Provable Learning of Overcomplete Latent Variable Models: Semi-supervised and Unsupervised Settings

Animashree Anandkumar
University of California
Irvine, CA
a.anandkumar@uci.edu

Rong Ge
Microsoft Research
Cambridge, MA
rongge@microsoft.com

Majid Janzamin
University of California
Irvine, CA
mjanzami@uci.edu

Abstract

We provide guarantees for learning latent variable models emphasizing on the overcomplete regime, where the dimensionality of the latent space can exceed the observed dimensionality. In particular, we consider spherical Gaussian mixtures and multiview mixtures models. Our algorithm is based on method of moments, and employs a tensor decomposition method for learning. In the semi-supervised setting, we exploit the label or prior information to get a rough estimate of the model parameters, and then refine it using the tensor method on unlabeled samples. We establish learning guarantees when the number of components scales as $k = o(d^{p/2})$, where d is the observed dimension, and p is the order of the observed moment employed in the tensor method. In the unsupervised setting, a simple initialization algorithm based on SVD of the tensor slices is proposed, and the guarantees are provided under the stricter condition that $k \leq Cd$ (where constant C can be larger than 1). We also provide tight sample complexity bounds through novel covering arguments.

Keywords: Unsupervised and semi-supervised learning, latent variable models, overcomplete representations, tensor decomposition.

1 Introduction

Tensor decompositions have been recently popular for unsupervised learning of a wide range of latent variable models (LVMs) such as topic models, Gaussian mixtures, independent component analysis, network community models, and so on [1, 2, 3]. It involves decomposition of a certain low order multivariate moment tensor (typically up to fourth order), and is guaranteed to provide a consistent estimate of the model parameters. In practice, the tensor decomposition techniques have been effective in a number of applications such as blind source separation [4], computer vision [5], topic modeling [6], and community detection [7].

The state of art for guaranteed tensor decomposition involves two steps: converting the input tensor to an orthogonal symmetric form, and then solving the orthogonal decomposition through tensor eigen decomposition [1, 8, 9]. While having efficient guarantees, this approach is unable to learn *overcomplete representations*, where the latent dimensionality exceeds the observed dimensionality. This is especially limiting given the recent popularity of overcomplete feature learning in many domains, e.g. [10, 11].

In this paper, we establish guarantees for learning overcomplete LVMs, assuming incoherent components, which can be viewed as a *soft orthogonality* constraint. Incoherent representations have been extensively considered, e.g., in compressed sensing [12] and sparse coding [13, 14]. They provide flexible modeling, and are robust to noise [11]. Moreover, when we have randomly constructed (multiview) features [15], the moment tensors have incoherent components, as assumed here.

Summary of results: In this paper, we provide semi-supervised and unsupervised learning guarantees for LVMs such as spherical Gaussian mixtures and multiview mixtures model[1]. We employ a tensor decomposition algorithm, which basically performs alternating asymmetric power updates on the input tensor modes. Under the semi-supervised setting, we establish that highly overcomplete models can be learned efficiently through tensor decomposition methods. The moment tensors are constructed using unlabeled samples, and the labeled samples are used to provide a rough initialization to the tensor decomposition algorithm. In the unsupervised setting, we propose a simple initialization strategy for the tensor method, and require stricter conditions on the extent of overcompleteness for guaranteed learning. We also provide tight sample complexity bounds.

We now summarize the results for learning spherical Gaussian mixtures. Let k be the number of Gaussian mixtures (hidden dimension), and d be the observed dimensionality. In the semi-supervised setting, we prove guaranteed learning when $k = o(d^{p/2})$, where p is the order of observed moment employed for tensor decomposition. We prove that in the "low" variance regime (where the expected radius of spherical Gaussian is of the same order as that of the Gaussian mean), having an extremely small number of labeled samples for each mixture is sufficient (scaling as $\mathrm{polylog}(d, k)$ independent of the final precision). This is far less than the number of unlabeled samples required. Note that in most applications, labeled samples are expensive/hard to obtain, while many more unlabeled samples are easily available, e.g., see [16, 17]. Furthermore, the sample complexity bounds for unlabeled samples is derived, which scales as $\tilde{\Omega}(k)$.

We also provide *unsupervised* learning guarantees when no label is available. Here, the initialization is obtained by performing a rank-1 SVD on the random slices of the moment tensor. This imposes additional conditions on rank and sample complexity. We prove that when $k \leq Cd$ (for arbitrary constant $C > 1$), the model parameters can be learned using a polynomial number of initializations (which depends on C as k^{C^2}) and sample complexity scales as $\tilde{\Omega}(kd)$. This is an improvement over existing results since we do not have dependence on the condition number of the component means and in addition, we can handle overcomplete models.

Notations: Let $[n]$ denote the set $\{1, 2, \dots, n\}$, and $\|v\|$ denote the ℓ_2 norm of vector v. We also use $\tilde{O}$ and $\tilde{\Omega}$ to hide $\mathrm{polylog}$ factors in asymptotic notations O and Ω, respectively.

2 Tensor Decomposition Algorithm

We propose the tensor[2] decomposition method in Algorithm 1. The main step in (1) basically performs alternating *asymmetric power updates*[3] on different tensor modes. Notice that the updates alternate among different modes of the tensor which can be viewed as a rank-1 form of the standard alternating least squares (ALS) method. Notice that in learning LVMs, the input tensor T is the higher order observed moment.

3 Learning Spherical Gaussian Mixtures

In this section, we exploit Algorithm 1 for learning spherical Gaussian mixtures, and provide sample complexity guarantees. Consider a mixture of k different Gaussian distributions with spherical covariances. Let $w_j, j \in [k]$ denote the proportion for choosing each mixture. For each Gaussian component $j \in [k]$, $a_j \in \mathbb{R}^d$ is the mean, and $\zeta_i^2 I$ is the spherical covariance. For simplicity, we restrict to the case where all the components have the same spherical variance, i.e., $\zeta_1^2 = \zeta_2^2 = \cdots = \zeta_k^2 = \zeta^2$. The generalization is discussed in Hsu and Kakade [18]. In addition, in order to generalize the learning result to the overcomplete setting, we assume that variance parameter ζ^2 is known. The

[1] According to limited space, the results for spherical Gaussian mixtures are provided in this 4-page draft. For other latent variable models including multiview mixtures, Independent Component Analysis (ICA) and sparse coding see Appendix B.

[2] See Appendix D for detailed tensor preliminaries and notations.

[3] We view a tensor $T \in \mathbb{R}^{d \times d \times d}$ as a multilinear form. For vectors $v, w \in \mathbb{R}^d$, we have $T(I, v, w) := \sum_{j,l \in [d]} v_j w_l T(:, j, l) \in \mathbb{R}^d$ as the multilinear combination of tensor fibers. Note that the proposed asymmetric form of the algorithm is particularly useful for learning multiview mixtures model which involves asymmetric tensor decomposition.

Algorithm 1 Tensor decomposition via alternating asymmetric power updates

```
Input: Tensor $T \in \mathbb{R}^{d \times d \times d}$, number of initializations $L$, number of iterations $N$.
Output: Estimates for the components of tensor $T = \sum_{j \in [k]} w_j a_j \otimes b_j \otimes c_j$.
  for $\tau = 1$ to $L$ do
    Initialize unit vectors $\hat{a}_\tau^{(0)} \in \mathbb{R}^d$, $\hat{b}_\tau^{(0)} \in \mathbb{R}^d$, and $\hat{c}_\tau^{(0)} \in \mathbb{R}^d$ as
      • Semi-supervised setting: label information is exploited.
      • Unsupervised setting: SVD-based technique in Procedure 3 when $k \leq Cd$.
    for $t = 0$ to $N - 1$ do
      Asymmetric power updates:
        $$\hat{a}_\tau^{(t+1)} = \frac{T\left(I, \hat{b}_\tau^{(t)}, \hat{c}_\tau^{(t)}\right)}{\left\| T\left(I, \hat{b}_\tau^{(t)}, \hat{c}_\tau^{(t)}\right) \right\|}, \quad \hat{b}_\tau^{(t+1)} = \frac{T\left(\hat{a}_\tau^{(t)}, I, \hat{c}_\tau^{(t)}\right)}{\left\| T\left(\hat{a}_\tau^{(t)}, I, \hat{c}_\tau^{(t)}\right) \right\|}, \quad \hat{c}_\tau^{(t+1)} = \frac{T\left(\hat{a}_\tau^{(t)}, \hat{b}_\tau^{(t)}, I\right)}{\left\| T\left(\hat{a}_\tau^{(t)}, \hat{b}_\tau^{(t)}, I\right) \right\|}. \quad (1)$$
    end for
    weight estimation:   $\hat{w}_\tau = T\left(\hat{a}_\tau^{(N)}, \hat{b}_\tau^{(N)}, \hat{c}_\tau^{(N)}\right)$.
  end for
  Cluster set $\left\{ \left(\hat{w}_\tau, \hat{a}_\tau^{(N)}, \hat{b}_\tau^{(N)}, \hat{c}_\tau^{(N)}\right), \tau \in [L] \right\}$ into $k$ clusters as in Procedure 2.
  return the center member of these $k$ clusters as estimates $(\hat{w}_j, \hat{a}_j, \hat{b}_j, \hat{c}_j), j \in [k]$.
```

Procedure 2 Clustering process

```
Input: Tensor $T \in \mathbb{R}^{d \times d \times d}$, set of 4-tuples $\left\{ (\hat{w}_\tau, \hat{a}_\tau, \hat{b}_\tau, \hat{c}_\tau), \tau \in [L] \right\}$, parameter $\epsilon$.
  for $i = 1$ to $k$ do
    Among the remaining 4-tuples, choose $\hat{a}, \hat{b}, \hat{c}$ which correspond to the largest $|T(\hat{a}, \hat{b}, \hat{c})|$.
    Do $N$ more iterations of alternating updates in (1) starting from $\hat{a}, \hat{b}, \hat{c}$.
    Let the output of iterations denoted by $(\hat{a}, \hat{b}, \hat{c})$ be the center of cluster $i$.
    Remove all the tuples with $\max\{|\langle \hat{a}_\tau, \hat{a} \rangle|, |\langle \hat{b}_\tau, \hat{b} \rangle|, |\langle \hat{c}_\tau, \hat{c} \rangle|\} > \epsilon/2$.
  end for
  return the $k$ cluster centers.
```

following lemma shows that the problem of estimating parameters of this mixture model can be formulated as a tensor decomposition problem.

Lemma 1 (Hsu and Kakade 18)**.** *If*

$$M_3 := \mathbb{E}[x \otimes x \otimes x] - \zeta^2 \sum_{i \in [d]} \left(\mathbb{E}[x] \otimes e_i \otimes e_i + e_i \otimes \mathbb{E}[x] \otimes e_i + e_i \otimes e_i \otimes \mathbb{E}[x] \right), \tag{2}$$

then $M_3 = \sum_{j \in [k]} w_j a_j \otimes a_j \otimes a_j$.

Without loss of generality, we assume that the mean vectors $a_j, j \in [k]$ have unit ℓ_2 norm, since we can always rescale them, and adjust the weights appropriately. Also, for simplicity we assume $a_j \in \mathbb{R}^d, j \in [k]$, are uniformly i.i.d. drawn from the unit d-dimensional sphere $\mathcal{S}^{d-1}$ (see Remark 1). In this work, we focus on learning in the challenging overcomplete regime where the number of components/mixtures is larger than observed dimension. Precisely, we assume $k \geq \Omega(d)$. Note that the results can be easily adapted to the highly undercomplete regime when $k \leq o(d)$.

For brevity, we consider the low variance regime where the expected radius of spherical Gaussian is bounded by a constant, i.e., $\zeta^2 d = O(1)$. Note that since means $a_j, j \in [k]$, have unit norm, low variance regime imposes that the expected norm of spherical radius is in the same order of norm of means (model parameters). Since w_j's are the mixture probabilities, for brevity we consider $w_j = \Theta(1/k), j \in [k]$.

Semi-supervised learning**:** In the semi-supervised setting, label information is exploited to build good initialization vectors for tensor decomposition Algorithm 1 as follows. For the Spherical Gaussian mixtures, let $x_j^{(l)}, j \in [k], l \in [m_j]$, denote $m = \sum_{j \in [k]} m_j$ samples of vectors corresponding to different mixtures, where the samples with subscript j are from mixture j. Then for any $j \in [k]$, we have the empirical estimate of Gaussian means as $\hat{a}_j := \frac{1}{m_j} \sum_{l \in [m_j]} x_j^{(l)}$.

Procedure 3 SVD-based initialization when $k \leq Cd$ for arbitrary constant C

```
Input: Tensor $T \in \mathbb{R}^{d\times d\times d}$.
  Draw a random standard Gaussian vector $\theta \sim \mathcal{N}(0, I_d)$.
  Compute $u_1$ and $v_1$ as the top left and right singular vectors of $T(I, I, \theta) \in \mathbb{R}^{d\times d}$.
  $\hat{a}^{(0)} \leftarrow u_1$, $\hat{b}^{(0)} \leftarrow v_1$, and initialize $\hat{c}^{(0)}$ by update formula in (1).
  return $\left(\hat{a}^{(0)}, \hat{b}^{(0)}, \hat{c}^{(0)}\right)$.
```

Given n unlabeled samples, let

$$\epsilon_T := \tilde{O}\left(\sqrt{k/n}\right) + \tilde{O}\left(\sqrt{k}/d\right) \tag{3}$$

denote the recovery error. We first provide the settings of Algorithm 1 which include input tensor T, number of iterations N and the initialization setting.

Settings of Algorithm 1 in Theorem 1: Given n unlabeled samples $x^{(i)} \in \mathbb{R}^d, i \in [n]$, consider the empirical estimate of 3rd order moment in (2) as the input to Algorithm 1 (with *symmetric* updates). Let the number of iterations $N = \Theta\left(\log\left(1/\epsilon_T\right)\right)$. Use the empirical estimates using labeled data as initialization.

Theorem 1 (Semi-supervised learning of spherical Gaussian mixtures)**.** *Assume the Algorithm settings mentioned above hold. Suppose the number of labeled samples with label $j \in [k]$, denoted by m_j, and the number of unlabeled samples n satisfy $m_j \geq \tilde{\Omega}\left(1\right), \; n \geq \tilde{\Omega}\left(k\right)$. If rank condition $\Omega(d) \leq k \leq o(d^{3/2})$ holds, then Algorithm 1 outputs estimates $\hat{w}_j, \hat{a}_j$, satisfying w.h.p.* [4]

$$\min_{z\in\{-1,1\}} \|z\hat{a}_j - a_j\| \leq \epsilon_T, \quad |\hat{w}_j - w_j| \leq O(\epsilon_T/k), \quad j \in [k]. \tag{4}$$

Note that the number of labeled samples required is much smaller than the number of unlabeled samples, i.e., $\sum_{j\in[k]} m_j \ll n$. Thus, we provide efficient learning guarantees for overcomplete spherical Gaussian mixtures in the semi-supervised setting under a small number of labeled samples.

The recovery error ϵ_T involves two terms. One arises due to empirical estimation of 3rd order moment (given by $\tilde{O}\big(\sqrt{k/n}\big)$) and is inevitable. The other term is due to non-orthogonality of columns of factor matrices (given by $\tilde{O}\big(\sqrt{k}/d\big)$) which is an approximation error in recovery of the tensor components. Note that the latter goes to zero for large enough d since we have $k \leq o(d^{3/2})$.

Remark 1 (Random assumption). In the above learning result, we assume that the mixture components are uniformly i.i.d. drawn from unit d-dimensional sphere $\mathcal{S}^{d-1}$. This is a reasonable assumption for continuous models including the spherical Gaussian mixtures model described here. But, it is not appropriate for discrete models where the non-negativity assumptions on the entries of factor matrices are required. Moreover, the random assumption is provided for simplicity, while the original conditions for the guarantees of Algorithm 1 are deterministic.

***Unsupervised learning*:** In the unsupervised setting, there is no label information available to build the initialization vectors. Here, the initialization is performed by doing rank-1 SVD on random slices of the moment tensor proposed in Procedure 3. The settings and conditions for unsupervised learning are stated as follows:

Settings of Algorithm 1 in Theorem 2: Consider the same settings as in Theorem 1 for the input tensor and the number of iterations N. But, the initialization in each run of Algorithm 1 is performed by SVD-based technique in Procedure 3, with the number of initializations as $L \geq k^{\Omega(k^2/d^2)}$.

Theorem 2 (Unsupervised learning of spherical Gaussian mixtures)**.** *Assume the Algorithm settings mentioned above hold. Suppose the number of unlabeled samples n satisfies $n \geq \tilde{\Omega}\left(k^2\right)$. If rank condition $k = \Theta(d)$ holds, then the same guarantees as in Theorem 1 are satisfied. See* (4).

An algorithm for learning mixture of spherical Gaussians in the undercomplete setting is provided in [18], which is a moment-based technique combined with a whitening step. When $k = d$, the sample complexity in [18] scales as $n \geq \tilde{\Omega}(d^3)$. But, our tight tensor concentration analysis leads to the better sample complexity of $n \geq \tilde{\Omega}(d^2)$.

[4] Note that recovery of components is up to sign. This is because a third order tensor is unchanged if the sign along one of the modes is fixed and the signs along the other two modes are flipped.

References

[1] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky. Tensor Methods for Learning Latent Variable Models. *Available at arXiv:1210.7559*, Oct. 2012.

[2] A. Anandkumar, R. Ge, D. Hsu, and S. M. Kakade. A Tensor Spectral Approach to Learning Mixed Membership Community Models. In *Conference on Learning Theory (COLT)*, June 2013.

[3] L. De Lathauwer, J. Castaing, and J.-F. Cardoso. Fourth-order cumulant-based blind identification of underdetermined mixtures. *Signal Processing, IEEE Transactions on*, 55(6):2965–2973, 2007.

[4] P. Comon. Tensor decompositions. *Mathematics in Signal Processing V*, pages 1–24, 2002.

[5] M. A. O. Vasilescu and D. Terzopoulos. Multilinear subspace analysis of image ensembles. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–93. IEEE, 2003.

[6] J. Y. Zou, D. Hsu, D. C. Parkes, and R. P. Adams. Contrastive learning using spectral methods. In *Advances in Neural Information Processing Systems*, pages 2238–2246, 2013.

[7] F. Huang, U. N. Niranjan, M. Hakeem, and A. Anandkumar. Fast Detection of Overlapping Communities via Online Tensor Methods. *ArXiv 1309.0787*, Sept. 2013.

[8] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, 1994.

[9] T. Zhang and G. Golub. Rank-one approximation to high order tensors. *SIAM Journal on Matrix Analysis and Applications*, 23:534–550, 2001.

[10] Y. Bengio, A. Courville, and P. Vincent. Unsupervised feature learning and deep learning: A review and new perspectives. *arXiv preprint arXiv:1206.5538*, 2012.

[11] M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural computation*, 12(2): 337–365, 2000.

[12] D. Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006.

[13] S. Arora, R. Ge, and A. Moitra. New Algorithms for Learning Incoherent and Overcomplete Dictionaries. *ArXiv e-prints*, August 2013.

[14] A. Agarwal, A. Anandkumar, P. Jain, P. Netrapalli, and R. Tandon. Learning Sparsely Used Overcomplete Dictionaries via Alternating Minimization. *Available on arXiv:1310.7991*, Oct. 2013.

[15] B. McWilliams, D. Balduzzi, and J. Buhmann. Correlated random features for fast semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 440–448, 2013.

[16] Q. V. Le, A. Karpenko, J. Ngiam, and A. Y. Ng. ICA with Reconstruction Cost for Efficient Overcomplete Feature Learning. In *NIPS*, pages 1017–1025, 2011.

[17] A. Coates, H. Lee, and A. Y. Ng. An analysis of single-layer networks in unsupervised feature learning. *Journal of Machine Learning Research - Proceedings Track*, 15:215–223, 2011.

[18] D. Hsu and S. M. Kakade. Learning Mixtures of Spherical Gaussians: Moment Methods and Spectral Decompositions. *arXiv preprint arXiv:1206.5766*, 2012.