Information geometry and minimum description length networks

Sun, Ke; Wang, Jun; Kalousis, Alexandros; Marchand-Maillet, Stéphane

Information Geometry and Minimum Description Length Networks

Ke Sun SUNK.EDU@GMAIL.COM
Viper Group, Computer Vision and Multimedia Laboratory, University of Geneva, Switzerland

Jun Wang JWANG1@EXPEDIA.COM
Expedia, Switzerland

Alexandros Kalousis ALEXANDROS.KALOUSIS@HESGE.CH
Business Informatics Department, University of Applied Sciences, Western Switzerland
Department of Computer Science, University of Geneva, Switzerland

Stéphane Marchand-Maillet STEPHANE.MARCHAND-MAILLET@UNIGE.CH
Viper Group, Computer Vision and Multimedia Laboratory, University of Geneva, Switzerland

Abstract

We study parametric unsupervised mixture learning. We measure the loss of intrinsic information from the observations to complex mixture models, and then to simple mixture models. We present a geometric picture, where all these representations are regarded as free points in the space of probability distributions. Based on minimum description length, we derive a simple geometric principle to learn all these models together. We present a new learning machine with theories, algorithms, and simulations.

1. Introduction

Knowledge is often gradually perceived from simple to complex. In the realm of mixture modeling (Hinton et al., 1995; Rasmussen, 2000; Vincent & Bengio, 2003), a simple representation with a few components and a complex representation with many components are both meaningful in the learning path. For example, the point cloud in fig. 1 (a) can either be perceived as 3 blobs, or as 9 blobs, which are both "correct".

A stack of mixture models, or clustering schemes, with a different number of components at each layer can be learned (Ward, 1963; Heller & Ghahramani, 2005; Goldberger & Roweis, 2005; Garcia et al., 2010; Liu et al., 2012; Schwander & Nielsen, 2012; Krishnamurthy et al., 2012). Often, such learning is split into stages, and the learners at different stages are not fully communicating with each other. For example, while successively merging small clusters (Ward, 1963), these clusters are often not re-adjusted based on the latter-learned hierarchy. As another example, consider a stage-① learner for complex mixture modeling using many components, and a stage-② learner for simplification (Goldberger & Roweis, 2005; Schwander & Nielsen, 2012). The communication from ① to ② is unidirectional. What are the intrinsic principles of stacked mixture learning, and how to implement full communication within this stack, so that different representation layers help each other to learn, are still open problems.

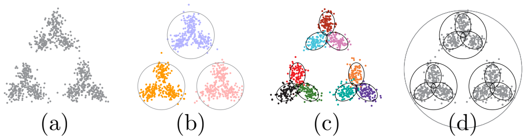

Figure 1. (a) a point cloud; (b) a simple perception (colors and circles denote mixture components); (c) a complex perception; (d) a stacked mixture model.

This paper studies such a stacked mixture model as in fig. 1 (d). This model fits in a *minimum description length* (MDL, Rissanen, 1978; 1989) *network*, where all mixture components at different layers are regarded as free points in the space of probability distributions. Based on a global cost function, these points are learned so that the whole network is compact in the geometric sense. Under this concept, we implement one specific method, where the mixture components are Gaussian distributions. We show empirical results on the effectiveness of the network-structured regularization.

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

The proposed MDL networks extend to stacked mixtures of any distribution in the exponential family. It unifies the concepts of geometric compactness and description conciseness. It provides new insights on the connections among machine learning, information geometry, and MDL.

In the following, section 2 recalls useful concepts of information geometry. Sections 3 and 4 present the method and how to implement it, respectively. Section 5 shows density estimation simulations. Section 6 presents an analysis on the learning theory. Section 7 concludes.

2. Information Geometry

This section introduces basic *known facts* about information geometry to make the manuscript self-contained. Readers are referred to other materials (Amari & Nagaoka, 2000; Nielsen, 2013) for a more complete understanding.

2.1. Exponential Family Statistical Manifolds

In an *exponential family* $\mathcal{S}$ dominated by certain measure $\sigma(\boldsymbol{x})$, any probability distribution can be written in the canonical form

$$p(\boldsymbol{x}\,|\,\boldsymbol{\theta}) = \exp\left(\boldsymbol{\theta}^T\boldsymbol{t}(\boldsymbol{x}) - \psi(\boldsymbol{\theta})\right), \tag{1}$$

where $\boldsymbol{\theta}$ is the canonical parameter, $\boldsymbol{t}(\boldsymbol{x})$ is a vector of sufficient statistics, and $\psi(\boldsymbol{\theta})$ is a strictly-convex potential function. The Hessian of $\psi(\boldsymbol{\theta})$, given by $g(\boldsymbol{\theta}) \overset{def}{=} \partial^2\psi/\partial\boldsymbol{\theta}^2$, must be positive definite and can be regarded as a Riemannian metric (Jost, 2011) of $\mathcal{S}$, known as the Fisher Information Metric (FIM) (Rao, 1945). Using Riemannian geometry as a tool, and based on the fundamentals defined by FIM, the discipline of *information geometry* (Rao, 1945; Čencov, 2000; Efron, 1975; Amari & Nagaoka, 2000; Nielsen & Nock, 2009) studies the measurements and dynamics of intrinsic information on $\mathcal{S}$ as a Riemannian manifold.

2.2. Expectation Parameters

Because of the strict convexity of $\psi(\boldsymbol{\theta})$, there is a one-to-one mapping between $\boldsymbol{\theta}$ and $\boldsymbol{\eta} \overset{def}{=} \partial\psi/\partial\boldsymbol{\theta}$. Differentiating both sides of $\int p(\boldsymbol{x}\,|\,\boldsymbol{\theta})d\sigma(\boldsymbol{x}) = 1$ w. r. t. $\boldsymbol{\theta}$, by eq. (1), we get $\boldsymbol{\eta} = \int p(\boldsymbol{x}\,|\,\boldsymbol{\theta})\boldsymbol{t}(\boldsymbol{x})d\sigma(\boldsymbol{x})$. Therefore $\boldsymbol{\eta}$ is called the "expectation parameter", i.e., the expectation of $\boldsymbol{t}(\boldsymbol{x})$. Both $\boldsymbol{\theta}$ and $\boldsymbol{\eta}$ play the role of a global coordinate system of $\mathcal{S}$. They are connected by the Legendre transformations (Amari & Nagaoka, 2000) $\boldsymbol{\eta} = \partial\psi/\partial\boldsymbol{\theta}$ and $\boldsymbol{\theta} = \partial\psi^\star/\partial\boldsymbol{\eta}$, where $\psi^\star \overset{def}{=} \int p(\boldsymbol{x}\,|\,\boldsymbol{\theta})\ln p(\boldsymbol{x}\,|\,\boldsymbol{\theta})d\sigma(\boldsymbol{x})$ is the negative entropy, the dual potential function which is convex w. r. t. $\boldsymbol{\eta}$. By eq. (1) and the definition of $\psi^\star$, ψ and $\psi^\star$ [1] have the fundamental relationship

$$\psi^\star - \boldsymbol{\eta}^T\boldsymbol{\theta} + \psi = 0. \tag{2}$$

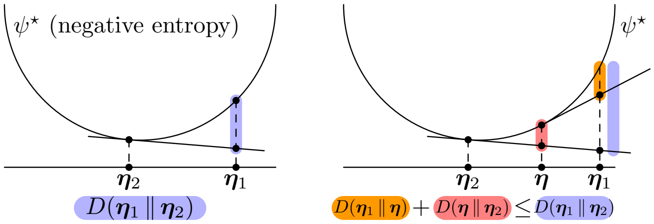
$\psi^\star$ (negative entropy)
$\boldsymbol{\eta}_2$
$\boldsymbol{\eta}_1$
$D(\boldsymbol{\eta}_1\,\|\,\boldsymbol{\eta}_2)$
$\psi^\star$
$\boldsymbol{\eta}_2$
$\boldsymbol{\eta}$
$\boldsymbol{\eta}_1$
$D(\boldsymbol{\eta}_1\,\|\,\boldsymbol{\eta}) + D(\boldsymbol{\eta}\,\|\,\boldsymbol{\eta}_2) \leq D(\boldsymbol{\eta}_1\,\|\,\boldsymbol{\eta}_2)$

Figure 2. (left) $D(\boldsymbol{\eta}_1\,\|\,\boldsymbol{\eta}_2)$ as a Bregman divergence. (right) An intuitive presentation of the "gain", to be used in section 6.

We give without proof that, $g(\boldsymbol{\eta})$, the FIM in the η-coordinates, is the Hessian of $\psi^\star$, and is equivalent to $g(\boldsymbol{\theta})$ under coordinate transformation. Therefore, $g(\boldsymbol{\eta}) = \partial^2\psi^\star/\partial\boldsymbol{\eta}^2 = \partial\boldsymbol{\theta}/\partial\boldsymbol{\eta}$, where $\partial\boldsymbol{\theta}/\partial\boldsymbol{\eta}$ denotes the Jacobi matrix of the mapping $\boldsymbol{\eta} \to \boldsymbol{\theta}$.

2.3. Divergence

By Riemannian geometry (Jost, 2011), the infinitesimal distance between $\boldsymbol{\eta}$ and $\boldsymbol{\eta} + d\boldsymbol{\eta}$ is $\sqrt{d\boldsymbol{\eta}^T g(\boldsymbol{\eta})d\boldsymbol{\eta}}$. The macroscopic distance, i.e., the length of the shortest path, between two points $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ on $\mathcal{S}$ does not have a closed form in general. As a practical way to measure their dissimilarity, the Bregman divergence (Bregman, 1967; Nielsen et al., 2010) induced by the function $\psi^\star$ is [2]

$$\begin{aligned} D(\boldsymbol{\eta}_1\,\|\,\boldsymbol{\eta}_2) &\overset{def}{=} \psi^\star(\boldsymbol{\eta}_1) - \psi^\star(\boldsymbol{\eta}_2) - \boldsymbol{\theta}_2^T(\boldsymbol{\eta}_1 - \boldsymbol{\eta}_2) \\ &= \psi^\star(\boldsymbol{\eta}_1) - \boldsymbol{\eta}_1^T\boldsymbol{\theta}_2 + \psi(\boldsymbol{\theta}_2), \quad \text{(by eq. (2))} \end{aligned} \tag{3}$$

which is illustrated by fig. 2 (left) and turns out to be the Kullback-Leibler divergence. By a Taylor expansion of $\psi^\star(\boldsymbol{\eta})$ at $\boldsymbol{\eta}_2$ up to the second order, we see that $D(\boldsymbol{\eta}_1\,\|\,\boldsymbol{\eta}_2) \approx (\boldsymbol{\eta}_1 - \boldsymbol{\eta}_2)^T g(\boldsymbol{\eta}_2)(\boldsymbol{\eta}_1 - \boldsymbol{\eta}_2)/2$ is half of the square distance between $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ w. r. t. the FIM at $\boldsymbol{\eta}_2$. This approximation is accurate when $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ are close enough.

2.4. The Gaussian Manifold

A multivariate Gaussian density with mean $\boldsymbol{\mu}$ and covariance matrix Σ can be written as $G(\boldsymbol{x}\,|\,\boldsymbol{\mu},\Sigma) = \exp\left(\boldsymbol{x}^T\boldsymbol{\theta}^{(1)} + tr(\boldsymbol{\theta}^{(2)}\boldsymbol{x}\boldsymbol{x}^T) - \psi\left(\boldsymbol{\theta}^{(1)},\boldsymbol{\theta}^{(2)}\right)\right)$, where $\boldsymbol{\theta}^{(1)} = \Sigma^{-1}\boldsymbol{\mu}$, $\boldsymbol{\theta}^{(2)} = -\Sigma^{-1}/2$, and $tr(\cdot)$ is the trace.

[1] In this paper, functions defined on $\mathcal{S}$, like $\psi(\boldsymbol{\theta})$ or $\psi^\star(\boldsymbol{\eta})$, and coordinate transformations of $\mathcal{S}$, like $\boldsymbol{\eta}(\boldsymbol{\theta})$ or $\boldsymbol{\theta}(\boldsymbol{\eta})$, can be notated without their arguments.

[2] To avoid confusion, $\psi^\star$ in this paper is denoted as φ in the textbook (Amari & Nagaoka, 2000), and $D(\cdot\,\|\,\cdot)$ in this paper is denoted as $D^{(-1)}(\cdot\,\|\,\cdot)$ or $D^\star(\cdot\,\|\,\cdot)$.

Therefore, the Gaussian manifold, i.e., the parameter space of all such $G(\boldsymbol{x}\,|\,\boldsymbol{\mu},\Sigma)$, is in the exponential family. Its dimensionality is $\dim(\mathcal{S})=\dim(\boldsymbol{x})(\dim(\boldsymbol{x})+3)/2$. The expectation parameters are given by $\boldsymbol{\eta}^{(1)}=E(\boldsymbol{x})=\boldsymbol{\mu}$ and $\boldsymbol{\eta}^{(2)}=E(\boldsymbol{x}\boldsymbol{x}^T)=\Sigma+\boldsymbol{\mu}\boldsymbol{\mu}^T$. The Legendre transformations $\boldsymbol{\theta}\leftrightarrow\boldsymbol{\eta}$ involve matrix inversions and can have a cubic complexity in $\dim(\boldsymbol{x})$. All diagonal Gaussian distributions with the same $\dim(\boldsymbol{x})$ form an embedded sub-manifold. There, the computational complexity of the Legendre transformations is linear in $\dim(\boldsymbol{x})$.

3. Minimum Description Length Networks

3.1. Divergence-induced Priors

For developing the proposed method, we introduce a new tool called *divergence-induced priors*. Using a reference set $\mathcal{B}=\{\boldsymbol{\eta}_1,\ldots,\boldsymbol{\eta}_m\}$ on an exponential family $\mathcal{S}$, and using some non-negative weights $\boldsymbol{\alpha}=(\alpha_1,\ldots,\alpha_m)$ so that $\sum_{i=1}^m\alpha_i=1$, we define a distribution of a random $\boldsymbol{\eta}\in\mathcal{S}$ as

$$p(\boldsymbol{\eta}\,|\,\mathcal{B},\,\boldsymbol{\alpha})=\frac{1}{N(\mathcal{B},\boldsymbol{\alpha})}\sum_{i=1}^m\alpha_i\exp\left(-D(\boldsymbol{\eta}\,\|\,\boldsymbol{\eta}_i)\right), \quad (4)$$

where $N(\mathcal{B},\boldsymbol{\alpha})=\int_{\boldsymbol{\eta}\in\mathcal{S}}\sum_{i=1}^m\alpha_i\exp\left(-D(\boldsymbol{\eta}\,\|\,\boldsymbol{\eta}_i)\right)d\boldsymbol{\eta}$. The intuition is that any $\boldsymbol{\eta}$ close to some $\boldsymbol{\eta}_i\in\mathcal{B}$ has a high probability, where closeness is defined by the information divergence. Although the η-coordinates are used here, $p(\boldsymbol{\eta}\,|\,\mathcal{B},\,\boldsymbol{\alpha})$ is a density function defined on $\mathcal{S}$ and is invariant to coordinate transformations. This $p(\boldsymbol{\eta}\,|\,\mathcal{B},\boldsymbol{\alpha})$ is an *informative prior*, as compared to the non-informative prior (Jeffreys, 1946) or weakly informative priors used in Bayesian learning (Ghahramani & Beal, 2000).

Using a divergence-induced prior, we can describe any $\boldsymbol{\eta}\in\mathcal{S}$ with a *code*. This description will be used later to measure the cost to describe the data. We first discretize $\mathcal{S}$ into tiny hyper-cubes with edge-length δ. By Riemannian geometry (Jost, 2011), the volume[3] of the cube containing $\boldsymbol{\eta}$ is $\sqrt{|g(\boldsymbol{\eta})|}\delta^{\dim\mathcal{S}}$. By the divergence-induced prior, the probability mass of this cube is $p(\boldsymbol{\eta}\,|\,\mathcal{B},\boldsymbol{\alpha})\sqrt{|g(\boldsymbol{\eta})|}\delta^{\dim\mathcal{S}}$. The optimal code length (Shannon, 1948) of $\boldsymbol{\eta}$ in *nats* is

$$\begin{aligned}&-\ln\left[p(\boldsymbol{\eta}\,|\,\mathcal{B},\boldsymbol{\alpha})\sqrt{|g(\boldsymbol{\eta})|}\delta^{\dim\mathcal{S}}\right]\\ =&-\ln\left(\sum_{i=1}^m\alpha_i\exp\left(-D(\boldsymbol{\eta}\,\|\,\boldsymbol{\eta}_i)\right)\right)+\ln N(\mathcal{B},\boldsymbol{\alpha})\\ &-\frac{1}{2}\ln|g(\boldsymbol{\eta})|-\dim\mathcal{S}\ln\delta. \qquad (5)\end{aligned}$$

We refer the reader to a good introduction of MDL (Hansen & Yu, 2001) for a related analysis.

In the four-part code on the right-hand-side of eq. (5), the first term measures the *novelty* of $\boldsymbol{\eta}$ or its difference to the prior knowledge $\mathcal{B}$ and $\boldsymbol{\alpha}$; the second term measures the *strength* of the prior knowledge or its similarity with $\mathcal{S}$ as a whole; the last two terms measure the description accuracy w. r. t. the discretization. Therefore, longer codes are assigned to accurate description of new knowledge.

By approximating $D(\boldsymbol{\eta}\,\|\,\boldsymbol{\eta}_i)$ to a square distance[4], $\ln N(\mathcal{B},\boldsymbol{\alpha})\approx\dim\mathcal{S}\ln(2\pi)/2$. By discretizing over a coordinate system $\boldsymbol{\nu}$ with more uniform $|g(\boldsymbol{\nu})|$ on different $\boldsymbol{\nu}\in\mathcal{S}$, $-\frac{1}{2}\ln|g(\boldsymbol{\nu})|$ can be regarded as constant. Therefore, it is reasonable to use only the first novelty term in eq. (5) for parameter learning, and regard the rest terms as constant, because they are less sensitive to the variation of the parameters.

We build an equivalence between a mixture model in the observation space and a divergence-induced prior defined on $\mathcal{S}$. Given an $\boldsymbol{x}$, its associated $\boldsymbol{t}(\boldsymbol{x})$ in the η-coordinates is not on $\mathcal{S}$ but on its boundary $\partial\mathcal{S}$. This is because a single observation is a deteriorated distribution with zero variance. The image of the observation space under the mapping $\boldsymbol{t}(\cdot)$ is $\mathcal{O}\overset{def}{=}\{\boldsymbol{t}(\boldsymbol{x})\}$, It is embedded in $\partial\mathcal{S}$, where FIM degenerates (Amari et al., 2006). We can approach $\mathcal{O}$ from inside $\mathcal{S}$ using a series of equal-entropy surfaces, or level sets of $\psi^\star$, with reducing entropy levels.

Proposition 1. *Given $\mathcal{B}$ and $\boldsymbol{\alpha}$,*

$$\sum_{i=1}^m\alpha_i p(\boldsymbol{x}\,|\,\boldsymbol{\eta}_i)\propto\lim_{\tilde{\boldsymbol{\eta}}\to\boldsymbol{t}(\boldsymbol{x})}p(\tilde{\boldsymbol{\eta}}\,|\,\mathcal{B},\boldsymbol{\alpha})\left(\overset{def}{=}p(\boldsymbol{t}(\boldsymbol{x})\,|\,\mathcal{B},\boldsymbol{\alpha})\right), \quad (6)$$

where the limit is taken through equal-entropy surfaces.

The left-hand-side of eq. (6) is a function in the observation space. On the right-hand-side, $p(\tilde{\boldsymbol{\eta}}\,|\,\mathcal{B},\boldsymbol{\alpha})$ is given by eq. (4), constrained on a level set of $\psi^\star$. By proposition 1, when this level set is close enough to $\partial\mathcal{S}$, these two functions become proportional. The proof is straightforward from eqs. (1), (3) and (4). The likelihood of a sample $\boldsymbol{x}$ can be measured by giving $\boldsymbol{x}$ a small variance, regarding it as a point on $\mathcal{S}$, and computing a divergence-induced prior.

3.2. MDL Networks

An unsupervised mixture modeling *MDL Network* $\mathcal{N}=\{\mathcal{L}_0,\ldots,\mathcal{L}_L\}$ on a statistical manifold is a collection of points (distributions) on $\mathcal{S}\cup\mathcal{O}$, referred to as "cells" and organized in pyramid-shaped layers, as shown in fig. 3. Each layer $\mathcal{L}_l=\{\boldsymbol{\eta}_{l1},\ldots,\boldsymbol{\eta}_{ln_l}\}$ consists of n_l cells, where $n=n_0\geq\cdots\geq n_L$. The ground layer $\mathcal{L}_0=\{\boldsymbol{\eta}_{0i}:\boldsymbol{\eta}_{0i}=\boldsymbol{t}(\boldsymbol{x}_i)\}\subset\mathcal{O}\subset\partial\mathcal{S}$ is fixed by the input samples $\{\boldsymbol{x}_i\}_{i=1}^n$. Any other layer $\mathcal{L}_l\subset\mathcal{S}$ $(1\leq l\leq L)$ consists of free points

[3] In this paper, depending on the context, "$|\cdot|$" denotes either the determinant or the cardinality of a set.

[4] Analyses and proofs are in the supplementary material.

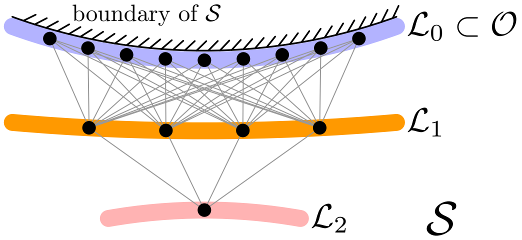
boundary of $\mathcal{S}$
$\mathcal{L}_0 \subset \mathcal{O}$
$\mathcal{L}_1$
$\mathcal{L}_2$
$\mathcal{S}$

Figure 3. An MDL network. The black dots denote the cells, i.e. points on $\mathcal{S}$ and $\mathcal{O}$. The links mean that attraction exists between the endpoints. The thick colored lines represent different layers.

on $\mathcal{S}$, which are to be learned. The size of $\mathcal{N}$ is denoted as $n_1 : \cdots : n_L$ with the sample size n_0 omitted.

Cells of consecutive layers interrelate with each other through pair-wise attraction forces, shown by the links in fig. 3. Using each layer $\mathcal{L}_l$ $(1 \le l \le L)$ as a reference set, and using some mixture weights $\boldsymbol{\alpha}^l = (\alpha_{l1}, \ldots, \alpha_{ln_l})$, we can define a divergence-induced prior $p(\boldsymbol{\eta} \,|\, \mathcal{L}_l, \boldsymbol{\alpha}^l)$. However, this "prior" is not known a priori but is to be learned from data. The learning goal is to reduce the description length, given by the first novelty term in eq. (5) after abandoning constants, of all the cells in $\mathcal{N}$ using a one-level-higher model by minimizing

$$E(\mathcal{N}, A) = -\sum_{l=0}^{L-1}\sum_{i=1}^{n_l} \ln\left(\sum_{j=1}^{n_{l+1}} \alpha_{l+1,j} \exp\left(-D(\boldsymbol{\eta}_{li} \,\|\, \boldsymbol{\eta}_{l+1,j})\right)\right), \tag{7}$$

where $A = (\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^L)$ consists of all the mixture weights. The learning result is a mixture model in the input space, and a stack of higher-level mixture models.

A first justification of the cost function $E(\mathcal{N}, A)$ is the theory of MDL (Rissanen, 1978; 1989), or minimum message length (Wallace & Boulton, 1968). After reasonable simplifications as discussed in subsection 3.1, $E(\mathcal{N}, A)$ measures the cost to gradually describe the data in a simple-to-complex manner. If a person would like to communicate the data, he or she can start from some vague common knowledge $\mathcal{L}_L$, and then tell the codes of all the cells in $\mathcal{L}_{L-1}$, which in turn give another coding scheme to describe $\mathcal{L}_{L-2}$, and so on.

For simplicity, the cost to describe the top-most layer $\mathcal{L}_L$ and the mixture weights A is not measured by $E(\mathcal{N}, A)$. This is because the high-entropy $\mathcal{L}_L$ is close to some common knowledge, e.g., uniform distribution, and the scalars $\{\alpha_{li}\}$ are negligible in storage compared to the often high-dimensional vectors $\{\boldsymbol{\eta}_{li}\}$.

An MDL network corresponds to a spawning process or a directed graphical model (Heckerman, 1995; Jordan et al., 1999), using $\mathcal{L}_L$ to generate $\mathcal{L}_{L-1}$, and using $\mathcal{L}_{L-1}$ to generate $\mathcal{L}_{L-2}$, and so on. Minimizing $E(\mathcal{N}, A)$ implements a maximum a posteriori (MAP) estimator. In the sum in eq. (7), the term corresponding to $l = 0$ measures the fitness of $\mathcal{L}_1$ to the input data in $\mathcal{L}_0$. The rest terms $(l = 1, \ldots, L-1)$ perform network-structured regularization on $\mathcal{L}_1$.

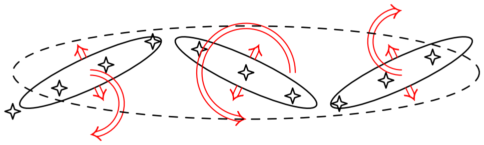

Figure 4. A toy example to show how an MDL network of size 3-1 reduces over-fitting on a dataset of 10 samples (the stars). The red arrows show the regularization strengths by the top-layer (the dashed circle) upon the 3 mixture components (the solid circles).

An immediate advantage of MDL networks is to avoid singular mixture components, which is a known problem of maximum likelihood mixture learning. For example, in fig. 4, a zig-zag pattern over-fits a "line structure". By adding an upper layer with one parent cell, the three mixture components are pulled toward this parent on $\mathcal{S}$, and thus can avoid the boundary $\partial\mathcal{S}$, where singularity occurs.

4. Implementations

4.1. HARDN

To tackle the log-sum terms in eq. (7), a simple way is to relax $\min E(\mathcal{N}, A)$ to $\min \hat{E}(\mathcal{N}, A)$, where $\hat{E}(\mathcal{N}, A)$ is an upper bound of $E(\mathcal{N}, A)$ given by

$$\hat{E}(\mathcal{N}, A) = \sum_{l=0}^{L-1}\sum_{i=1}^{n_l} \min_j \left(-\ln \alpha_{l+1,j} + D(\boldsymbol{\eta}_{li} \,\|\, \boldsymbol{\eta}_{l+1,j})\right). \tag{8}$$

This is exactly the strategy used in "hard" assignment mixture learning (Bishop, 2006; Nielsen, 2012). The algorithm is therefore named HARDN, whose outline is given by alg. 1. Each cell $\boldsymbol{\eta}_{li}$ in the layers $\mathcal{L}_0, \ldots, \mathcal{L}_{L-1}$ is associated with a unique parent cell $\boldsymbol{\eta}_{l+1,j}$, which minimizes $\left(-\ln \alpha_{l+1,j} + D(\boldsymbol{\eta}_{li} \,\|\, \boldsymbol{\eta}_{l+1,j})\right)$. Unlike fig. 3, the links are sparse in HARDN. The algorithm alternates between updating the child-parent associations, and updating all cells in $\mathcal{N}$ with these associations fixed. Its advantage is being simple and efficient. The computational complexity w. r. t the size of $\mathcal{N}$ is $O(\sum_{l=0}^{L-1} n_l n_{l+1})$, which is $O(n)$ if the scale of $\mathcal{L}_1, \ldots, \mathcal{L}_L$ can be neglected as compared to the sample size n. The main memory cost is on storing $\mathcal{N}$.

Alg. 1: $\mathcal{N}$, A = HARDN $(\{\boldsymbol{x}_i\}_{i=1}^n,\, n_1,\, \ldots,\, n_L,\, \gamma)$

```
Fix $\mathcal{L}_0 = \{\boldsymbol{t}(\boldsymbol{x}_i)\}_{i=1}^n$; randomly initialize $A$ and
  $\{\mathcal{L}_l : 1 \le l \le L\}$; store the result into $\mathcal{N}^{(0)}$; $t \leftarrow 0$;
repeat
    // Establish the connections of the network $\mathcal{N}^{(t)}$
    for $l \leftarrow 0$ to $L-1$ do
        foreach cell $\boldsymbol{\eta}_{li}$ in $\mathcal{L}_l$ do
            $j^\star \leftarrow$
              $\arg\min_j \left(-\ln\alpha_{l+1,j} + D(\boldsymbol{\eta}_{li} \,\|\, \boldsymbol{\eta}_{l+1,j})\right)$;
            Create the link $\boldsymbol{\eta}_{li} \rightarrow \boldsymbol{\eta}_{l+1,j^\star}$;
    // Adjust the cells in $\mathcal{N}^{(t)}$ based on the connections
    for $l \leftarrow 1$ to $L$ do
        foreach cell $\boldsymbol{\eta}_{li}$ in $\mathcal{L}_l$ do
            $\boldsymbol{\eta}_{li} \leftarrow$
              relocate($\boldsymbol{\eta}_{li}$, pred($\boldsymbol{\eta}_{li}$), succ($\boldsymbol{\eta}_{li}$), $\gamma$);
            // $\gamma$ is a learning rate
            $\alpha_{li} \leftarrow$ percentage of pred($\boldsymbol{\eta}_{li}$) in $\mathcal{L}_{l-1}$;
    Copy all cells (without connections) from $\mathcal{N}^{(t)}$
      to $\mathcal{N}^{(t+1)}$; set $t \rightarrow t+1$; optionally update $\gamma$;
until convergence;
```

pred$(\cdot)$ denotes the predecessors
succ$(\cdot)$ denotes the successors

4.2. SOFTN

In analogy to "soft" v.s. "hard" assignment (Bishop, 2006), an alternative implementation called SOFTN is to minimize a variational upper bound (Jordan et al., 1999) of $E(\mathcal{N},\, A)$ given by

$$\bar{E}(\mathcal{N},\, A,\, B) = \sum_{l=0}^{L-1}\sum_{i=1}^{n_l}\sum_{j=1}^{n_{l+1}} \beta_{li}^j \left(\ln\frac{\beta_{li}^j}{\alpha_{l+1,j}} + D(\boldsymbol{\eta}_{li} \,\|\, \boldsymbol{\eta}_{l+1,j})\right), \tag{9}$$

where $B = \{\beta_{li}^j\}$, $\forall l,\, i,\, j$, $\beta_{li}^j \ge 0$ and $\sum_{j=1}^{n_{l+1}} \beta_{li}^j = 1$. This $\left(\beta_{li}^1, \cdots, \beta_{li}^{n_{l+1}}\right)$ models a random parent cell of $\boldsymbol{\eta}_{li}$ in $\mathcal{L}_{l+1}$. By simple derivations, a minimizer of $\bar{E}(\mathcal{N},\, A,\, B)$ must satisfy

$$\beta_{li}^j = \frac{\alpha_{l+1,j}\exp\left(-D(\boldsymbol{\eta}_{li} \,\|\, \boldsymbol{\eta}_{l+1,j})\right)}{\sum_{j=1}^{n_{l+1}} \alpha_{l+1,j}\exp\left(-D(\boldsymbol{\eta}_{li} \,\|\, \boldsymbol{\eta}_{l+1,j})\right)}. \tag{10}$$

The corresponding algorithm is omitted due to space limit. Basically, it alternates among updating β_{li}^j, $\alpha_{l+1,j}$ and $\boldsymbol{\eta}_{li}$. Because updating each cell $\boldsymbol{\eta}_{li}$ requires all the cells in $\mathcal{L}_{l-1}$ and $\mathcal{L}_{l+1}$, the computational complexity w. r. t. the size of $\mathcal{N}$ is $O(\sum_{l=1}^{L-1} n_l(n_{l-1}+n_{l+1}))$, which is slower than HARDN but still approximates to $O(n)$ if $n_1,\, \cdots,\, n_L$ are relatively small. Storing $\{\beta_{li}^j\}$, a real vector of size n_{l+1} for each cell $\boldsymbol{\eta}_{li}$, adds a memory overhead as compared to HARDN.

Alg. 2: relocate $\left(\boldsymbol{\eta}, \{(\boldsymbol{\eta}_i^L, w_i^L)\}, \{(\boldsymbol{\eta}_i^R, w_i^R)\}, \gamma\right)$

```
If $\{w_i^L\}$ and $\{w_j^R\}$ are missing, set all of them to 1;
$\boldsymbol{\eta}^L \leftarrow$ weighted arithmetic average of $\{(\boldsymbol{\eta}_i^L, w_i^L)\}$;
Obtain $\boldsymbol{\theta}_1^R, \boldsymbol{\theta}_2^R, \ldots$ by Legrendre transformations;
$\boldsymbol{\theta}^R \leftarrow$ weighted arithmetic average of $\{(\boldsymbol{\theta}_i^R, w_i^R)\}$;
Compute the wavy underlined term in eq. (12);
Compute grad$C = \Delta_\eta^T \partial/\partial\boldsymbol{\eta}$ by proposition 2;
$\boldsymbol{\eta} \leftarrow \boldsymbol{\eta} - \gamma\Delta_\eta$;
```

The underlined procedures are relatively expensive.

4.3. A Key Procedure

Both HARDN and SOFTN minimize a sum of divergences. This is a key procedure to implement MDL networks. If we see $\hat{E}(\mathcal{N},\, A)$ in eq. (8) or $\bar{E}(\mathcal{N},\, A,\, B)$ in eq. (9) as a function of a single cell $\boldsymbol{\eta}$, while fixing all the other parameters, the problem reduces to

$$\begin{aligned}&\min C(\boldsymbol{\eta}),\\ &C(\boldsymbol{\eta}) = \sum_i w_i^L D(\boldsymbol{\eta}_i^L \,\|\, \boldsymbol{\eta}) + \sum_j w_j^R D(\boldsymbol{\eta} \,\|\, \boldsymbol{\eta}_j^R)\end{aligned} \tag{11}$$

w. r. t. some given $\{(\boldsymbol{\eta}_i^L, w_i^L)\}$, $\{(\boldsymbol{\eta}_j^R, w_j^R)\} \subset \mathcal{S} \times \Re^+$. Minimizing the first term on the right-hand-side of eq. (11) leads $\boldsymbol{\eta}$ to a linear combination of $\{\boldsymbol{\eta}_i^L\}$. This interprets the M-step in Estimation-Maximization (EM, Amari, 1995) mixture learning. Minimizing the second term alone leads $\boldsymbol{\theta}(\boldsymbol{\eta})$ to a linear combination of $\{\boldsymbol{\theta}_j^R\}$. Minimizing both terms introduces non-linearity, resulting in a weighted *symmetrized Bregman centroid*, and forming a key difference with EM. A binary searching algorithm for the case $\sum_i w_i^L = \sum_j w_j^R$ was given (Nielsen & Nock, 2009). We seek more general, albeit slower, solutions.

We use natural gradient (Amari, 1998), which defines w. r. t. a cost function an intrinsic gradient flow on $\mathcal{S}$, and shows good properties in machine learning optimization (Amari et al., 2006). The natural gradient of $C(\boldsymbol{\eta})$ is grad$C \overset{def}{=} \left[g^{-1}(\boldsymbol{\eta})\partial C/\partial\boldsymbol{\eta}\right]^T \partial/\partial\boldsymbol{\eta}$, where $\partial/\partial\boldsymbol{\eta}$ denotes the local velocities, i.e., tangent vectors (Jost, 2011), along the η-coordinate curves. By subsection 2.2, $g(\boldsymbol{\eta}) = \partial\boldsymbol{\theta}/\partial\boldsymbol{\eta}$. Therefore, grad$C = (\partial C/\partial\boldsymbol{\theta})^T \partial/\partial\boldsymbol{\eta}$, leading to

Proposition 2.

$$\mathtt{grad}C = \left[w^L(\boldsymbol{\eta}-\boldsymbol{\eta}^L) + w^R \frac{\partial\boldsymbol{\eta}}{\partial\boldsymbol{\theta}}(\boldsymbol{\theta}-\boldsymbol{\theta}^R)\right]^T \frac{\partial}{\partial\boldsymbol{\eta}}, \tag{12}$$

where $w^L = \sum_i w_i^L$, $w^R = \sum_j w_j^R$, $\boldsymbol{\eta}^L = \sum_i w_i^L \boldsymbol{\eta}_i^L / w^L$, *and* $\boldsymbol{\theta}^R = \sum_j w_j^R \boldsymbol{\theta}_j^R / w^R$.

The proof is straightforward from eqs. (3) and (11). Assume the model is represented and updated in the η-coordinates during learning. The underlined term in eq. (12) means to *push-forward* an increment in the θ-coordinates to the η-coordinates. By eq. (3), this term can also be written as $\partial D(\boldsymbol{\eta} \,\|\, \boldsymbol{\eta}^R)/\partial \boldsymbol{\theta}$. A simpler expression can be derived based on the choice of $\mathcal{S}$. Alg. 2 gives one gradient descent step in an iterative procedure to minimize $C(\boldsymbol{\eta})$.

5. Simulations on Gaussian MDL Networks

We implement an MDL network $\mathcal{N}$, where $\mathcal{S}$ is the Gaussian manifold [5]. The purpose is to show how higher levels in $\mathcal{N}$ help *regularize a mixture model* in $\mathcal{L}_1$. This demonstrates a key advantage of a fully communicating hierarchy (see section 1) over the bottom-up approach, where $\mathcal{L}_1$ is learned from $\mathcal{L}_0$ but not from the higher levels.

MDL networks are a family of methods. How the other family members perform, how they compare to other hierarchical modeling approaches, and how an MDL network with descent scale and depth can be useful in real learning tasks, are beyond the scope of this paper.

For simplicity, we fix $\boldsymbol{\alpha}^l$ ($2 \le l \le L$) to be uniform. Only $\boldsymbol{\alpha}^1$ consisting of the weights of $\mathcal{L}_1$ is to be learned. Each cell $\boldsymbol{\eta}_{0i}$ in $\mathcal{L}_0$ is fixed to be a small spherical Gaussian $G(\cdot \,|\, \boldsymbol{x}_i,\, \epsilon I)$, where $\epsilon = 10^{-3}$. This means that, in practice, $\boldsymbol{\eta}_{0i}$ is placed near the boundary $\partial\mathcal{S}$ rather than on $\partial\mathcal{S}$. This *blurring trick* is known to give better empirical results.

In alg. 1, the $\boldsymbol{\mu}$'s in $\mathcal{L}_l$ ($1 \le l \le L$) are initialized (line 1) by the k-means (Arthur & Vassilvitskii, 2007) centroids of the $\boldsymbol{\mu}$'s in $\mathcal{L}_{l-1}$; the Σ's are either initialized by the covariance of the k-means clusters, or the global data covariance. The learning rate γ is adjusted (line 11) online by a bold driver (Battiti, 1989).

In alg. 2, because the $\boldsymbol{\mu}$ of the weighted centroid in eq. (11) can be solved in closed form, the lines $5 \sim 7$ are adapted to

$$
\begin{aligned}
&H \leftarrow \Sigma(\Sigma^R)^{-1};\\
&\boldsymbol{\mu} \leftarrow \left(w^L I + w^R H\right)^{-1}\left(w^L \boldsymbol{\mu}^L + w^R H \boldsymbol{\mu}^R\right);\\
&\Sigma \leftarrow \Sigma + \gamma w^L (\Sigma^L + (\boldsymbol{\mu}^L - \boldsymbol{\mu})(\boldsymbol{\mu}^L - \boldsymbol{\mu})^T - \Sigma)\\
&\qquad + \gamma w^R (\Sigma - H\Sigma).
\end{aligned}
$$

This is based on the same natural gradient as in proposition 2 (derivations omitted). If all cells in $\mathcal{N}$ are full Gaussians, alg. 2 has a cubic complexity in $\dim(\boldsymbol{x})$ due to the matrix inversions. If the cells are diagonal Gaussians, alg. 2 has a linear complexity in $\dim(\boldsymbol{x})$.

Table 1. Datasets used. The columns, in order, are data name, number of samples, dimensionality, the number of instance datasets, and the size of the MDL network.

Name	# samples	$\dim(\boldsymbol{x})$	# datasets	size of $\mathcal{N}$
`faithful`	272	2	10^5	$2:1$
Old Faithful Geyser Data				
`2moons`	10^4	2	10^5	$8:2:1$
From scikit-learn library				
`9blobs`	10^4	2	10^5	$9:3:1$
Synthesized data similar to fig. 1				
`iris`	150	4	10^5	$3:1$
`wine`	178	13	10^5	$3:1$
Both from the UCI repository [6]				
`digit1`	7877	784	10	$n_1:1$
Hand-written digits of "1" in MNIST [7]				

The methods compared, in order, are `GMM` — a vanilla Gaussian mixture model by the scikit-learn machine learning library (Pedregosa et al., 2011); `DP` — a Dirichlet process Gaussian mixture model (Blei & Jordan, 2006) implemented by Haines (Haines & Xiang, 2014), which does not need a pre-specified number of components; `HARD1` — a flat Gaussian mixture model based on the same implementation as `HARDN`, with only $\mathcal{L}_0$ and $\mathcal{L}_1$ but no higher levels; `HARDN`; `SOFT1` — `SOFTN` with only $\mathcal{L}_0$ and $\mathcal{L}_1$; `SOFTN`. Among these methods, `GMM` and `SOFT1` are both variations of EM with the following difference. The E-step in `SOFT1` is based on eq. (10), or the Gaussian-to-Gaussian assignment of $\boldsymbol{\eta}_{0i}$ (with a small variance ϵI) to $\boldsymbol{\eta}_{1j}$. The E-step in `GMM` is based on the sample-to-Gaussian assignment. A similar blurring trick is used by `GMM` by adding ϵI ($\epsilon = 10^{-3}$) to the learned covariances in each iteration.

The datasets used are listed in table 1. For each dataset, a large number of instance datasets are generated based on different random seeds and different splits of training and testing data. The size of the network, given by the last column, is chosen empirically based on prior knowledge. This is only to show the effect of regularization by $\mathcal{N}$ on $\mathcal{L}_1$, under similar configurations with a flat mixture model with only $\mathcal{L}_1$. The model selection will be discussed in section 6.

Figure 5 shows the testing errors measured by the average negative log-likelihood. For each dataset, a small training size and a big training size are used. The results are stable based on the large number of instance datasets. The key observation is that, in all cases, `HARDN` (resp. `SOFTN`) performs better than `HARD1` (resp. `SOFT1`), meaning that the regularization by $\mathcal{N}$ is effective. This improvement is more obvious on a smaller training size. `HARD1` and `HARDN`, although being inconsistent (Bishop, 2006), can gain better performance than `SOFT1` and `SOFTN` if the dataset has explicit clustering structures. `HARDN` is recommended for its

[5] The codes are at `https://git.unige.ch/gitweb/marchand/mdlnetworks`

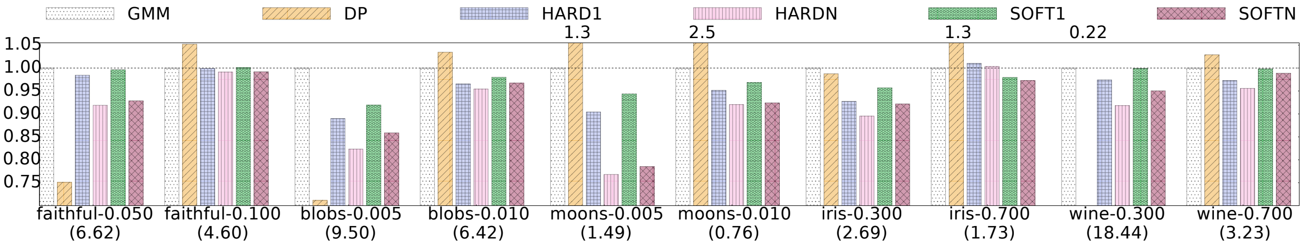

Figure 5. Average testing error over all instance datasets. The labels on the x-axis are in the format "data name–ratio of training set" followed by "(testing error of GMM)". The y-axis shows the testing error of all methods divided by the testing error of GMM.

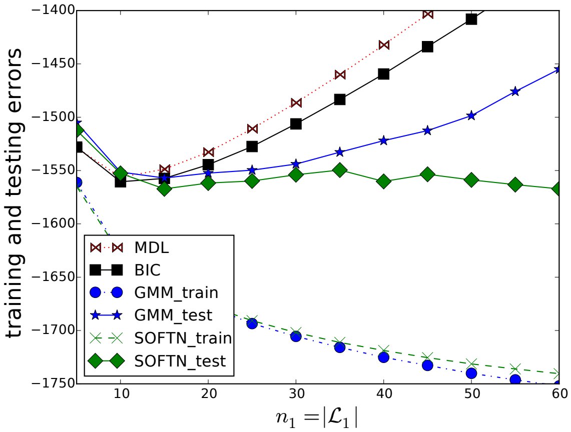

Figure 6. Average training (resp. testing) errors over the training (resp. testing) samples on digit1 against n_1 (n_2 is fixed to 1) with a training : testing ratio of 1 : 9. BIC and MDL are also normalized to be an average value over the training samples.

good performance and its simplicity. GMM performs a bit worse than SOFT1 due to the implementation difference as discussed earlier. The advantage of DP is that, it does not need to be told a "correct" number of components, and the results are more stable across different configurations. In several cases with small training sizes, it performs significantly better. When the training size scales up, it does not catch up with the other methods, because it has less information and it is biased by its priors.

The dataset digit1 has a large dimensionality and an unknown number of clusters. To reduce the model flexibility, we constrain the network cells to be diagonal Guassians. Figure 6 shows the training and testing errors by GMM and SOFTN against n_1, the size of $\mathcal{L}_1$. The testing errors of SOFTN are consistently smaller than GMM. When n_1 is large, SOFTN is much less affected by over-fitting as compared to GMM. The effective regularization means less dependence on choosing a "correct" model scale.

6. Analysis

We investigate the basic properties of an MDL network as a parameter estimator. The following theorem shows that it uncovers certain "truth" given enough samples.

Theorem 3. *If the true distribution is a finite mixture model with the components $\{\boldsymbol{\eta}_i^t\}$, then as $n \to \infty$, $\mathcal{L}_1^\star$ is exactly $\{\boldsymbol{\eta}_i^t\}$ in an optimal MDL network $\mathcal{N}^\star$ which minimizes eq. (7).*

The structural regularization by an MDL network *biases* the cells in $\mathcal{L}_1$ towards the cells in $\mathcal{L}_2$. This strength increases as the size of $\mathcal{L}_2$ decreases. If $\mathcal{L}_2$ does not correspond to the data, the performance will go down. This effect is reduced by MDL networks, because the whole hierarchy adapts to the data. This is different from Bayesian learning, where some vague priors should be fixed manually.

Why a hierarchy leads to a better description as compared to a flat model? It is clear from eq. (7) that

$$\sum_{l=0}^{L-1}\sum_{i=1}^{n_l}\min_j D(\boldsymbol{\eta}_{li} \,\|\, \boldsymbol{\eta}_{l+1,j}) \le E(\mathcal{N},\, A)$$
$$\le \sum_{l=0}^{L-1}\sum_{i=1}^{n_l}\sum_{j=1}^{n_{l+1}} \alpha_{l+1,j} D(\boldsymbol{\eta}_{li} \,\|\, \boldsymbol{\eta}_{l+1,j}). \qquad (13)$$

By eqs. (7) to (9) and (13), $E(\mathcal{N}, A)$ is essentially a nonlinear integration of the divergences from each $\boldsymbol{\eta}_{li}$ to a set $\mathcal{L}_{l+1}$, by converting those divergences to similarities on $\mathcal{S}$ and then converting back. Then, how a "routing network" helps save the total divergence from $\mathcal{L}_0$ to $\mathcal{L}_L$?

Horizontally, a representation layer $\mathcal{L}_l$ could have clustering structures on $\mathcal{S}$. If $\{\boldsymbol{\eta}_{li}\} \subset \mathcal{L}_l$ are clustered around a centroid $\bar{\boldsymbol{\eta}}$, it is more economical to route from $\{\boldsymbol{\eta}_{li}\}$ to $\bar{\boldsymbol{\eta}}$, then from $\bar{\boldsymbol{\eta}}$ to higher layers. It "saves words" to describe the common $\bar{\boldsymbol{\eta}}$ first, then describe the difference of each individual $\boldsymbol{\eta}_{li}$.

Vertically, it is a fundamental property of information divergence to favor more representation layers. Consider a simplified scenario to route from $\boldsymbol{\eta}_1 \in \mathcal{S}$ to $\boldsymbol{\eta}_2 \in \mathcal{S}$.

Theorem 4. *$\forall \boldsymbol{\eta}_1, \boldsymbol{\eta}_2 \in \mathcal{S}$, $\boldsymbol{\eta}_1 \neq \boldsymbol{\eta}_2$, then ① $\exists \boldsymbol{\eta} \in \mathcal{S}$, s.t. $D(\boldsymbol{\eta}_1 \,\|\, \boldsymbol{\eta}) + D(\boldsymbol{\eta} \,\|\, \boldsymbol{\eta}_2) < D(\boldsymbol{\eta}_1 \,\|\, \boldsymbol{\eta}_2)$; ② $\exists \boldsymbol{\eta} \in \mathcal{S}$, s.t.*

$$\begin{aligned} gain(\boldsymbol{\eta}) &\overset{def}{=} D(\boldsymbol{\eta}_1 \,\|\, \boldsymbol{\eta}_2) - D(\boldsymbol{\eta}_1 \,\|\, \boldsymbol{\eta}) - D(\boldsymbol{\eta} \,\|\, \boldsymbol{\eta}_2) \\ &\geq \max\{D(\boldsymbol{\eta}_{lc} \,\|\, \boldsymbol{\eta}_1) + D(\boldsymbol{\eta}_1 \,\|\, \boldsymbol{\eta}_{lc}), \\ &\qquad D(\boldsymbol{\eta}_{rc} \,\|\, \boldsymbol{\eta}_2) + D(\boldsymbol{\eta}_2 \,\|\, \boldsymbol{\eta}_{rc})\}, \end{aligned} \tag{14}$$

where $\boldsymbol{\eta}_{lc}$ in the θ-coordinates is $\boldsymbol{\theta}_{lc} = (\boldsymbol{\theta}_1 + \boldsymbol{\theta}_2)/2$, and $\boldsymbol{\eta}_{rc} = (\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2)/2$.

Example Consider a Bernoulli distribution $p(x=1) = \eta = \exp\theta/(1+\exp\theta)$. Let $\eta_1 = 0.1$, $\eta_2 = 0.5$. Then $\eta_{rc} = 0.3$. By theorem 4, $\exists \eta$, s. t. $gain(\eta) \geq 0.5 \ln \frac{0.5}{0.3} + 0.5 \ln \frac{0.5}{0.7} + 0.3 \ln \frac{0.3}{0.5} + 0.7 \ln \frac{0.7}{0.5} \approx 0.17$.

The triangle inequality of information divergence was known to be not satisfied (Amari & Nagaoka, 2000; Nielsen & Nock, 2009). Theorem 4 is not new in information geometry but presents new meanings in machine learning. By ①, one can always gain a smaller sum of divergence through intermediate stops, as shown in fig. 2 (right). By ②, if $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ are distant, this gain can be large. The same principle holds in an MDL network, or other layered statistical models, where the cost in measured by divergence. Intuitively, a stage-wise incremental description is better than a one-step description, because it costs less divergence on $\mathcal{S}$.

With respect to some given data, $E(\mathcal{N}, A)$ decreases as the size of $\mathcal{N}$ increases. However, adding a new cell to $\mathcal{N}$ involves a cost given by the second to forth terms on the right-hand-side of eq. (5). They are regarded as constant during parameter learning, but has to be considered in post-learning model selection. Within these terms, only the forth term $-\dim\mathcal{S}\ln\delta$ scales with the number of observations of $\boldsymbol{\eta}$, which is explained as follows.

By Rissanen's proposition (Rissanen, 1989), it is reasonable to discretize a parameter space $\mathcal{S}$ up to a precision $\delta \propto 1/\sqrt{n}$, where n is the sample size. A justification (Hansen & Yu, 2001) is that, the error magnitude in parameter estimation scales with $1/\sqrt{n}$ by Cramér-Rao's bound (1946). By regarding $\mathcal{L}_l$, $l = 0, \ldots, L-1$, as the samples of $p(\boldsymbol{\eta} \,|\, \mathcal{L}_{l+1}, \boldsymbol{\alpha}^{l+1})$, we get a criterion $\text{MDL} = E(\mathcal{N}, A) + \frac{\dim\mathcal{S}}{2} \sum_{l=1}^{L} n_l \ln n_{l-1}$, which corrects $E(\mathcal{N}, A)$ by considering the constant terms in eq. (5). This criterion is similar to Bayesian Information Criterion (BIC, Schwarz, 1978) or the two-stage MDL (Hansen & Yu, 2001), except that it measures a stacked representation as a whole. In Figure 6, both MDL and BIC select a 10-component mixture model, which achieves a relatively low testing error with a small model size.

7. Remarks

We propose a novel approach to learn a stacked mixture model. We picture this model as a pyramid-shaped network on a statistical manifold. This network is learned to be tight in the sense of information divergence. This learning is not gradual, layer-by-layer, but at once, through minimizing a global cost function. This fully communicating stack distinguishes from traditional hierarchical learning in letting adjacent layers to regularize each other.

On the intersection of machine learning and information geometry (Banerjee et al., 2005; Garcia et al., 2010; Schwander & Nielsen, 2012; Nielsen, 2012; Liu et al., 2012), a novel step is using symmetrized Bregman centroids (Nielsen & Nock, 2009) as basic learning units, which communicate with both higher-level and lower-level units. As compared to EM seeking information geometric compactness in a two-body system (Amari, 1995), MDL networks seek such compactness in a multi-body system.

On the intersection of information geometry and MDL (Balasubramanian, 1997; Myung et al., 2000), a unified view is demonstrated between the concepts of a small divergence and a short description.

Bayesian mixture learning (Blei & Jordan, 2006; Ghahramani & Beal, 2000) and MDL networks both learn an intermediate representation between the priors and the observations. The former is based on Bayes' rule. The latter is based on information geometric quantities. MDL networks do not need heavy integrations as in Bayesian methods, and are faster in theory and in our practice. The basic cells are all same-type distributions in a common space $\mathcal{S}$. This is simpler to understand and easier to implement.

MDL networks are not neural networks, e.g. (Salakhutdinov & Hinton, 2009), where the non-linearity among a set of neurons is explicitly formalized. Instead, the non-linearity among the cells is implicitly introduced by information divergence. This concept could be interesting to architect learning machines.

The proposed theory is extensible. Different choices of $\mathcal{S}$, different divergence measures, and different network structures, lead to a pool of learning methods.

Acknowledgments

This work has been partly supported by the Swiss Secretariat for Research and Education (SERI) under grant C11.0043 for participation in the COST Action MUMIA.

References

Amari, S. Information geometry of the EM and em algorithms for neural networks. *Neural Networks*, 8(9):

1379–1408, 1995.

Amari, S. Natural gradient works efficiently in learning. *Neural Comput.*, 10(2):251–276, 1998.

Amari, S. and Nagaoka, H. *Methods of Information Geometry*, volume 191 of *Translations of Mathematical Monographs*. AMS and OUP, 2000. (Published in Japanese in 1993).

Amari, S., Park, H., and Ozeki, T. Singularities affect dynamics of learning in neuromanifolds. *Neural Comput.*, 18(5):1007–1065, 2006.

Arthur, D. and Vassilvitskii, S. k-means++: The advantages of careful seeding. In *18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1027–1035, 2007.

Balasubramanian, V. Statistical inference, Occam's razor, and statistical mechanics on the space of probability distributions. *Neural Comput.*, 9(2):349–368, 1997.

Banerjee, A., Merugu, S., Dhillon, I. S., and Ghosh, J. Clustering with Bregman divergences. *JMLR*, 6(Oct): 1705–1749, 2005.

Battiti, R. Accelerated backpropagation learning: Two optimization methods. *Complex systems*, 3(4):331–342, 1989.

Bishop, C. M. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer-Verlag New York, Inc., 2006.

Blei, D. M. and Jordan, M. I. Variational inference for Dirichlet process mixtures. *Bayesian Anal.*, 1(1):121–143, 2006.

Bregman, L. M. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Comput. Math. Math. Phys.*, 7(3):200–217, 1967.

Čencov, N. N. *Statistical Decision Rules and Optimal Inference*, volume 53 of *Translations of Mathematical Monographs*. AMS, reprint edition, 2000. (Published in Russian in 1972).

Cramér, H. *Mathematical Methods of Statistics*. Princeton Univ. Press., 1946.

Efron, B. Defining the curvature of a statistical problem (with applications to second order efficiency). *Ann. Stat.*, 3(6):1189–1242, 1975.

Garcia, V., Nielsen, F., and Nock, R. Hierarchical Gaussian mixture model. In *ICASSP*, pp. 4070–4073, 2010.

Ghahramani, Z. and Beal, M. J. Variational inference for Bayesian mixtures of factor analysers. In *NIPS 12*, pp. 449–455. MIT Press, 2000.

Goldberger, J. and Roweis, S. Hierarchical clustering of a mixture model. In *NIPS 17*, pp. 505–512. MIT Press, 2005.

Haines, T. S. F. and Xiang, T. Background subtraction with Dirichlet processes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(4):670–683, 2014.

Hansen, M. H. and Yu, B. Model selection and the principle of minimum description length. *JASA*, 96(454):746–774, 2001.

Heckerman, D. A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, 1995.

Heller, K. A. and Ghahramani, Z. Bayesian hierarchical clustering. In *ICML*, pp. 297–304, 2005.

Hinton, G. E., Revow, M., and Dayan, P. Recognizing handwritten digits using mixtures of linear models. In *NIPS 7*, pp. 1015–1022. MIT Press, 1995.

Jeffreys, H. An invariant form for the prior probability in estimation problems. *Proc. R. Soc. A*, 186(1007):453–461, 1946.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. *Mach. Learn.*, 37(2):183–233, 1999.

Jost, J. *Riemannian Geometry and Geometric Analysis*. Universitext. Springer, 6th edition, 2011.

Krishnamurthy, A., Balakrishnan, S., Xu, M., and Singh, A. Efficient active algorithms for hierarchical clustering. In *ICML*, pp. 887–894, 2012.

Liu, M., Vemuri, B. C., Amari, S., and Nielsen, F. Shape retrieval using hierarchical total Bregman soft clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(12):2407–2419, 2012.

Myung, J., Balasubramanian, V., and Pitt, M. A. Counting probability distributions: differential geometry and model selection. *PNAS*, 97(21):11170–11175, 2000.

Nielsen, F. k-MLE: A fast algorithm for learning statistical mixture models. *CoRR*, abs/1203.5181, 2012.

Nielsen, F. Cramer-Rao lower bound and information geometry. *CoRR*, abs/1301.3578, 2013.

Nielsen, F. and Nock, R. Sided and symmetrized Bregman centroids. *IEEE Trans. Inf. Theory*, 55(6):2882–2904, 2009.

Nielsen, F., Boissonnat, J. D., and Nock, R. Bregman Voronoi diagrams: Properties, algorithms and applications. *Discrete Comput. Geom.*, 44(2):281–307, 2010.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *JMLR*, 12: 2825–2830, 2011.

Rao, C. R. Information and accuracy attainable in the estimation of statistical parameters. *Bull. Cal. Math. Soc.*, 37(3):81–91, 1945.

Rasmussen, C. E. The infinite Gaussian mixture model. In *NIPS 12*, pp. 554–560. MIT Press, 2000.

Rissanen, J. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

Rissanen, J. *Stochastic Complexity in Statistical Inquiry Theory*. World Scientific Publishing, 1989.

Salakhutdinov, R. and Hinton, G. E. Deep Boltzmann machines. In *AISTATS*, pp. 448–455, 2009.

Schwander, O. and Nielsen, F. Learning mixtures by simplifying kernel density estimators. In *Matrix Information Geometry*, pp. 403–426. Springer, 2012.

Schwarz, G. Estimating the dimension of a model. *Ann. Stat.*, 6(2):461–464, 1978.

Shannon, C. E. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

Vincent, P. and Bengio, Y. Manifold Parzen windows. In *NIPS 15*, pp. 825–832. MIT Press, 2003.

Wallace, C. S. and Boulton, D. M. An information measure for classification. *Comput. J.*, 11(2):185–194, 1968.

Ward, J. H. Hierarchical grouping to optimize an objective function. *J. Amer. Statist. Assoc.*, 58(301):236–244, 1963.