
Learning Exponential Families

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 Recently much attention has been paid to implicit probabilistic models – models
2 defined by mapping a simple random variable through a complex transformation,
3 often a deep neural network. These models have been used to great success for
4 variational inference, generation of complex data types, and more. In most all of
5 these settings, the goal has been to find a *particular member* of that model family:
6 optimized parameters index a distribution that is close (via a divergence or clas-
7 sification metric) to a target distribution (such as a posterior or data distribution).
8 Much less attention, however, has been paid to the problem of *learning a model*
9 *itself*. Here we define implicit probabilistic models with specific deep network
10 architecture and optimization procedures in order to learn intractable exponential
11 family models (*not* a single distribution from those models). These exponential
12 families, which are central to some of the most fundamental problems in probabilis-
13 tic inference, are learned accurately and scalably, allowing operations like posterior
14 inference to be executed directly and generically by an input choice of natural
15 parameters, rather than performing inference via optimization for each particular
16 realization of a distribution within that model. We demonstrate this ability across
17 a number of non-conjugate exponential families that appear often in the machine
18 learning literature.

1 Introduction

19 IPMs are used a lot; they matter but aren't perfect. Set context:

- 21 • generative probabilistic models are the fundamental object of bayesian modeling [1].
- 22 • classic issue has been tractability-expressivity tradeoff. choosing and even defining a
23 statistical model is hard [2, 1]
- 24 • However, these models are chosen to be generic and flexible, rather than in the classic sense
25 of instantiating a set of statistical assumptions concerning the process of generating some
26 data. somehow offering an explanation or a structured assumption about data. This is not
27 bad per se, but leaves much to be desired in terms of modeling.
- 28 • recently implicit probabilistic models have been used a lot, and for VI in particular [3, 4, 5]
29 (more blei stuff here)
- 30 • while offering many advantages, two shortcomings: represent a potentially too-flexible
31 model, and are used to find single posterior distributions (often on local variables).
- 32 • VI has to re-learn on every dataset; yes it can amortize across points from the same dataset,
33 but not across datasets in the same model. Given the frequency of certain non-conjugate
34 models appearing – hierarchies of Dirichlet distributions, log Gaussian Poisson models, etc –
35 this seems needless to continue considering this as an “intractable” exp fam.

- recently much attention has been paid to bijective neural networks, networks that admit tractable density calculations. An old idea with new options.
- Also we always sample from intractable families via some transformations [6]; the fact that some have known constructions (ratio of gammas, Bartlett decomposition, etc) should not distract from the fundamental nature of this process.

Here we learn an exp fam *model*:

- We investigate the problem of learning exp fams, not individual distributions. Inherent in all the above approaches is an algorithmic procedure to select a *single* distribution $q_\theta(z)$ from among the *model* \mathcal{Q} . Implicit in this effort is the belief that \mathcal{Q} is suitably general to contain the true distribution of interest, or at least an adequately close approximation.
- Many models are exp fams, though intractable. [7]. It is worth revisiting whence that intractability arises, often just because hard work has not yet been put into deriving transformation samplers. Many intractable distributions encountered in machine learning belong to exponential families. In rare cases these distributions are tractable due to either known conjugacy in the problem setup (such as the normal-inverse-Wishart), or due to careful numerical work historically that has made these distributions computationally indistinguishable from tractable (eg the Dirichlet). [6]. not a known mapping from other simpler distributions (eg the Wishart via the Bartlett decomposition), an inversion, transformation-rejection algorithm, or similar custom numerical solution [6]. It is intriguing then to reflect upon the success that deep neural networks have offered to function approximation, and ask to what extent we can automate this numerical process, widening the class of effectively tractable exponential family distributions.
- EFNs allow the embodiment of modeling assumptions without sacrificing expressivity
- EFNs include neural net observation models in many cases, so don't despair. (like a VAE generator)
- concept here is to learn something we care about already and get the usual benefits of learning a restricted model space [8, §7, for example]
- we parameterize a network whose input is the natural parameters of the exponential family being learned
- the output of this *parameter* network is the parameters ϕ of a bijective neural network that allows density to be calculated.
- Can use this as an initializer if more specific training is required.

Our contributions include:

- novel architecture to learn a model, not a particular member
- stochastic optimization that samples over the model space: sampling both natural parameters (the family member to be learned) and data points (the observed density points)
- our choice of exp fam produces a linear regression type problem in KL divergence. We leverage the natural parameterization of exponential families to derive a novel objective that is amenable to stochastic optimization.
- empirical results confirming against ground truth in known “tractable” families like the Dirichlet, inverse Wishart, and Gaussian.
- empirical results demonstrating inference performance in common “intractable” families including the hierarchical dirichlet, the log Gaussian Poisson.
- Demonstration that there is surprisingly little performance loss training a single posterior vs an entire model, advocating its broader use, at least as an initializer if not as an amortizer.

People use lots of implicit probability models:

Across machine learning, including ABC [9], GANs [10], VAEs [3, 4], density estimation [11], and their many follow-ons (too numerous to cite in any detail), models that specify a distribution via the nonlinear transformation of latent random variable. Such implicit probability models have a rich history and newer contributions (density networks all the way through [12]). Some equations:

$$q_{\theta}(z)$$

Also use the proper notation of the density implied by the pushforward measure of the function $f_{\theta_{\#}}$ if useful. The two central uses are at present generative distributions of interesting data types (as in GANs), and for variational inference. Regardless, all of these use cases specify a *model* (or variational family) $\mathcal{Q} = \{q_{\theta} : \theta \in \Theta\}$, and then minimize a suitable loss $\mathcal{L}(q, p)$ over $q \in \mathcal{Q}$. In the case of VI p is the posterior (or the unnormalized log joint) and \mathcal{L} is the KL divergence (or so called ELBO), in GAN p is the sample density of a (large) dataset and \mathcal{L} is the adversarial objective whose details do not matter here.

VI in this style can be seen and is often referred to as amortized inference, in the sense that it is in fact less expressive than full mean field (eg the VAE), but (1) it offers the usual benefits of a restricted model space in terms of bias-variance tradeoff [8] (Fig 7.2), and (2) it offers test time speed up by pretraining (hence the term amortization). Here we offer what can be seen as a different sort of amortization, over datasets themselves. The exp fam may be challenging to learn, but then it can be used at trivial cost. We will focus more on the distinction with variational inference later. We use IPM vs generative to clarify that we are not simply dealing in the inference case, but the more general problem of learning probabilistic models (nor just single members of these models).

2 Exponential family networks

2.1 Implicit probability models via density networks

bants. defines a \mathcal{Q} . Why this is coherent Θ defines quite a big \mathcal{Q} , and indeed the subject of compressibility, generalization, etc is of keen interest to many [?]. So actually the space of distributions is quite large, and in many cases certainly larger than it needs be. Why? Well, we know precisely the parameter space of the exponential family; it is defined by the *natural* parameters $\eta \in \mathbb{R}^p$ (or whatever we choose there).

Density networks are an old idea [13], as are neural networks to fit a probability model to data [14, 15].

We choose flow networks [16]. And "implicit generative models aka density networks" (or rather, density networks are the instantiation of an IGM with deep nets, which is effectively synonymous these days. And invertible networks In that vein probably definitely cite invertible/bijective deep nets in general [17, 18, 16, 19, 11, 20]. Note that what norm flows [16] did is make it tractable and scalable and in the modern VAE style, and even that is probably overstating the case. That makes these comparisons legitimate and apples to apples. Gaussianization is an old idea that this is basically the inverse of [21]; same idea in more depth and that argues for the normal prior in [22]. Really the norm flow is not so special as this is a well established classic idea.

More generally there has been a lot of attention to making these more flexible in structured variational inference. Any generalization of this is also dandy though, so could use a mean field approach (standard) or any of the things that go beyond mean field, either classically [23, 24]; this is called structured variational inference or newer stuff [25] [26], to name but a few.

2.2 Exponential families

bants. Pitman-Koopman Lemma [27, §3.3.3] Defines an M .

Why this is important. Exp fams are awesome and fundamental. Also [7] rightly point out that many many inference problems can be cast as exponential families. Can we cast the VAE encoder network as a suitable exp fam... sure I think that's right; the network parameters of z form the statistics, and then the observations are η 's.

Common examples in the ML community include hierarchical Dirichlet and log Gaussian Poisson.

Note briefly that one common model that this does not conveniently include is local latent variable models like LDA and logistic regression, as they define larger and larger exp fams as they go (yes they are exp fams, but not of a fixed parameterization under sampling).

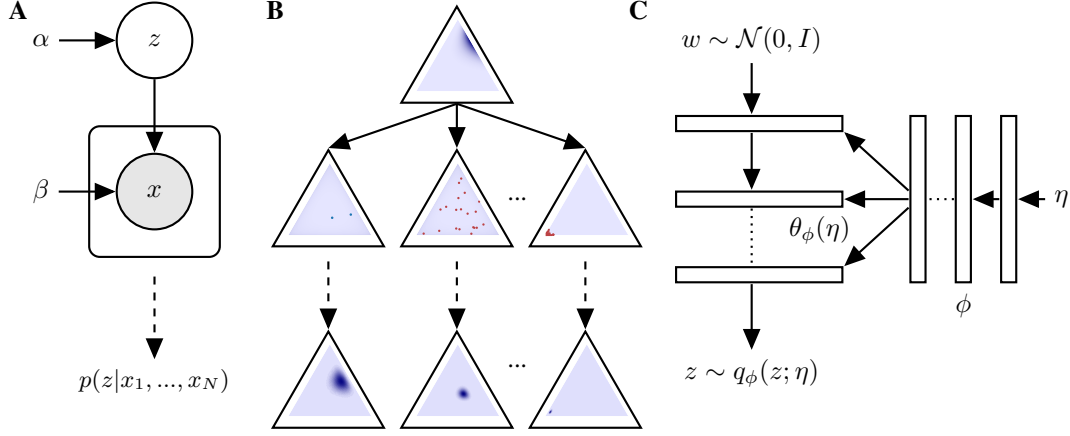


Figure 1: Learning exponential families. A shows the graphical model, emphasizing conditional iid sampling. B shows Dirichlet prior (a density), conditional Dirichlet observations (some observed points in the simplex), and then the posteriors learned by an EFN. SRB to fill in these triangles. C shows the EFN network schematic.

132 Note somewhere that the natural parameter space needs to be considered in general. That is, not all η
 133 lead to a valid distribution (standard fact, see for example [7]). In practice that's not often a problem,
 134 as the space is known for most distributions one uses, and when one composes them in a posterior
 135 scheme (for example), this is inherited (eg the normal covariance...). So we skip that here. But yes in
 136 general that needs to be considered.

137 2.3 Exponential family networks

138 includes the network definition of Fig 1c, the objective, and the optimization algorithm.
 139 This should not be confused with "Learning to learn by gradient descent by gradient descent" [28]
 140 Another related work is that this is somehow the dual of MEFN [29], or a generalization of the dual
 141 problem. In the wainwright and jordan sense of forward and backward mappings. Stuff on sampling
 142 from Gibbs distributions (max ent models), and sampling from exp fams generally, with MCMC and
 143 such.
 144 Note that this objective can also produce approximations of the log partition, via essentially linear
 145 regression; more nuanced schemes are recommended [30]. We don't explore that here.

146 2.4 Relation to variational inference

147 We have already covered related work; here we scrutinize EFNs in terms of VI.
 148 We are interested in perhaps the most classic inference problem:

$$p(z|x) \propto p(z) \prod_{i=1}^n p(x_i|z)$$

149 shown with the attached plate model (not local latents). Supposing as is often the case that the
 150 likelihood is a member of the s exp fam, we have:

$$p(z|x) \propto \exp \left\{ \left[\sum_{i=1}^n s(x_i) \right]^\top [t(z)] + g_0(\alpha, z) \right\}$$

Important to distinguish carefully from VI. In a sense VI does parameterize a family: given data, you get local variational parameters and that parameterizes a density (like a regular VAE). Inference networks are exclusively used to data to amortize with a global set of parameters a variational distribution, not a model. Of course it is in a sense a model, but that's a bunch of normals. The sampling mechanism is easy (Gaussian).

where the natural parameters of the sampling distribution are indexed by the latent parameter on which we want to inference (z). Here I've written the prior as arbitrary, and possibly not exp fam, which is fine, since this is still an exp fam in the sense of, for a fixed α , the function g_0 can just be viewed as a sufficient statistic. Even if α is not fixed though, we can sample over that too to learn the whole fam (but maybe not if we want to infer it?). Regardless, life is simpler to make sense of if we take an exp fam prior $g_0(\alpha, z) = \alpha^\top t_0(z)$, and then the desired posterior is an intractable exp fam, but still just an exp fam.

Note: consider changing all z to θ to remind the average reader that we're doing real bayesian inference and not just run of the mill VI with local latents in a nonlinear dimension reduction setting. Perhaps an important reminder that most all of VAE and such are for inference of local latents, and that's a little bit too bad. We fix that.

Another key idea that EFNs enable is to ask if learning the $\theta(\eta)$ network leads to better VI in terms of inference networks, since it is apparently appropriately regularized and can just take suff stats. That's testable if we have time.

In a restricted technical sense, rather close: VAE and other black box VI that uses reparameterization results in a conditional density $q_\phi(z|x)$. If we consider η as x , then sure yes the previous stuff specifies a model $\mathcal{Q}_{VAE} = \{q_\phi(z|x) : x \in X\}$. But that's a little silly, and any way that is very often a normal family with variational parameters specified by (a deep function of) x . Much closer is Figure 2 in Rezende and Mohamed, where like here they use a network to index the *parameters* of the normalizing flow. In that case it's a function of x the observation, and as such that network is an inference network; here it's a function of η and as such is a parameter network. That's just nomenclature, so naturally the next question is do they differ at some other level. Yes, distinctly. The other term implied in a VI (or norm flow VAE style as they use) is the expected log joint $E_{q_\phi(x)}(\log p_\theta(x, z))$. Now sure that's a loss function on x, z , so then when we look at that same term in EFN we see $E_{q_\phi(\eta)}(\eta^\top t(z))$, which sure also looks like a loss function on η, z . And yes, they are both unnormalized (in the sense that VI is an ELBO / joint $p(x, z)$ and EFN lacks the normalizer because it's constant, so we're not getting a KL estimate). A picky difference is that the exp family doesn't really correspond to a proper unnormalized log joint (though I suppose it could), as there is not a prior on η in the objective (but is that just ignoring $p(\eta)$ in our sampling scheme?). But yes if we want to be reductionist and pedantic [use nicer words] in general we could see this as a specific case where $x = \eta$ and thus we are learning a family just as in the inference case. Or rather, we are putting the data in as sufficient stat (computation of natural parameters), but that's nonobvious. And for example we are giving in the bayesian logistic regression example full datasets for inference instead of single data points. To make this as close as possible, we write $p(\eta|z) = \frac{1}{A(t(z))} \exp\{\eta^\top t(z)\}$. That's the "likelihood" of an EFN in some wonky sense. So this reveals the mechanical differences: first, $t(z)$ is not a deep generative model with parameters θ , but rather it is a fixed set of sufficient statistics that define the exp fam. Next, there is no clear prior $p(z)$, which is critical to understanding how VI behaves (see Hoffman and Johnson ELBO surgery paper, also Duvenaud's <https://arxiv.org/pdf/1801.03558.pdf>). So yes there is a hand wavy sense in which EFN is a specific case of norm flow, but of course it is. And anyway norm flow is a specific case of a DNN architecture or Helmholtz machine or deep density network (Ripple and Adams). This is just rambling but good to have all perspective here. Ok so what to do? First, then we need to produce really compelling results focusing on when learning an exp fam is key. Second we need some very tight language to draw this distinction without seeming a small tweak on normalizing flows. One way to do this is the restricted model class argument, a la Fig 7.2 in Hastie and Tibshirani. Another is to actually produce a conditional exp fam, as in something indexed on both x and η . Third, possible novelties in norm flows, like triple spinners or other better choices than planar flows (yuck).

Another point is that it's unknown if posterior contraction can be well modeled. As in, we know that most VI NF type things are conditioned on a single data point, so the posterior variance can tend to be rather homogenous. One more contribution is to offer that contraction study; as we get more data points we will get more posterior contraction, so this tests the ability of this model to learn that.

207 Key distinctions:

- 208 • narrow mechanical sense this is VI with an observation of the natural parameters, namely
209 the sample exp fam over all data. but that's pedantic.
- 210 • no generative model in the usual sense: yes, we can consider a prior and then some
211 observation model as the genrative model, but in any event it's not a neural net.
- 212 • we lack a finite data set X , so the objective is technically different. We stipulate a distribution
213 and then this is expectation over that model space, a KL or a KL to the broader joint with η .
214 This is concretely different, as we typically use a fixed size dataset X so we can calculate
215 the ELBO over the

216 3 Results

217 Introductory remarks and then comments about architectural particulars, including planar flow
218 networks of [16]. Note Number of panar flows is always D (intrinsic dimensionality of flows), units
219 per layer ramping is always the same function of D . The number of layers in the theta network is
220 always a function of D - will probably just always use 8 layers. Remember

221 3.1 Tractable exponential families

222 *Chapter 2: Fig 2 and 3 and 4* Ground truth toy examples, etc.

223 Figure 2.

224 Single EFN:

225 Panel A: r^2 throughout training

226 Panel B: KL throughout training

227 Panel C: Distributin of MMD p values

228

229 Figure 3:

230 EFN performance by dimensionality

231 Panel A: Dir KL for NF1 and EFN

232 Panel B NIW KL for NF1 and EFN

233 Panel C: Gaussian KL for NF1 and EFN

234

235 3.2 Intractable exponential families

236 *concept and detail of hiearchical Dir and TNP*

237 **Hierarchical Dirichlets** Hierarchical dirichlets are useful and have some history; most notable is
238 with the Hierarchical Dirichlet Process [31], but historically this was done in the finite case also [32].
239 Here is some math. Note that this matters for multi-corpus LDA generally as well [33, 34].

240 **Truncated- and log-normal Poisson** used a lot [35][36][37, 38] Figure 4:

241 EFN performance by dimensionality on Dir-Dir

242

243 3.3 Truncated-Gaussian and log-Gaussian Poisson

244 Figure 4:

245 Truncated

246

247 Fig 5. The intractable posterior inference example. **Key real data result.** Learn the full posterior
248 family for some problem (see ideas below). Then get some data X . Then find the posterior distribution
249 for that data by indexing the natural parameters (as in, just plugging in the correct choice of η , which
250 is after all some function of the prior and X). That gives the EFN posterior $q(z|X)$. (Possible
251 preceeding figure: show its properties, show a low-d picture, show its non-Gaussianity). Now, as

Alternative 1 do full norm flow variational inference (explore all of ϕ space with the full flow network model \mathcal{Q}), which is to say $\arg \min_{\phi} KL(q_{\phi}||p)$: the key difference here is that, while you have the *same exact* flow network architecture, now you have to optimize over ϕ with a limited single dataset. As Alternative 2, be literal to Figure 2 of the Norm Flow VI paper, give the sufficient statistics of that $K=1$ dataset, and learn an EFN from scratch. This alternative is important because it is the most specific (but kind of annoying, hence alternative 1) interpretation of norm flow VI paper.

Now, PANEL A of this figure shows performance as a size of the dataset. This will likely show that when the dataset gets small, this "traditional VI" will get arbitrarily bad (can't learn a network); eventually, there will be so much data that the VI will match or outperform the EFN... outperform because VI can focus specifically on this distribution rather than over the whole family, so the EFN has less effective data for this η (but not because it has a broader range of models, since we believe the EFN contains the closest member). Alternative 2 should do shittier across the board than alt 1, I think? Performance metric should be ELBO on some held out data or something like that (it's a posterior, so log likelihood doesn't really make sense). Test data anyway. Check VI papers for usual metrics. PANEL B of this figure shows performance as dimension of the problem grows. Pick some middle dataset size, then repeat same performance metric as in Panel A for a range of dimensionalities of the exponential family. VI will generalize to test data worse and worse as dimensionality grows, but EFN will learn the family less well on its computational budget. This could go either way but will be interesting regardless. I suppose we should also have those panels for training data. A key point to make here is that one great virtue of EFNs is is learning a restricted model, which should demonstrate the usual bias-variance tradeoff (see for example Hastie and Tibshirani book, Fig 7.2). Maybe that's Panel A. Or Figure 4 is bias-variance and some sample posteriors in 2-d (showing how nicely it works), and then Fig 5 is the above performance, with both train and test. Notice one pain here is that Panel B requires training a new EFN at every dimensionality. Sorry.

This will be for one real example X . As such, to get error bars, just take a big dataset and randomly subsample. Then the posterior performance is really for that very dataset, so the sem is coherent and the right thing to calculate/show. Important to clarify that doing so *does not* test how well this does across the entire exp fam, but just this one posterior. To test that, we do it in simulation: generate *many datasets* X , then do the above for every one of them. Same computation for EFN (since its just plugging in a dataset), but VI alternatives 1 and 2 now need to be rerun for every dataset. And it's still simulated data, not really offering something fundamentally more than Fig 3 (well ok it's an intractable model, but I'm not sure that offers so much).

284 4 Appendix

285 Exponential form of posterior for Dirichlet-Dirichlet

286 $\mathbf{z} \sim \text{Dir}(\boldsymbol{\alpha}_0)$

287 $\mathbf{x}_i \sim \text{Dir}(\beta \mathbf{z})$

288 $p(\mathbf{z}) \propto \exp(\boldsymbol{\alpha}_0^T \log(\mathbf{z}) - \sum_{d=1}^D \log(z_d))$

289 $p(\mathbf{x}_i | \mathbf{z}) \propto \exp(\beta \mathbf{z}^T \log(\mathbf{x}_i) - \sum_{d=1}^D \log(x_{i,d}) - (\sum_{d=1}^D \log(\Gamma(\beta z_d)) - \log(\Gamma(\beta \sum_{d=1}^D z_d))))$

290 $p(X | \mathbf{z}) \propto \exp(\beta \mathbf{z}^T [\sum_{i=1}^N \log(\mathbf{x}_i)] - \sum_{i,d=1}^{N,D} \log(x_{i,d}) - N(\sum_{d=1}^D \log(\Gamma(\beta z_d)) - \log(\Gamma(\beta \sum_{d=1}^D z_d))))$

291

292 $p(\mathbf{z} | X) \propto p(\mathbf{z})p(X | \mathbf{z})$

293 $\propto \exp(\boldsymbol{\alpha}_0^T \log(\mathbf{z}) - \sum_{d=1}^D \log(z_d))$

294 $\exp(\beta \mathbf{z}^T [\sum_{i=1}^N \log(\mathbf{x}_i)] - \sum_{i,d=1}^{N,D} \log(x_{i,d}) - N(\sum_{d=1}^D \log(\Gamma(\beta z_d)) - \log(\Gamma(\beta \sum_{d=1}^D z_d))))$

295 We don't care about the term that just has x in it.

296 $p(\mathbf{z} | X) \propto \exp(\boldsymbol{\alpha}_0^T \log(\mathbf{z}) + \beta [\sum_{i=1}^N \log(\mathbf{x}_i)]^T \mathbf{z} - \sum_{d=1}^D \log(z_d) - N(\sum_{d=1}^D \log(\Gamma(\beta z_d)) - \log(\Gamma(\beta \sum_{d=1}^D z_d))))$

297 $p(\mathbf{z} | X) \propto \exp\left(\begin{pmatrix} \boldsymbol{\alpha}_0 - \mathbf{1} \\ \sum_{i=1}^N \log(\mathbf{x}_i) \\ -N \\ -N \end{pmatrix}^T \begin{pmatrix} \log(\mathbf{z}) \\ \beta \mathbf{z} \\ \log(\Gamma(\beta \mathbf{z})) \\ \log(\Gamma(\beta \sum_{d=1}^D z_d)) \end{pmatrix}\right)$

298 This seems right to me. I moved β for the second element of the natural parameters to be over with
299 his other β -friends in the sufficient statistics.

300 Here's a more cleaned up version:

$$p(\mathbf{z} | X) \propto \exp\left\{\left[\begin{pmatrix} \boldsymbol{\alpha}_0 - \mathbf{1} \\ \sum_{i=1}^N \log(\mathbf{x}_i) \\ -N\mathbf{1} \\ -N \end{pmatrix}\right]^\top \begin{bmatrix} \log(\mathbf{z}) \\ \beta \mathbf{z} \\ \log(\Gamma(\beta \mathbf{z})) \\ \log(\Gamma(\beta \mathbf{1}^\top \mathbf{z})) \end{bmatrix}\right\} \triangleq \exp\{\boldsymbol{\eta}^\top t(\mathbf{z})\}$$

301 or just using the Beta function:

$$p(\mathbf{z} | X) \propto \exp\left\{\left[\begin{pmatrix} \boldsymbol{\alpha}_0 - \mathbf{1} \\ \sum_{i=1}^N \log(\mathbf{x}_i) \\ -N \end{pmatrix}\right]^\top \begin{bmatrix} \log(\mathbf{z}) \\ \beta \mathbf{z} \\ \log(B(\beta \mathbf{z})) \end{bmatrix}\right\} \triangleq \exp\{\boldsymbol{\eta}^\top t(\mathbf{z})\}$$

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments go at the end of the paper. Do not include acknowledgments in the anonymized submission, only in the final paper.

References

References

- [1] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*, volume 2. CRC press Boca Raton, FL, 2014.
- [2] Peter McCullagh. What is a statistical model? *The Annals of Statistics*, 30(5):1225–1267, 2002.
- [3] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv*, 12 2013.
- [4] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.
- [5] Michalis Titsias and Miguel Lázaro-Gredilla. Doubly stochastic variational bayes for non-conjugate inference. In *International Conference on Machine Learning*, pages 1971–1979, 2014.
- [6] Luc Devroye. *Non-uniform random variate generation*. Springer-Verlag, New York, 1986.
- [7] Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.
- [8] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.
- [9] Michael U Gutmann, Ritabrata Dutta, Samuel Kaski, and Jukka Corander. Statistical inference of intractable generative models via classification. *arXiv preprint arXiv:1407.4981*, 2014.
- [10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.
- [11] George Papamakarios, Iain Murray, and Theo Pavlakou. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pages 2335–2344, 2017.
- [12] Shakir Mohamed and Balaji Lakshminarayanan. Learning in implicit generative models. *arXiv*, 10 2016.
- [13] Wenda Zhou, Victor Veitch, Morgane Austern, Ryan P Adams, and Peter Orbanz. Compressibility and generalization in large-scale deep learning. *arXiv preprint arXiv:1804.05862*, 2018.
- [14] D. J. C. MacKay and M. N. Gibbs. Density networks. In *Statistics and Neural Networks*, pages 129–146. Oxford, 1997.
- [15] Peter Dayan, Geoffrey E Hinton, Radford M Neal, and Richard S Zemel. The helmholtz machine. *Neural computation*, 7(5):889–904, 1995.
- [16] Geoffrey E Hinton, Peter Dayan, Brendan J Frey, and Radford M Neal. The "wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995.
- [17] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.
- [18] Leemon Baird, David Smalenberger, and Shawn Ingkiriwang. One-step neural network inversion with pdf learning and emulation. In *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on*, volume 2, pages 966–971. IEEE, 2005.

- [19] Oren Rippel and Ryan Prescott Adams. High-dimensional probability estimation with deep density models. *arXiv preprint arXiv:1302.5125*, 2013.
- [20] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.
- [21] Jörn-Henrik Jacobsen, Arnold Smeulders, and Edouard Oyallon. i-revnet: Deep invertible networks. *arXiv preprint arXiv:1802.07088*, 2018.
- [22] Scott Saobing Chen and Ramesh A Gopinath. Gaussianization. In *Advances in neural information processing systems*, pages 423–429, 2001.
- [23] Esteban G Tabak, Eric Vanden-Eijnden, et al. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217–233, 2010.
- [24] Lawrence K Saul and Michael I Jordan. Exploiting tractable substructures in intractable networks. In *Advances in neural information processing systems*, pages 486–492, 1996.
- [25] David Barber and Wim Wiegierinck. Tractable variational structures for approximating graphical models. In *Advances in Neural Information Processing Systems*, pages 183–189, 1999.
- [26] Matthew Hoffman and David Blei. Stochastic structured variational inference. In *Artificial Intelligence and Statistics*, pages 361–369, 2015.
- [27] Dustin Tran, David Blei, and Edo M Airoldi. Copula variational inference. In *Advances in Neural Information Processing Systems*, pages 3564–3572, 2015.
- [28] Christian Robert. *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media, 2007.
- [29] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, pages 3981–3989, 2016.
- [30] Gabriel Loaiza-Ganem, Yuanjun Gao, and John P Cunningham. Maximum entropy flow networks. *International Conference on Learning Representations*, 2017.
- [31] George Papamakarios and Iain Murray. Distilling intractable generative models. In *Probabilistic Integration Workshop at Neural Information Processing Systems*, 2015.
- [32] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.
- [33] David JC MacKay and Linda C Bauman Peto. A hierarchical dirichlet language model. *Natural language engineering*, 1(3):289–308, 1995.
- [34] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
- [35] Jonathan K Pritchard, Matthew Stephens, and Peter Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, 2000.
- [36] Yuanjun Gao, Evan W Archer, Liam Paninski, and John P Cunningham. Linear dynamical neural population models through nonlinear embeddings. In *Advances in Neural Information Processing Systems*, pages 163–171, 2016.
- [37] Ryan Prescott Adams, Iain Murray, and David JC MacKay. Tractable nonparametric bayesian inference in poisson processes with gaussian process intensities. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 9–16. ACM, 2009.
- [38] John P Cunningham, Krishna V Shenoy, and Maneesh Sahani. Fast gaussian process methods for point process intensity estimation. In *Proceedings of the 25th international conference on Machine learning*, pages 192–199. ACM, 2008.

389 [39] John P Cunningham, M Yu Byron, Krishna V Shenoy, and Maneesh Sahani. Inferring neural
390 firing rates from spike trains using gaussian processes. In *Advances in neural information*
391 *processing systems*, pages 329–336, 2008.

392 From Blei review on VI. The development of variational techniques for Bayesian inference followed
393 two parallel, yet separate, tracks. Peterson and Anderson (1987) is arguably the first variational
394 procedure for a particular model: a neural network. This paper, along with insights from statistical
395 mechanics (Parisi, 1988), led to a flurry of variational inference procedures for a wide class of models
396 (Saul et al., 1996; Jaakkola and Jordan, 1996, 1997; Ghahramani and Jordan, 1997; Jordan et al.,
397 1999). In parallel, Hinton and Van Camp (1993) proposed a variational algorithm for a similar neural
398 network model. Neal and Hinton (1999) (first published in 1993) made important connections to the
399 expectation maximization (EM) algorithm (Dempster et al., 1977), which then led to a variety of
400 variational inference algorithms for other types of models (Waterhouse et al., 1996; MacKay, 1997).

401 Salimans, T. and Knowles, D. (2014). On using control variates with stochastic approximation for
402 variational Bayes. arXiv preprint arXiv:1401.1022.

403 Salimans, T., Kingma, D., and Welling, M. (2015). Markov chain Monte Carlo and variational
404 inference: Bridging the gap. In International Conference on Machine Learning, pages 1218? 1226.

405 Ranganath, R., Gerrish, S., and Blei, D. (2014). Black box variational inference. In Artificial
406 Intelligence and Statistics.

407 Hoffman, M. D. and Blei, D. M. (2015). Structured stochastic variational inference. In Artificial
408 Intelligence and Statistics.

409 Possibly some of Burda, Y., Grosse, R., & Salakhutdinov, R. (2016). Importance Weighted Autoen-
410 coders. In International Conference on Learning Representations. Damianou, A. C., & Lawrence, N.
411 D. (2013). Deep Gaussian Processes. In Artificial Intelligence and Statistics. Dayan, P., Hinton, G. E.,
412 Neal, R. M., & Zemel, R. S. (1995). The Helmholtz Machine. *Neural Computation*, 7(5), 889?904.
413 <http://doi.org/10.1162/neco.1995.7.5.889> Dinh, L., Sohl-Dickstein, J., & Bengio, S. (2016). Density
414 estimation using Real NVP. arXiv.org. Harville, D. A (1977). Maximum likelihood approaches
415 to variance component estimation and to related problems. *Journal of the American Statistical*
416 *Association*, 72(358):320?338. Hinton, G. and Van Camp, D (1993). Keeping the neural networks
417 simple by minimizing the description length of the weights. In *Computational Learning Theory*,
418 pp. 5?13. ACM. Johnson, M. J., Duvenaud, D., Wiltchko, A. B., Datta, S. R., & Adams, R. P.
419 (2016). Composing graphical models with neural networks for structured representations and fast
420 inference. arXiv.org. Kingma, D. P., & Welling, M. (2014). Auto-Encoding Variational Bayes. In
421 International Conference on Learning Representations. Kingma, D. P., Salimans, T., & Welling, M.
422 (2016). Improving Variational Inference with Inverse Autoregressive Flow. arXiv.org. Louizos, C.,
423 & Welling, M. (2016). Structured and Efficient Variational Deep Learning with Matrix Gaussian
424 Posteriors. In International Conference on Machine Learning. Maaloe, L., Sonderby, C. K., Sonderby,
425 S. K., & Winther, O. (2016). Auxiliary Deep Generative Models. In International Conference
426 on Machine Learning. MacKay, D. J., & Gibbs, M. N. (1999). Density networks. *Statistics and*
427 *neural networks: advances at the interface*. Oxford University Press, Oxford, 129-144. Mnih, A., &
428 Rezende, D. J. (2016). Variational inference for Monte Carlo objectives. In International Conference
429 on Machine Learning. Ranganath, R., Tran, D., & Blei, D. M. (2016). Hierarchical Variational
430 Models. In International Conference on Machine Learning. Rezende, D. J., Mohamed, S., & Wierstra,
431 D. (2014). Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In
432 International Conference on Machine Learning. Salimans, T., Kingma, D. P., & Welling, M. (2015).
433 Markov Chain Monte Carlo and Variational Inference: Bridging the Gap. In International Conference
434 on Machine Learning. Salakhutdinov, R., Tenenbaum, J. B., and Torralba, A (2013). Learning with
435 hierarchical-deep models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35
436 (8):1958?1971. Stuhlmuller, A., Taylor, J., & Goodman, N. (2013). Learning Stochastic Inverses.
437 In *Neural Information Processing Systems*. Tran, D., Blei, D. M., & Airoldi, E. M. (2015). Copula
438 variational inference. In *Neural Information Processing Systems*. Tran, D., Ranganath, R., & Blei, D.
439 M. (2016). The Variational Gaussian Process. International Conference on Learning Representations.
440 Waterhouse, S., MacKay, D., and Robinson, T (1996). Bayesian methods for mixtures of experts. In
441 *Neural Information Processing Systems*.