Nonlinear Stein Variational Gradient Descent for Learning Diversified Mixture Models

Dilin Wang [1] **Qiang Liu** [1]

Abstract

Diversification has been shown to be a powerful mechanism for learning robust models in non-convex settings. A notable example is learning mixture models, in which enforcing diversity between the different mixture components allows us to prevent the model collapsing phenomenon and capture more patterns from the observed data. In this work, we present a variational approach for diversity-promoting learning, which leverages the entropy functional as a natural mechanism for enforcing diversity. We develop a simple and efficient functional gradient-based algorithm for optimizing the variational objective function, which provides a significant generalization of Stein variational gradient descent (SVGD). We test our method on various challenging real world problems, including deep embedded clustering and deep anomaly detection. Empirical results show that our method provides an effective mechanism for diversity-promoting learning, achieving substantial improvement over existing methods.

1. Introduction

Modern machine learning abounds with challenging non-convex optimization. A typical source of non-convexity comes from learning composite models consisting of symmetric units, such as latent variable or mixture models, ensemble models and additive models. In these problems, we are asked to estimate an un-ordered list of parameters, denoted by $\Theta := \{\theta_i\}_{i=1}^m$, by maximizing a permutation-invariant objective function that does not depend on the order of the elements of Θ:

$$\max_{\Theta} F(\{\theta_1, \dots, \theta_m\}),$$

where θ_i can be the component parameters in mixture models or additive models, or the posterior samples in Bayesian ensemble prediction; $F(\cdot)$ can be the log-likelihood function or negative loss function on the training data.

Due to the non-convex nature of the objective function, directly solving these problems with standard local search methods, such as gradient ascent, suffers from the risk of obtaining sub-optimal solutions. A common degenerate case is when multiple elements of θ_i converge to similar values (a.k.a. mode collapsing), while other important modes are missed due to insufficient exploration. In practice, it was found that this problem can be alleviated by adding diversification constraints to encourage the diversity among $\{\theta_i\}$, which can yield better and more robust empirical performance (e.g., Kwok & Adams, 2012; Mariet & Sra, 2016; Cogswell et al., 2015; Xie et al., 2016b; 2017). However, because repulsive regularization functions are inherently highly non-convex, directly adding them into the objective function casts additional difficulty in the optimization problem.

Main Algorithm We propose a simple yet powerful *functional approach* for learning diversified parameters. Our main practical result is a general "*diversified gradient ascent*" algorithm that learns the parameters by iteratively updating $\Theta := \{\theta_i\}_{i=1}^m$ in parallel by

$$\theta_i \leftarrow \theta_i + \varepsilon \sum_{j=1}^{m} \Big[\underbrace{\nabla_{\theta_j} F(\Theta)}_{gradient}\ k(\theta_j, \theta_i) + \frac{\alpha}{m} \underbrace{\nabla_{\theta_j} k(\theta_j, \theta_i)}_{repulsive} \Big], \tag{1}$$

where $k(\theta, \theta')$ is any positive definite kernel that specifies the similarity between θ and θ'. The two terms of update (1) yield intuitive interpretations: the first term performs an (averaged) gradient ascent update to increase the value of the objective F, while the second term corresponds to a repulsive force that enforces $\{\theta_i\}$ to be mutually different, which theoretically corresponds to a functional entropy regularization, as we show in the sequel. The balance of these two terms is controlled by a coefficient α, which correlates to a temperature parameter of the entropy regularization.

[1]Department of Computer Science, UT Austin. Correspondence to: Dilin Wang <dilin@cs.utexas.edu>, Qiang Liu <lqiang@cs.utexas.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

It is easy to see that our algorithm is a generalization of Stein variational gradient descent (SVGD) (Liu & Wang, 2016), which corresponds to the special case of (1) when F has a simple form of $F(\{\theta_i\}) = \sum_{i=1}^m f(\theta_i)/m$ for some function f; in this case, the update (1) yields an accurate particle approximation for distribution $\rho^*(\theta) \propto \exp(f(\theta)/\alpha)$. Our algorithm applies to more general classes of non-linear functions F, and hence significantly extends the scope of SVGD, especially in learning complex composite models in order to obtain particles with complex structures.

Theoretical Framework The derivation of our method leverages a general variational optimization framework that generalizes the underlying principle of SVGD. As a brief overview, note that any permutation-invariant function $F(\{\theta_i\})$ depends on $\{\theta_i\}$ only through its empirical measure $\rho(\theta) := \sum_{i=1}^m \delta(\theta - \theta_i)/m$, where δ denotes the Dirac Deta function. Therefore, we can consider $F(\{\theta_i\})$ as a *functional* of ρ, which we rewrite as $F[\rho]$ with an abuse of notation (where we use $F[\cdot]$ for functionals and $F(\cdot)$ for functions). We then relax the optimization of $\{\theta_i\}$ to the optimization of distribution ρ in the space of all distributions, on which an entropy regularization can be naturally defined:

$$\max_{\rho} F[\rho] \; + \; \alpha\mathbb{H}[\rho], \tag{2}$$

where the entropy $\mathbb{H}[\rho] = -\int \rho(\theta)\log\rho(\theta)\mathrm{d}\theta$ enforces ρ to distribute its probability more uniformly, and hence encourages the diversity on the particles $\{\theta_i\}$ drawn from ρ. In the special case when $F[\rho]$ is a linear functional of ρ, that is, $F[\rho] = \mathbb{E}_{\theta\sim\rho}[f(\theta)]$, the optimization reduces to minimizing a KL divergence between ρ and $\rho^*(\theta) \propto \exp(f(\theta)/\alpha)$:

$$\min_{\rho} \big\{\mathrm{KL}(\rho \,||\, \rho^*) := -\mathbb{E}_{\theta\sim\rho}[f(\theta)/\alpha] \; - \; \mathbb{H}[\rho]\big\}; \tag{3}$$

this is the variational optimization solved by SVGD, which returns a particle distribution $\rho(\theta) = \sum_i \delta(\theta - \theta_i)/m$ that weakly converges to the optimal solution of (3) when $m \to \infty$.[1] By extending the basic idea of SVGD, we develop a principled Stein variational optimization framework for (2) with more general nonlinear functionals $F[\rho]$, which yields the simple update in (1) (see also Algorithm 1).

Empirical Results Our master algorithmic framework is generally applicable to a broad range of problems, and can be implemented easily by plugging standard parametric gradients into the simple update rule in (1). In this work, we particularly focus on learning diversified mixture models with our method and demonstrate its effectiveness on deep unsupervised clustering (Dizaji et al., 2017) and deep anomaly detection (Zong et al., 2018), for both of which we obtain significant improvements with minor implementation and computation overhead over baseline approaches. Our code is avalable at `https://github.com/dilinwang820/nonlinear_svgd`.

[1] Note that $\mathbb{H}[\rho]$ is undefined for particle distribution of form $\rho(\theta) = \sum_i \delta(\theta - \theta_i)/m$, but the particle distributions returned by SVGD and our method approximates the optimal solutions of (2) and (3) in the sense of weak convergence as $m \to \infty$.

2. Motivation: Learning Diversified Mixture Models

We use learning diversified mixture models as a major example to demonstrate our general framework. Consider a mixture model $p(x \mid \Theta) := \frac{1}{m}\sum_{i=1}^m p(x \mid \theta_i)$ consisting of m components $p(x \mid \theta_i)$, where $\theta_i \in \mathbb{R}^d$ is the parameter of the i-th component and $\Theta = \{\theta_i\}_{i=1}^m$. Given an observation dataset $\mathcal{D} = \{x_\ell\}_{\ell=1}^n$, the parameters Θ are often estimated with the maximum likelihood estimator:

$$\max_{\Theta} \; \mathbb{E}_{x\sim\mathcal{D}}\left[\log\left(\frac{1}{m}\sum_{i=1}^m p(x \mid \theta_i)\right)\right] \; + \; \Phi(\Theta), \tag{4}$$

where $\Phi(\Theta)$ is a regularization term that encodes prior information. For example, repulsive priors such as determinantal point processes (Borodin, 2009) can be used to learn diversified models in which different component parameters θ_i are forced to be different from each other (e.g., Kwok & Adams, 2012; Xie et al., 2016b). Unfortunately, it is well known that (4) yields a challenging non-convex optimization, which significantly deteriorate with the addition of the repulsive regularization.

In this work, we take a variational approach for learning diversified models, which allows us to flexibly encode population-level priors such as diversity. Our framework considers the class of *infinite mixture models* of form

$$p(x \mid \rho) = \int \rho(\theta)p(x \mid \theta)d\theta,$$

where ρ is a distribution that parameterizes the mixture model of (potentially) infinite number of components θ. It reduces to the finite mixture model when $\rho = \sum_{i=1}^m \delta(\theta - \theta_i)/m$. We then formulate the estimation of the optimal ρ as the following variational optimization on the space $\mathcal{P}(\mathbb{R}^d)$ of all possible distributions on $\mathbb{R}^d$:

$$\rho^* = \arg\max_{\rho}\left\{J[\rho] \; := \; F[\rho] + \alpha\mathbb{H}[\rho]\right\}, \qquad \text{with } \; F[\rho] = \mathbb{E}_{x\sim\mathcal{D}}\log(\mathbb{E}_{\theta\sim\rho}[p(x;\theta)]), \tag{5}$$

where $F[\rho]$ denotes the likelihood functional of mixture model $p(x \mid \rho)$, and $\mathbb{H}[\rho] := -\int \rho(\theta)\log\rho(\theta)d\theta$ denotes the entropy of ρ, which is introduced to encourage ρ to

spread out its probability mass more uniformly. It promotes exploration in the parameter space so that local optima are more easily visited. Eq. (5) can be further extended to a PAC-Bayesian like framework (McAllester, 1999), by replacing $\mathbb{H}[\rho]$ with $\mathrm{KL}[\rho \,||\, \rho_0]$, where ρ_0 is a prior of ρ.

Solving Eq. (5) exactly is both intractable and unnecessary because $p(x \mid \rho)$ can be computationally infeasible for a complex ρ. The main technical contribution of this work is to derive an efficient *particle-based* approximation of (5) that outputs a finite set of component parameters $\{\theta_i\}$ (referred to as particles), such that their empirical measure $\rho = \sum_{i=1}^m \delta(\theta - \theta_i)/m$ approaches to the optimal solution ρ^* as we use more particles ($m \to \infty$). This yields practically-useful finite mixture models, but with the advantage of having more well-explored diversified components thanks to the functional entropic regularization.

3. Nonlinear Stein Variational Gradient Descent

It is highly non-trivial to construct a particle-based approximation for nonlinear variational optimization problems of the form (5). An immediate obscurity is that the entropy $\mathbb{H}[\rho]$ equals negative infinity for any empirical measures $\rho = \sum_{i=1}^m \delta(\theta - \theta_i)/m$, making direct optimization of $\{\theta_i\}_{i=1}^m$ impossible. We sidestep this problem by extending the key idea from Stein variational gradient descent (SVGD) (Liu & Wang, 2016), which iteratively pushes the particles $\{\theta_i\}$ to minimize the variational objective as fast as possible, following a functional gradient descent direction constrained on reproducing kernel Hilbert space (RKHS).

Specifically, our idea is to iteratively apply optimal transport on a set of particles to descend the functional objective as fast as possible. Starting from $\rho(\theta) := \sum_i \delta(\theta - \theta_i)/m$, we want to iteratively move $\{\theta_i\}$ towards the optimal solution ρ^* of (5). To achieve this, we initialize the particles to some initial locations and iteratively move them with a transform map that resembles the gradient ascent update:

$$T(\theta) = \theta + \varepsilon \boldsymbol{\phi}(\theta),$$

where ε is a small step size and $\boldsymbol{\phi}\colon \mathbb{R}^d \to \mathbb{R}^d$ is a vector-valued function (velocity field) that defines the moving direction of the particles. The key question is to choose an optimal $\boldsymbol{\phi}$ such that it brings the particle distribution ρ towards the optimum as fast as possible. Motivated by SVGD, we explicitly formulate this into an optimization problem from a predefined candidate function set $\mathcal{H}$:

$$\begin{aligned} \boldsymbol{\phi}^* &:= \arg\max_{\boldsymbol{\phi} \in \mathcal{H}} \{\mathcal{G}J[\rho;\ \boldsymbol{\phi}] \quad s.t. \quad ||\boldsymbol{\phi}||_{\mathcal{H}} \le 1\} \\ \text{where} \qquad & \mathcal{G}J[\rho;\ \boldsymbol{\phi}] := \frac{\mathrm{d}}{\mathrm{d}\varepsilon}(J[\rho_T] - J[\rho])\Big|_{\varepsilon=0}, \end{aligned} \tag{6}$$

where ρ_T denotes the distribution of the updated particles $\theta' = T(\theta)$ as $\theta \sim \rho$, and $\mathcal{G}J[\rho;\ \boldsymbol{\phi}]$ denotes the increasing rate of $J[\rho]$ as we move θ with $\boldsymbol{\phi}$ using a infinitesimal step size. We want to find the best $\boldsymbol{\phi}^*$ that maximizes $\mathcal{G}J[\rho;\ \boldsymbol{\phi}]$.

Following Liu & Wang (2016), we choose the candidate function set $\mathcal{H}$ to be a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ which allows us to derive a simple closed form solution; the norm constraint $\|\boldsymbol{\phi}\|_{\mathcal{H}} \le 1$ is introduced to prevent the issue of arbitrary scaling.

Our key result, which we now present, derives a simple closed form solution of (6), hence yielding a generalization of SVGD for optimizing the general entropy regularized nonlinear variational optimization in (2).

Definition 1. *Let $T(\theta) = \theta + \varepsilon\boldsymbol{\phi}(\theta)$ and ρ_T the distribution of $T(\theta)$ as $\theta \sim \rho$. For a functional $F[\rho]$, a vector-valued function $\mathcal{D}F[\rho]\colon \mathbb{R}^d \to \mathbb{R}^d$ is said to be its* transportation derivative *(T-derivative) if it satisfies*

$$F[\rho_T] - F[\rho] = \varepsilon\, \mathbb{E}_{\theta\sim\rho}[\mathcal{D}F[\rho](\theta)^\top \boldsymbol{\phi}(\theta)] \ + \ O(\varepsilon^2).$$

Equivalently, we have

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}(F[\rho_T] - F[\rho])\Big|_{\varepsilon=0} = \mathbb{E}_{\theta\sim\rho}[\mathcal{D}F[\rho](\theta)^\top \boldsymbol{\phi}(\theta)].$$

Here the concept of T-derivative is closely related to optimal transport and Wasserstein gradient flow (Vinh et al., 2010; Santambrogio, 2014); it specifies how $F[\rho]$ changes under transform $T(\theta) = \theta + \varepsilon\boldsymbol{\phi}(\theta)$. It is easy, for example, to show that the T-derivative of a linear functional $F[\rho] := \mathbb{E}_\rho[f]$ equals $\mathcal{D}F[\rho](\theta) = \nabla f(\theta)$. Further discussion of T-derivative is deferred to Section 3.1, where we show that T-derivative in fact coincides with a typical parametric gradient function when ρ is a empirical distribution, which allows convenient calculation in practice.

Theorem 1. *I) Let $J[\rho] = F[\rho] + \alpha\mathbb{H}[\rho]$, where $F[\rho]$ is a functional with a T-derivative of $\mathcal{D}F[\rho]$ and $\mathbb{H}[\rho]$ is the differential entropy. Assume ρ is absolutely continuous and $\mathbb{H}[\rho] < \infty$, and $\boldsymbol{\phi}$ is continuous differentiable. We have*

$$\mathcal{G}J[\rho;\ \boldsymbol{\phi}] = \mathbb{E}_{\theta\sim\rho}\left[\mathcal{D}F[\rho](\theta)^\top \boldsymbol{\phi}(\theta) + \alpha\nabla_\theta^\top \boldsymbol{\phi}(\theta)\right], \tag{7}$$

where $\mathcal{G}J[\rho;\ \boldsymbol{\phi}]$ is defined in (6)*.*

II) Let $\mathcal{H}_0$ be a reproducing kernel Hilbert space (RKHS) of scalar-valued functions with a positive definite kernel $k(\theta, \theta')$ on $\mathbb{R}^d \times \mathbb{R}^d$, and $\mathcal{H} := \mathcal{H}_0 \times \cdots \times \mathcal{H}_0$ be the vector-valued RKHS of functions of form $\boldsymbol{\phi} = [\phi_1, \ldots, \phi_d]^\top$, where $\phi_i \in \mathcal{H}_0$. Assume $\nabla_{\theta,\theta'}k(\theta, \theta')$ exits and continuous. Then the optimal solution $\boldsymbol{\phi}^*$ of (6) *satisfies*

$$\boldsymbol{\phi}^*(\cdot) \propto \mathbb{E}_{\theta\sim\rho}\left[\mathcal{D}F[\rho](\theta)k(\theta, \cdot) + \alpha\nabla_\theta k(\theta, \cdot)\right]. \tag{8}$$

This result generalizes Theorem 3.1 of Liu & Wang (2016) for SVGD. Specifically, when $F[\rho] = \mathbb{E}_{\theta\sim\rho}[\log p(\theta)]$, we can show that $\mathcal{D}F[\rho](\theta) = \nabla_\theta \log p(\theta)$, and the left side of (7) reduces to the *Stein operator* in Eq (5) of Liu & Wang (2016). In general, we may formally view $\mathcal{D}F[\rho]+\alpha\nabla_\theta$ as a functional operator, which generalizes the Stein operator in Liu & Wang (2016).

The optimal $\boldsymbol{\phi}^*$ in (8) defines the optimal velocity field that increases $J[\rho]$ as fast as possible w.r.t. ρ. If we recursively apply this optimal transform $T(\theta) \leftarrow \theta + \varepsilon\boldsymbol{\phi}^*(\theta)$ starting from some initial values, we obtain a sequence of particles that form increasingly better approximation of the optimal solution. In practice, we take ρ to be an empirical measure of a set of particles, so that the updates on ρ reduces to its corresponding particles. This yields our main algorithm, *Nonlinear Stein Variational Gradient Descent (NSVGD)*, which iteratively performs the following updates until convergence:

$$\theta_i^{t+1} \leftarrow \theta_i^t + \varepsilon\boldsymbol{\phi}^*(\theta_i^t), \quad \forall i = 1,\dots,m,$$
$$\boldsymbol{\phi}^*(\theta_i^t) = \mathbb{E}_{\theta\sim\rho_m^t}\big[\underbrace{\mathcal{D}F[\rho_m^t](\theta)k(\theta,\,\theta_i^t)}_{gradient} + \alpha\underbrace{\nabla_\theta k(\theta,\,\theta_i^t)}_{repulsive}\big],$$

where $\{\theta_i^t\}_{i=1}^m$ are the particles at the t-th iteration and $\rho_m^t(\theta) := \sum_i \delta(\theta-\theta_i^t)/m$ is their empirical distribution. Similar to SVGD, this update is simple and intuitive: the first term in $\boldsymbol{\phi}^*$ drives the particles to increase $F[\rho]$ following its functional gradient $\mathcal{D}F[\rho]$, while the second term serves to maximize the entropy $\mathbb{H}[\rho]$, and can be interpreted as a repulsive force that enforces diversity among the particles.

3.1. More on the Transportation Derivative

It remains to be determined what the T-derivative $\mathcal{D}F[\rho]$ is for a general nonlinear functional $F[\rho]$ and how to calculate it practically. It turns out that $\mathcal{D}F[\rho]$ simply equals the typical notion of parametric derivative, if $F[\rho]$ reduces to a differentiable parametric function when ρ is an empirical distribution of a set of particles.

Theorem 2. *Assume $\rho_\Theta := \sum_{i=1}^m \delta(\theta-\theta_i)/m$ is the empirical measure of a set of particles $\Theta = \{\theta_i\}_{i=1}^m$. With an abuse of notation, let $F(\Theta) = F[\rho_\Theta]$ be the (parametric) function that maps Θ to $F[\rho_\Theta]$. If $F(\Theta)$ is differentiable, we have*

$$\mathcal{D}F[\rho_\Theta](\theta_i) = m\nabla_{\theta_i}F(\Theta). \tag{9}$$

Proof. Let $T(\theta) = \theta + \varepsilon\boldsymbol{\phi}(\theta)$, and $\rho_{\Theta,T}$ the distribution of $T(\theta)$ when $\theta\sim\rho_\Theta$. We have

$$F[\rho_{\Theta,T}] = F(\{\theta_i + \varepsilon\boldsymbol{\phi}(\theta_i)\}_{i=1}^m).$$

Therefore, by Taylor expansion,

$$\begin{aligned} F[\rho_{\Theta,T}] - F[\rho_\Theta] &= F(\{\theta_i + \varepsilon\boldsymbol{\phi}(\theta_i)\}_{i=1}^m) - F(\{\theta_i\}_{i=1}^m) \\ &= \epsilon\sum_{i=1}^m \nabla_{\theta_i}F(\{\theta_i\}_{i=1}^m)^\top\boldsymbol{\phi}(\theta_i) + O(\epsilon^2) \\ &= \epsilon\mathbb{E}_{\theta\sim\rho_\Theta}\left[m\nabla_\theta F(\Theta)^\top\boldsymbol{\phi}(\theta)\right] + O(\epsilon^2). \end{aligned}$$

Comparing this with the definition of $\mathcal{D}F$ in Definition 1, we have $\mathcal{D}F[\rho_\Theta](\theta_i) = m\nabla_{\theta_i}F(\Theta)$. □

Plugging (9) into the nonlinear SVGD update yields our main algorithm (1), which can be easily implemented based on the existing auto-differentiation tools that have been developed for gradient-based optimization algorithms.

For the sake of theoretical completeness, the following result characterizes T-derivative when ρ is absolutely continuous; it shows that T-derivative equals the derivative of the typical notion of *first variation*.

Theorem 3. *Define the* first variation *of functional $F[\rho]$ to be the scalar-valued function $\mathcal{L}F[\rho]\colon \mathbb{R}^d \to \mathbb{R}$ that satisfies*

$$F[\rho + \varepsilon v] = F[\rho] + \varepsilon\,\mathbb{E}[\mathcal{L}F[\rho](\theta)v(\theta)] + O(\varepsilon^2),$$

for $\varepsilon\in\mathbb{R}$ and function $v\colon \mathbb{R}^d\to\mathbb{R}$. Assume ρ is a continuous differentiable density function supported on $\mathbb{R}^d$ and $\mathcal{L}F[\rho](\theta)$ is differentiable w.r.t. θ. We have

$$\mathcal{D}F[\rho](\theta) = \nabla_\theta(\mathcal{L}F[\rho](\theta)).$$

As an example, consider the entropy functional $\mathbb{H}[\rho] = -\mathbb{E}_\rho[\log\rho]$. We can show that $\mathcal{L}\mathbb{H}[\rho](\theta) = -\log\rho(\theta)$, and hence $\mathcal{D}\mathbb{H}[\rho](\theta) = \nabla_\theta(\mathcal{L}\mathbb{H}[\rho](\theta)) = -\nabla_\theta\log\rho(\theta)$. Note that $\mathbb{H}[\rho]$, $\mathcal{D}\mathbb{H}[\rho]$ and $\mathcal{L}\mathbb{H}[\rho]$ are all infinite when ρ is an empirical delta measure, and hence Theorem 2 does not apply. Using Stein's identity (see e.g., Stein et al. (2004); Liu & Wang (2016)), we have

$$\begin{aligned} \mathbb{E}_\rho[\nabla_\theta^\top\boldsymbol{\phi}(\theta)] &= -\mathbb{E}_\rho[\nabla_\theta\log\rho(\theta)^\top\boldsymbol{\phi}(\theta)] \\ &= \mathbb{E}_\rho[\mathcal{D}\mathbb{H}[\rho](\theta)^\top\boldsymbol{\phi}(\theta)], \end{aligned}$$

which suggests that the differential operator ∇_θ can be formally viewed as $\mathcal{D}\mathbb{H}[\rho]$, which is consistent with the result in (7) of Theorem 1.

4. Related Work

Diversity-promoting learning has been exploited in various contents (e.g. Xie et al., 2016b; 2017; Cogswell et al., 2015; Kathuria et al., 2016; Affandi et al., 2013; Kuncheva & Whitaker, 2003; Banfield et al., 2005; Partalas et al., 2008; Yu et al., 2011; Kwok & Adams, 2012). For example, determinant point process (DPP) has been widely exploited

Algorithm 1 Nonlinear SVGD for Learning Mixture Models

Input: A collection of data $\mathcal{D}$

Goal: Learn a mixture model $\sum_{i=1}^m p(x \mid \theta_i)/m$ with diversified parameters $\Theta = \{\theta_i\}_{i=1}^m$, by optimizing the functional objective in (5).

Initialize a set of particles $\{\theta_i^0\}_{i=1}^m$; pick a positive define kernel $k(\theta, \theta')$ and a stepsize scheme $\{\varepsilon_t\}$.

for iteration t **do**

$$\theta_i^{t+1} \leftarrow \theta_i^t \ + \ \varepsilon_t \boldsymbol{\phi}^*(\theta_i^t), \qquad \text{with} \quad \boldsymbol{\phi}^*(\theta_i^t) = \sum_{j=1}^m \left[\nabla_{\theta_j^t} F(\Theta) k(\theta_j^t, \theta_i^t) + \frac{\alpha}{m} \nabla_{\theta_j^t} k(\theta_j^t, \theta_i^t) \right],$$

where $F(\Theta) = \mathbb{E}_{x \sim \mathcal{D}}[\log(\sum_{i=1}^m p(x \mid \theta_i)/m)]$ is the standard parametric log-likelihood of the mixture model.

end for

Remark. When the size of the dataset $\mathcal{D}$ is large, we may subsample a mini-batch $\mathcal{M} = \{x_1, \cdots, x_m\}$ from the data $\mathcal{D}$ at each iteration and use it to calculate $F(\Theta)$.

as a diversity regularization mechanism (e.g., Gong et al., 2014; Kulesza et al., 2012; Borodin, 2009; Hough et al., 2009; Chen et al., 2018; Zhang et al., 2017; Mariet et al., 2018). A few works have been developed in the particular content of learning diversified mixture models; examples include (Kwok & Adams, 2012; Xie et al., 2016b; 2017). Our algorithm differs from these previous works in that we take a functional view. Our method is also more computationally efficient than methods based on DPP, which require calculating the matrix determinant.

5. Experiments

In this section, we present experiments on learning diversified mixture models. We start with a toy Gaussian mixture example and then present results on deep embedded clustering (Dizaji et al., 2017) and deep anomaly detection (Zong et al., 2018). We also provide additional results on variational inference in Appendix C. In all cases, we find that our method provides an effective approach for promoting diversity, hence increasing the robustness of learning.

For all of our experiments, we use the Gaussian RBF kernel $k(\theta, \theta') = \exp(-\frac{||\theta - \theta'||^2}{2h^2})$ following Liu & Wang (2016), and take the bandwidth h to be the median of the pairwise distances of the set of particles at each iteration.

5.1. Toy Gaussian Mixture Models

We first study a toy Gaussian mixture model (GMM) case. Suppose that we observe n i.i.d. 1-dimensional samples $\{x_\ell\}_{\ell=1}^n$ drawn from an unknown GMM $p(x) = \frac{1}{m}\sum_{i=1}^m \mathcal{N}(x; \mu_i^*, 1)$ whose variance of each mixture component is fixed to be 1. We draw $n = 2000 \times m$ samples from p and learn a GMM $q(x) = \frac{1}{m}\sum_{i=1}^m \mathcal{N}(x; \mu_i, \sigma_i^2)$ to best fit the observations. The log-likelihood objective is given by

$$F(\Theta) = \frac{1}{n}\sum_{\ell=1}^n \log\left(\frac{1}{m}\sum_{i=1}^m \mathcal{N}(x_\ell | \mu_i, \sigma_i^2)\right),$$

where model parameters $\Theta = \{\theta_i\}_{i=1}^m$ with each particle $\theta_i = [\mu_i, \sigma_i]$.

In this case, the difficulty of estimating GMM parameters depends on the amount of separation between the mean vectors of the different components of the unknown target distribution p. We construct the true mean vectors μ_i^* to be on a uniform grid $\mu_i^* = i\delta$, where δ is *separation distance* of the different components of the mixture model. For the learning algorithms, we initialize the mean estimates in all the algorithms by $\mu_i \sim \text{Uniform}[-m, m]$; therefore, when δ is very large, the initialization is much more concentrated than the true values, creating a challenging situation for iterative learning algorithms to find all the true mixture components. The idea is that by using the diversification mechanisms like the ones in our algorithm, we can push the components apart to avoid redundant and degenerate solutions.

We compare our method with standard maximum likelihood optimization without repulsive regularization (denoted as **GMM**). We also compare our method with two standard diversity regularizers as baselines: 1) **GMM-PT**, an MLE estimator regularized by a repulsive force defined by the pairwise sum of a kernel function; the training objective is

$$\max_{\Theta} \{F(\Theta) + \alpha \sum_{i \neq j} k(\theta_i, \theta_j)\},$$

where $k(\theta_i, \theta_j)$ is a kernel function that measures the similarity between θ_i and θ_j; 2) **GMM-LogDet**, an MLE estimator regularized by a log-determinant divergence term,

$$\max_{\Theta} \{F(\Theta) + \alpha \log\det \mathcal{K}_\Theta\};$$

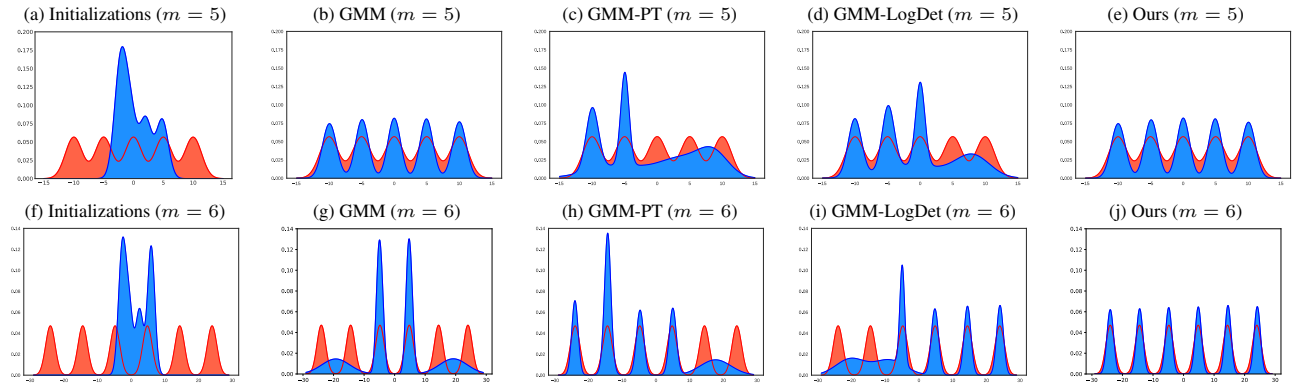

Figure 1. Visualization of the densities of the true distributions (red) and the learned densities (blue) of each method. (a)-(e) show the case when we have $m = 5$ components, and a separation distribution of $\delta = 2$, and (f)-(j) show the case of $m = 6$ and $\delta = 4$.

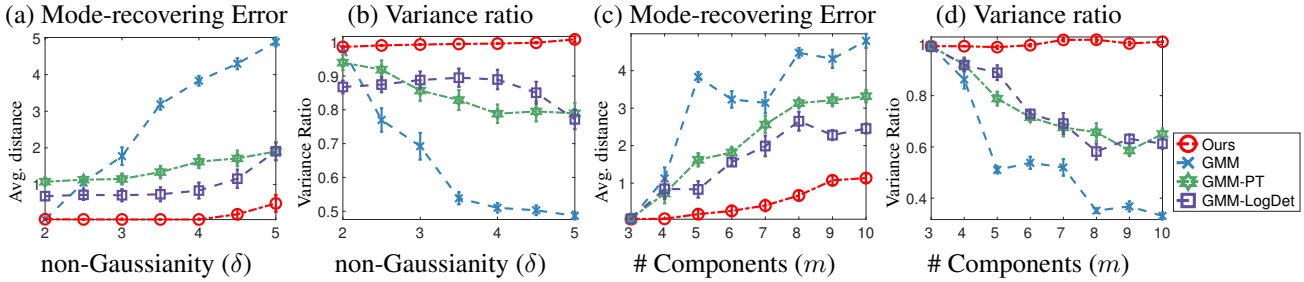

Figure 2. (a)-(b) plot the mode-recovering errors and the variance ratios $\mathrm{var}(q)/\mathrm{var}(p)$ between the learned density q and the true density p. (a)-(b) show the case when we increase the separation distribution δ with fixed number of components $m = 6$; (c)-(d) show the case when we increase the number of components m, with fixed $\delta = 4$. All the results are averaged over 20 random trials.

with $\mathcal{K}_\Theta = [k(\theta_i, \theta_j)]_{ij} \in \mathbb{R}^{m\times m}$ representing the kernel matrix over the particles. Both regularizers act as a role to prevent collapsed components.

To evaluate the *mass-covering* property of different methods, we calculate a *mode-recovering error* between p and q following Wang et al. (2018). Specifically, for each Gaussian component i of the target p, we measure its distance to its nearest Gaussian component j in the distribution q, which is defined as $d_i = \min_j\{||\mu_i^* - \mu_j||_2, \ \forall j\}$, where μ_i^* denotes the true mean parameter of component i and μ_j the estimated parameter of component j. The overall mode-recovering error between p and q is the average over all distance d_i over all i. We also calculate the ratio of the variance of the estimated model q and the true model p, that is, $\mathrm{var}(q)/\mathrm{var}(p)$.

In our experiments, all methods are optimized using Adagrad with a constant learning rate of 0.05. For each model, we train $50,000$ iterations with a mini-batch size of 256. We clip the logarithm values of the variance σ_i to $[-3, 3]$ to avoid singularities. We choose the best temperature parameter α from $\{1.0, 0.1, 0.5, 0.05, 0.01\}$ for GMM-PT, GMM-LogDet and our method to minimize the mode-recovering error. We use RBF kernel for both regularizers and set the bandwidth as the median distance.

Figure 1 shows example density functions learned by our methods and the baselines. We can see that our method recovers the density well, while the other methods tend to degenerate as a consequence of the poor initialization in our setting.

Figure 2 (a)-(b) show the result when we fix the number of components $m = 6$ and gradually increase the value of δ so that p is increasingly multi-modal. We can see from Figure 2 (a)-(b) that GMM generally underestimates the variance, and the variance ratio between q and p is relative small when δ is large and the true target p is highly multi-modal, which suggests that GMM fails to learn diverse components and only places mass in small regions. We can see from Figure 2 (c)-(d) that, as we increase the number of components, our approach still yields robust performance and outperforms other baseline approaches significantly.

5.2. Deep Embedded Clustering

We demonstrate an application of our method on deep embedded clustering, for which a latent space representation

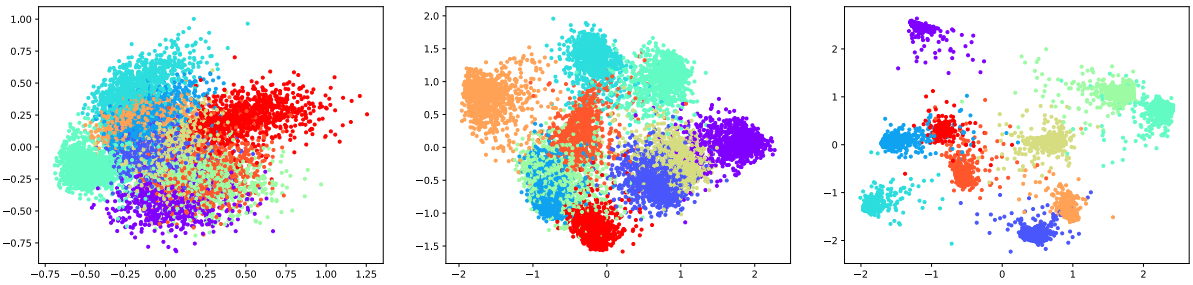

Figure 3. Visualization of the learned low-dimensional representations by different methods on MNIST. PCA is used to project the latent representations to a 2-D space. Different colors correspond to different clusters.

	k-means	AE+k-means	DEC*	JULE*	DEPICT	DEPICT+ Ours
NMI	0.500	0.805	0.816	0.913	0.917	**0.933**
ACC	0.534	0.899	0.844	0.964	0.965	**0.974**

Table 1. Clustering performance (NMI and ACC) of different algorithms on MNIST (higher values are better for both metrics). The results labeled by (*) are baselines reported in Dizaji et al. (2017). Our results are averaged over 10 random trials.

of high dimension data is trained jointly with clustering parameters. Specifically, we build our method on DEPICT (Dizaji et al., 2017), which joint trains a deep convolutional auto-encoder to map the data to a low-dimensional feature space and simultaneously fits a mixture model in the feature space for clustering. We apply our method as an inter loop to learn the parameters of the mixture model. Experimental results show our approach yields more diversified clusters and achieves better clustering performance.

DEPICT Consider the problem of clustering a set of n points $\{x_1, \cdots, x_n\}$ into m categories. DEPICT consists of two main parts: a deep convolutional autoencoder AE parameterized with Ψ that learns clustering oriented representations $\{z_1, \cdots, z_n\}$; a softmax layer parameterized with $\Theta = \{\theta_1, \cdots, \theta_m\}$ stacked on the latent representations to give cluster assignments. Specifically, the probability that x_ℓ with latent representation z_ℓ belongs to the i-th cluster can be written as

$$c_{\ell i} = \exp(\theta_i^\top z_\ell) / \sum_{j=1}^{m} \exp(\theta_j^\top z_\ell). \tag{10}$$

DEPICT jointly minimizes the reconstruction error and a clustering-related regularization term:

$$\min_{\Psi,\Theta} F(\Psi, \Theta) = \frac{1}{n} \sum_{\ell=1}^{n} ||x_\ell - \mathrm{AE}(x_\ell; \Psi)||_2^2 + R(\Theta, \{z_\ell\}),$$

where $\mathrm{AE}(x_\ell; \Psi)$ denotes the reconstruction of x_ℓ from the antoencoder, and $C = [c_{\ell i}]_{\ell i} \in \mathbb{R}^{n \times m}$ relates to the clustering probabilities. The regularizer $R(\Theta, \{z_\ell\})$ defines a nonlinear optimization objective over Θ that refines the cluster assignments by placing more emphasis on data points assigned with high confidence and prevents large clusters from distorting the hidden feature space in the meantime. We refer the reader to Appendix (or Dizaji et al. (2017)) for more detailed discussions.

DEPICT with Nonlinear SVGD We replace the softmax layer with a Gaussian mixture model, $p_g(z;\ \Theta) = \frac{1}{m} \sum_{i=1}^{m} \mathcal{N}(z; \theta_i, \mathrm{I})$ where $\{\theta_i\}$ are the mean vectors and and the convariance matrices are fixed to be identity. In this way, the corresponding probability that x_ℓ belongs to the cluster i is changed from (10) to

$$c_{\ell i} = \mathcal{N}(z_\ell; \theta_i,\ \mathrm{I}) / \sum_{j=1}^{m} \mathcal{N}(z_\ell; \theta_i,\ \mathrm{I}).$$

We keep the updates of the autoencoder parameters Ψ the same, but update the GMM parameters Θ using Algorithm 1 by taking $F(\Theta) = -R(\Theta, \{z_\ell\})$.

Empirical Results We evaluate our approach on the MNIST dataset (LeCun et al., 1998), which consists of 70,000 handwritten digits of 28-by-28 pixel size. For fair comparison, we use the same network architectures and hyper-parameters settings as reported in DEPICT. Specifically, we use Adam with a constant learning rate 0.0001 for optimization and fix the dimension of the feature space to 10 and the temperature α to 0.5.

We compare with several baseline methods: 1) k-means, the naive k-means (Lloyd, 1982) algorithm applied on the raw pixel as inputs; 2) AE+k-means, which first pre-trains a deep autoencoder with the same structure as DEPICT and then applies k-means on learned representations; 3) DEC (Xie et al., 2016a) and JULE (Yang et al., 2016): simi-

Method	Precision	Recall	F1
DSEBM (Zhai et al., 2016)*	0.7369	0.7477	0.7423
DCN (Yang et al., 2017)*	0.7696	0.7829	0.7762
DAGMM-p (Zong et al., 2018)*	0.7579	0.7710	0.7644
DAGMM-NVI (Zong et al., 2018)*	0.9290	0.9447	0.9368
DAGMM (Zong et al., 2018)*	0.9297	0.9442	0.9369
DAGMM+ Ours	**0.9659**	**0.9490**	**0.9573**

Table 2. Average precision, recall and F1 scores on DAGMM. The results labeled by (*) were baselines reported in Zong et al. (2018).

lar to DEPICT, the idea of DEC and JULE is to connect a clustering module to the output layer of a deep neural network (DNN) and jointly learn DNN parameters and clusters. DEC achieves this by minimizing a Kullback-Leibler (KL)-like clustering loss, and JULE minimizing an agglomerative clustering loss.

We visualize in Figure 3 the learned embeddings on the MNIST dataset using PCA. We can see that our method learns more diversified clusters compared to the baselines. Table 1 shows the average normalized mutual information (NMI) and clustering accuracy (ACC) scores on MNIST. We can see that our approach improves DEPICT and achieves the best NMI and ACC scores.

5.3. Deep Anomaly Detection

We evaluate our approach on an anomaly detection task. We use a recently proposed deep autoencoding Gaussian mixture model (DAGMM) (Zong et al., 2018) as our base model. DAGMM utlizes a deep autoencoder to learn low-dimensional representations for the inputs, which are further fed into a GMM. During testing time, data points with GMM densities smaller than a threshold are classifed as anomalies. On the KDDCUP benchmark dataset, we show our method improves DAGMM by simply replacing the gradients of the GMM parameters with our method.

DAGMM Assume we have a collection of observations $\{x_1, \cdots, x_n\}$ and its corresponding low dimensional latent representations $\{z_1, \cdots, z_n\}$ generated by a deep autoencoder. DAGMM proposes to simultaneously optimize the deep autoencoder network $\mathrm{AE}(x; \Psi)$ that learns feature representations and a GMM $p_g(z; \Theta)$ to fit the low-dimensional latent features z. The objective is to minimize the reconstruction error and the negative log-likelihood jointly,

$$\min_{\Psi,\Theta} L(\Psi, \Theta) = \frac{1}{n}\sum_{\ell=1}^{n} ||x_\ell - \mathrm{AE}(x_\ell; \Psi)||_2^2 - E(\Theta,\ \{z_\ell\})$$

$$\text{where} \quad E(\Theta,\ \{z_\ell\}) = \frac{\lambda_1}{n}\sum_{\ell=1}^{n} \log p_g(z_\ell; \Theta) - \lambda_2 R(\Theta).$$

Here $\mathrm{AE}(x_\ell; \Psi)$ denotes the reconstructed counterpart of x_ℓ, $R(\Theta)$ is a regularization term, which prevents the covariance matrix of each GMM component to be singular. λ_1, λ_2 are hyperparameters. See Zong et al. (2018) for more details.

DAGMM with Nonlinear SVGD We propose to update the GMM parameters Θ using our Algorithm 1 with $F(\Theta) = E(\Theta,\ \{z_\ell\})$ at each iteration. For the GMM model, we assume

$$p_g(z;\ \Theta) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{N}(z; \mu_i, S_i^\top S_i),$$

where $\Theta = \{\theta_1, \cdots, \theta_m\}$ with each particle $\theta_i = \{\mu_i, S_i\}$, representing the mean vectors $\{\mu_i\}$ and square root of covariance matrices $\{S_i\}$. We keep S_i a lower triangular matrix in our implementation.

Empirical Results For fair comparison, we follow exactly the implementation details in Zong et al. (2018). We evaluate on the KDDCUP99 10 percent dataset[2], which contains 494,021 instances and an anomaly ratio of 0.2. We replace the categorical features with one-hot representation as suggested in Zong et al. (2018). We take 50% of the normal data for training and the rest for testing. For testing, the top 20% samples with the lowest likelihood are classified as anomalies. We fix the temperature α to be 0.5. Table 2 reports the average precision, recall and F1 score after 20 runs for our approach and the baselines. We can see that our approach achieves the best performance.

6. Conclusions

In this paper, we generalize Stein variational gradient descent (SVGD) for learning entropy regularized diversified mixture models. Our algorithm has a straightforward and simple form, which closely mimics the typical gradient updates but with a repulsive force for encouraging exploration in the parameter space. Empirical results on the deep embedded clustering and the deep anomaly detection task demonstrate the practical effectiveness of our approach.

[2] `http://archive.ics.uci.edu/ml/datasets/kdd+cup+1999+data`

Acknowledgement

We would like to thank Yan Zheng and Ziyang Tang for their helpful comments on the paper. This work is supported in part by NSF CRII 1830161 and NSF CAREER 1846421. We would like to acknowledge Google Cloud for their support.

References

Raja Hafiz Affandi, Emily Fox, and Ben Taskar. Approximate inference in continuous determinantal processes. In *Advances in Neural Information Processing Systems*, pp. 1430–1438, 2013.

Robert E Banfield, Lawrence O Hall, Kevin W Bowyer, and W Philip Kegelmeyer. Ensemble diversity measures and their application to thinning. *Information Fusion*, 6 (1):49–62, 2005.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Alexei Borodin. Determinantal point processes. *arXiv preprint arXiv:0911.1153*, 2009.

Laming Chen, Guoxin Zhang, and Eric Zhou. Fast greedy map inference for determinantal point process to improve recommendation diversity. In *Advances in Neural Information Processing Systems*, pp. 5627–5638, 2018.

Michael Cogswell, Faruk Ahmed, Ross Girshick, Larry Zitnick, and Dhruv Batra. Reducing overfitting in deep networks by decorrelating representations. *arXiv preprint arXiv:1511.06068*, 2015.

Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5747–5756. IEEE, 2017.

Boqing Gong, Wei-Lun Chao, Kristen Grauman, and Fei Sha. Diverse sequential subset selection for supervised video summarization. In *Advances in Neural Information Processing Systems*, pp. 2069–2077, 2014.

John Ben Hough, Manjunath Krishnapur, Yuval Peres, et al. *Zeros of Gaussian analytic functions and determinantal point processes*, volume 51. American Mathematical Soc., 2009.

Tarun Kathuria, Amit Deshpande, and Pushmeet Kohli. Batched gaussian process bandit optimization via determinantal point processes. In *Advances in Neural Information Processing Systems*, pp. 4206–4214, 2016.

Alex Kulesza, Ben Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

Ludmila I Kuncheva and Christopher J Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2): 181–207, 2003.

James T Kwok and Ryan P Adams. Priors for diversity in generative latent variable models. In *Advances in Neural Information Processing Systems*, pp. 2996–3004, 2012.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Qiang Liu. Stein variational gradient descent as gradient flow. In *Advances in neural information processing systems*, pp. 3115–3123, 2017.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances In Neural Information Processing Systems*, pp. 2378–2386, 2016.

S Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

Zelda Mariet and Suvrit Sra. Diversity networks: Neural network compression using determinantal point processes. *International Conference on Learning Representations*, 2016.

Zelda E Mariet, Suvrit Sra, and Stefanie Jegelka. Exponentiated strongly rayleigh distributions. In *Advances in Neural Information Processing Systems*, pp. 4464–4474, 2018.

David A McAllester. Pac-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pp. 164–170. ACM, 1999.

Ioannis Partalas, Grigorios Tsoumakas, and Ioannis P Vlahavas. Focused ensemble selection: A diversity-based method for greedy ensemble selection. In *European Conference on Artificial Intelligence*, pp. 117–121, 2008.

Filippo Santambrogio. Introduction to optimal transport theory. *chapter in "Optimal Transportation, theory and applications", London Math. Soc.*, 2014.

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

Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11 (Oct):2837–2854, 2010.

Dilin Wang, Hao Liu, and Qiang Liu. Variational inference with tail-adaptive f-divergence. In *Advances in Neural Information Processing Systems*, pp. 5742–5752, 2018.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning*, pp. 478–487, 2016a.

Pengtao Xie, Jun Zhu, and Eric Xing. Diversity-promoting bayesian learning of latent variable models. In *International Conference on Machine Learning*, pp. 59–68, 2016b.

Pengtao Xie, Aarti Singh, and Eric P Xing. Uncorrelation and evenness: a new diversity-promoting regularizer. In *International Conference on Machine Learning*, pp. 3811–3820, 2017.

Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *International Conference on Machine Learning*, pp. 3861–3870, 2017.

Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5147–5156, 2016.

Yang Yu, Yu-Feng Li, and Zhi-Hua Zhou. Diversity regularized machine. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, pp. 1603, 2011.

Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. Deep structured energy based models for anomaly detection. In *International Conference on Machine Learning*, pp. 1100–1109, 2016.

Cheng Zhang, Hedvig Kjellstrom, and Stephan Mandt. Determinantal point processes for mini-batch diversification. *UAI*, 2017.

Ding-Xuan Zhou. Derivative reproducing properties for kernel methods in learning theory. *Journal of computational and Applied Mathematics*, 220(1-2):456–463, 2008.

Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018.