Score Function Features for Discriminative Learning: Matrix and Tensor Frameworks

Majid Janzamin* Hanie Sedghi† Anima Anandkumar‡

February 7, 2015

Abstract

Feature learning forms the cornerstone for tackling challenging learning problems in domains such as speech, computer vision and natural language processing. In this paper, we consider a novel class of matrix and tensor-valued features, which can be pre-trained using unlabeled samples. We present efficient algorithms for extracting discriminative information, given these pre-trained features and labeled samples for any related task. Our class of features are based on higher-order score functions, which capture local variations in the probability density function of the input. We establish a theoretical framework to characterize the nature of discriminative information that can be extracted from score-function features, when used in conjunction with labeled samples. We employ efficient spectral decomposition algorithms (on matrices and tensors) for extracting discriminative components. The advantage of employing tensor-valued features is that we can extract richer discriminative information in the form of an overcomplete representations. Thus, we present a novel framework for employing generative models of the input for discriminative learning.

Keywords: Feature learning, semi-supervised learning, self-taught learning, pre-training, score function, spectral decomposition methods, tensor methods.

1 Introduction

Having good features or representations of the input data is critical to achieving good performance in challenging machine learning tasks in domains such as speech, computer vision and natural language processing (Bengio et al., 2013). Traditionally, feature engineering relied on carefully hand-crafted features, tailored towards a specific task: a laborious and a time-consuming process. Instead, the recent trend has been to automatically learn good features through various frameworks such as deep learning (Bengio et al., 2013), sparse coding (Raina et al., 2007), independent component analysis (ICA) (Le et al., 2011), Fisher kernels (Jaakkola et al., 1999), and so on. These approaches are unsupervised and can thus exploit the vast amounts of unlabeled samples, typically present in these domains.

A good feature representation incorporates important prior knowledge about the input, typically through a probabilistic model. In almost every conceivable scenario, the probabilistic model needs to

*University of California, Irvine. Email: mjanzami@uci.edu

†University of Southern California. Email: hsedghi@usc.edu

‡University of California, Irvine. Email: a.anandkumar@uci.edu

incorporate latent variables to fit the input data. These latent factors can be important explanatory variables for classification tasks associated with the input. Thus, incorporating generative models of the input can hugely boost the performance of discriminative tasks.

Many approaches to feature learning focus on unsupervised learning, as described above. The hypothesis behind employing unsupervised learning is that the input distribution is related to the associative model between the input and the label of a given task, which is reasonable to expect in most scenarios. When the distribution of the unlabeled samples, employed for feature learning, is the same as the labeled ones, we have the framework of *semi-supervised* learning. A more general framework, is the so-called *self-taught* learning, where the distribution of unlabeled samples is different, but related to the labeled ones (Raina et al., 2007). Variants of these frameworks include transfer learning, domain adaptation and multi-task learning (Bengio, 2011), and involve labeled datasets for related tasks. These frameworks have been of extensive interest to the machine learning community, mainly due to the scarcity of labeled samples for many challenging tasks. For instance, in computer vision, we have a huge corpus of unlabeled images, but a more limited set of labeled ones. In natural language processing, it is extremely laborious to annotate the text with syntactic and semantic parses, but we have access to unlimited amounts of unlabeled text.

It has been postulated that humans mostly learn in an unsupervised manner (Raina et al., 2007), gathering "common-sense" or "general-purpose" knowledge, without worrying about any specific goals. Indeed, when faced with a specific task, humans can quickly and easily extract relevant information from the accrued general-purpose knowledge. Can we design machines with similar capabilities? Can we design algorithms which succinctly summarize information in unlabeled samples as general-purpose features? When given a specific task, can we efficiently extract relevant information from general-purpose features? Can we provide theoretical guarantees for such algorithms? These are indeed challenging questions, and we provide some concrete answers in this paper.

1.1 Summary of Results

In this paper, we consider a class of matrix and tensor-valued "general-purpose" features, pre-trained using unlabeled samples. We assume that the labels are not present at the time of feature learning. When presented with labeled samples, we leverage these pre-trained features to extract discriminative information using efficient spectral decomposition algorithms. As a main contribution, we provide theoretical guarantees on the nature of discriminative information that can be extracted with our approach.

We consider the class of features based on higher-order score functions of the input, which involve higher-order derivatives of the probability density function (pdf). These functions capture "local manifold structure" of the pdf. While the first-order score function is a vector (assuming a vector input), the higher-order functions are matrices and tensors, and thus capture richer information about the input distribution. Having access to these matrix and tensor-valued features allows to extract better discriminative information, and we characterize its precise nature in this work.

Given score-function features and labeled samples, we extract discriminative information based on the method of moments. We construct cross-moments involving the labels and the input score features. Our main theoretical result is that these moments are equal to the expected derivatives of the label, as a function of the input or some model parameters. In other words, these moments capture variations of the label function, and are therefore informative for discriminative tasks.

We employ spectral decomposition algorithms to find succinct representations of the moment

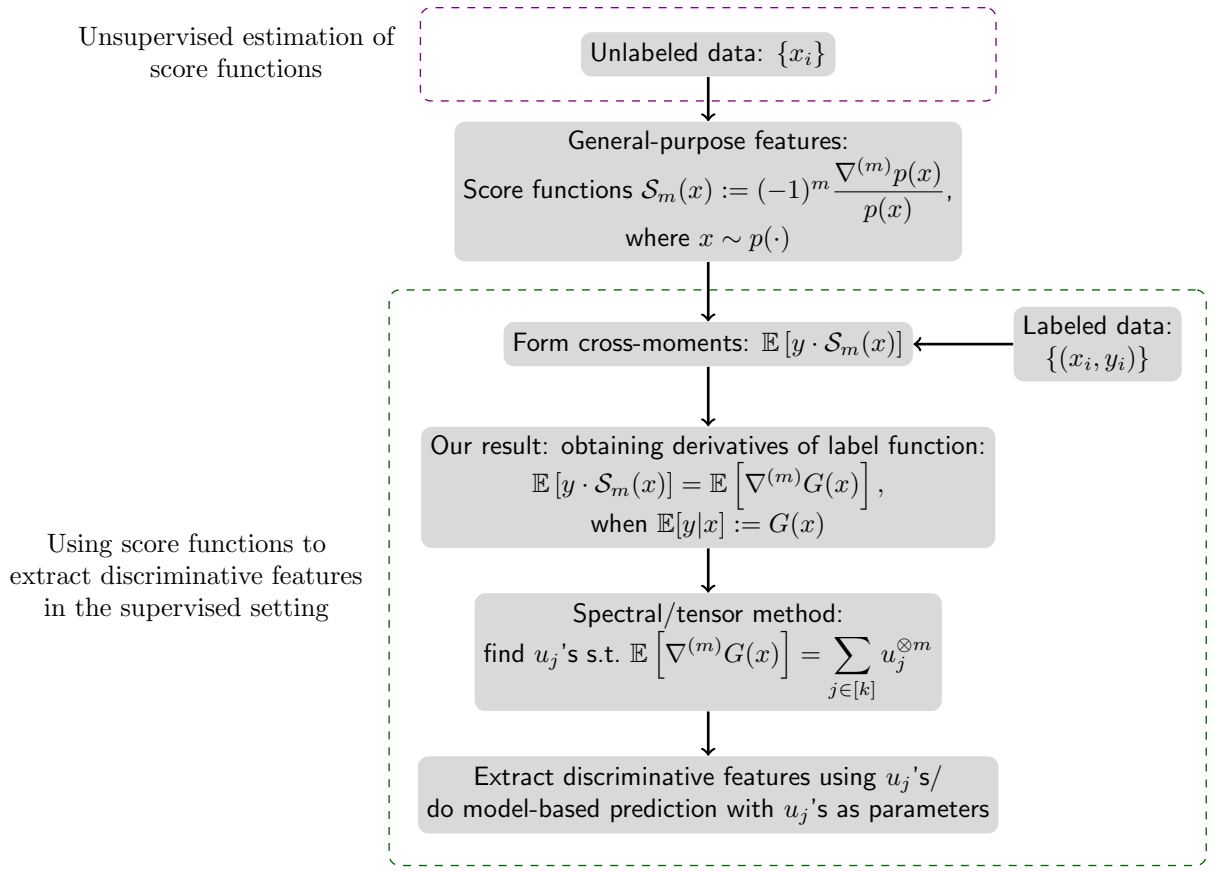

Figure 1: Overview of the proposed framework of using the general-purpose features to generate discriminative features through spectral methods.

matrices/tensors. These algorithms are fast and embarrassingly parallel. See (Anandkumar et al., 2014a,b,c) for details, where we have developed and analyzed efficient tensor decomposition algorithms (along with our collaborators). The advantage of the tensor methods is that they do not suffer from spurious local optima, compared to typical non-convex problems such as expectation maximization or backpropagation in neural networks. Moreover, we can construct overcomplete representations for tensors, where the number of components in the representation can exceed the data dimensionality. It has been argued that having overcomplete representations is crucial to getting good classification performance (Coates et al., 2011). Thus, we can leverage the latest advances in spectral methods for efficient extraction of discriminative information from moment tensors.

In our framework, the label can be a scalar, a vector, a matrix or even a tensor, and it can either be continuous or discrete. We can therefore handle a variety of regression and classification settings such as multi-task, multi-class, and structured prediction problems. Thus, we present a unified and an efficient end-to-end framework for extracting discriminative information from pre-trained features. An overview of the entire framework is presented in Figure 1 which is fully explained later in Section 1.2.

We now provide some important observations below.

Are the expected label function derivatives informative? Our analysis characterizes the discriminative information we can extract from score function features. As described above, we prove that the cross-moments between the label and the score function features are equal to the expected derivative of the label as a function of the input or model parameters. But when are these expected label derivatives informative? Indeed, in trivial cases, where the derivatives of the label function vanish over the support of the input distribution, these moments carry no information. However, such cases are pathological, since then, either there is no variation in the label function or the input distribution is nearly degenerate. Another possibility is that a certain derivative vanishes, when averaged over the input distribution, even though it is not zero everywhere. If this is the case, then the next derivative cannot be averaged out to zero, and will thus carry information about the variations of the label function. Thus, in practical scenarios, the cross-moments contain useful discriminative information. In fact, for many discriminative models which are challenging to learn, such as multi-layer neural networks and mixtures of classifiers, we establish that these moments have an intimate relationship with the parameters of the discriminative model in subsequent works (Sedghi and Anandkumar, 2014a,b). Spectral decomposition of the moments provably recovers the model parameters. These are the first results for guaranteed learning of many challenging discriminative latent variable models.

Contrasting with previous approaches: We now contrast our approach to previous approaches for incorporating generative models in discriminative tasks. Typically, these approaches directly feed the pre-trained features to a classifier. For example, in the Fisher kernel framework, the Fisher score features are fed to a kernel classifier (Jaakkola et al., 1999). The reasoning behind this is that the features obtained from unsupervised learning have information about all the classes, and the task of finding class-specific differences in the learnt representation is left to the classifier. However, in practice, this may not be the case, and a common complaint is that these generative features are not discriminative for the task at hand. Previous solutions have prescribed joint training discriminative features using labeled samples, in conjunction with unlabeled samples (Mairal et al., 2009; Maaten, 2011; Wang et al., 2013). However, the resulting optimization problems are complex and expensive to run, may not converge to good solutions, and have to be re-trained for each new task. We present an alternative approach to extract discriminative features using efficient spectral decomposition algorithms on moment matrices and tensors. These methods are light weight and fast, and we theoretically quantify the nature of discriminative features they can extract. These discriminative features can then be fed into the classification pipeline. Thus, the advantage of our approach is that we can quickly generate discriminative features for new classification tasks without going through the laborious process of re-training for new features.

We now contrast our approach with previous moment-based approaches for discriminative learning, which consider moments between the label and raw input, e.g. (Karampatziakis and Mineiro, 2014). Such methods have no theoretical guarantees. In contrast, we construct cross-moments between the label and the score function features. We show that using score function features is crucial to mining discriminative information with provable guarantees.

Extension to self-taught learning: We have so far described our framework under the semi-supervised setting, where the unlabeled and labeled samples have the same input distribution. We can also handle the framework of self-taught learning, where the two distributions are related but may not be the same. We prescribe some simple pre-processing to transfer the parameters and to

re-estimate the score function features for the input of the labeled data set. Such parameter transfer frameworks have been considered before, e.g. (Raina et al., 2007), except here we present a general latent-variable framework and focus on transferring parameters for computing score functions, since we require them for subsequent operations. Our framework can also be applied to scenarios where we have different input sources with different distributions, but the classification task is the same, and thus, the associative model between the label and the input is fixed. Consider for instance, crowdsourcing applications, where the same task is presented to different groups of individuals. In our approach, we can then construct different score function features for different input sources and the different cross-moments provide information about the variations in the label function, averaged over different input distributions. We can thus leverage the diversity of different input sources for improved performance on common tasks. Thus, our approach is applicable in many challenging practical scenarios.

1.2 Overview of our framework

In this section, we elaborate on the end-to-end framework presented in Figure 1.

Background: The problem of supervised learning consists of learning a predictor, given labeled training samples $\{(x_i, y_i)\}$ with input x_i and corresponding label y_i. Classical frameworks such as SVMs are purely discriminative since they make no distributional assumptions. However, when labeled data is limited and classification tasks are challenging, incorporating distributional information can improve performance. In an associative model-based framework, we posit a conditional distribution for the label given the input $p(y|x)$. However, learning this model is challenging, since maximum-likelihood estimation of $p(y|x)$ is non-convex and NP-hard to solve in general, especially if it involves hidden variables (e.g., associative mixtures, multi-layer neural networks). In addition, incorporating a generative model for input x often leads to improved discriminative performance.

Label-function derivatives are discriminative: Our main focus in this work is to extract useful information about $p(y|x)$ without attempting to learn it in its entirety. In particular, we extract information about the local variations of conditional distribution $p(y|x)$, as the input x (or some model parameter) is changed. For the classification setting, it suffices to consider[1] $\mathbb{E}[y|x] := G(x)$. In this paper, we present mechanisms to estimate its expected higher order derivatives[2]

$$\mathbb{E}[\nabla_x^{(m)} G(x)],\ m \geq 1, \tag{1}$$

where $\nabla_x^{(m)}$ denotes the m-th order derivative operator w.r.t. variable x. By having access to expected derivatives of the label function $G(x)$ in (1), we gain an understanding of how the label y varies as we change the input x locally, which is valuable discriminative information.

Score functions yield label-function derivatives: One of the main contributions of this paper is to obtain these expected derivatives in (1) using features denoted by $\mathcal{S}_m(x)$, for $m \geq 1$ (learnt

[1] In the classification setting, powers of y, e.g., y^2 contain no additional information, and hence, all the information of the associative model is in $\mathbb{E}[y|x] := G(x)$. However, in the regression setting, we can compute additional functions, e.g., $\mathbb{E}[\nabla^{(m)} H(x)]$, where $\mathbb{E}[y^2|x] := H(x)$. Our approach can also compute these derivatives.

[2] Note that since we are computing the expected derivatives, we also assume a distribution for the input x.

from unlabeled samples) and the labeled data. In particular, we form the cross-moment between the label y and the features $\mathcal{S}_m(x)$, and show that they yield the derivatives as[3]

$$\mathbb{E}[y \cdot \mathcal{S}_m(x)] = \mathbb{E}[\nabla^{(m)} G(x)], \quad \text{when } \mathbb{E}[y|x] := G(x). \tag{2}$$

We establish a simple form for features $\mathcal{S}_m(x)$, based on the derivatives of the probability density function $p(\cdot)$ of the input x as

$$\mathcal{S}_m(x) = (-1)^m \frac{\nabla^{(m)} p(x)}{p(x)}, \quad \text{when } x \sim p(\cdot). \tag{3}$$

In fact, we show that the feature $\mathcal{S}_m(x)$ defined above is a function of higher order score functions $\nabla_x^{(n)} \log p(x)$ with $n \leq m$, and we derive an explicit relationship between them. This is basically why we also call these features as (higher order) score functions. Note that the features $\mathcal{S}_m(x)$ can be learnt using unlabeled samples, and we term them as general-purpose features since they can be applied to any labeled dataset, once they are estimated. Note the features $\mathcal{S}_m(x)$ can be vectors, matrices or tensors, depending on m, for multi-variate x. The choice of order m depends on the particular setup: a higher m yields more information (in the form of higher order derivatives) but requires more samples to compute the empirical moments accurately.

We then extend the framework to parametric setting, where we obtain derivatives $\mathbb{E}[\nabla_\theta^{(m)} G(x;\theta)]$ with respect to some model parameter θ when $\mathbb{E}[y|x;\theta] := G(x;\theta)$. These are obtained using general-purpose features denoted by $\mathcal{S}_m(x;\theta)$ which is a function of higher order Fisher score functions $\nabla_\theta^{(n)} \log p(x;\theta)$ with $n \leq m$. Note that by using the parametric framework we can now incorporate discrete input x, while this is not possible with the previous framework.

Spectral decomposition of derivative matrices/tensors: Having obtained the derivatives $\mathbb{E}[\nabla^{(m)} G(x)]$ (which are matrices or tensors), we then find efficient representations using spectral/tensor decomposition methods. In particular, we find vectors u_j such that

$$\mathbb{E}[\nabla^{(m)} G(x)] = \sum_{j \in [k]} \overbrace{u_j \otimes u_j \otimes \cdots \otimes u_j}^{m \text{ times}}, \tag{4}$$

where $\otimes$ refers to the tensor product notation. Note that since the higher order derivative is a symmetric matrix/tensor, the decomposition is also symmetric. Thus, we decompose the matrix/tensor at hand into sum of rank-1 components, and in the matrix case, this reduces to computing the SVD. In the case of a tensor, the above decomposition is termed as CP decomposition (Kruskal, 1977). In a series of works (Anandkumar et al., 2014a,b,c), we have presented efficient algorithms for obtaining (4), and analyzed their performance in detail.

The matrix/tensor in hand is decomposed into a sum of k rank-1 components. Unlike matrices, for tensors, the rank parameter k can be larger than the dimension. Therefore, the decomposition problems falls in to two different regimes. One is the undercomplete regime: where k is less than the dimension, and the overcomplete one, where it is not. The undercomplete regime leads to dimensionality reduction, while the overcomplete regime results in richer representation.

Once we obtain components u_j, we then have several options to perform further processing. We can extract discriminative features such as $\sigma(u_j^\top x)$, using some non-linear function $\sigma(\cdot)$, as

[3] We drop subscript x in the derivative operator $\nabla_x^{(m)}$ saying $\nabla^{(m)}$ when there is no ambiguity.

performed in some of the earlier works, e.g., (Karampatziakis and Mineiro, 2014). Alternatively, we can perform model-based prediction and incorporate u_j's as parameters of a discriminative model. In a subsequent paper, we show that u_j's correspond to significant parameters of many challenging discriminative models such as multi-layer feedforward neural networks and mixture of classifiers, under the *realizable* setting.

Extension to self-taught learning: The results presented so far assume the semi-supervised setting, where the unlabeled samples $\{\tilde{x}_i\}$ used to estimate the score functions are drawn from the same distributions as the input $\{x_i\}$ of the labeled samples $\{(x_i, y_i)\}$. We present simple mechanisms to extend to the self-taught setting, where the distributions of $\{\tilde{x}_i\}$ and $\{x_i\}$ are related, but not the same. We assume latent-variable models for $\tilde{x}$ and x, e.g., sparse coding, independent component analysis (ICA), mixture models, restricted Boltzmann machine (RBM), and so on. We assume that the conditional distributions $p(\tilde{x}|\tilde{h})$ and $p(x|h)$, given the corresponding latent variables $\tilde{h}$ and h are the same. This is reasonable since the unlabeled samples $\{\tilde{x}_i\}$ are usually "rich" enough to cover all the elements. For example, in the sparse coding setting, we assume that all the dictionary elements can be learnt through $\{\tilde{x}_i\}$, which is assumed in a number of previous works, e.g (Raina et al., 2007; Zhang et al., 2008). Under this assumption, estimating the score function for new samples $\{x_i\}$ is relatively straightforward, since we can transfer the estimated conditional distribution $p(\tilde{x}|\tilde{h})$ (using unlabeled samples $\{\tilde{x}_i\}$) as the estimation of $p(x|h)$, and we can re-estimate the marginal distribution $p(h)$ easily. Thus, the use of score functions allows for easy transfer of information under the self-taught framework. The rest of the steps can proceed as before.

1.3 Related Work

Due to limitation of labeled samples in many domains such as computer vision and natural language processing, the frameworks of domain adaptation, semi-supervised, transfer and multi-task learning have been popular in domains such as NLP (Blitzer et al., 2006), computer vision (Quattoni et al., 2007; Yang et al., 2007; Hoffman et al., 2013), and so on. We now list the various approaches below.

Non-probabilistic approaches: Semi-supervised learning has been extensively studied via non-probabilistic approaches. Most works attempt to assign labels to the unlabeled samples, e.g. (Ando and Zhang, 2005), either through bootstrapping (Yarowsky, 1995; Blum and Mitchell, 1998), or by learning good functional structures (Szummer and Jaakkola, 2002; Ando and Zhang, 2005). Related to semi-supervised learning is the problem of domain adaptation, where the source domain has labeled datasets on which classifiers have been trained, e.g. (Ben-david et al., 2006; Huang et al., 2006; Mansour et al., 2009; Blitzer et al., 2009; Ben-David et al., 2010; Gong et al., 2013), and there may or may not be labeled samples in the target domain. The main difference is that in this paper, we consider the source domain to have only unlabeled samples, and we pre-train general-purpose features, which are not tied to any specific task.

A number of recent works have investigated transfer learning using deep neural networks, e.g. (Bengio, 2011; Socher et al., 2013; Zeiler and Fergus, 2013; Sermanet et al., 2013; Donahue et al., 2014; Yosinski et al., 2014) and obtain state-of-art performance on various tasks.

Probabilistic approaches (Fisher kernels): A number of works explore probabilistic approaches to semi-supervised and transfer learning, where they learn a generative model on the

input and use the features from the model for discriminative tasks. Fisher kernels fall into this category, where the Fisher score is pre-trained using unlabeled samples, and then used for discriminative tasks through a kernel classifier (Jaakkola et al., 1999). Our paper proposes higher order extensions of the Fisher score, which yield matrix and tensor score features, and we argue that they are much more informative for discriminative learning, since they yield higher order derivatives of the label function. Moreover, we provide a different mechanism to utilize the score features: instead of directly feeding the score features to the classifier, we form cross-moments between the score features and the labels, and extract discriminative features through spectral decomposition. These discriminative features can then be used in the standard classification frameworks. This allows us to overcome a common complaint that the pre-trained features, by themselves may not be discriminative for a particular task. Instead there have been attempts to construct discriminative features from generative models using labeled samples (Maaten, 2011; Mairal et al., 2009; Wang et al., 2013). However, this is time-consuming since the discriminative features need to be re-trained for every new task.

Probabilistic approaches (latent representations): A number of works learn latent representations to obtain higher level features for classification tasks. Popular models include sparse coding (Raina et al., 2007), independent component analysis (ICA) (Le et al., 2011), and restricted Boltzmann machines (RBM) (Swersky et al., 2011). It has been argued that having overcomplete latent representations, where the latent dimensionality exceeds the observed dimensionality, is crucial to getting good classification performance (Coates et al., 2011). We also note here that the authors and others have been involved in developing guaranteed and efficient algorithms for unsupervised learning of latent representations such as mixture models, ICA (Anandkumar et al., 2014a,c), sparse coding (Agarwal et al., 2014; Arora et al., 2014), and deep representations (Arora et al., 2013). There have been various other probabilistic frameworks for information transfer. Raina et al. (2006) consider learning priors over parameters in one domain and using it in the other domain in the Bayesian setting. McCallum et al. (2006) argue that incorporating generative models for $p(x|y)$ acts as a regularizer and leads to improved performance.

Raina et al. (2007) introduce the framework of self-taught learning, where the distribution of unlabeled samples is related but not the same as the input for labeled samples. They employ a sparse coding model for the input, and assume that both the datasets share the same dictionary elements, and only the distribution of the coefficients which combine the dictionary elements are different. They learn the dictionary from the unlabeled samples, and then use it to decode the input of the labeled samples. The decoded dictionary coefficients are the features which are fed into SVM for classification. In this paper, we provide an alternative framework for learning features in a self-taught framework by transferring parameters for score function estimation in the new domain.

Probabilistic approaches (score matching): We now review the score matching approaches for learning probabilistic models. These methods estimate parameters using score matching criteria rather than a likelihood-based one. Since we utilize score function features, it makes sense to estimate parameters based on the score matching criterion. Hyvärinen (2005) introduce the criterion of minimizing the Fisher divergence, which is the expected square loss between the model score function and the data score function. Note that the score function $\nabla_x \log p(x)$ does not involve the partition function, which is typically intractable to compute, and is thus tractable in scenarios where the likelihood cannot be computed. Lyu (2009) further provide a nice theoretical result that

the score matching criterion is more robust than maximum likelihood in the presence of noise. Swersky et al. (2011) apply the score matching framework for learning RBMs, and extract features for classification, which show superior performance compared to auto-encoders.

Probabilistic approaches (regularized auto-encoders): Another class of approaches for feature learning are the class of regularized auto-encoders. An auto-encoder maps the input to a code through an encoder function, and then maps back using a decoder function. The training criterion is to minimize the reconstruction loss along with a regularization penalty. They have been employed for pre-training neural networks. See Bengio et al. (2013) for a review. Vincent (2011) established that a special case of denoising auto-encoder reduces to a score matching criterion for an appropriately chosen energy function. Alain and Bengio (2012) establish that the denoising auto-encoders estimate the score function (of first and second order), in the limit as the noise variance goes to zero. In this paper, we argue that the score function are the appropriate features to learn for transferring information to various related tasks. Thus, we can employ auto-encoders for estimating the score functions.

Stein's identity: Our results establishing that the higher order score functions in (3) yield derivatives of the label function in (2) is novel. The special case of the first derivative reduces to Stein's identity in statistics (Stein, 1986; Ley and Swan, 2013), which is essentially obtained through integration by parts. We construct higher order score functions in a recursive manner, and then show that it reduces to the simple form in (3).

Orthogonal polynomials: For the special case of Gaussian input $x \sim \mathcal{N}(0, I)$, we show that the score functions in (3) reduce to the familiar multivariate *Hermite* polynomials, which are orthogonal polynomials for the Gaussian distribution (Grad, 1949; Holmquist, 1996). However, for general distributions, the score functions need not be polynomials.

2 Problem Formulation

In this section, we first review different learning settings; in particular semi-supervised and self-taught learning settings which we consider in this paper. Then, we state the main assumptions to establish our theoretical guarantees.

2.1 Learning settings and assumptions

First, we describe different learning settings and clarify the differences between them by giving some image classification examples, although the framework is general and applicable to other domains as well.

Semi-supervised learning: In the semi-supervised setting, we have both labeled samples $\{(x_i, y_i)\}$ and unlabeled samples $\{\tilde{x}_i\}$ in the training set. For instance, consider a set of images containing cats and dogs, where a fraction of them are labeled with the binary output y_i specifying if the image contains cat or dog. The main assumption is that the input in the labeled and unlabeled datasets have the same distribution.

Multi-task learning: In the multi-task setting, we have labeled samples $\{(x_i, y_i)\}$ and $\{(x_i, \tilde{y}_i)\}$ where the same set of inputs x_i are labeled for two different tasks. For instance, consider a set of images containing cats, dogs, monkeys and humans where the first task is to label them as {human, not human}, while the other task is to label them as {cat, not cat}.

Transfer learning: In transfer learning, we want to exploit the labeled information of one task to perform other related tasks. This is also known as knowledge transfer since the goal is to transfer the knowledge gained in analyzing one task to another. Concretely, we have access to labeled samples $\{(x_i, y_i)\}$ and $\{(\tilde{x}_i, \tilde{y}_i)\}$ of two related tasks. For instance, imagine a set of images $\{(x_i, y_i)\}$ containing cats and dogs, another set of images $\{(\tilde{x}_i, \tilde{y}_i)\}$ containing monkeys and humans, each of them with the corresponding labels. The goal is to use the information in source labeled data $\{(\tilde{x}_i, \tilde{y}_i)\}$ for classifying new samples of target data x_i.

Self-taught learning: In self-taught learning, we further assume that the related dataset $\{\tilde{x}_i\}$ in the transfer learning setting does not have labels. Concretely, we have labeled samples of the original task as $\{(x_i, y_i)\}$, and other unlabeled data $\{\tilde{x}_i\}$ from a related distribution. For instance, consider a set of images $\{(x_i, y_i)\}$ containing cats and dogs with labels, and assume we have lots of unlabeled images $\{\tilde{x}_i\}$ which can be any type of images, say downloaded from internet.

In this paper, we focus on semi-supervised and self-taught learning settings, and other related learning frameworks mentioned above are also treated in these two settings, i.e. we consider first training score function features from input, without using labels, and then use the labels in conjunction with score function features.

We first give general assumptions we use in both semi-supervised and self-taught settings. Then, we state additional assumptions for the self-taught learning framework.

Probabilistic input x: We assume a generative model on input x where it is randomly drawn from some continuous distribution $p(x)$ satisfying mild regularity condition[4]. It is known that incorporating such generative assumption on x usually results in better performance on discriminative tasks.

Probabilistic output y: We further assume a probabilistic model on output (label) y where it is randomly drawn according to some distribution $p(y|x)$ given input x, satisfying mild regularity conditions[5]. In our framework, the output (label) can be scalar, vector, matrix or even tensor, and it can be continuous or discrete, and we can handle a variety of regression and classification settings such as multi-class, multi-label, and structured prediction problems.

Assumptions under the self-taught learning framework: We now state the assumptions that tie the distribution of unlabeled samples with labeled ones. We consider latent-variable models for $\tilde{x}$ and x, e.g., sparse coding, independent component analysis, mixture models, restricted Boltzmann machine, and so on. We assume that the conditional distributions $p(\tilde{x}|\tilde{h})$ and $p(x|h)$, given the corresponding latent variables $\tilde{h}$ and h, are the same. This is reasonable since the unlabeled samples $\{\tilde{x}_i\}$ are usually "rich" enough to cover all the elements. For example, in the sparse

[4]The exact form of regularity conditions are provided in Theorem 6.

[5]The exact form of regularity conditions are provided in Theorem 6.

coding setting, we assume that all the dictionary elements can be learnt through $\{\tilde{x}_i\}$, which is assumed in a number of previous works, e.g (Raina et al., 2007; Zhang et al., 2008). In particular, in the image classification task mentioned earlier, consider a set of images $\{x_i\}$ containing cats and dogs, and assume we also have lots of unlabeled images $\{\tilde{x}_i\}$, which can be any type of images, say downloaded from internet. Under the sparse coding model, the observed images are the result of a sparse combination of dictionary elements. The coefficients for combining the dictionary elements correspond to hidden variables h and $\tilde{h}$ for the two datasets. It is reasonable to expect that the two datasets share the same dictionary elements, i.e., once the coefficients are fixed, it is reasonable to assume that the conditional probability of drawing the observed pixels in images is the same for both labeled images (including only cats and dogs) and unlabeled images (including all random images). But the marginal probability of the coefficients, denoted by $p(h)$ and $p(\tilde{h})$, will be different, since they represent two different data sets. In Section 5.4, we show that this assumption leads to simple mechanisms to transfer knowledge about the score function to the new dataset.

2.2 A snapshot of our approach

We now succinctly explain our general framework and state how the above assumptions are involved in our setting. In general, semi-supervised learning considers access to both labeled and unlabeled samples in the training data set. When the labeled data is limited and the learning task mostly relies on the unlabeled data (which is the case in many applications), the task is more challenging, and assuming distributional assumptions as above can improve the performance of learning task.

Note that maximum likelihood estimation of $p(y|x)$ is non-convex and NP-hard in general, and our goal in this work is to extract useful information from $p(y|x)$ without entirely recovering it. We extract information about the local variations of conditional distribution $p(y|x)$, when input x is changed. In particular, for the classification task, we extract useful information from derivatives (including local variation information) of the first order conditional moment of output y given input x denoted as [6]

$$\mathbb{E}[y|x] := G(x).$$

More concretely, we provide mechanisms to compute $\mathbb{E}[\nabla_x^{(m)} G(x)]$, where $\nabla_x^{(m)}$ denotes the m-th order derivative operator w.r.t. variable x. We usually limit to a small m, e.g., $m = 3$.

Note that in computing $\mathbb{E}[\nabla_x^{(m)} G(x)]$, we also apply the expectation over input x, and thus, the generative model of x comes into the picture. This derivative is a vector/matrix/tensor, depending on m. Finally, we decompose this derivative matrix/tensor to rank-1 components to obtain discriminative features.

Now the main question is how to estimate the derivatives $\mathbb{E}[\nabla^{(m)} G(x)]$. One of our main contributions in this work is to show that the *score functions* yield such label-function derivatives. For $m = 1$, it is known that the (first order) score function yields the derivative as

$$-\mathbb{E}[y \cdot \nabla \log p(x)] = \mathbb{E}[\nabla G(x)], \quad \text{when } \mathbb{E}[y|x] := G(x),$$

where $-\nabla \log p(x)$ is the (usual first order) score function. More generally, we introduce (m-th

[6] In the classification setting, powers of y, e.g., y^2 contain no additional information, and hence, all the information of the associative model is in the first order conditional moment $\mathbb{E}[y|x] := G(x)$. However, in the regression setting, we can involve higher order conditional moments, e.g., $\mathbb{E}[y^2|x] := H(x)$. Our approach can also compute the derivatives of these higher order conditional moments.

order) score functions denoted by $\mathcal{S}_m(x)$ which also yield the desired derivatives as

$$\mathbb{E}[y \cdot \mathcal{S}_m(x)] = \mathbb{E}[\nabla^{(m)} G(x)].$$

The estimation of score functions $\mathcal{S}_m(x)$ is performed in an unsupervised manner using unlabeled samples $\{x_i\}$; see Section 5 for the details.

Computing cross-moment between label y and score function $\mathcal{S}_m(x)$: After estimating the score function, we form the cross moment $\mathbb{E}[y \cdot \mathcal{S}_m(x)]$ between labels y and (higher order) score functions $\mathcal{S}_m(x)$ using labeled data. Here, we assume that we can compute the *exact* form of these moments. Perturbation analysis of the computed empirical moment depends on the setting of the probabilistic models on input x and output y which is the investigation direction in the subsequent works applying the proposed framework in this paper to specific learning tasks.

3 Score Functions Yield Label Function Derivatives: Informal Results

In this section, we provide one of our main contributions in this paper, which is showing that higher order score functions yield *differential operators*. Here, we present informal statements of the main result, along with with detailed discussions, while the formal lemmas and theorems are stated in Section 6.

3.1 First order score functions

We first review the existing results on score functions and their properties as yielding first order differential operators. The score function is the derivative of the logarithm of density function. The derivative is w.r.t. either variable x or the parameters of the distribution. The latter one is usually called *Fisher score* in the literature. We provide the properties of score functions as yielding differential operators in both cases.

Stein identity: We start with Stein's identity (or Stein's lemma) which is the building block of our work. The original version of Stein's lemma is for the Gaussian distribution. For a standard random Gaussian vector $x \sim \mathcal{N}(0, I_{d_x})$, it states that for all functions[7] $G(x)$ satisfying mild regularity conditions, we have (Stein, 1972)

$$\mathbb{E}[G(x) \otimes x] = \mathbb{E}[\nabla_x G(x)], \tag{5}$$

where $\otimes$ denotes the tensor product (note that if $G(x)$ is a vector (or a scalar), the notation $G(x) \otimes x$ is equivalent to $G(x)x^\top$), and ∇_x denotes the usual gradient operator w.r.t. variable x. For details on tensor and gradient notations, see Section 6.1.

The above result for the Gaussian distribution can be generalized to other random distributions as follows. For a random vector $x \in \mathbb{R}^{d_x}$, let $p(x)$ and $\nabla_x \log p(x)$ respectively denote the joint density function and the corresponding *score function*. Then, under some mild regularity conditions, for all functions $G(x)$, we have

$$\mathbb{E}[G(x) \otimes \nabla_x \log p(x)] = -\mathbb{E}[\nabla_x G(x)]. \tag{6}$$

[7] We consider general tensor valued functions $G(x) : \mathbb{R}^{d_x} \to \bigotimes^r \mathbb{R}^{d_y}$. For details on tensor notation, see Section 6.1.

See Lemma 4 for a formal statement of this result including description of regularity conditions. Note that for the Gaussian random vector $x \sim \mathcal{N}(0, I_{d_x})$ with joint density function $p(x) = \frac{1}{\left(\sqrt{2\pi}\right)^{d_x}} e^{-\|x\|^2/2}$, the score function is $\nabla_x \log p(x) = -x$, and the above equality reduces to the special case in (5).

Parametric Stein identity: A parametric approach to Stein's lemma is introduced in (Ley and Swan, 2013), which we review here. We first define some notations. Let Θ denote the set of parameters such that for $\theta \in \Theta$, $p(x;\theta)$ be a valid θ-parametric probability density function. In addition, consider any $\theta_0 \in \Theta$ for which specific regularity conditions for a class of functions $G(x;\theta)$ hold over a neighborhood of θ_0. See Definition 2 for a detailed description of the regularity conditions, which basically allows us to change the order of derivative w.r.t. to θ and integration on x. We are now ready to state the informal parametric Stein's identity as follows.

For a random vector $x \in \mathbb{R}^{d_x}$, let $p(x;\theta)$ and $\nabla_\theta \log p(x;\theta)$ respectively denote the joint θ-parametric density function and the corresponding *parametric score function*. Then, for all functions $G(x;\theta)$ satisfying the above (mild) regularity conditions, we have

$$\mathbb{E}[G(x;\theta) \otimes \nabla_\theta \log p(x;\theta)] = -\mathbb{E}[\nabla_\theta G(x;\theta)] \quad \text{at } \theta = \theta_0. \tag{7}$$

See Theorem 5 for a formal statement of this result including description of regularity conditions.

Contrasting the above Stein identities: We provide Stein's identity and the parametric form in (6) and (7), respectively. The two identities mainly differ in taking the derivative w.r.t. either the variable x or the parameter θ. We now provide an example assuming the mean vector as the parameter of the distribution to elaborate the parametric result and contrast it with the original version, where we also see how the two forms are closely related in this case.

Consider the random Gaussian vector $x \in \mathbb{R}^d$ with mean parameter μ and known identity covariance matrix. Hence,

$$p(x;\mu) = \frac{1}{\left(\sqrt{2\pi}\right)^{d}} e^{-(x-\mu)^\top (x-\mu)/2}$$

denotes the corresponding joint parametric density function with mean parameter $\theta = \mu$. Thus, the parametric score function is $\nabla_\mu \log p(x;\mu) = x - \mu$ and applying parametric Stein's identity in (7) to functions of the form $G(x;\mu) = G_0(x-\mu)$ leads to

$$\mathbb{E}[G_0(x-\mu_0) \otimes (x-\mu_0)] = -\mathbb{E}[\nabla_\mu G_0(x-\mu)|_{\mu=\mu_0}].$$

Setting $\mu_0 = 0$, this identity is the same as the original Stein's identity in (5) since $\nabla_\mu G_0(x-\mu) = -\nabla_x G_0(x-\mu)$.

Note that this relation is true for any distribution and not just Gaussian, i.e., for the joint parametric density function $p(x;\mu)$ with mean parameter μ, we have the Stein identities from (6) and (7) respectively as

$$\begin{aligned}
\mathbb{E}[G_0(x-\mu_0) \otimes \nabla_x \log p(x;\mu)] &= -\mathbb{E}[\nabla_x G_0(x-\mu_0)], \\
\mathbb{E}[G_0(x-\mu) \otimes \nabla_\mu \log p(x;\mu)] &= -\mathbb{E}[\nabla_\mu G_0(x-\mu)] \quad \text{at } \mu = \mu_0,
\end{aligned}$$

which are the same since $\nabla_\mu G_0(x-\mu) = -\nabla_x G_0(x-\mu)$ and $\nabla_x \log p(x;\mu) = -\nabla_\mu \log p(x;\mu)$.

3.2 Higher order score functions

The first order score functions and their properties as yielding differential operators are reviewed in the previous section. Such differential property is called the Stein's identity. In this section, we generalize such differential properties to higher orders by introducing higher order score functions as matrices and tensors.

3.2.1 Our contribution: higher order extensions to Stein's identities

Let $p(x)$ denote the joint probability density function of random vector $x \in \mathbb{R}^d$. We denote $\mathcal{S}_m(x)$ as the m-th order score function, which we establish is given by [8]

$$\mathcal{S}_m(x) = (-1)^m \frac{\nabla_x^{(m)} p(x)}{p(x)}, \tag{8}$$

where $\nabla_x^{(m)}$ denotes the m-th order derivative operator w.r.t. variable x. Note that the first order score function $\mathcal{S}_1(x) = -\nabla_x \log p(x)$ is the same as the score function in (6). Furthermore, we show that $\mathcal{S}_m(x)$ is equivalently constructed from the recursive formula

$$\mathcal{S}_m(x) = -\mathcal{S}_{m-1}(x) \otimes \nabla_x \log p(x) - \nabla_x \mathcal{S}_{m-1}(x), \tag{9}$$

with $\mathcal{S}_0(x) = 1$. Here $\otimes$ denotes the tensor product; for details on tensor notation, see Section 6.1. Thus, $\mathcal{S}_m(x)$ is related to higher order score functions $\nabla_x^{(n)} \log p(x)$ with $n \leq m$, which is the reason we also call $\mathcal{S}_m(x)$'s as higher order score functions. These functions $\mathcal{S}_m(x)$ enable us to generalize the Stein's identity in (6) to higher order derivatives, i.e., they yield higher order differential operators.

Theorem 1 (Higher order differential operators, *informal statement*)**.** *For random vector x, let $p(x)$ and $\mathcal{S}_m(x)$ respectively denote the joint density function and the corresponding m-th order score function in* (8)*. Then, under some mild regularity conditions, for all functions $G(x)$, we have*

$$\mathbb{E}\left[G(x) \otimes \mathcal{S}_m(x)\right] = \mathbb{E}\left[\nabla_x^{(m)} G(x)\right],$$

where $\nabla_x^{(m)}$ denotes the m-th order derivative operator w.r.t. variable x.

See Theorem 6 for a formal statement of this result including description of regularity conditions.

Comparison with orthogonal polynomials: In the case of standard multivariate Gaussian distribution as $x \sim \mathcal{N}(0, I_d)$, the score functions defined in (8) turn out to be multivariate *Hermite* polynomials (Grad, 1949; Holmquist, 1996) $\mathcal{H}_m(x)$ defined as

$$\mathcal{H}_m(x) := (-1)^m \frac{\nabla_x^{(m)} p(x)}{p(x)}, \quad p(x) = \frac{1}{\left(\sqrt{2\pi}\right)^d} e^{-\|x\|^2/2}. \tag{10}$$

[8] Since $\mathcal{S}_m(x)$ is related to m-th order derivative of the function $p(x)$ with input vector x, it represents a tensor of order m, i.e., $\mathcal{S}_m \in \bigotimes^m \mathbb{R}^d$. For details on tensor notation, see Section 6.1.

It is also worth mentioning that the Hermite polynomials satisfy the orthogonality property as (Holmquist, 1996, Theorem 5.1)

$$\mathbb{E}\left[\mathcal{H}_m(x) \otimes \mathcal{H}_{m'}(x)\right] = \begin{cases} m! I^{\otimes m}, & m = m', \\ 0, & \text{otherwise}, \end{cases}$$

where the expectation is over the standard multivariate Gaussian distribution. The application of higher order Hermite polynomials as yielding differential operators has been known before (Goldstein and Reinert, 2005, equation (37)), but applications have mostly involved scalar variable $x \in \mathbb{R}$.

Thus, the proposed higher order score functions $\mathcal{S}_m(x)$ in (8) coincides with the orthogonal Hermite polynomials $\mathcal{H}_m(x)$ in case of multivariate Gaussian distribution. However, this is not necessarily the case for other distributions; for instance, it is convenient to see that the Laguerre polynomials which are orthogonal w.r.t. Gamma distribution are different from the score functions proposed in (8), although the Laguerre polynomials have a differential operator interpretation too; see Goldstein and Reinert (2005) for the details. Note that the proposed score functions need not be polynomial functions in general.

3.2.2 Parametric higher order Stein identities

In this section, we provide the generalization of first order parametric differential property in (7) to higher orders. In order to do this, we introduce the parametric form of higher order score functions in (8). Let $\mathcal{S}_m(x;\theta)$ be the m-th order parametric score function given by

$$\mathcal{S}_m(x;\theta) = (-1)^m \frac{\nabla_\theta^{(m)} p(x;\theta)}{p(x;\theta)}. \tag{11}$$

Similar to the previous section, we can construct $\mathcal{S}_m(x;\theta)$ as a function of higher order Fisher score functions $\nabla_\theta^{(n)} \log p(x;\theta)$, $n \leq m$, as

$$\mathcal{S}_m(x;\theta) := -\mathcal{S}_{m-1}(x;\theta) \otimes \nabla_\theta \log p(x;\theta) - \nabla_\theta \mathcal{S}_{m-1}(x;\theta), \tag{12}$$

with $\mathcal{S}_0(x;\theta) = 1$.

Note that the first order parametric score function $\mathcal{S}_1(x;\theta) = -\nabla_\theta \log p(x;\theta)$ is the same as Fisher score function exploited in (7). These higher order score functions enable us to generalize the parametric Stein's identity in (7) to higher orders as follows.

Theorem 2 (Higher order parametric differential operators, *informal statement*)**.** *For random vector $x \in \mathbb{R}^{d_x}$, let $p(x;\theta)$ and $\mathcal{S}_m(x;\theta)$ respectively denote the joint θ-parametric density function and the corresponding m-th order score function in* (11)*. Then, for all functions $G(x;\theta)$ satisfying the regularity conditions, we have*

$$\mathbb{E}[G(x;\theta) \otimes \mathcal{S}_m(x;\theta)] = \mathbb{E}[\nabla_\theta^{(m)} G(x;\theta)] \quad \text{at } \theta = \theta_0.$$

See Theorem 7 for a formal statement of this result including description of regularity conditions.

The advantage of this parametric form is it can be applied to both discrete and continuous random variables.

Algorithm 1 Tensor decomposition via tensor power iteration (Anandkumar et al., 2014b)

Require: 1) Rank-k tensor $T = \sum_{j\in[k]} u_j \otimes u_j \otimes u_j \in \mathbb{R}^{d\times d\times d}$, 2) L initialization vectors $\hat{u}_\tau^{(1)}$, $\tau \in [L]$, 3) number of iterations N.

for $\tau = 1$ **to** L **do**

for $t = 1$ **to** N **do**

Tensor power updates (see (16) for the definition of the multilinear form):

$$\hat{u}_\tau^{(t+1)} = \frac{T\left(I, \hat{u}_\tau^{(t)}, \hat{u}_\tau^{(t)}\right)}{\left\|T\left(I, \hat{u}_\tau^{(t)}, \hat{u}_\tau^{(t)}\right)\right\|}, \tag{14}$$

end for

end for

return the cluster centers of set $\left\{\hat{u}_\tau^{(N+1)} : \tau \in [L]\right\}$ (by Procedure 2) as estimates u_j.

4 Spectral Decomposition Algorithm

As part of the framework we introduced in Figure 1, we need a spectral/tensor method to decompose the higher order derivative tensor $\mathbb{E}[\nabla^{(m)} G(x)]$ to its rank-1 components denoted by u_j. Let us first consider the case that the derivative tensor is a matrix [9]. Then the problem of decomposing this matrix to the rank-1 components reduces to the usual Principle Component Analysis (PCA), where the rank-1 directions are the eigenvectors of the matrix.

More generally, we can form higher order derivatives ($m > 2$) of the label function $G(x)$ and extract more information from their decomposition. The higher order derivatives are represented as *tensors* which can be seen as multi-dimensional arrays. There exist different tensor decomposition frameworks, but the most popular one is the CP decomposition where a (symmetric) rank-k tensor $T \in \mathbb{R}^{d\times d\times d}$ is written as the sum of k rank-1 tensors [10]

$$T = \sum_{j\in[k]} u_j \otimes u_j \otimes u_j, \quad u_j \in \mathbb{R}^d. \tag{13}$$

Here notation $\otimes$ represents the tensor (outer) product; see Section 6.1 for a detailed discussion on the tensor notations.

We now state a tensor decomposition algorithm for computing decomposition forms in (13). The Algorithm 1 is considered by Anandkumar et al. (2014b) where the generalization to higher order tensors can be similarly introduced. The main step in (14) performs *power iteration* [11]; see (16) for the multilinear form definition. After running the algorithm for all different initialization vectors, the clustering process from Anandkumar et al. (2014b) ensures that the best converged vectors are returned as the estimates of true components u_j. Detailed analysis of the tensor decomposition algorithm and its convergence properties are provided by Anandkumar et al. (2014b). We briefly summarize the initialization and convergence guarantees of the algorithm below.

[9] For instance, it happens when the label function y is a scalar, and $m = 2$ for vector input x. Then, $\mathbb{E}[\nabla^{(2)} G(x)]$ is a matrix (second order tensor).

[10] The decomposition for an asymmetric tensor is similarly defined as $T = \sum_{j\in[k]} u_j \otimes v_j \otimes w_j,\ u_j, v_j, w_j \in \mathbb{R}^d$.

[11] This is the generalization of matrix power iteration to 3rd order tensors.

Procedure 2 Clustering process (Anandkumar et al., 2014b)

Require: Tensor $T \in \mathbb{R}^{d \times d \times d}$, set $S := \left\{ \hat{u}_\tau^{(N+1)} : \tau \in [L] \right\}$, parameter ν.
  while S is not empty **do**
    Choose $u \in S$ which maximizes $|T(u, u, u)|$.
    Do N more iterations of power updates in (14) starting from u.
    Let the output of iterations denoted by $\tilde{u}$ be the center of a cluster.
    Remove all the $u \in S$ with $|\langle u, \tilde{u} \rangle| > \nu/2$.
  end while
  return the cluster centers.

Initialization: Since tensor decomposition is a non-convex problem, different initialization lead to different solutions. Anandkumar et al. (2014b) introduce two initialization methods for the above algorithm. One is random initialization and the other is a SVD-based technique, where the convergence analysis is provided for the latter one.

Convergence guarantees: Tensor power iteration is one of the key algorithms for decomposing rank-k tensor T in (13) into its rank-1 components u_j's. Zhang and Golub (2001) provide the convergence analysis of tensor power iteration in the orthogonal setting where the tensor components u_j's are orthogonal to each other, and Anandkumar et al. (2014a) analyze the robustness of this algorithm to noise.

Note that the rank-k tensor decomposition can still be unique even if the rank-1 components are not orthogonal (unlike the matrix case). Anandkumar et al. (2014b) provide local and global convergence guarantees for Algorithm 1 in the non-orthogonal and overcomplete, (where the tensor rank k is larger than the dimension d) settings. The main assumption in their analysis is the incoherence property which imposes soft-orthogonality conditions on the components u_j's; see Anandkumar et al. (2014b) for details.

Whitening (orthogonalization): In the non-orthogonal and undercomplete (where the tensor rank k is smaller than the dimension d), instead of direct application of tensor power iteration for tensor decomposition as in Algorithm 1, we first orthogonalize the tensor and then apply the tensor power iteration, which requires different perturbation analysis; see for instance Song et al. (2013). In the orthogonalization step also known as whitening, the tensor modes are multiplied by whitening matrix such that the resulting tensor has an orthogonal decomposition.

5 Unsupervised Estimation of Score Functions

In this section, we discuss further on the score function and its estimation. First, we discuss the form of score function for exponential family, and for models with latent variables. Next, we review the frameworks which estimate the score function and discuss the connection with auto-encoders. Note that these frameworks can be also extended to learning score functions of a nonlinear transformation of the data.

5.1 Score function for exponential family

The score function expression proposed in Equation (8) can be further simplified when the random vector x belongs to the exponential family distributions where we have $p(x;\theta) \propto \exp(-E(x;\theta))$. Here, $E(x;\theta)$ is known as the energy function. Then, we have

$$\mathcal{S}_m(x) = (-1)^m \sum_{\alpha_1,\ldots,\alpha_t} \nabla_x^{(\alpha_1)} E(x,\theta) \otimes \nabla_x^{(\alpha_2)} E(x,\theta) \otimes \cdots \otimes \nabla_x^{(\alpha_t)} E(x,\theta),$$

$$\text{where } \{\alpha_i \in \mathbb{Z}^+,\ i \in [t] : \sum_{i=1}^{t} \alpha_i = m\}.$$

Thus, in case of the exponential family, the higher order score functions are compositions of the derivatives of the energy function.

5.2 Score function for Latent Variable Models

For a latent variable model $p(x,h;\theta)$, let h denote the vector of latent variables and x the vector of observed ones. It is well known that the (first order) Fisher score function of x is the marginalized of the joint Fisher score (Tsuda et al., 2002)

$$\nabla_\theta \log p(x;\hat{\theta}) = \sum_h p(h|x;\hat{\theta}) \nabla_\theta \log p(x,h;\hat{\theta}).$$

Given this marginalized form, Tsuda et al. (2002) also show that Fisher kernel is a special case of marginalized kernels (where the joint kernel over both observed and latent variables is marginalized over the posterior distribution).

The higher order score functions can be readily calculated by marginalization as

$$\mathcal{S}_m(x) = (-1)^m \frac{\sum_h \nabla^{(m)} p(x,h)}{\sum_h p(x,h)}.$$

For the special case of Gaussian mixtures, we can simplify the above general form in the following manner. Let $x = Ah + z$, where z has multivariate standard normal distribution for simplicity. In this case, the score function $\nabla_x \log p(x)$ is equal to $x - A\mathbb{E}[h|x]$. Here $A\mathbb{E}[h|x]$ is the posterior estimation of the mean, where the mean vectors a_j's are weighted with posterior estimation of the hidden state h, i.e., variable x is centered according to the contribution of each mixture component. Note that if h were observed, we would do the centering based on the mean corresponding to that component. But since h is hidden, we center based on the posterior estimation of h. More generally, the higher order score functions for Gaussian mixtures can be simplified as

$$\mathcal{S}_m(x) = \mathbb{E}_{h|x}\left[\mathcal{H}_m(x - Ah)|x\right], \quad \text{where } x \text{ is a mixture of Gaussians with hidden mixture } h.$$

Here $\mathcal{H}_m(y)$ is the multivariate Hermite polynomial defined in (10). Recall that the multivariate Hermite polynomials are the same as the higher order score functions for the standard multivariate Gaussian vector $y \sim \mathcal{N}(0, I)$.

5.3 Efficient Estimation of the Score Function

There are various efficient methods for computing the score function. In deep learning, the framework of auto-encoders attempts to find encoding and decoding functions which minimize the reconstruction error under noise (the so-called denoising auto-encoders or DAE). This is an unsupervised framework involving only unlabeled samples. Alain and Bengio (2012) argue that the DAE approximately learns the score function of the input, as the noise variance goes to zero. Moreover, they also describe ways to estimate the second order score function.

Score matching: The framework of score matching is popular for parameter estimation in probabilistic models (Hyvärinen, 2005; Swersky et al., 2011), where the criterion is to fit parameters based on matching the data score function. We now review the score matching framework and analysis in Lyu (2009). Let [12] $p(x)$ denote the pdf of x, and the goal is to find a parametric probabilistic model $q_\theta(x)$ with model parameter θ that best matches $p(x)$. Lyu (2009) formulate the score matching framework introduced by (Hyvärinen, 2005) as minimizing the Fisher divergence between two distributions $p(x)$ and $q_\theta(x)$, defined as

$$D_F(p\|q_\theta) := \int_x p(x) \left\| \frac{\nabla_x p(x)}{p(x)} - \frac{\nabla_x q_\theta(x)}{q_\theta(x)} \right\|^2 dx.$$

Note that $\frac{\nabla_x p(x)}{p(x)} = \nabla_x \log p(x)$ is the first order score function (up to sign). Lyu (2009) also show that the Fisher divergence can be equivalently written as

$$D_F(p\|q_\theta) = \int_x p(x) \left(\|\nabla \log p(x)\|^2 + \|\nabla \log q_\theta(x)\|^2 + 2 \triangle \log q_\theta(x) \right) dx.$$

where $\triangle$ denotes the Laplacian operator $\triangle := \sum_{i\in[d]} \frac{\partial^2}{\partial x_i^2}$. Then, they also provide a nice interpretation of Fisher divergence relating that to the usual KL (Kulback-Leibler) divergence $D_{KL}(p\|q_\theta) := \int_x p(x) \log \frac{p(x)}{q_\theta(x)} dx$ in the sense of robustness to Gaussian noise as follows. Note that this also gives the relation between score matching and maximum-likelihood (ML) estimation since ML is achieved by minimizing the KL-divergence.

Lemma 3 (Lyu 2009)**.** *Let $y = x + \sqrt{t}w$, for $t \geq 0$ and w a zero-mean white Gaussian vector. Denote $\tilde{p}_t(y)$ and $\tilde{q}_t(y)$ as the densities of y when x has distribution $p(x)$ and $q(x)$, respectively. Then, under some mild regularity conditions*[13]*, we have*

$$\frac{d}{dt} D_{KL}(\tilde{p}_t(y)\|\tilde{q}_t(y)) = -\frac{1}{2} D_F(\tilde{p}_t(y)\|\tilde{q}_t(y)).$$

This provides us the interpretation that score matching (by minimizing Fisher divergence $D_F(p\|q_\theta)$) looks for stability, where the optimal parameter θ leads to least changes in the KL divergence between the two models when a small amount of noise is added to the training data.

The above framework can be extended to matching the higher order score functions $\mathcal{S}_m(x)$ introduced in this paper, where the derivative is replaced by the m-th order derivative leading to

[12] For the sake of notation simplicity, we also refer to $p(x)$ and $q_\theta(x)$ as p and q_θ respectively, i.e., dropping the dependence on x.

[13] See Lyu (2009) for the details of regularity conditions

minimizing [14]

$$D_{\mathcal{L}}(p\|q_\theta) = \int_x p(x) \left\| \frac{\nabla_x^{(m)} p(x)}{p(x)} - \frac{\nabla_x^{(m)} q_\theta(x)}{q_\theta(x)} \right\|^2 dx.$$

Note that $\frac{\nabla_x^{(m)} p(x)}{p(x)}$ is exactly the m-th order score function $\mathcal{S}_m(x)$ up to sign.

In addition, Swersky et al. (2011) analyze the score matching for latent energy-based models with the joint distribution $p(x, h; \theta) = \frac{1}{Z(\theta)} \exp(-E_\theta(x, h))$, and provide the closed-form estimation for the parameters. Finally, Sasaki et al. (2014) point out that the score function can be estimated efficiently through non-parametric methods without the need to estimate the density function. In fact, the solution is closed form, and the hyper-parameters (such as the kernel bandwidth and the regularization parameter) can be tuned easily through cross validation.

Estimation of the score function for $\phi(x)$: In some applications we need to compute the score function of a nonlinear mapping of the input, i.e., for some function $\phi(x)$. This can be done by first estimating the joint density function of transformed variable and then computing its score function. Let $t = \phi(x)$ and $D_t(i, j) := \left[\frac{\partial x_i}{\partial t_j}\right]$. Then, we know

$$p_{\phi(x)}(t_1, \ldots, t_r) = p_x(\phi_1^{-1}(t), \ldots, \phi_r^{-1}(t)) \cdot |\det(D_t)|,$$

and the score function is defined as

$$\mathcal{S}_m(t) = (-1)^m \frac{\nabla_t^{(m)} p_{\phi(x)}(t)}{p_{\phi(x)}(t)}.$$

5.4 Score function estimation in self-taught setting

Now we discuss score function computation in the self-taught setting. Recall that in the self-taught setting, the distribution of unlabeled samples $\{\tilde{x}_i\}$ is different from the input of the labeled samples $\{x_i\}$. In Section 2, we assume that the conditional distributions $p(\tilde{x}|\tilde{h})$ and $p(x|h)$, given the corresponding latent variables $\tilde{h}$ and h, are the same and give justifications for this assumption.

Under this assumption, estimating the score function for new samples $\{x_i\}$ is relatively straightforward, since we can transfer the estimated conditional distribution $p(\tilde{x}|\tilde{h})$ (using unlabeled samples $\{\tilde{x}_i\}$) as the estimate for $p(x|h)$, and we can re-estimate the marginal distribution $p(h)$ easily. Thus, the use of score functions allows for easy transfer of information under the self-taught framework with latent-variable modeling.

More concretely, for the estimation of higher order score function $\mathcal{S}_m(x)$, we need to estimate the joint probability density function of x denoted by $p(x)$. We have

$$p(x) = \sum_h p(h)p(x|h) = \sum_h p(h)p(\tilde{x}|h),$$

where we also used the above assumption that the conditional distribution of target data x given hidden variables can be substituted by the conditional distribution of unlabeled data $\tilde{x}$ given hidden variables. Note that $p(\tilde{x}|h)$ can be estimated using unlabeled data $\{\tilde{x}_i\}$. The unsupervised

[14] Subscript notation $\mathcal{L}$ is from Lyu (2009) where the Fisher divergence is generalized to any linear operator, e.g., higher order derivatives in our case.

estimation of $p(\tilde{x}|h)$ can be done in different ways, e.g., using spectral methods, score matching and so on.

6 Formal Statement of the Results

In this section, we provide formal statement of the theorems characterizing the differential properties of the score functions. Before that, we propose an overview of notations mostly including tensor preliminaries.

6.1 Notations and tensor preliminaries

Let $[n]$ denote the set $\{1, 2, \ldots, n\}$.

Tensor: A real *r-th order tensor* $T \in \bigotimes_{i=1}^{r} \mathbb{R}^{d_i}$ is a member of the outer product of Euclidean spaces $\mathbb{R}^{d_i}$, $i \in [r]$. For convenience, we restrict to the case where $d_1 = d_2 = \cdots = d_r = d$, and simply write $T \in \bigotimes^{r} \mathbb{R}^{d}$. As is the case for vectors (where $r = 1$) and matrices (where $r = 2$), we may identify a r-th order tensor with the r-way array of real numbers $[T_{i_1,i_2,\ldots,i_r} : i_1, i_2, \ldots, i_r \in [d]]$, where $T_{i_1,i_2,\ldots,i_r}$ is the $(i_1, i_2, \ldots, i_r)$-th coordinate of T with respect to a canonical basis. For convenience, we limit to third order tensors ($r = 3$) in our analysis, while the results for higher order tensors are also provided.

Tensor as multilinear form: We view a tensor $T \in \mathbb{R}^{d\times d\times d}$ as a multilinear form. Consider matrices $M_l \in \mathbb{R}^{d\times d_l}, l \in \{1, 2, 3\}$. Then tensor $T(M_1, M_2, M_3) \in \mathbb{R}^{d_1} \otimes \mathbb{R}^{d_2} \otimes \mathbb{R}^{d_3}$ is defined as

$$T(M_1, M_2, M_3)_{i_1,i_2,i_3} := \sum_{j_1,j_2,j_3\in[d]} T_{j_1,j_2,j_3} \cdot M_1(j_1, i_1) \cdot M_2(j_2, i_2) \cdot M_3(j_3, i_3). \tag{15}$$

In particular, for vectors $u, v, w \in \mathbb{R}^d$, we have [15]

$$T(I, v, w) = \sum_{j,l\in[d]} v_j w_l T(:, j, l) \ \in \mathbb{R}^d, \tag{16}$$

which is a multilinear combination of the tensor mode-1 fibers. Similarly $T(u, v, w) \in \mathbb{R}$ is a multilinear combination of the tensor entries, and $T(I, I, w) \in \mathbb{R}^{d\times d}$ is a linear combination of the tensor slices.

CP decomposition and tensor rank: A 3rd order tensor $T \in \mathbb{R}^{d\times d\times d}$ is said to be rank-1 if it can be written in the form

$$T = w \cdot a \otimes b \otimes c \Leftrightarrow T(i, j, l) = w \cdot a(i) \cdot b(j) \cdot c(l), \tag{17}$$

where notation $\otimes$ represents the *tensor (outer) product*, and $a \in \mathbb{R}^d$, $b \in \mathbb{R}^d$, $c \in \mathbb{R}^d$ are unit vectors (without loss of generality). A tensor $T \in \mathbb{R}^{d\times d\times d}$ is said to have a CP rank $k \geq 1$ if it can be written as the sum of k rank-1 tensors

$$T = \sum_{i\in[k]} w_i a_i \otimes b_i \otimes c_i, \quad w_i \in \mathbb{R},\ a_i, b_i, c_i \in \mathbb{R}^d. \tag{18}$$

[15] Compare with the matrix case where for $M \in \mathbb{R}^{d\times d}$, we have $M(I, u) = Mu := \sum_{j\in[d]} u_j M(:, j) \in \mathbb{R}^d$.

Derivative of tensor-valued functions: Consider function $F(x) \in \bigotimes^r \mathbb{R}^d$ as a tensor-valued function with vector input $x \in \mathbb{R}^d$. The gradient of $F(x)$ w.r.t. variable x is defined as a higher order tensor $\nabla_x F(x) \in \bigotimes^{r+1} \mathbb{R}^d$ such that

$$\nabla_x F(x)_{i_1,\ldots,i_r,j} := \frac{\partial F(x)_{i_1,\ldots,i_r}}{\partial x_j}. \tag{19}$$

In addition, the m-th order derivative is denoted by $\nabla_x^{(m)} F(x) \in \bigotimes^{r+m} \mathbb{R}^d$.

Finally, the transposition of a tensor with respect to a permutation vector is defined as follows.

Definition 1 (Tensor transposition)**.** *Consider tensor $A \in \bigotimes^r \mathbb{R}^d$ and permutation vector $\pi = [\pi_1, \pi_2, \ldots, \pi_r] \in \mathbb{R}^r$ as a permutation of index vector $1:r$. Then, the π-transpose of A denoted by $A^{\langle \pi \rangle}$ is defined such that it satisfies*

$$A^{\langle \pi \rangle}(j_{\pi_1}, \ldots, j_{\pi_r}) = A(j_1, \ldots, j_r).$$

In other words, the i-th mode of tensor $A^{\langle \pi \rangle}$ corresponds to the π_i-th mode of tensor A.

6.2 Stein identity

The following lemma states Stein's identity saying how first order score functions yield differential properties.

Lemma 4 (Stein's lemma (Stein et al., 2004))**.** *Let $x \in \mathbb{R}^{d_x}$ be a random vector with joint density function $p(x)$. Suppose the score function $\nabla_x \log p(x)$ exists. Consider any continuously differentiable tensor function $G(x) : \mathbb{R}^{d_x} \to \bigotimes^r \mathbb{R}^{d_y}$ such that all the entries of $p(x) \cdot G(x)$ go to zero on the boundaries of support of $p(x)$. Then, we have*

$$\mathbb{E}[G(x) \otimes \nabla_x \log p(x)] = -\mathbb{E}[\nabla_x G(x)],$$

Note that it is also assumed that the above expectations exist (in the sense that the corresponding integrals exist).

The proof follows integration by parts; the result for the scalar x and scalar-output functions $g(x)$ is provided in Stein et al. (2004).

6.3 Parametric Stein identity

We first recall and expand some parametric notations mentioned earlier. Let Θ denote the set of parameters such that for $\theta \in \Theta$, $p(x;\theta)$ be a valid θ-parametric probability density function. For $\theta_0 \in \Theta$, let $\mathcal{P}(\mathbb{R}^{d_x}, \theta_0)$ be the collection of θ-parametric probability density functions on $\mathbb{R}^{d_x}$ for which there exists a bounded neighborhood $\Theta_0 \subset \Theta$ of θ_0 and an integrable function $l : \mathbb{R}^{d_x} \to \mathbb{R}^+$ such that $p(x;\theta) \leq l(x)$ over $\mathbb{R}^{d_x}$ for all $\theta \in \Theta_0$. Given $\theta_0 \in \Theta$ and $p \in \mathcal{P}(\mathbb{R}^{d_x}, \theta_0)$, we write $x \sim p(\cdot;\theta_0)$ to denote the joint density function of x.

The following *regularity conditions* is defined along the lines of (Ley and Swan, 2013, Definition 2.1).

Definition 2. *Let θ_0 be an interior point of Θ and $p \in \mathcal{P}(\mathbb{R}^{d_x}, \theta_0)$. Define $S_\theta := \{x \in \mathbb{R}^{d_x} | p(x;\theta) > 0\}$ as the support of $p(\cdot,\theta)$. We define the class $\mathcal{G}(p, \theta_0)$ as the collection of functions $G : \mathbb{R}^{d_x} \times \Theta \to \bigotimes^r \mathbb{R}^{d_y}$ such that there exits Θ_0 some neighborhood of θ_0 where the following conditions are satisfied:*

1. *There exists a constant $c_g \in \mathbb{R}$ (not depending on θ) such that $\int G(x;\theta)_{i_1,\ldots,i_r} p(x;\theta)dx = c_g$ for all $\theta \in \Theta_0$. Note that the equality is entry-wise.*

2. *For all $x \in S_\theta$ the mapping $\theta \to G(\cdot;\theta)p(\cdot;\theta)$ is differentiable in the sense of distributions over Θ_0, and in addition the order of derivative w.r.t. θ and integration over x can be changed.*

Finally, we state the parametric characterization of Stein's lemma as follows which is the result of Ley and Swan (2013, Theorem 2.1) generalized to tensor-output functions.

Theorem 5 (Parametric Stein characterization)**.** *Let $x \in \mathbb{R}^{d_x}$ be a random vector with joint θ-parametric density function $p(x;\theta)$. If the parametric score function $\nabla_\theta \log p(x;\theta)$ exists, then for all $G(x;\theta) \in \mathcal{G}(p;\theta_0)$ defined in Definition 2, we have*

$$\mathbb{E}[G(x;\theta) \otimes \nabla_\theta \log p(x;\theta)] = -\mathbb{E}[\nabla_\theta G(x;\theta)] \quad \text{at } \theta = \theta_0.$$

See Appendix A for the proof. The above result also holds for discrete random vectors; see Ley and Swan (2013, Section 2.4) for the details.

6.4 Higher order Stein identities

We first provide the formal definition of higher order Score functions $\mathcal{S}_m(x)$, and then their differential properties are stated.

Definition 3 (Higher order score functions)**.** *Let $p(x)$ denote the joint probability density function of random vector $x \in \mathbb{R}^d$. We denote $\mathcal{S}_m(x) \in \bigotimes^m \mathbb{R}^d$ as the m-th order score function which is defined based on the recursive differential relation*

$$\mathcal{S}_m(x) := -\mathcal{S}_{m-1}(x) \otimes \nabla_x \log p(x) - \nabla_x \mathcal{S}_{m-1}(x), \tag{20}$$

with $\mathcal{S}_0(x) = 1$.

By induction on m we can prove that the above definition is equivalent to (see the proof in the appendix)

$$\mathcal{S}_m(x) = (-1)^m \frac{\nabla_x^{(m)} p(x)}{p(x)}. \tag{21}$$

Note that the first order score function $\mathcal{S}_1(x) = -\nabla_x \log p(x)$ is the same as score function in Stein's lemma; see Lemma 4. These higher order score functions enable us to generalize the Stein's identity in Lemma 4 to higher orders as follows.

Theorem 6 (Yielding higher order differential operators)**.** *Let $x \in \mathbb{R}^{d_x}$ be a random vector with joint density function $p(x)$. Suppose the m-th order score function $\mathcal{S}_m(x)$ defined in (20) exists. Consider any continuously differentiable tensor function $G(x) : \mathbb{R}^{d_x} \to \bigotimes^r \mathbb{R}^{d_y}$ satisfying the regularity condition such that all the entries of $\nabla_x^{(i)} G(x) \otimes \mathcal{S}_{m-i-1}(x) \otimes p(x)$, $i \in \{0, 1, \ldots, m-1\}$, go to zero on the boundaries of support of $p(x)$. Then, we have*

$$\mathbb{E}\left[G(x) \otimes \mathcal{S}_m(x)\right] = \mathbb{E}\left[\nabla_x^{(m)} G(x)\right].$$

The result can be proved by iteratively applying the recursion formula of score functions in (20) and Stein's identity in Lemma 4, see Appendix A for the details.

6.5 Parametric higher order Stein identities

Let [16] $\mathcal{S}_m(x;\theta) \in \bigotimes^m \mathbb{R}^{|\theta|}$ be the m-th order parametric score function which is defined based on the recursive differential relation

$$\mathcal{S}_m(x;\theta) := -\mathcal{S}_{m-1}(x;\theta) \otimes \nabla_\theta \log p(x;\theta) - \nabla_\theta \mathcal{S}_{m-1}(x;\theta), \tag{22}$$

with $\mathcal{S}_0(x;\theta) = 1$. By induction on m we can prove that the above definition is equivalent to

$$\mathcal{S}_m(x;\theta) = (-1)^m \frac{\nabla_\theta^{(m)} p(x;\theta)}{p(x;\theta)}.$$

Theorem 7 (Yielding higher order parametric differential operators)**.** *Let $x \in \mathbb{R}^{d_x}$ be a random vector with joint θ-parametric density function $p(x;\theta)$. If the m-th order parametric score function $\mathcal{S}_m(x;\theta)$ defined in* (22) *exists, then for all $G(x;\theta) \in \mathcal{G}(p;\theta_0)$ defined in Definition 2, we have*

$$\mathbb{E}[G(x;\theta) \otimes \mathcal{S}_m(x;\theta)] = \mathbb{E}[\nabla_\theta^{(m)} G(x;\theta)] \quad \text{at } \theta = \theta_0.$$

Acknowledgements

M. Janzamin thanks Rina Panigrahy for useful discussions. M. Janzamin is supported by NSF Award CCF-1219234. H. Sedghi is supported by ONR Award N00014-14-1-0665. A. Anandkumar is supported in part by Microsoft Faculty Fellowship, NSF Career award CCF-1254106, NSF Award CCF-1219234, ARO YIP Award W911NF-13-1-0084 and ONR Award N00014-14-1-0665.

Appendix

A Proof of Theorems

Proof of Theorem 5: Let us denote by $\nabla_\theta h(\theta_0)$ the derivative of function $h(\theta)$ w.r.t. θ evaluated at point θ_0. We have

$$\begin{aligned}
\int \nabla_\theta \big(G(x;\theta_0)p(x;\theta_0)\big)dx &= \int \nabla_\theta G(x;\theta_0) \cdot p(x;\theta_0)dx + \int G(x;\theta_0) \otimes \nabla_\theta p(x;\theta_0)dx \\
&= \int \nabla_\theta G(x;\theta_0) \cdot p(x;\theta_0)dx + \int G(x;\theta_0) \otimes \nabla_\theta \log p(x;\theta_0) \cdot p(x;\theta_0)dx \\
&= \mathbb{E}[\nabla_\theta G(x;\theta_0)] + \mathbb{E}[G(x;\theta_0) \otimes \nabla_\theta \log p(x;\theta_0)],
\end{aligned}$$

where the first step is concluded from product rule. On the other hand, we have

$$\int \nabla_\theta \big(G(x;\theta_0)p(x;\theta_0)\big)dx = \nabla_\theta \int G(x;\theta_0)p(x;\theta_0)dx = \nabla_\theta c_g = 0,$$

where the second and first regularity conditions are respectively exploited in the above steps. Combining the above two inequalities, the result is proved. □

[16] Here, $|\theta|$ denote the dimension of parameter θ.

Proof of Theorem 6: The proof is done by iteratively applying the recursion formula of score functions in (20) and Stein's identity in Lemma 4. First, we provide the first order analysis as follows:

$$\begin{aligned}\mathbb{E}\left[G(x)\otimes\mathcal{S}_m(x)\right] &\overset{(e_1)}{=} -\mathbb{E}\left[G(x)\otimes\mathcal{S}_{m-1}(x)\otimes\nabla_x\log p(x)\right]-\mathbb{E}\left[G(x)\otimes\nabla_x\mathcal{S}_{m-1}(x)\right]\\ &\overset{(e_2)}{=} \mathbb{E}\left[\nabla_x\left(G(x)\otimes\mathcal{S}_{m-1}(x)\right)\right]-\mathbb{E}\left[G(x)\otimes\nabla_x\mathcal{S}_{m-1}(x)\right]\\ &\overset{(e_3)}{=} \mathbb{E}\left[\nabla_x G(x)\otimes\mathcal{S}_{m-1}(x)\right]^{\langle\pi\rangle}+\mathbb{E}\left[G(x)\otimes\nabla_x\mathcal{S}_{m-1}(x)\right]-\mathbb{E}\left[G(x)\otimes\nabla_x\mathcal{S}_{m-1}(x)\right]\\ &= \mathbb{E}\left[\nabla_x G(x)\otimes\mathcal{S}_{m-1}(x)\right]^{\langle\pi\rangle},\end{aligned}$$

where recursion formula (20) is used in equality (e_1), equality (e_2) is concluded by applying Stein's identity in Lemma 4 for which we also used the regularity condition that all the entries of $G(x)\otimes\mathcal{S}_{m-1}(x)\otimes p(x)$ go to zero on the boundaries of support of $p(x)$. Finally, the product rule in Lemma 8 is exploited in (e_3) with appropriate permutation vector π to put the differentiating variable in the last mode of the resulting tensor.

By iteratively applying above steps, the result is proved. Note that the permutation in the final step does not affect on the tensor $\nabla_x^{(m)}G(x)$ which is symmetric along the involved modes in the permutation. □

A.1 Auxiliary lemmas

Lemma 8 (Product rule for gradient)**.** *Consider $F(x)$ and $G(x)$ as tensor-valued functions*

$$F(x):\mathbb{R}^n\to\bigotimes^{p_1}\mathbb{R}^n,$$
$$G(x):\mathbb{R}^n\to\bigotimes^{p_2}\mathbb{R}^n.$$

Then, we have

$$\nabla(F(x)\otimes G(x))=(\nabla F(x)\otimes G(x))^{\langle\pi\rangle}+F\otimes\nabla G(x),$$

for permutation vector $\pi=[1,2,\ldots,p_1,p_1+2,p_1+3,\ldots,p_1+p_2+1,p_1+1]$.

Proof: The lemma is basically the product rule for derivative with the additional transposition applied to the first term. The necessity for transposition is argued as follows.

Note that for tensor-valued function $F(x)$, the gradient $\nabla F(x)$ is defined in (19) such that the last mode of the gradient tensor $\nabla F(x)$ corresponds to the entries of derivation argument or variable x. This is immediate to see that the transposition applied to first term $\nabla F(x)\otimes G(x)$ is required to comply with this convention. This transposition enforced by the specified permutation vector π puts the last mode of $\nabla F(x)$ (mode number p_1+1) to the last mode of whole tensor $\nabla F(x)\otimes G(x)$. Note that such transposition is not required for the other term $F\otimes\nabla G(x)$ since the last mode of $\nabla G(x)$ is already the last mode of $F\otimes\nabla G(x)$ as well. □

We also prove the explicit form of score functions in (21) as follows.

$$\mathcal{S}_m(x)=(-1)^m\frac{\nabla_x^{(m)}p(x)}{p(x)}.$$

Proof of explicit score function form in (21)**:** The result is proved by induction. It is easy to verify that the basis of induction holds for $m = 0, 1$. Now we argue the inductive step assuming that the result holds for $m-1$ and showing that it also holds for m. Substituting the induction assumption in the recursive form of $\mathcal{S}_m(x)$ defined in (20), we have

$$\begin{aligned}
\mathcal{S}_m(x) &= (-1)^m \frac{\nabla_x^{(m-1)} p(x)}{p(x)} \otimes \nabla_x \log p(x) + (-1)^m \nabla_x \left(\frac{\nabla_x^{(m-1)} p(x)}{p(x)} \right) \\
&= (-1)^m \frac{\nabla_x^{(m-1)} p(x) \otimes \nabla_x p(x)}{p(x)^2} + (-1)^m \frac{p(x) \nabla_x^{(m)} p(x) - \nabla_x^{(m-1)} p(x) \otimes \nabla_x p(x)}{p(x)^2} \\
&= (-1)^m \frac{\nabla_x^{(m)} p(x)}{p(x)},
\end{aligned}$$

where the quotient rule for derivative is used in the second equality. □

References

A. Agarwal, A. Anandkumar, P. Jain, P. Netrapalli, and R. Tandon. Learning Sparsely Used Overcomplete Dictionaries. In *Conference on Learning Theory (COLT)*, June 2014.

Guillaume Alain and Yoshua Bengio. What regularized auto-encoders learn from the data generating distribution. *arXiv preprint arXiv:1211.4246*, 2012.

A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky. Tensor Methods for Learning Latent Variable Models. *J. of Machine Learning Research*, 15:2773–2832, 2014a.

Anima Anandkumar, Rong Ge, and Majid Janzamin. Guaranteed Non-Orthogonal Tensor Decomposition via Alternating Rank-1 Updates. *arXiv preprint arXiv:1402.5180*, Feb. 2014b.

Anima Anandkumar, Rong Ge, and Majid Janzamin. Sample Complexity Analysis for Learning Overcomplete Latent Variable Models through Tensor Methods. *arXiv preprint arXiv:1408.0553*, Aug. 2014c.

Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *The Journal of Machine Learning Research*, 6:1817–1853, 2005.

Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. *arXiv preprint arXiv:1310.6343*, 2013.

Sanjeev Arora, Rong Ge, and Ankur Moitra. New algorithms for learning incoherent and overcomplete dictionaries. In *Proceedings of The 27th Conference on Learning Theory*, pages 779–806, 2014.

Shai Ben-david, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 137–144, 2006.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *ICML Workshop on Unsupervised and Transfer Learning*, 2011.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1798–1828, 2013.

John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128. Association for Computational Linguistics, 2006.

John Blitzer, Dean P Foster, and Sham M Kakade. Zero-shot domain adaptation: A multi-view approach. Technical report, Technical Report TTI-TR-2009-1, Toyota Technological Institute Chicago, 2009.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. of the eleventh annual conference on computational learning theory*, pages 92–100, 1998.

Adam Coates, Andrew Y Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*, pages 215–223, 2011.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proc. of ICML*, 2014.

Larry Goldstein and Gesine Reinert. Distributional transformations, orthogonal polynomials, and stein characterizations. *arXiv preprint arXiv:0510240*, 2005.

Boqing Gong, Kristen Grauman, and Fei Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *Proceedings of The 30th International Conference on Machine Learning*, pages 222–230, 2013.

Harold Grad. Note on n-dimensional hermite polynomials. *Communications on Pure and Applied Mathematics*, 2(4):325–330, Dec. 1949.

Judy Hoffman, Erik Rodner, Jeff Donahue, Trevor Darrell, and Kate Saenko. Efficient learning of domain-invariant image representations. *arXiv preprint arXiv:1301.3224*, 2013.

Bjorn Holmquist. The d-variate vector hermite polynomial of order k. *Linear Algebra and its Applications*, 237–239:155–190, April 1996.

Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*, pages 601–608, 2006.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

Tommi Jaakkola, David Haussler, et al. Exploiting generative models in discriminative classifiers. In *Advances in neural information processing systems*, pages 487–493, 1999.

Nikos Karampatziakis and Paul Mineiro. Discriminative features via generalized eigenvectors. In *Proceedings of The 31st International Conference on Machine Learning*, pages 494–502, 2014.

J.B. Kruskal. 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.

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.

Christophe Ley and Yvik Swan. Parametric stein operators and variance bounds. *arXiv preprint arXiv:1305.5067*, 2013.

Siwei Lyu. Interpretation and generalization of score matching. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 359–366. AUAI Press, 2009.

Laurens Maaten. Learning discriminative fisher kernels. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 217–224, 2011.

Julien Mairal, Jean Ponce, Guillermo Sapiro, Andrew Zisserman, and Francis R Bach. Supervised dictionary learning. In *Advances in neural information processing systems*, pages 1033–1040, 2009.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *Advances in Neural Information Processing Systems*, pages 1041–1048, 2009.

Andrew McCallum, Chris Pal, Greg Druck, and Xuerui Wang. Multi-conditional learning: Generative/discriminative training for clustering and classification. In *Proc. of AAAI*, 2006.

Ariadna Quattoni, Michael Collins, and Trevor Darrell. Learning visual representations using images with captions. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

Rajat Raina, Andrew Y Ng, and Daphne Koller. Constructing informative priors using transfer learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 713–720. ACM, 2006.

Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*, pages 759–766. ACM, 2007.

Hiroaki Sasaki, Aapo Hyvärinen, and Masashi Sugiyama. Clustering via mode seeking by direct estimation of the gradient of a log-density. *arXiv preprint arXiv:1404.5028*, 2014.

Hanie Sedghi and Anima Anandkumar. Provable methods for training neural networks with sparse connectivity. *NIPS workshop on Deep Learning and Representation Learning*, Dec. 2014a.

Hanie Sedghi and Anima Anandkumar. Provable tensor methods for learning mixtures of classifiers. *Unpublished*, 2014b.

Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems*, pages 935–943, 2013.

L. Song, A. Anandkumar, B. Dai, and B. Xie. Nonparametric estimation of multi-view latent variable models. *Available on arXiv:1311.3287*, Nov. 2013.

Charles Stein. 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: Probability Theory*, pages 583–602, Berkeley, Calif., 1972. University of California Press.

Charles Stein. Approximate computation of expectations. *Lecture Notes-Monograph Series*, 7: i–164, 1986.

Charles Stein, Persi Diaconis, Susan Holmes, Gesine Reinert, et al. Use of exchangeable pairs in the analysis of simulations. In *Stein's Method*, pages 1–25. Institute of Mathematical Statistics, 2004.

Kevin Swersky, David Buchman, Nando D Freitas, Benjamin M Marlin, et al. On autoencoders and score matching for energy based models. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1201–1208, 2011.

Martin Szummer and Tommi Jaakkola. Partially labeled classification with markov random walks. In *Advances in Neural Information Processing Systems*, pages 945–952, 2002.

Koji Tsuda, Taishin Kin, and Kiyoshi Asai. Marginalized kernels for biological sequences. *Bioinformatics*, 18(suppl 1):S268–S275, 2002.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Hua Wang, Feiping Nie, and Heng Huang. Robust and discriminative self-taught learning. In *Proceedings of The 30th International Conference on Machine Learning*, pages 298–306, 2013.

Jun Yang, Rong Yan, and Alexander G Hauptmann. Cross-domain video concept detection using adaptive svms. In *Proceedings of the 15th international conference on Multimedia*, pages 188–197. ACM, 2007.

David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 189–196, 1995.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Proc. of NIPS*, 2014.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional neural networks. *arXiv preprint arXiv:1311.2901*, 2013.

Jian Zhang, Zoubin Ghahramani, and Yiming Yang. Flexible latent variable models for multi-task learning. *Machine Learning*, 73(3):221–242, 2008.

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