
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, allowing operations like posterior inference
14 to be executed directly and generically by an input choice of natural parameters,
15 rather than performing inference via optimization for each particular realization
16 of a distribution within that model. We demonstrate this ability across a number
17 of non-conjugate exponential families that appear often in the machine learning
18 literature.

1 Introduction

19
20 Probability models, the fundamental object of Bayesian machine learning, have long challenged
21 researchers with the tradeoff between tractability and expressivity. Though well understood that a
22 model should be chosen to instantiate a set of assumptions and capture existing domain knowledge
23 [1, 2, 3], for many years too-simple models were chosen for their practical advantages (such as
24 conditional conjugacy), which left much to be desired in terms of expressive performance and
25 scalability of these models.

26 More recently the pendulum has swung, via a resurgence in models which map a latent random
27 variable $z \sim \mathbb{P}_0$ through a member of a highly expressive function family $\mathcal{G} = \{g_\theta : \theta \in \Theta\}$, the
28 composition resulting in an *implicit probability model* $\mathcal{M} = \{\mathbb{P}(g_\theta \circ z) : \theta \in \Theta\}$. Choosing \mathcal{G} to be
29 a parameter-indexed family of deep neural *density networks* has both a rich history [4, 5], and has
30 recently been used to produce exciting results for density estimation [6, 7, 8], generation of complex
31 data [9], variational inference [10, 11, 12], and more. A noted advantage of these implicit density
32 network models is that in many cases they make minimal assumptions about the data generative (or
33 posterior inference) process. On the other hand, since these models have been chosen to be generic
34 and flexible, they can lack the classic stipulation that a model instantiates existing domain knowledge.
35 The downsides of a too-flexible model with finite data (albeit large) – and the corresponding bias-
36 variance benefit of a restricted model – are textbook knowledge [13, §7.3], and work on generalization
37 and compressibility in deep networks suggests that this broad class of function families are indeed
38 quite large, perhaps problematically so [14].

Is all the flexibility of an implicit density network model \mathcal{M} always necessary? Consider the case of variational inference, where a generative model $p(z)p_\beta(X|z)$ (latent z , observed data X) is stipulated in the classic sense to embody modeling assumptions (hierarchical model, topic model, Bayesian logistic regression, etc.). When such a model is intractable, it is increasingly common to deploy an implicit “recognition network” model for variational inference [10], which finds a $q_{\theta^*}(z) \in \mathcal{M}$ such that an evidence bound is optimized with respect to the true posterior $p(z|X)$. However, note the widely recognized fact [15] that many such true posteriors $p(z|X)$ belong to models that can be written as exponential families (albeit intractable, due to the choice of sufficient statistics $t(z)$), of the form: $\mathcal{P} = \left\{ \frac{h(z)}{A(\eta)} \exp \{ \eta^\top t(z) \} : \eta \in H \right\}$.

Should we be able to learn a tractable approximation to this exponential family model (not simply members thereof), we would in the very least get the bias-variance benefits of an intelligently restricted model space, and at best would get inference “for free” in the sense that we could evaluate approximate posteriors directly without case by case optimization. In this paper we aim to learn a restricted model $\mathcal{Q} = \{q(z; \eta : \eta \in H)\}$ that will be a strict subset of the deep implicit model \mathcal{M} and will closely approximate a target exponential family \mathcal{P} . Note the critical difference between this aim and much of the literature that seeks to learn a density $q_\theta^* \in \mathcal{M}$ (we explore this distinction in depth both algorithmically and empirically).

To proceed, we must first specify a set of models $\mathcal{Q} = \{Q_\phi : \phi \in \Phi\}$, from which we can learn a single model Q_{ϕ^*} , and we must second define a sensible parameter space H of each model. To the first, we restrict Θ , the parameter space of \mathcal{M} , to be itself the image of a second deep *parameter network* family $\mathcal{F} = \{f_\phi : \phi \in \Phi\}$, such that $\{f_\phi(\eta) : \eta \in H\} \subset \Theta$. The second part is answered immediately by our choice of target \mathcal{P} , an exponential family which by definition has *natural* parameterization $\eta \in H$. Thus, appealingly, we know that H is precisely the correct parameter space for \mathcal{Q} (as it defines \mathcal{P}), and that the image of H under f_ϕ will be of the correct dimensionality within the codomain Θ ; approximation error between \mathcal{Q} and \mathcal{P} will be caused by the flexibility and learnability of the parameter network f_ϕ and the density network $g_{f_\phi(\eta)}$.

We define this two-network architecture, which we term an *exponential family network* (EFN), and we specify a stochastic optimization procedure over a variant of the typical Kullback-Leibler divergence. We then demonstrate the ability of EFNs to approximately learn exponential families, both known tractable families and well-used intractable families, including hierarchical Dirichlet and truncated normal Poisson families. Finally we demonstrate the utility of this approach in an example inferring the posterior distribution of the latent intensity of a point-process, given neural spike train data. In all, our contributions include:

- a novel implicit model: a two-network deep architecture to learn a probability model along with a doubly stochastic optimization that samples over both natural parameters (the family member to be learned) and data points (observations of the target density);
- analysis of the connections between approximately learning a model and approximate variational inference, and an empirical study that gives insight to possible improvements to variational inference;
- empirical results confirming performance against ground truth in known tractable exponential families and in common intractable exponential families.

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 [14]. 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 [5], as are neural networks to fit a probability model to data [4, 16].

We choose flow networks [18]. And “implicit generative models aka density networks” (or rather, density networks are the instantiation of an IGM with deep nets, which is effectively synonymous

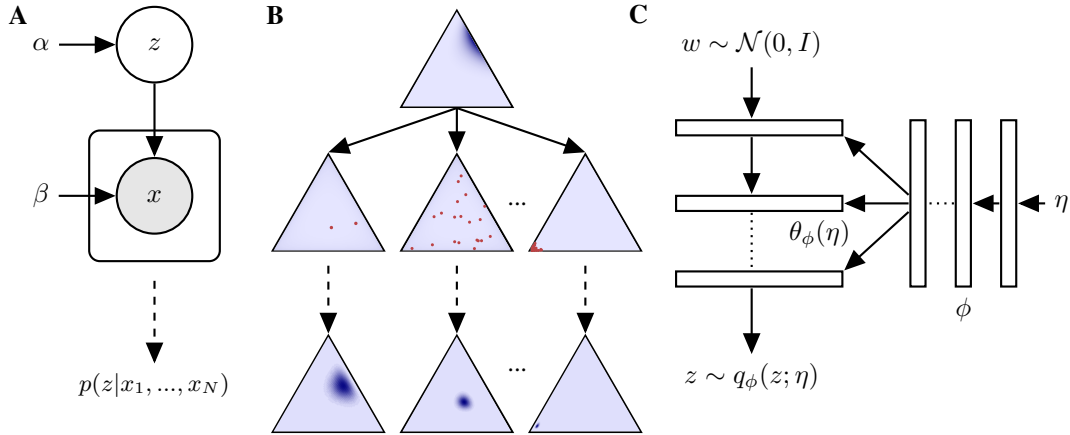


Figure 1: Learning exponential families. A shows the graphical model, emphasizing conditional iid sampling. B shows Dirichlet prior (a density), conditional Dirichlet observations (some observed points in the simplex), and then the posteriors learned by an EFN. SRB to fill in these triangles. C shows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

these days. And invertible networks In that vein probably definitely cite invertible/bijective deep nets in general [19, 7, 6, 18, 20, 8, 21]. Note that what norm flows [18] 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 [22]; same idea in more depth and that argues for the normal prior in [23]. 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 [24, 25]; this is called structured variational inference or newer stuff [26] [27], to name but a few.

2.2 Exponential families

bants. Pitman-Koopman Lemma [28, §3.3.3] Defines an M.

Why this is important. Exp fams are awesome and fundamental. Also [15] 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).

Note somewhere that the natural parameter space needs to be considered in general. That is, not all η lead to a valid distribution (standard fact, see for example [15]). In practice that's not often a problem, as the space is known for most distributions one uses, and when one composes them in a posterior scheme (for example), this is inherited (eg the normal covariance...). So we skip that here. But yes in general that needs to be considered.

2.3 Exponential family networks

includes the network definition of Fig 1c, the objective, and the optimization algorithm.

This should not be confused with "Learning to learn by gradient descent by gradient descent" [29]

Another related work is that this is somehow the dual of MEFN [30], or a generalization of the dual problem. In the wainwright and jordan sense of forward and backward mappings. Stuff on sampling from Gibbs distributions (max ent models), and sampling from exp fams generally, with MCMC and such.

Note that this objective can also produce approximations of the log partition, via essentially linear regression; more nuanced schemes are recommended [31]. We don't explore that here.

2.4 Relation to variational inference

We have already covered related work; here we scrutinize EFNs in terms of VI.

We are interested in perhaps the most classic inference problem:

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

shown with the attached plate model (not local latents). Supposing as is often the case that the 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.

Key distinctions:

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

Latest key distinctions:

- prior is in parameter network, unlike essentially all others, even if you take a narrow view that $\sum_i t(x_i)$ is a single data point. Prior has been recognized for mattering in the ELBO, though this sentence is a dubious distinction [32, 33] (dubious need for these refs)
- data is given via an assumption of sufficiency, namely in natural form [28], not in x form. Of course this is sensible as in some settings we don't know the natural form of the generative model, but that's a key difference with SVI; plenty of those models are not deep nets (and shouldn't be, if there is an intent of statistical inference rather than nonlinear dimensionality reduction / autoencoding) and there we *do* know the natural parameters.

3 Results

Introductory remarks and then comments about architectural particulars, including planar flow networks of [18]. Note Number of panar flows is always D (intrinsic dimensionality of flows), units per layer ramping is always the same function of D. The number of layers in the theta network is always a function of D - will probably just always use 8 layers. Remember

NF1: do full norm flow "variational inference" (explore all of ϕ space with the full flow network model \mathcal{Q}), which is to say $\arg \min_{\phi} KL(q_{\phi}||p)$.

EFN1: be literal to Figure 1C, 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.

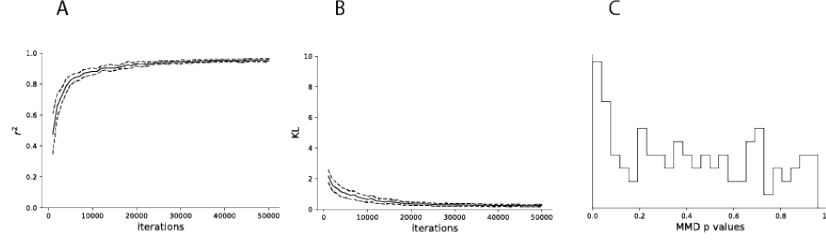


Figure 2: 25-dimensional Dirichlet exponential family network. A.) Distribution of r^2 between the sufficient statistics and log-probability across choices of η throughout optimization. B.) Distribution of KL divergence across choices of η throughout optimization. C.) Distribution of maximum mean discrepancy p-values between EFN samples and ground truth after optimization [34].

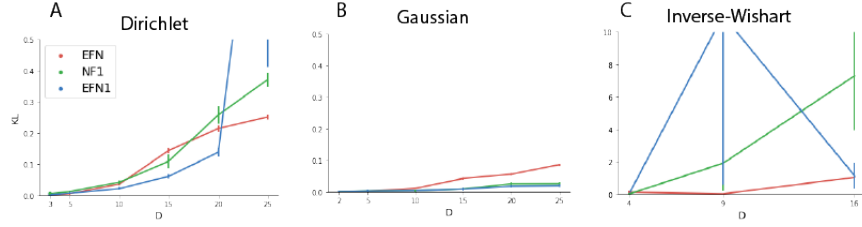


Figure 3: Scaling exponential family networks. A.) Dirichlet. B.) Gaussian C.) Inverse-Wishart

3.1 Tractable exponential families

3.2 Intractable exponential families

Hierarchical Dirichlets Hierarchical dirichlets are useful and have some history; most notable is with the Hierarchical Dirichlet Process [35], but historically this was done in the finite case also [36]. Here is some math. Note that this matters for multi-corpus LDA generally as well [37, 38].

Truncated- and log-normal Poisson used a lot [39][40][41, 42]

Figure 4:

EFN in intractable exp fams (connecting to above, but with hard distribs and the ELBO)

Panel A: Dir-Dir ELBO by dimensionality for NF1 and EFN and EFN1

Panel B: Dir-Dir ELBO by dimensionality for EFN1 vs EFN1a vs 1b vs 1c vs NF1 (with $N = 1$ data point)

3.3 Neural spike train analysis

Figure 5:

Panel A TNP picture example of prior and posterior with a few samples, just for feel good

PANEL B: ELBO on held out data as a function of R , for a middle choice of training dataset size N and D .

PANEL C: ELBO on held out data as a function of N , for a middle choice of number of samples in the posterior R .

PANEL D (optional): (ELBO EFN - ELBO NF1) as a surface plot as a function of R, N . That is, positive places is where EFN outperforms, negative NF1.

The key point with these is that, while you have the *same exact* flow network architecture, now you have to optimize over ϕ with a limited single dataset. Learning a restricted model space is good for the bias-variance tradeoff! Do this many times so that variance will become clear.

—other thoughts— Real data analysis and posterior inference. **Key real data result on TNP.**

Get some data from CRCNS that has many spike trains x_i for $i = 1, \dots, N$ (ask Gabriel, as he has

done some poking around recently; or look at some of the above TNP/LNP refs).
Those spike trains should be conditionally independent draws from the same underlying intensity
function z . (for example, trials under the same stimulus)
Bin the length of time T into $\approx 20 - 30$ equally spaced time bins. Thus z is now a vector in \mathbb{R}^{20} .
Now each spike train x_i is a conditionally independent Poisson vector observation, with rate vector z .
Learn the 20 dimensional TNP exp fam, without any regard to this dataset X .
No: Panel No: TNP ELBO by dimensionality for NF1 and EFN and EFN1
Panel A TNP picture example of prior and posterior with a few samples, just for feel good

Now we want to learn the posterior $p(z | \text{some fixed number } R \text{ of data points})$.
To do this for an EFN, just plug in those R points x_{i_1}, \dots, x_{i_R} and the prior as a natural parameter,
and job done.
To do this for an NF1, train a VI model by taking the log joint with R data points, then go through
and resample R points every time from your training dataset with N data points.
**PANEL A: ELBO on held out data as a function of R , for a middle choice of training dataset
size N .**
**PANEL B: ELBO on held out data as a function of N , for a middle choice of number of
samples in the posterior R .**
**PANEL C: (ELBO EFN - ELBO NF1) as a surface plot as a function of R, N . That is, positive
places is where EFN outperforms, negative NF1.**
The key point with these is that, while you have the *same exact* flow network architecture, now you
have to optimize over ϕ with a limited single dataset. Learning a restricted model space is good
for the bias-variance tradeoff! Do this many times so that variance will become clear. **Panel C v2:
Possibly want to explicitly plot variance of EFN and NF1 to focus on the variance tradeoff**
Panel C v3: change time bin granularity from 10 to 50 to show how this story changes in D .
My thought is that all will be exhausted by dimensionality sweeps by this point, so no.
also Notice one pain here is that these panels requires training a new EFN1 at every choice of N and
 R (but only one EFN). Sorry.

We hope and expect this will 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). 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. 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). 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.

This will be for one real example X . As such, to get error bars, just take a big dataset and randomly
subsample the test set. 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 would 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)...let's skip that altogether)).

4 Conclusion

Snappy closing remarks!

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] Joshua B Tenenbaum, Thomas L Griffiths, and Charles Kemp. Theory-based bayesian models of inductive learning and reasoning. *Trends in cognitive sciences*, 10(7):309–318, 2006.
- [3] Peter McCullagh. What is a statistical model? *The Annals of Statistics*, 30(5):1225–1267, 2002.
- [4] Peter Dayan, Geoffrey E Hinton, Radford M Neal, and Richard S Zemel. The helmholtz machine. *Neural computation*, 7(5):889–904, 1995.
- [5] D. J. C. MacKay and M. N. Gibbs. Density networks. In *Statistics and Neural Networks*, pages 129–146. Oxford, 1997.
- [6] Benigno Uribe, Iain Murray, and Hugo Larochelle. Rnade: The real-valued neural autoregressive density-estimator. In *Advances in Neural Information Processing Systems*, pages 2175–2183, 2013.
- [7] Oren Rippel and Ryan Prescott Adams. High-dimensional probability estimation with deep density models. *arXiv preprint arXiv:1302.5125*, 2013.
- [8] George Papamakarios, Iain Murray, and Theo Pavlakou. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pages 2335–2344, 2017.
- [9] 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.
- [10] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv*, 12 2013.
- [11] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.
- [12] Michalis Titsias and Miguel Lázaro-Gredilla. Doubly stochastic variational bayes for non-conjugate inference. In *International Conference on Machine Learning*, pages 1971–1979, 2014.
- [13] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.
- [14] 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.
- [15] 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.
- [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] Luc Devroye. *Non-uniform random variate generation*. Springer-Verlag, New York, 1986.
- [18] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.
- [19] 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.
- [20] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

- 334 [21] Jörn-Henrik Jacobsen, Arnold Smeulders, and Edouard Oyallon. i-revnet: Deep invertible
335 networks. *arXiv preprint arXiv:1802.07088*, 2018.
- 336 [22] Scott Saobing Chen and Ramesh A Gopinath. Gaussianization. In *Advances in neural informa-*
337 *tion processing systems*, pages 423–429, 2001.
- 338 [23] Esteban G Tabak, Eric Vanden-Eijnden, et al. Density estimation by dual ascent of the log-
339 likelihood. *Communications in Mathematical Sciences*, 8(1):217–233, 2010.
- 340 [24] Lawrence K Saul and Michael I Jordan. Exploiting tractable substructures in intractable
341 networks. In *Advances in neural information processing systems*, pages 486–492, 1996.
- 342 [25] David Barber and Wim Wiegierinck. Tractable variational structures for approximating graphical
343 models. In *Advances in Neural Information Processing Systems*, pages 183–189, 1999.
- 344 [26] Matthew Hoffman and David Blei. Stochastic structured variational inference. In *Artificial*
345 *Intelligence and Statistics*, pages 361–369, 2015.
- 346 [27] Dustin Tran, David Blei, and Edo M Airolidi. Copula variational inference. In *Advances in*
347 *Neural Information Processing Systems*, pages 3564–3572, 2015.
- 348 [28] Christian Robert. *The Bayesian choice: from decision-theoretic foundations to computational*
349 *implementation*. Springer Science & Business Media, 2007.
- 350 [29] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom
351 Schaul, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. In
352 *Advances in Neural Information Processing Systems*, pages 3981–3989, 2016.
- 353 [30] Gabriel Loaiza-Ganem, Yuanjun Gao, and John P Cunningham. Maximum entropy flow
354 networks. *International Conference on Learning Representations*, 2017.
- 355 [31] George Papamakarios and Iain Murray. Distilling intractable generative models. In *Probabilistic*
356 *Integration Workshop at Neural Information Processing Systems*, 2015.
- 357 [32] Matthew D Hoffman and Matthew J Johnson. Elbo surgery: yet another way to carve up the
358 variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference,*
359 *NIPS*, 2016.
- 360 [33] Chris Cremer, Xuechen Li, and David Duvenaud. Inference suboptimality in variational
361 autoencoders. *arXiv preprint arXiv:1801.03558*, 2018.
- 362 [34] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander
363 Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773,
364 2012.
- 365 [35] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical dirichlet
366 processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.
- 367 [36] David JC MacKay and Linda C Bauman Peto. A hierarchical dirichlet language model. *Natural*
368 *language engineering*, 1(3):289–308, 1995.
- 369 [37] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of*
370 *machine Learning research*, 3(Jan):993–1022, 2003.
- 371 [38] Jonathan K Pritchard, Matthew Stephens, and Peter Donnelly. Inference of population structure
372 using multilocus genotype data. *Genetics*, 155(2):945–959, 2000.
- 373 [39] Yuanjun Gao, Evan W Archer, Liam Paninski, and John P Cunningham. Linear dynamical
374 neural population models through nonlinear embeddings. In *Advances in Neural Information*
375 *Processing Systems*, pages 163–171, 2016.
- 376 [40] Ryan Prescott Adams, Iain Murray, and David JC MacKay. Tractable nonparametric bayesian
377 inference in poisson processes with gaussian process intensities. In *Proceedings of the 26th*
378 *Annual International Conference on Machine Learning*, pages 9–16. ACM, 2009.

- 379 [41] John P Cunningham, Krishna V Shenoy, and Maneesh Sahani. Fast gaussian process methods
380 for point process intensity estimation. In *Proceedings of the 25th international conference on*
381 *Machine learning*, pages 192–199. ACM, 2008.
- 382 [42] John P Cunningham, M Yu Byron, Krishna V Shenoy, and Maneesh Sahani. Inferring neural
383 firing rates from spike trains using gaussian processes. In *Advances in neural information*
384 *processing systems*, pages 329–336, 2008.

385 5 Appendix

386 Exponential form of posterior for Dirichlet-Dirichlet

387 $\mathbf{z} \sim \text{Dir}(\boldsymbol{\alpha}_0)$

388 $\mathbf{x}_i \sim \text{Dir}(\beta \mathbf{z})$

389 $p(\mathbf{z}) \propto \exp(\boldsymbol{\alpha}_0^T \log(\mathbf{z}) - \sum_{d=1}^D \log(z_d))$

390 $p(\mathbf{x}_i | \mathbf{z}) \propto \exp(\beta \mathbf{z}^T \log(\mathbf{x}_i) - \sum_{d=1}^D \log(x_{i,d}) - (\sum_{d=1}^D \log(\Gamma(\beta z_d)) - \log(\Gamma(\beta \sum_{d=1}^D z_d))))$

391 $p(X | \mathbf{z}) \propto \exp(\beta \mathbf{z}^T [\sum_{i=1}^N \log(\mathbf{x}_i)] - \sum_{i,d=1}^{N,D} \log(x_{i,d}) - N(\sum_{d=1}^D \log(\Gamma(\beta z_d)) - \log(\Gamma(\beta \sum_{d=1}^D z_d))))$

392

393 $p(\mathbf{z} | X) \propto p(\mathbf{z})p(X | \mathbf{z})$

394 $\propto \exp(\boldsymbol{\alpha}_0^T \log(\mathbf{z}) - \sum_{d=1}^D \log(z_d))$

395 $\exp(\beta \mathbf{z}^T [\sum_{i=1}^N \log(\mathbf{x}_i)] - \sum_{i,d=1}^{N,D} \log(x_{i,d}) - N(\sum_{d=1}^D \log(\Gamma(\beta z_d)) - \log(\Gamma(\beta \sum_{d=1}^D z_d))))$

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

397 $p(\mathbf{z} | X) \propto \exp(\boldsymbol{\alpha}_0^T \log(\mathbf{z}) + \beta [\sum_{i=1}^N \log(\mathbf{x}_i)]^T \mathbf{z} - \sum_{d=1}^D \log(z_d) - N(\sum_{d=1}^D \log(\Gamma(\beta z_d)) - \log(\Gamma(\beta \sum_{d=1}^D z_d))))$

398 $p(\mathbf{z} | X) \propto \exp\left(\begin{pmatrix} \boldsymbol{\alpha}_0 - \mathbf{1} \\ \sum_{i=1}^N \log(\mathbf{x}_i) \\ -N \end{pmatrix}^T \begin{pmatrix} \log(\mathbf{z}) \\ \beta \mathbf{z} \\ \log(\Gamma(\beta \mathbf{z})) \\ \log(\Gamma(\beta \sum_{d=1}^D z_d)) \end{pmatrix}\right)$

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

401 Here's a more cleaned up version:

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

402 or just using the Beta function:

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