Professor Gheorghe RUXANDA, PhD
E-mail: gheorghe.ruxanda@csie.ase.ro
Sorin OPINCARIU, PhD Candidate
The University of Economic Studies, Bucharest

BAYESIAN NEURAL NETWORKS WITH DEPENDENT DIRICHLET PROCESS PRIORS. APPLICATION TO PAIRS TRADING

Abstract*. Bayesian neural networks combine the universality of the neural networks with the principled uncertainty quantification of the Bayesian approach. The black-box character of neural networks makes it difficult establishing appropriate priors for the weights of the neural network. In this paper we propose a hierarchical model where the prior distribution of the network weights is drawn from a Dirichlet process mixture model. We further extend the model to dependent Dirichlet process mixtures to allow the model to account for non-stationarity in the data. The neural network with dependent Dirichlet priors is used to model a pairs trading experiment.*

Keywords*: Learning Machine, Neural Networks, Bayesian Neural Networks, Dirichlet Process, Mixture distribution*.

JEL Classification : C02, C11, C45, C46, C63 .

1. Introduction

The past decade has seen a revival and spectacular successes of neural networks, revival brought about by advances in training algorithms and in hardware capabilities (especially parallel architectures like GPU). Despite their successes the neural network architecture is capable of outputting only point-wise forecasts without being able of offering any quantification of uncertainty of network's forecast. The early works of MacKay [1] and Neal [2] proposed a principled Bayesian model of neural networks where priors are placed on the weights of connections between the neurons. While the backpropagation introduced by Rumelhart in [3] has transformed the problem of training of neural networks into a problem of optimization of a cost function, the

DOI: 10.24818/18423264/52.4.18.01

training of the Bayesian neural networks (BNN) is a problem of Bayesian inference of the posterior distribution of network's weights.

The majority of the Bayesian neural network literature has focused on the problem of inference of the posterior distribution of the network's weights. In [2] Neal proposed a modified Markov chain Monte Carlo (MCMC) sampling scheme based on Hamiltonian dynamics to estimate the posterior distribution of network's weights. Although asymptotically guaranteed to correctly estimate the posterior distribution, the sampling-based methods are computationally expensive which limits their applicability to relatively small networks and datasets.

To overcome the shortcomings of MCMC schemes, various variational inference procedures have been introduced in the literature. The variational inference introduced by Jordan in [4] was used by Graves in [5] to approximate the posterior distribution. The variational inference replaces the computationally expensive sample-based approach with the approximation of posterior distribution with a parametric family of easier to sample distribution. The variational inference thus consists in finding a distribution among the parametric family that minimizes a measure of discrepancy (usually Kullback-Leibler divergence) between the posterior distribution and the approximating distribution. The literature on variational inference has focused either on finding more expressive approximation families (see for example the work on normalizing flows of Rezende et al in [6]) or on extending measure of divergence between probability distribution (see for example Hernadez-Lobato in [7] using the α-divergences or operator variational inference of Ranganath in [8]).

Although a considerable literature was devoted to the problem of inference of Bayesian neural networks relatively few papers addressed the problem of selecting the prior distribution of neural network's weights. Since MacKay and Neal the default choice for BNN priors seems to be the normal distribution (N(0,1)) However the black box character of neural network makes it difficult to impose an informative prior distribution on network weights. Following this intuition Lee in [9] proposed using the non-informative priors introduced by Bernardo in its reference analysis [10], positing that since the neural networks are black boxes our priors for the weights should reflect our ignorance. Another approach in model selection of BNN is offered by Gohsh et al in [11] who uses a horseshoe prior to enforce the sparsity among the network's connection.

In this paper we address the problem of choosing the prior distribution for network's weights from a Bayesian nonparametric point of view. The Dirichlet

processes introduced by Ferguson in [12] acts as a distribution of probability on the space of distributions of probability. Using a Dirichlet process prior in our opinion delegates the problem of selecting the prior distribution to the data itself without forcing the analyst to make unnecessary assumptions. The Dirichlet processes are specified by a base measure (probability distribution) and a concentration parameter. The expectation of a Dirichlet process is equal to the base measure and the concentration parameter controls how far from the base measure are each draw from the Dirichlet process.

In this paper we discuss the following:

- **Section 2** How to specify a neural network with Dirichlet Process priors. Being discrete processes the Dirichlet processes are not suited as priors for BNN. We use the Dirichlet process mixture model which extends to continuous case the Dirichlet process. Choosing as base measure the normal distribution we ensure that the prior of the neural network is closed to the normal distribution (closeness controlled by the concentration parameter) while at the same time retaining the flexibility of deviations from normality should the data require. The model proposed is a hierarchical one in the sense that we endow the concentration parameter with a prior distribution of its own.
- **Section 3** The specification of dependent Dirichlet process priors (see MacEachern [13]) for neural network weights to account for non-stationarity of the data. We use a Dirichlet process driven by a Gaussian process (see Rassmusen [14]) to account for the temporal evolution of financial data's regimes.
- **Section 5** Application of the BNN with Dirichlet process priors to model the evolution of correlated pairs of financial assets (pairs trading)

1 Dirichlet Process Mixture Priors

1.1 Dirichlet Process Mixture Model

Dirichlet distribution: Following Ruxanda in [15] we describe Dirichlet distribution as the multivariate generalization of the beta distribution whose probability density function parametrized by the vector α is:

$$f(x;\alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^{K} x_i^{\alpha_i - 1},$$

where: $B(\alpha) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{K} \alpha_i)}$.

The Dirichlet distribution is conjugate to the multinomial distribution which makes them very useful to the Bayesian analyses.

Dirichlet Process: The Dirichlet Process was first constructed by Ferguson in [16] as a probability measure on the space of probability distributions

Definition 2.1. Let G_0 be a finite measure on a Polish space X. A random measure P is said to be a Dirichlet process if for every finite measurable partition $\{B_1...,B_n\}$ the joint distribution of $(P(B_1),...,P(B_n))$ is a Dirichlet distribution $Dir(G_0(B_1),...G_0(B_n))$

The Dirichlet process thus defined is a discrete measure with probability 1. Ferguson also proves that a Dirichlet process has an expectation $E[P] = G_0$ that is the expectation of a Dirichlet process is the base measure of that process.

Sethuraman in [17] has defined an alternative constructive definition of the Dirichlet process called the stick breaking construction:

Definition 2.2. Let $\alpha > 0$ and G_0 a probability measure on X. The random discrete probability measure:

$$P = \sum_{k=1}^{\infty} C_k \delta_{\Phi_k} \quad (1)$$

where $\Phi_1,\Phi_2 \cdots \sim_{iid} G_0$, δ is the Dirac measure and the weights are constructed as:

$$C_k = V_k \prod_{j=1}^{k-1}(1 - V_j) \quad (2)$$

with $V_1,V_2,\cdots \sim_{iid} Beta(1,\alpha)$.

The random probability measure P is said to be sampled from a Dirichlet process $DP(\alpha G_0)$. In Figure 1 we present some samples extracted from a Dirichlet process with a standard normal base distribution. We can observe the concentration of samples around base measure(bold) and the discrete character of the samples.

The almost sure discrete character of Dirichlet process makes them inconvenient in many modeling situations. Following the definition of mixtures of probability distributions for a mixing measure θ:

$$p(x) = \sum_{k \in N} c_k p(x|\phi_k) = \int p(x|\phi)\theta(d\phi) \quad (3)$$

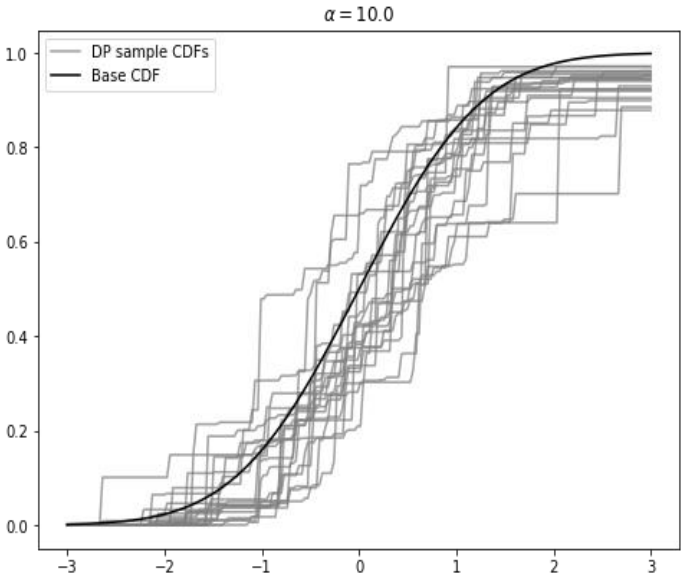


Figure 1: Samples extracted from a Dirichlet process

Ferguson in [12], Lo in [18] and Escobar et al [19] define the Dirichlet Process Mixture model (DPM) using the stick-breaking measure (1) as a mixing measure:

$$P \sim DP(\alpha G_0) \tag{4}$$

$$\Phi_1, \Phi_2, \ldots | P \sim_{iid} P \tag{5}$$

$$X_i \sim p(x|\Phi_i) \tag{6}$$

where $p(x|\Phi)$ is parametric family of continuous density functions.

Therefore, we can write the density of a DPM making use of the weights (2):

$$p(x) = \sum_{k \in N} C_k p(x|\Phi_k) \tag{7}$$

1.2 DPM priors for Bayesian neural networks

Neal in [2] defines the Bayesian neural network (BNN) by attaching prior probability distribution to the weights *W* of the neural network defined by a repeated application of a nonlinear transfer function to an affine transform. A neural network is constructed by stacking layers of neurons of the form:

$$y_i = h\left(\sum_{j=1}^{n} w_j \cdot x_{ij}\right) \quad i = 1, \dots, n \tag{8}$$

where n is the number of neurons per layer and x_{ij} are inputs to neuron i.

A BNN is defined by attaching a prior probability to the network's weights $W \sim p(W)$, the learning process for a dataset $D = \{(x_i, y_i)\}$ of pairs of inputs and target values is the Bayesian inference:

$$p(W|D) \frac{\prod_i p\left(y_i \middle| f(x_i, W)\right) p(W)}{p(D)}, \tag{9}$$

where f is the nonlinear likelihood function obtained by composing the layers. The problem of selection a prior distribution for the weights of Bayesian neural networks is complicated by the black box character of neural networks. While Neal [2] and MacKay [1] has proposed the use of normal distributions as prior probabilities for the weights of the network we propose using a DPM scale prior.

We formalize the Bayesian neural network with Dirichlet process mixtures (DP-BNN) as follows:

$$\begin{aligned}
&\alpha \sim \Gamma(a, a); \quad V_1, V_2, \dots \sim_{iid} \text{Beta}(1, \alpha); \\
&C_i = V_i \prod_{j=1}^{i-1} (1 - V_j); \\
&\tau_1, \tau_2, \dots \sim_{iid} \Gamma(b, b); \quad \lambda_1, \lambda_2, \dots \sim \text{Uniform}(0, c); \\
&w \sim \sum_{i=1}^{\infty} C_i \mathcal{N}(0, (\lambda_i \tau_i)^{-1}); \\
&\qquad\qquad \sigma_\epsilon \sim C^+(0, c); \quad y \sim \mathcal{N}(f(x, w), \sigma_\epsilon^2),
\end{aligned} \tag{10}$$

where $f(x,w)$ is the nonlinear transfer function from inputs to outputs obtained by composing the neural network's layers.

The DP-BNN model above proposes a scale mixture prior distribution for the network's weights. The parametric mixture components are normal distribution of mean 0, here we follow the literature on neural networks which suggest that the mean of weights should be normalized to 0 mean. The base measure of the Dirichlet process mixture is a product of a gamma and a uniform distribution, here we follow the guidelines of Gelman [20] for choosing the priors for variance parameters.

The DP-BNN model proposes a hierarchical model where the complexity of the prior distribution is allowed to increase alongside the complexity of the data.

The stick breaking weights v_i although not strictly decreasing are stochastically decreasing making the contribution of a large number of components unlikely enforcing a stochastic parsimony. The stochastically decreasing behavior of stick breaking weights can be seen in Figure 2 where only the first two components are significantly greater than zero.

We also assume a independent normally distributed noise term whose standard deviation σ is HalfCauchy distributed $C^{+}(0,c)$.

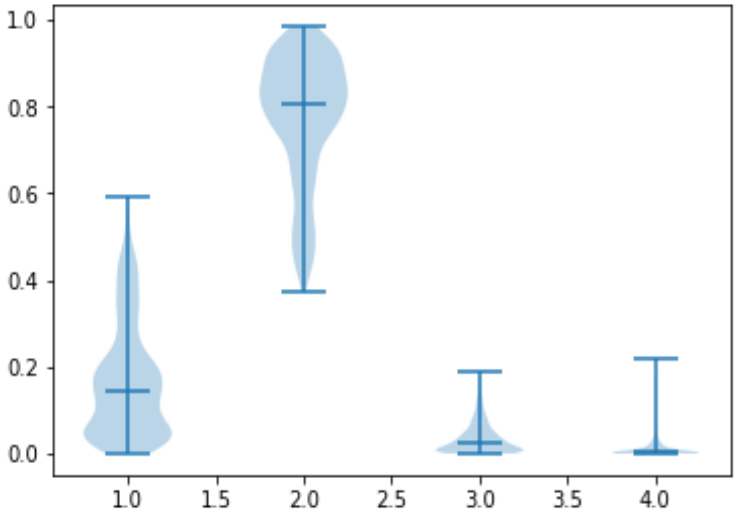


Figure 2: Stochastic decreasing weights

2 Dependent Dirichlet Process Priors

The Dirichlet process mod prior of the Bayesian neural network assumes that all the weights are drawn iid from the base measure G_0 hence an implicit assumption of strong stationarity is being placed on the dataset D.

To account for the non-stationarity typical of financial datasets we will use the dependent Dirichlet process mixtures introduced by MacEachern in [13] as priors for the weights of Bayesian neural networks.

In the dependent Dirichlet process model the stick breaking construction is extended to account for the dependence on a set of covariates X. The following generalization of the Sethuraman is being obtained:

$$G_x = \sum_{k=1}^{\infty} c_k \delta_{\phi_{x,k}}, \quad (11)$$

where $\varphi_k = \{\varphi_{x,k} x \in X\}$ are realizations from a stochastic process. If we ensure that $c_{x,k}$ are independent given x then G is marginally a Dirichlet process.

The model (11) is called the single p model and is a simplification of the general dependent Dirichlet model of MacEachern [13] where only the locations $\varphi(x,k)$ depend on the covariate x while the stick breaking weights are the same as in the Dirichlet process model. We consider that the single p model introduces a form of dependency that can be exploited while at the same time enjoying computational efficiencies over the general dependent Dirichlet process

Our strategy to introduce dependency of networks weights on a set of covariates (e.g. time) is to put a dependent Dirichlet prior on the network weights. Our choice of driving stochastic process mirrors MacEachern's a Gaussian process. Our reason for choosing a Gaussian process to model the stochastic process φ_k is motivated by Gaussian processes' property of being universal priors over a large space of functions and hence many types of dependency can be modeled Furthermore a Gaussian process allows us to experiment with various degrees of smoothness of the covariate curve. In this work we choose an exponential squared covariance function introduced by Rasmussen in [14]. The Gaussian process with a squared exponential covariance function samples smooth functions whose amplitude of change from one point to another is controlled by a parameter of the covariance function called length scale Using exponential squared covariance functions allows us to specify BNN priors that don't change too much for different values of the covariate thus offering some stability to the BNN. One can envision specifying a prior distribution to the length scale although this is an approach that is not pursued in this work.

We specify the following time dependent Gaussian process with a squared exponential covariance function:

$$k(t,t') = \exp\left(\frac{\left||t-t'|\right|^2}{2l^2}\right)$$

$$g_t \sim \mathcal{G}P\left(0, k(t,t')\right) \quad (12)$$

where l length scale controlling the amount of covariate movement necessary for f_t to change significantly.

With the Gaussian process thus specified, we can proceed to introduce dependence on time to the model (10). Therefore, we get the following formal specification for the Bayesian neural network with dependent Dirichlet prior (DDP-BNN):

$$\alpha \sim \Gamma(a,a);\ \ \beta_1,\beta_2,\ldots \sim_{iid} \mathrm{Beta}(1,\alpha)$$
$$v_i = \beta_i \prod_{j=1}^{i-1}\left(1-\beta_j\right)$$
$$\tau_1,\tau_2,\ldots \sim_{iid} \Gamma(b,b);\ \ \lambda_1,\lambda_2,\ldots \sim \mathrm{Uniform}(0,c);$$
$$w \sim \sum_{i=1}^{\infty} v_i \mathcal{N}(0,(\lambda_i\tau_i)^{-1}g_t);\ \ \sigma_\epsilon \sim C^+(0,c);$$
$$y \sim \mathcal{N}(f(x,w),\sigma_\epsilon^2). \tag{13}$$

The model (13) is a minimal modification of model (10) and is also relatively parsimonious to the constant weights model; only a single parameter is supplementary introduced: the length scale *l*.

3 Related work

Since the seminal work of the Neal [2] and MacKay [1] most of the work has focused beginning with Graves [5] on various variational inference scheme to address the computational tractability of inference in Bayesian neural network. Various variational inference schemes were proposed of notice we mention the normalized flows approach of Rezende [6] the generalization to operator-based schemes of Ranganath [8] and the use of α-divergences by Hernandez-Lobato [7].

On the issue of prior selection for Bayesian neural network we mention the use of horseshoe prior to impose sparsity by Ghosh et al [11] and the imposition of noise at the input level in the noise contrastive prior by Hafner et al [21].

4 Applications to pairs trading

In this section we illustrate the applicability of the Bayesian neural networks models developed in the sections above to a challenging financial data situation.

While much of academic literature has been devoted to forecasting financial datasets understood as time series (see for example Krollner in [22]) we will devote this section to present the application of Bayesian neural networks with Dirichlet process priors to pairs trading.

Pairs trading is one of the most popular trading strategies in the financial markets and it is based according to Gatev in [23] on finding two assets that have historically moved together and trading the spread between the prices of those two assets. When

the spread between the prices widens beyond a statistical measure of significance (usually Z-scores) one will sell the overpriced asset and buy the underpriced asset in the expectation that the spread between them will revert to a historical mean.

The traditional approach to modeling the joint evolution of pairs of financial assets is to find pairs of assets whose returns are cointegrated and model their evolution with linear models. We consider this approach to include in the universe of tradeable pairs, only to pairs that are linearly correlated and their relationship stationary as highly restrictive. As showed by Cont in [24] the evolution of financial returns is characterized by the lack of linear dependence, non-stationarity, relatively low signal to noise ratio.

We posit that Bayesian neural network with Dirichlet process prior are suitable to model evolution of pair of assets because:

- The universal approximator property of neural networks allows us to approximate a large class of nonlinear dependencies.
- Being a Bayesian model one can easily quantify and propagate the uncertainty of the network's forecasts. This quantification of uncertainty allows the analyst to better cope with low noise to signal ratios and to devise better risk management tools. Therefore, when the confidence of the network is high, we can place a larger bet relative to the situation when the confidence is low and one should not trade on the forecast made.
- The dependent Dirichlet process priors allow the network to adapt to the non-stationarity encountered in the data.

To illustrate the points above we offer the example of a classic pair trading strategy: an ETF of gold miners (GFI) against an ETF replicating the evolution of the price of gold (GLD).

When looking to Figure 3 one can easily identify various temporal regimes of the joint evolution of GFI relative to GLD. Although the evolution seems tobe piecewise linear or weakly nonlinear we can easily spot the non-stationarity in the evolution of the relationship. Because of the non-stationarity one cannot find a single function that can approximate well the data as it can be inspected graphically in Figure 3.

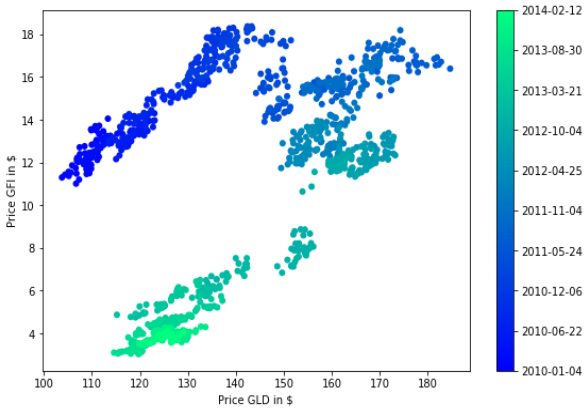


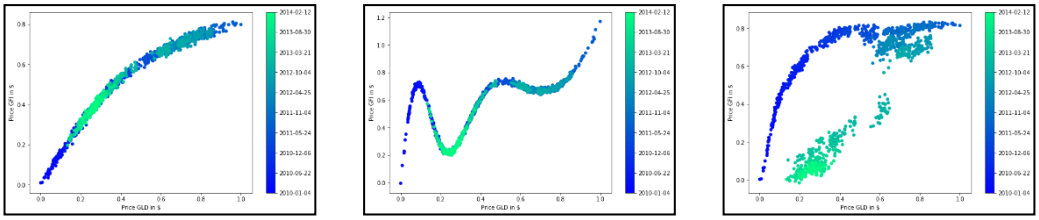
Figure 3: The time joint evolution of GFI vs GLD

We propose the following types of Bayesian neural networks to model the evolution of GFI vs GLD:

- Classical Bayesian feedforward neural network. Here we assign a N(0,1) prior to each layer of the network. This model will serve as a baseline against which the models introduced in this paper.
- DP-BNN Bayesian neural network with Dirichlet process prior. Here we endow each weight of the network with a Dirichlet mixture prior. As mixtures can approximate arbitrarily complex distribution, this prior allows us to specify priors as complex as data requires. A major drawback of this model is that like the classical BNN it cannot account for the non-stationarity in the evolution of the pair.
- DDP-BNN Bayesian neural network with dependent Dirichlet process prior. Here we endow each weight with a dependent Dirichlet process mixture driven by a Gaussian process. This model should be able to model the non-stationarity in the data.

Table 1: Root mean square error for Bayesian neural networks

Model	RMSE
BNN	0.08754
DP-BNN	0.059467
DDP-BNN	0.012111



(a) BNN (b) DP-BNN (c) DDP-BNN

Figure 4: The fit of the GFI/GLD data

All the experiments were done using pymc3 [25] a probabilistic programming package running in python. For the Dirichlet process models we decided to cutoff the infinite sum to 6 components but as we can see from the Figure 2 only the first two components have significant weight, hence the finite truncation does not adversely impact the results of the inference.

For the inference we used the No U Turn sampler of Hoffman et al [26] a variant of the Hamiltonian Monte Carlo algorithm presented by Neal in [2]. The table 1 presents the root mean square error for all 3 Bayesian neural network considered; we used the expectation of the posterior distribution as the point forecast, as we consider the expectation as representative of the typical set of the posterior distribution.

If we examine graphically the fit of each model in Figure 4, we observe that the dependent Dirichlet mixture priors manage to recover the multiple temporal regimes in the data while the simple Dirichlet mixture prior tries to fit a single nonlinear curve that tries to mediate among the different temporal regimes. The simple Bayesian neural network presents the worst fit that is also evident in the RMSE presented in table 1.

5 Conclusion

In this paper we showed how we can specify more complex priors for Bayesian neural networks than the normal distribution prior usually found in the literature. Our approach is to use a Bayesian nonparametric mixture priors whose complexity is allowed to grow with the complexity of the data. Moreover, we showed how one can introduce the dependence on one covariate (time) with the minimum increase of complexity of the Bayesian inference (only one supplementary parameter) by using a dependent Dirichlet mixture process driven by a Gaussian process. We consider that Bayesian neural networks are best fitted for financial data modeling task as their

principled approach to quantifying and propagating the uncertainty is a natural fit to the requirements of risk management usually found in the financial sector.

REFERENCES

[1] **Bernardo M. Jose (2005),** ***Reference Analysis***. *Handbook of statistics*, 25:17–90;

[2] **Cont Rama (2001),** ***Empirical Properties of Asset Returns: Stylized Facts and Statistical Issues***; *Quantitative Finance*, Volume 1/2001;

[3] **Dungaciu Dan, Cristea Darie, Dumitrescu Diana Alexandra, Pop Stefan Zaharie (2018)**, ***Stratfor vs. Reality (1995-2025). Dilemmas in Global Forecasting***, *Romanian Journal of Economic Forecasting*, Volume 21, Issue 1/ 2018;

[4] **Escobar D. Michael,West Mike (1995),** ***Bayesian Density Estimation and Inference Using Mixtures***. *Journal of the American statistical association*, 90(430):577– 588;

[5] **Ferguson S Thomas (1973),** ***A Bayesian Analysis of some Nonparametric Problems***. *The annals of statistics*, pages 209–230;

[6] **Ferguson S. Thomas (1983),** ***Bayesian Density Estimation by Mixtures of Normal Distributions***. In *Recent advances in statistics*, pages 287–302. *Elsevier*;

[7] **Gatev Evan, Goetzmann N. William, Rouwenhorst K Geert (2006),** ***Pairs Trading: Performance of a Relative-value Arbitrage Rule***. *The Review of Financial Studies*, 19(3):797–827;

[8] **Gelman Andrew et al. (2006),** ***Prior Distributions for Variance Parameters in Hierarchical Models****;* (Comment on Article by Browne and Draper). *Bayesian analysis*, 1(3):515–534;

[9] **Ghosh Soumya,Doshi-Velez Finale (2017),** ***Model Selection in Bayesian Neural Networks via Horseshoe Priors***. *arXiv preprint arXiv:1705.10388*;

[10] **Graves Alex (2011),** ***Practical Variational Inference for Neural Networks***. In *Advances in neural information processing systems*, pages 2348–2356;

[11] **Hafner Danijar, Tran Dustin, Irpan Alex, Lillicrap Timothy, Davidson James** (**2018**), ***Reliable Uncertainty Estimates in Deep Neural Networks Using Noise Contrastive Priors***. *arXiv preprint arXiv:1807.09289*;

[12] **Hernandez-Lobato Jose Miguel, Yingzhen Li, Rowland Mark, Hernandez-Lobato Daniel, Thang Bui,Turner Richard Eric (2016),** ***Black-box α Divergence Minimization,*** *arXiv:1511.03243v2 [stat.ML];*

[13] **Hoffman D. Matthew, Gelman Andrew (2014),** ***The no-u-turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo***.*Journal of Machine Learning Research*, 15(1):1593–1623;

[14] **Krollner Bjoern, Vanstone J. Bruce, Finnie R. Gavin (2010),** ***Financial Time Series Forecasting with Machine Learning Techniques: A Survey***. In *ESANN*;

[15] **Lee KH Herbert (2003),** ***A Noninformative Prior for Neural Networks***. *Machine Learning*, 50(1-2):197–212;

[16] **Lo Y. Albert (1984),** ***On a Class of Bayesian Nonparametric Estimates: I. Density Estimates***. *The annals of statistics*, pages 351–357;

[17] **MacEachern N. Steven(2000),** ***Dependent Dirichlet Processes***. *Unpublished manuscript, Department of Statistics; The Ohio State University*, pages 1–40;

[18] **MacKay J.C. David(2003),*****Information Theory, Inference and Learning Algorithms***. *Cambridge University Press*;

[19] **Neal M. Radford (2012),*****Bayesian Learning for Neural Networks***, volume 118. *Springer Science & Business Media*;

[20] **Scheau Mircea Constantin, Pop Stefan Zaharie (2017)**, ***Methods of Laundering Money Resulted from Cyber-crime***, *Economic Computation and Economic Cybernetics Studies and Research*, Volume 51, Issue 3/2017**;** Academy of Economic Studies, Bucharest;

[21] **Ranganath Rajesh, Tran Dustin, Altosaar Jaan,Blei David (2016),** ***Operator Variational Inference***. In *Advances in Neural Information Processing Systems*, pages 496–504;

[22] **Rasmussen Carl Edward (2000),** ***The Infinite Gaussian Mixture Model***. In *Advances in neural information processing systems*, pages 554–560;

[23] **Rezende Danilo Jimenez, Shakir Mohamed (2015),** ***Variational Inference with Normalizing Flows***. *arXiv preprint arXiv:1505.05770*;

[24] **Rumelhart E. David, Hinton E. Geoffrey, Williams J. Ronald (1986),** ***Learning Representations by Back-Propagating Errors***. *Nature*, 323(6088):533;

[25] **Ruxanda Gheorghe (2001),** ***Analiza datelor***; *ASE Publishing*; Bucharest;

[26] **Salvatier John, Wiecki V.Thomas,Fonnesbeck Christopher (2016),** ***Probabilistic Programming in Python Using pymc3***. *PeerJ Computer Science*, 2:e55;

[27] **Sethuraman Jayaram (1994),** ***A Constructive Definition of Dirichlet Priors***. *Statistica sinica*, pages 639–650;

[28] **Wainwright J. Martin, Jordan I. Michael, et al (2008),** ***Graphical Models, Exponential Families, and Variational Inference***. *Foundations and Trends*R *in Machine Learning*, 1(1–2):1–305;