \boldsymbol{a}_1^* - \boldsymbol{a}_2^*, \boldsymbol{x} \rangle} = f(\langle \boldsymbol{w}^*, \boldsymbol{x} \rangle).$$

Note that since $\boldsymbol{x}$ is Gaussian, $\langle \boldsymbol{a}_1^* - \boldsymbol{a}_2^*, \boldsymbol{x} \rangle$ is non-zero with probability 1. Hence, to recover $\boldsymbol{w}^*$, in view of Stein's lemma, we may write

$$\begin{aligned}\mathbb{E}\left[\left(\frac{y - \langle \boldsymbol{a}_2^*, \boldsymbol{x} \rangle}{\langle \boldsymbol{a}_1^* - \boldsymbol{a}_2^*, \boldsymbol{x} \rangle}\right) \cdot \boldsymbol{x}\right] &\overset{\times}{=} \mathbb{E}\left[f(\langle \boldsymbol{w}^*, \boldsymbol{x} \rangle) \cdot \boldsymbol{x}\right] \\ &= \mathbb{E}\left[f'(\langle \boldsymbol{w}^*, \boldsymbol{x} \rangle)\right] \cdot \boldsymbol{w}^* \\ &= \mathbb{E}_{Z \sim \mathcal{N}(0,1)} f'(\|\boldsymbol{w}^*\|\, Z) \cdot \boldsymbol{w}^* \\ &\propto \boldsymbol{w}^*.\end{aligned}$$

However, it turns out that the above chain of equalities does not hold. Surprisingly, the first equality, which essentially is the law of iterated expectations, is not valid in this case as $\frac{y - \langle \boldsymbol{a}_2^*, \boldsymbol{x} \rangle}{\langle \boldsymbol{a}_1^* - \boldsymbol{a}_2^*, \boldsymbol{x} \rangle}$ is not integrable since it is a mixture of two Cauchy distributions, as proved in Appendix D.4. Thus the above analysis highlights the difficulty of learning the gating parameters even in the simplest setting of two linear mixtures. Can we still learn $\boldsymbol{w}^*$ using method of moments (MoM)? In Theorem 3, we precisely address this question and show that we can still provably recover the gating parameters using MoM, by designing clever transformations on the data to infer the parameters of a Cauchy mixture distribution.

While Theorem 3 highlights that gating parameters can be learnt using the method of moments for 2-MoE, we still need a principled approach to learn these parameters for a more generic setting of k-MoE. Recall that the traditional joint-EM algorithm randomly initializes both the regressors and the gating parameters and updates them iteratively. Figure 3b and Figure 3c highlight that this procedure is prone to spurious minima. Can we still learn the gating parameters with *global initializations*? To address this question, we utilize the regressors learnt from Algorithm 1. In particular, we use EM algorithm to update *only* the gating parameters, while fixing the regressors $\hat{\boldsymbol{a}}_1, \dots, \hat{\boldsymbol{a}}_k$. We show in Theorem 2 that, with *global/random* initializations, this variant of EM algorithm learns the true parameters. To the best of our knowledge, this is the first global convergence result for EM for $k > 2$ mixtures. This motivates the following algorithm ($\varepsilon > 0$ is some error tolerance):

Algorithm 2 Learning the gating parameter

```
Input: Samples $(\boldsymbol{x}_i, y_i), i \in [n]$ and regressors $\hat{\boldsymbol{a}}_1, \dots, \hat{\boldsymbol{a}}_k$ from Algorithm 1
$t \leftarrow 0$
Initialize $\boldsymbol{w}_0$ uniformly randomly in its domain $\Omega$
while (Estimation error $< \varepsilon$ ) do
    Compute the posterior $p^{(i)}_{\boldsymbol{w}_t}$ according to (9) for each $j \in [k]$ and $i \in [n]$
    Compute $Q(\boldsymbol{w}|\boldsymbol{w}_t)$ according to (8) using empirical expectation
    Set $\boldsymbol{w}_{t+1} = \mathrm{argmax}_{\boldsymbol{w} \in \Omega} Q(\boldsymbol{w}|\boldsymbol{w}_t)$
    $t \leftarrow t + 1$
    Estimation error $= \|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|$
end while
```

3. Theoretical analysis

In this section, we provide the theoretical guarantees for our algorithms in the population setting. We first formally state our assumptions and justify the rationale behind them:

1. $\boldsymbol{x}$ follows standard Gaussian distribution, i.e. $\boldsymbol{x} \sim \mathcal{N}(0, I_d)$.
2. $\|\boldsymbol{a}_i^*\|_2 = 1$ for $i \in [k]$ and $\|\boldsymbol{w}_i^*\|_2 \leq R$ for $i \in [k-1]$, with some $R > 0$.
3. $\boldsymbol{a}_i^*, i \in [k]$ are linearly independent and $\boldsymbol{w}_i^*$ is orthogonal to span$\{\boldsymbol{a}_1^*, \dots, \boldsymbol{a}_k^*\}$ for $i \in [k-1]$.
4. The non-linearity $g : \mathbb{R} \to \mathbb{R}$ is (α, β, γ)-valid, which we define in Appendix C. For example, this class includes g =linear, sigmoid and ReLU.

Remark. We note that the Gaussianity of the input distribution and norm constraints on the parameters are standard assumptions in the learning of neural networks literature (Janzamin et al., 2015; Li & Yuan, 2017; Ge et al., 2017; Zhong et al., 2017; Du et al., 2017; Safran & Shamir, 2017) and also that of M-GLMs (Sedghi et al., 2014; Yi et al., 2016; Zhong et al., 2016; Balakrishnan et al., 2017). An interpretation behind Assumption 3 is that if we think of $\boldsymbol{x}$ as a high-dimensional feature vector, distinct sub-features of $\boldsymbol{x}$ are used to perform the two distinct tasks of classification (using $\boldsymbol{w}_i^*$'s) and regression (using $\boldsymbol{a}_i^*$'s). We note that we need the above assumptions only for the technical analysis. In Section 4.1 and Section 4.2, we empirically verify that our algorithms work well in practice even under the relaxation of these assumptions. Thus we believe that the assumptions are merely technical artifacts.

We are now ready to state our results.

Theorem 1 (Recovery of regression parameters)**.** *Let* $(\boldsymbol{x}, y)$ *be generated according the true model (1). Under the above assumptions, we have that*

$$\mathcal{T}_2 \triangleq \mathbb{E}[\mathcal{P}_2(y) \cdot \mathcal{S}_2(\boldsymbol{x})] = \sum_{i=1}^{k} c'_g \mathbb{E}[P_{i|\boldsymbol{x}}] \cdot \boldsymbol{a}_i^* \otimes \boldsymbol{a}_i^*,$$

$$\mathcal{T}_3 \triangleq \mathbb{E}[\mathcal{P}_3(y) \cdot \mathcal{S}_3(\boldsymbol{x})] = \sum_{i=1}^{k} c_{g,\sigma} \mathbb{E}[P_{i|\boldsymbol{x}}] \cdot \boldsymbol{a}_i^* \otimes \boldsymbol{a}_i^* \otimes \boldsymbol{a}_i^*,$$

where c'_g and $c_{g,\sigma}$ are two non-zero constants depending on g and σ. Hence the regressors $\boldsymbol{a}_i^*$'s can be learnt through tensor decomposition on $\mathcal{T}_2$ and $\mathcal{T}_3$.

Proof. (Sketch) To highlight the central ideas behind the proof, first let g =linear. From (1) we get that

$$\mathbb{E}[y|\boldsymbol{x}] = \sum_{i \in [k]} p_i^*(\boldsymbol{x}) \langle \boldsymbol{a}_i^*, \boldsymbol{x} \rangle,$$

where $p_i^*(\boldsymbol{x}) \triangleq P_{i|\boldsymbol{x}}$ for $i \in [k]$. Taking the cross moment of y with $\mathcal{S}_3(\boldsymbol{x})$ and using Lemma 2 we obtain that

$$\begin{aligned}\mathbb{E}[y \cdot \mathcal{S}_3(\boldsymbol{x})] &= \sum_{i \in [k]} \mathbb{E}[p_i^*(\boldsymbol{x}) \langle \boldsymbol{a}_i^*, \boldsymbol{x} \rangle \cdot \mathcal{S}_3(\boldsymbol{x})] \\ &= \sum_{i \in [k]} \mathbb{E}[\nabla^{(3)}_{\boldsymbol{x}} (p_i^*(\boldsymbol{x}) \langle \boldsymbol{a}_i^*, \boldsymbol{x} \rangle)].\end{aligned}$$

Notice that had $p_i^*(\boldsymbol{x})$ been a constant in the above equation, we would obtain a supersymmetric tensor easily as is the case with M-GLMs. However, $\mathbb{E}[\nabla^{(3)}_{\boldsymbol{x}} (p_i^*(\boldsymbol{x}) \langle \boldsymbol{a}_i^*, \boldsymbol{x} \rangle)]$ now contains all the third-order rank-1 terms involving the tensor product of $\boldsymbol{w}_1^*, \dots, \boldsymbol{w}_{k-1}^*$ and $\boldsymbol{a}_i^*$ for any fixed i. Our key insight is that this issue can be avoided if we cleverly transform y. In particular, we consider a cubic transformation $\mathcal{P}_3(y) = y^3 - 3y(1+\sigma^2)$ and obtain that

$$\mathbb{E}[\mathcal{P}_3(y)|\boldsymbol{x}] = \sum_{i\in[k]} p_i^*(\boldsymbol{x})(\langle \boldsymbol{a}_i^*, \boldsymbol{x}\rangle^3 - 3\langle \boldsymbol{a}_i^*, \boldsymbol{x}\rangle)$$

Now it turns out that after using the orthogonality of $\boldsymbol{w}_i^*$ and $\boldsymbol{a}_i^*$, and the fact $\mathbb{E}[p(Z)] = \mathbb{E}[p'(Z)] = \mathbb{E}[p''(Z)] = 0$ for 3^{rd}-Hermite polynomial $p(z) = z^3 - 3z$ and $Z \sim \mathcal{N}(0,1)$, we can nullify the cross-moments between $\boldsymbol{w}_i^*$'s and $\boldsymbol{a}_i^*$'s to obtain that

$$\mathbb{E}[\mathcal{P}_3(y)\cdot \mathcal{S}_3(\boldsymbol{x})] = 6\sum_{i\in[k]} \mathbb{E}[p_i^*(\boldsymbol{x})].(\boldsymbol{a}_i^*)^{\otimes 3}.$$

Similarly, we can show that $\mathbb{E}[\mathcal{P}_2(y)\cdot\mathcal{S}_2(\boldsymbol{x})] = 2\sum_{i\in[k]}\mathbb{E}[p_i^*(\boldsymbol{x})].(\boldsymbol{a}_i^*)^{\otimes 2}$. For a general non-linearity $g:\mathbb{R}\to\mathbb{R}$, we can similarly design cubic and quadratic polynomials $\mathcal{P}_3 = y^3 + \alpha y^2 + \beta y$ and $\mathcal{P}_2 = y^2 + \gamma y$ such that we can still construct supersymmetric tensors involving the regressors. In order to obtain the unique set of coefficients (α, β, γ), we need to solve a linear system of equations, which we describe in Appendix C. □

Once we obtain $\mathcal{T}_2$ and $\mathcal{T}_3$, the recovery gurantees for the regressors $\boldsymbol{a}_i^*$ follow from the standard tensor decomposition guarantees, for example, Theorem 4.3 and Theorem 5 of (Anandkumar et al., 2014). We assume that the learnt regressors $\boldsymbol{a}_i$ are such that $\max_{i\in[k]}\|\boldsymbol{a}_i - \boldsymbol{a}_i^*\|_2 = \sigma^2\varepsilon$ for some $\varepsilon > 0$. Now we present our theoretical results for global convergence of EM. First we briefly recall the algorithm. Let Ω denote the domain of our gating parameters, defined as

$$\Omega = \{\boldsymbol{w} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{k-1}) : \|\boldsymbol{w}_i\|_2 \le R, \forall i \in [k-1]\}.$$

Then the population EM for the mixture of experts consists of the following two steps:

- **E-step:** Using the current estimate $\boldsymbol{w}_t$ to compute the function $Q(\cdot|\boldsymbol{w}_t)$,
- **M-step:** $\boldsymbol{w}_{t+1} = \text{argmax}_{\boldsymbol{w}\in\Omega} Q(\boldsymbol{w}|\boldsymbol{w}_t)$,

where the function $Q(.|\boldsymbol{w}_t)$ is the expected log-likelihood of the complete data distribution with respect to current posterior distribution. Mathematically,

$$\begin{aligned} Q(\boldsymbol{w}|\boldsymbol{w}_t) &\triangleq \mathbb{E}_{(\boldsymbol{x},y)}\mathbb{E}_{P_{z|\boldsymbol{x},y,\boldsymbol{w}_t}}[\log P_{\boldsymbol{w}}(\boldsymbol{x}, z, y)] \\ &= \mathbb{E}_{(\boldsymbol{x},y)}\mathbb{E}_{P_{z|\boldsymbol{x},y,\boldsymbol{w}_t}}[\log P(\boldsymbol{x})P_{\boldsymbol{w}}(z|\boldsymbol{x})P(y|\boldsymbol{x}, z)] \\ &= \mathbb{E}_{(\boldsymbol{x},y)}\mathbb{E}_{P_{z|\boldsymbol{x},y,\boldsymbol{w}_t}}[\log P_{\boldsymbol{w}}(z|x)] + \text{const.} \\ &= \mathbb{E}[\sum_{i\in[k-1]} p_{\boldsymbol{w}_t}^{(i)}(\boldsymbol{w}_i^\top \boldsymbol{x}) - (1 + \sum_{i\in[k-1]} e^{\boldsymbol{w}_i^\top \boldsymbol{x}})] \\ &\quad + \text{const.} \end{aligned} \tag{8}$$

where const refers to terms not depending on $\boldsymbol{w}$, $P_{\boldsymbol{w}}(z = i|x) = \exp(\boldsymbol{w}_i^\top\boldsymbol{x})/\sum_j \exp(\boldsymbol{w}_j^\top \boldsymbol{x})$ and $p_{\boldsymbol{w}_t}^{(i)} \triangleq \mathbb{P}[z = i|\boldsymbol{x}, y, \boldsymbol{w}_t]$ corresponds to the posterior probability for the i^{th} expert, given by

$$p_{\boldsymbol{w}_t}^{(i)} = \frac{p_{i,t}(\boldsymbol{x})\mathcal{N}(y|g(\boldsymbol{a}_i^\top\boldsymbol{x}), \sigma^2)}{\sum_{j\in[k]} p_{j,t}(\boldsymbol{x})\mathcal{N}(y|g(\boldsymbol{a}_j^\top\boldsymbol{x}), \sigma^2)}, \tag{9}$$

$$p_{i,t}(\boldsymbol{x}) = \frac{e^{(\boldsymbol{w}_t)_i^\top \boldsymbol{x}}}{1 + \sum_{j\in[k-1]} e^{(\boldsymbol{w}_t)_j^\top \boldsymbol{x}}}.$$

In (8), the expectation is with respect to the true distribution of $(\boldsymbol{x}, y)$, given by (1). Thus the EM can be viewed as a deterministic procedure which maps $\boldsymbol{w}_t \mapsto M(\boldsymbol{w}_t)$ where

$$M(\boldsymbol{w}) = \text{argmax}_{\boldsymbol{w}'\in\Omega} Q(\boldsymbol{w}'|\boldsymbol{w}).$$

When the estimated regressors $\boldsymbol{a}_i$ equal the true parameters $\boldsymbol{a}_i^*$, it follows from the self-consistency property of the EM that the true parameter $\boldsymbol{w}^*$ is a fixed-point for the EM operator M, i.e. $M(\boldsymbol{w}^*) = \boldsymbol{w}^*$ (McLachlan & Krishnan, 2007). However, this does not guarantee that EM converges to $\boldsymbol{w}^*$. In the following theorem, we show that even when the regressors are known approximately, EM algorithm converges to the true gating parameters at a geometric rate upto an additive error, under *global* initializations. For the error metric, we define $\|\boldsymbol{w} - \boldsymbol{w}'\| \triangleq \max_{i\in[k-1]}\|\boldsymbol{w}_i - \boldsymbol{w}_i'\|_2$ for any $\boldsymbol{w}, \boldsymbol{w}' \in \Omega$. We assume that $R = 1$ for simplicity. (Our results extend straightforwardly to general R).

Theorem 2. *Let $\varepsilon > 0$ be such that $\max_i \|\boldsymbol{a}_i - \boldsymbol{a}_i^*\|_2 = \sigma^2\varepsilon$. There exists a constant $\sigma_0 > 0$ such that whenever $0 < \sigma < \sigma_0$, for any random initialization $\boldsymbol{w}_0 \in \Omega$, the population-level EM updates on the gating parameter $\{\boldsymbol{w}\}_{t\ge 0}$ converge almost geometrically to the true parameter $\boldsymbol{w}^*$ upto an additive error, i.e.*

$$\|\boldsymbol{w}_t - \boldsymbol{w}^*\| \le (\kappa_\sigma)^t \|\boldsymbol{w}_0 - \boldsymbol{w}^*\| + \kappa\varepsilon\sum_{i=0}^{t-1}\kappa_\sigma^i,$$

where κ_σ, κ are dimension-independent constant depending on g and σ such that $\kappa_\sigma \xrightarrow{\sigma\to 0} 0$ and $\kappa \le (k-1)\frac{\sqrt{6(2+\sigma^2)}}{2}$ for g =linear, sigmoid and ReLU.

Proof. (Sketch) One can show that the $Q(\cdot|\boldsymbol{w}_t)$ defined in (8) is a strongly concave function. Moreover, if we let $\varepsilon = 0$ and $\boldsymbol{w}_t = \boldsymbol{w}^*$, we have from the self-consistency of EM that $\text{argmax} Q(\cdot|\boldsymbol{w}^*) = \boldsymbol{w}^*$. Thus if we can show that the functions are $Q(\cdot|\boldsymbol{w}_t)$ and $Q(\cdot|\boldsymbol{w}^*)$ "sufficiently close" whenever $\boldsymbol{w}_t$ and $\boldsymbol{w}^*$ are close, we can use the EM convergence analysis tools from (Balakrishnan et al., 2017) to show that their corresponding maximizers also stay close upto a scaling factor determined by κ_σ above. Then it follows that the EM updates converge geometrically. □

Remark. In the M-step of the EM algorithm, the next iterate is chosen so that the function $Q(\cdot|\boldsymbol{w}_t)$ is maximized. Instead we can perform an ascent step in the direction of the gradient of $Q(\cdot|\boldsymbol{w}_t)$ to produce the next iterate, i.e. $\boldsymbol{w}_{t+1} = \Pi_\Omega(\boldsymbol{w}_t + \alpha \nabla Q(\boldsymbol{w}_t|\boldsymbol{w}_t))$, where $\Pi_\Omega(\cdot)$ is the projection operator. This variant of EM algorithm is known as *Gradient EM*. In Appendix G, we show that Gradient EM also enjoys similar convergence guarantees.

MoM to learn gating parameters. In Theorem 2, we proved that EM algorithm provably recovers the true gating parameters for any $k \geq 2$ mixtures. In this section, we show that for the special case of $k = 2$, we can learn $\boldsymbol{w}^*$ (upto the unit direction) using an alternative procedure involving MoM. First we define

$$\mathrm{Ratio}(\boldsymbol{x}, y) \triangleq \frac{y - \langle \boldsymbol{a}_2, \boldsymbol{x} \rangle}{\langle \boldsymbol{a}_1 - \boldsymbol{a}_2, \boldsymbol{x} \rangle} \tag{10}$$

The following theorem establishes that the the CDF of the random variable $\mathrm{Ratio}(\boldsymbol{x}, y)$, when regressed on input $\boldsymbol{x}$, is proportional to $\boldsymbol{w}^*$.

Theorem 3. *Suppose that* $(\boldsymbol{a}_1, \boldsymbol{a}_2) = (\boldsymbol{a}_1^*, \boldsymbol{a}_2^*)$. *Then we have that*

$$\mathbb{E}[\mathbb{1}\{\mathrm{Ratio}(\boldsymbol{x}, y) \leq 0.5\} \cdot \boldsymbol{x}] = \alpha \boldsymbol{w}^*,$$

where $\alpha \in \mathbb{R}$ *is a scalar given by* $\alpha = \mathbb{E}[f'(\langle \boldsymbol{w}^*, \boldsymbol{x} \rangle) \cdot \left(1 - 2\Phi\left(\frac{|\langle \boldsymbol{a}_1 - \boldsymbol{a}_2, \boldsymbol{x} \rangle|}{2\sigma}\right)\right)]$.

Proof. (Sketch) We first show that $\mathrm{Ratio}(\boldsymbol{x}, y)$ is a mixture of Cauchy distributions. Then we show that $\mathbb{E}[\mathbb{1}\{\mathrm{Ratio}(\boldsymbol{x}, y) \leq z\}|\boldsymbol{x}] = \mathbb{P}\left[\mathrm{Ratio} \leq z|\boldsymbol{x}\right] = f(\boldsymbol{w}^\top \boldsymbol{x})\Phi\left((z-1)\frac{|\Delta_x|}{\sigma}\right) + (1 - f(\boldsymbol{w}^\top \boldsymbol{x}))\Phi\left(z\frac{|\Delta_x|}{\sigma}\right)$ where $\Delta_x = (\boldsymbol{a}_1 - \boldsymbol{a}_2)^\top \boldsymbol{x}$. Then our result follows from taking the first moment of the indicator random variable with $\boldsymbol{x}$ and Stein's lemma. □

4. Experiments

In this section, we empirically validate our algorithm in various settings and compare its performance to that of EM on both synthetic and real world datasets [1]. In both the scenarios, we found that our algorithm consistently outperforms the existing approaches. For the tensor decomposition in our Algorithm 1, we use the Orth-ALS package by (Sharan & Valiant, 2017). In all the synthetic experiments, we first draw the regressors $\{\boldsymbol{a}_i^*\}_{i=1}^k$ i.i.d uniformly from the unit sphere $\mathbb{S}^{d-1}$. The input distribution $P_{\boldsymbol{x}}$ and the generation of $\boldsymbol{w}_i^*$'s are detailed for each experiment. Then the labels y_i are generated according to the true k-MoE model in (1) for linear activation. Additional experiments in this setting with non-linear activations are detailed in Appendix H.1. Experiments with real world data are provided in Section 4.4.

4.1. Non-gaussian inputs

In this section we let the input distribution to be mixtures of Gaussians (GMM). We let $k = 2, d = 10$ and $\sigma = 0.1$. The gating parameter $\boldsymbol{w}^* \in \mathbb{R}^{10}$ is uniformly chosen from the unit sphere $\mathbb{S}^9$. To generate the input features, we first randomly draw $\mu_1, \mu_2 \in \mathbb{S}^9$, and generate n i.i.d. samples $\boldsymbol{x}_i \sim p\mathcal{N}(\mu_1, I_d) + (1-p)\mathcal{N}(\mu_2, I_d)$, where $p \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. Here $n = 2000$. Since $\boldsymbol{x}$ is a 2-GMM, its score functions $\mathcal{S}_3(\boldsymbol{x}), \mathcal{S}_2(\boldsymbol{x})$ are computed using the densities of Gaussian mixtures (Janzamin et al., 2014). To gauge the performance of our algorithm, we measure the correlation of our learned parameters $\boldsymbol{a}_1, \boldsymbol{a}_2$ and $\boldsymbol{w}$ with the ground truth, i.e.

$$\mathsf{Regressor\ Fit}(\boldsymbol{a}_1, \boldsymbol{a}_2) = \max_{\pi} \min_{i \in \{1,2\}} |\langle \boldsymbol{a}_{\pi(i)}, \boldsymbol{a}_i^* \rangle|, \tag{11}$$

where $\pi : \{1, 2\} \to \{1, 2\}$ is a permutation. Similarly, for the gating parameter, we define

$$\mathsf{Gating\ Fit}(\boldsymbol{w}) = |\langle \boldsymbol{w}, \boldsymbol{w}^* \rangle|. \tag{12}$$

Here we assume that all the parameters are unit-normalized. The closer the values of fit are to 1, the closer the learnt parameters are to the ground truth. As shown in Table 1, our algorithms are able to learn the ground truth very accurately in a variety of settings, as indicated by the measured fit. This highlights the fact that our algorithms are robust to the input distributions.

4.2. Non-orthogonal parameters

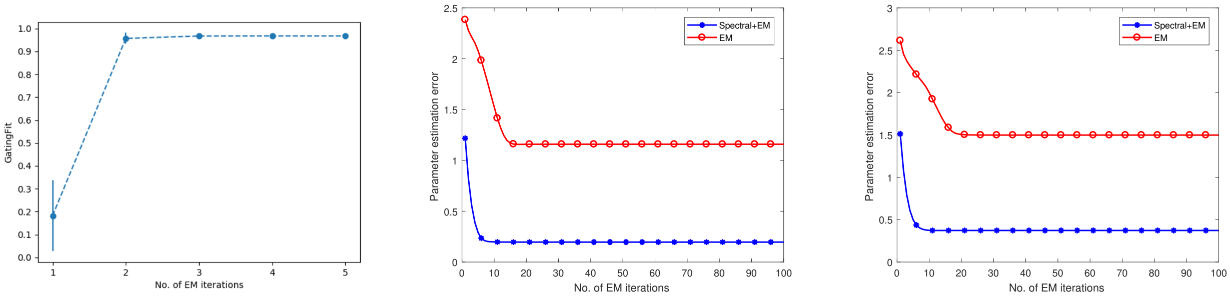
In this section we verify that our algorithms still work well in practice even under the relaxation of Assumption 3. For the experiments, we consider the similar setting as before with $k = 2, d = 10, \sigma = 0.1$ and the gating parameter $\boldsymbol{w}^*$ is drawn uniformly from $\mathbb{S}^9$ without the orthogonality restriction. We let $\boldsymbol{x}_i \overset{i.i.d.}{\sim} \mathcal{N}(0, I_d)$. We choose $n = 2000$. We use $\mathsf{RegressorFit}$ and $\mathsf{GatingFit}$ defined in (11) and (12) respectively, as our performance metrics. From Table 2, we can see that the performance of our algorithms is almost the same across both the settings. In both the scenarios, our fit is consistently greater than 0.9.

In Figure 3a, we plotted $\mathsf{GatingFit}(\boldsymbol{w}_t)$ vs. the number of iterations t, as $\boldsymbol{w}_t$ is updated according to Algorithm 2, over 10 independent trials. We observe that the learned parameters converge to the true parameters in less than 5 iterations.

4.3. Comparison to joint-EM

Here we compare the performance of our algorithm with that of the joint-EM. We let the number of mixture components be $k = 3$ and $k = 4$. We let $\boldsymbol{x} \sim \mathcal{N}(0, I_d)$ and the gating parameters are drawn uniformly from $\mathbb{S}^9$. If $\boldsymbol{A} = [\boldsymbol{a}_1 \ldots \boldsymbol{a}_k]$

[1] Codes are available at this repository MoE codes.



(a) Non-orthogonality (b) $k = 3$ (c) $k = 4$

Figure 3: (a): GatingFit for our algorithm under non-orthogonality setting. (b),(c): Estimation error $\mathcal{E}(\boldsymbol{A}, \boldsymbol{W})$ of our algorithm vs. joint-EM algorithm. Our algorithm is significantly better than the joint-EM under random initializations.

Table 1: Fit of our learned parameters for non-Gaussian inputs

	$p = 0.1$	$p = 0.3$	$p = 0.5$	$p = 0.7$	$p = 0.9$
Regressor Fit	0.93 ± 0.06	0.94 ± 0.02	0.92 ± 0.04	0.92 ± 0.02	0.91 ± 0.06
Gating Fit	0.9 ± 0.1	0.97 ± 0.01	0.93 ± 0.04	0.96 ± 0.03	0.97 ± 0.01

Table 2: Performance of our algorithm under orthogonal and non-orthogonal settings

	Regressor Fit	Gating Fit
Non-orthogonal	0.9 ± 0.08	0.96 ± 0.02
Orthogonal	0.93 ± 0.03	0.96 ± 0.03

and $\boldsymbol{W} = [\boldsymbol{w}_1 \dots \boldsymbol{w}_{k-1}\ 0]$ denote the estimated expert and gating parameters respectively, our evaluation metric is $\mathcal{E}$, the Frobenious norm of the parameter error accounting for the best possible permutation $\pi : [k] \to [k]$, i.e. $\mathcal{E}(\boldsymbol{A}, \boldsymbol{W}) = \inf_\pi \|\boldsymbol{A} - \boldsymbol{A}^*_\pi\|_F + \|\boldsymbol{W} - \boldsymbol{W}^*_\pi\|_F$, where $\boldsymbol{A}^*_\pi = [\boldsymbol{a}^*_{\pi(1)} \dots \boldsymbol{a}^*_{\pi(k)}]$ denotes the permuted regression parameter matrix and similarly for $\boldsymbol{W}^*_\pi$. In Figure 3b and Figure 3c, we compare the performance of our algorithm with the joint-EM algorithm for $n = 8000, d = 10, \sigma = 0.5$. The plotted estimation error $\mathcal{E}(\boldsymbol{A}, \boldsymbol{W})$ is averaged for 10 trials. It is clear that our algorithm is able to recover the true parameters thus resulting in much smaller parameter error than the joint-EM which often gets stuck in local optima. In addition, our algorithm is able to learn these parameters in very few iterations, often less than 10 iterations. We also find that our algorithm consistently outperforms the joint-EM for different choices of non-linearities, number of samples, number of mixtures, etc. (details provided in Appendix H). Note that the above error metric $\mathcal{E}(\boldsymbol{A}, \boldsymbol{W})$ is close to zero if and only if Regressor Fit and Gating Fit is close to one.

4.4. Real data

To highlight the generalizability of our algorithm, in Appendix H.2 of the supplement, we compare the performance of our algorithm to that of the standard approaches on a variety of real world datasets. Results from these experiments highlight the fact that in the real world scenario, where the underlying data is not generated according to a MoE model, our approach still learns a superior set of parameters as opposed to the existing algorithms. This fact is reflected in the lowest prediction errors obtained by our algorithm.

5. Discussion

In this paper we provided the first provable and globally consistent algorithm that can learn the true parameters of a MoE model. We believe that ideas from (Sedghi et al., 2014) can be naturally extended for the finite sample complexity analysis of the tensor decomposition to learn the regressors and similarly, techniques from (Balakrishnan et al., 2017) can be extended to the finite sample EM convergence analysis for the gating parameters. While we have focused here on parameter recovery, however, there are no statistical bounds on output prediction error when the data is not generated from the model. MoE models are known to be capable of fitting general functions, and getting statistical guarantees on learning in such regimes is an interesting direction for future work.

Acknowledgements

This work is partly supported by NSF grants 1927712 and 1815535, NSF awards CNS-1718270, 1651236, 1703403, and the Army Research Office under grant W911NF1810332.

References

Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M., and Telgarsky, M. Tensor decompositions for learning latent variable models. *J. Mach. Learn. Res.*, 15(1):2773–2832, January 2014. ISSN 1532-4435.

Balakrishnan, S., Wainwright, M. J., and Yu, B. Statistical guarantees for the EM algorithm: From population to sample-based analysis. *The Annals of Statistics*, 45(1): 77–120, 2017.

Brooks, T., Pope, D., and Marcolini., A. Airfoil self-noise and prediction. Technical report, NASA, 1989. URL https://archive.ics.uci.edu/ml/datasets/Airfoil+Self-Noise.

Chen, Y., Yi, X., and Caramanis, C. A convex formulation for mixed regression with two components: Minimax optimal rates. In *Conference on Learning Theory*, pp. 560–604, 2014.

Chung, J., Gülçehre, Ç., Cho, K., and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. abs/1412.3555, 2014.

Collobert, R., Bengio, S., and Bengio, Y. A parallel mixture of SVMs for very large scale problems. *Neural Computing*, 2002.

Du, S. S., Lee, J. D., Tian, Y., Poczos, B., and Singh, A. Gradient descent learns one-hidden-layer cnn: Don't be afraid of spurious local minima. *arXiv preprint arXiv:1712.00779*, 2017.

Ge, R., Lee, J. D., and Ma, T. Learning one-hidden-layer neural networks with landscape design. *arXiv preprint arXiv:1711.00501*, 2017.

Gross, S., Szlam, A., et al. Hard mixtures of experts for large scale weakly supervised vision. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pp. 5085–5093. IEEE, 2017.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. Adaptive mixtures of local experts. *Neural Computation*, 1991.

Janzamin, M., Sedghi, H., and Anandkumar, A. Score function features for discriminative learning: Matrix and tensor framework. abs/1412.2863, 2014. URL http://arxiv.org/abs/1412.2863.

Janzamin, M., Sedghi, H., and Anandkumar, A. Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods. *arXiv preprint arXiv:1506.08473*, 2015.

Jin, C., Zhang, Y., Balakrishnan, S., Wainwright, M. J., and Jordan, M. Local maxima in the likelihood of gaussian mixture models: Structural results and algorithmic consequences. *arXiv preprint arXiv:1609.00978*, 2016.

Jordan, M. I. and Jacobs, R. A. Hierarchical mixtures of experts and the EM algorithm. *Neural computation*, 6(2): 181–214, 1994.

Jordan, M. I. and Xu, L. Convergence results for the EM approach to mixtures of experts architectures. *Neural Networks*, 8(9):1409–1431, 1995.

Kruskal, J. B. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications*, 18(2):95–138, 1977.

Le, P., Dymetman, M., and Renders, J.-M. Lstm-based mixture-of-experts for knowledge-aware dialogues. *arXiv preprint arXiv:1605.01652*, 2016.

Li, Y. and Yuan, Y. Convergence analysis of two-layer neural networks with relu activation. In *Advances in Neural Information Processing Systems*, pp. 597–607, 2017.

Liu, Y.-C. and Yeh, I.-C. Using mixture design and neural networks to build stock selection decision support systems. *Neural Computing and Applications*, 28(3): 521–535, 2017. doi: 10.1007/s00521-015-2090-x. URL https://archive.ics.uci.edu/ml/datasets/Stock+portfolio+performance.

Masoudnia, S. and Ebrahimpour, R. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42(2): 275, 2014.

McLachlan, G. and Krishnan, T. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

Ng, J. W. and Deisenroth, M. P. Hierarchical mixture-of-experts model for large-scale gaussian process regression. *arXiv preprint arXiv:1412.3078*, 2014.

Safran, I. and Shamir, O. Spurious local minima are common in two-layer relu neural networks. *arXiv preprint arXiv:1712.08968*, 2017.

Sedghi, H., Janzamin, M., and Anandkumar, A. Provable tensor methods for learning mixtures of classifiers. *arXiv preprint arXiv:1412.3046*, 2014.

Sharan, V. and Valiant, G. Orthogonalized ALS: A theoretically principled tensor decomposition algorithm for practical use. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3095–3104, 06–11 Aug 2017. URL http://proceedings.mlr.press/v70/sharan17a.html.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Stein, C. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability*, volume 2, pp. 583–602. University of California Press, 1972.

Sun, X., Peng, X., Ren, F., and Xue, Y. Human-machine conversation based on hybrid neural network. In *Computational Science and Engineering (CSE) and Embedded and Ubiquitous Computing (EUC), 2017 IEEE International Conference on*, volume 1, pp. 260–266. IEEE, 2017.

Sun, Y., Ioannidis, S., and Montanari, A. Learning mixtures of linear classifiers. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pp. 721–729, 2014.

Theis, L. and Bethge, M. Generative image modeling using spatial lstms. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pp. 1927–1935, Cambridge, MA, USA, 2015. MIT Press.

Tresp, V. Mixtures of gaussian processes. NIPS, 2001.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Wang, X., Yu, F., Wang, R., Ma, Y.-A., Mirhoseini, A., Darrell, T., and Gonzalez, J. E. Deep mixture of experts via shallow embedding. *arXiv preprint arXiv:1806.01531*, 2018.

Yeh, I.-C. Modeling of strength of high performance concrete using artificial neural networks. *Cement and Concrete Research*, 28(12):1797–1808, 1998. URL https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength.

Yi, X., Caramanis, C., and Sanghavi, S. Alternating minimization for mixed linear regression. In *International Conference on Machine Learning*, pp. 613–621, 2014.

Yi, X., Caramanis, C., and Sanghavi, S. Solving a mixture of many random linear equations by tensor decomposition and alternating minimization. *arXiv preprint arXiv:1608.05749*, 2016.

Yuksel, S. E., Wilson, J. N., and Gader, P. D. Twenty years of mixture of experts. *IEEE Transactions on Neural Networks and Learning Systems*, 23(8):1177–1193, 2012.

Zhong, K., Jain, P., and Dhillon, I. S. Mixed linear regression with multiple components. pp. 2190–2198. 2016.

Zhong, K., Song, Z., Jain, P., Bartlett, P. L., and Dhillon, I. S. Recovery guarantees for one-hidden-layer neural networks. *arXiv preprint arXiv:1706.03175*, 2017.