Learning Exponential Families

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

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

1 Introduction

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

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

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

Consider specifically using implicit models for variational inference.

- while offering many advantages, two shortcomings: represent a potentially too-flexible model, and are used to find single posterior distributions (often on local variables).
- VI has to re-learn on every dataset; yes it can amortize across points from the same dataset, but not across datasets in the same model. Given the frequency of certain non-conjugate models appearing – hierarchies of Dirichlet distributions, log Gaussian Poisson models, etc – this seems needless to continue considering this as an "intractable" exp fam.

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. [15]. 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). [16]. 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 [16]. 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. Also we always sample from intractable families via some transformations [16]; the fact that some have known constructions (ratio of gammas, Bartlett decomposition, etc) should not distract from the fundamental nature of this process.
- We leverage old and new work recently much attention has been paid to bijective neural networks, networks that admit tractable density calculations. An old idea with new options.
- EFNs allow the embodiment of modeling assumptions without sacrificing expressivity
- concept here is to learn something we care about already and get the usual benefits of learning a restricted model space [13, §7, for example]
- EFNs include neural net observation models in many cases, so don't despair. (like a VAE generator)

Mechanically:

- 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. 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).
- offer insights into parameterizing existing VI and similar to increase performance (Fig 4)
- careful treatment distinguishing this from VI. The similarities to VI are clear.s

2 Exponential family networks

2.1 Implicit probability models via density networks

bants. defines a 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, 17].

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 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 variatonal 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 eta'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]

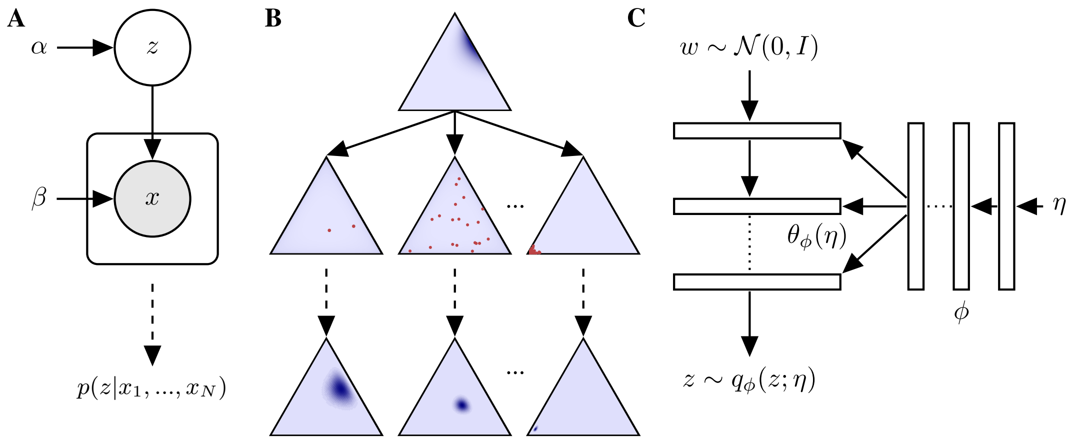

Figure 1: Learning exponential families. A shows the graphical model, emphasizing conditional iid sampling. B shows Dirichlet prior (a density), conditional Dirichlet observations (some observed points in the simplex), and then the posteriors learned by an EFN. SRB to fill in these triangles. C shows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

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 parmaterizes 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 (Guassian).

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 $E_{q_{\phi(x)}}\left(\log p_\theta(x, z)\right)$. Now sure that's a loss function on x, z, so then when we look at that same term in EFN we see $E_{q_{\phi(\eta)}}\left(\eta^\top t(z)\right)$, 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 $p(\eta|z) = \frac{1}{A(t(z))} \exp\left\{\eta^\top t(z)\right\}$. 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

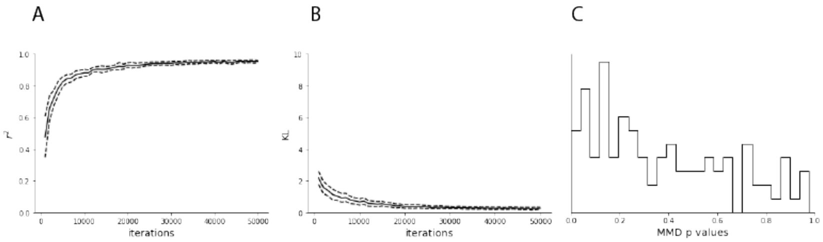

Figure 2: 25-dimensional Dirichlet exopnential family network. A.) Distribution of r^2 between the sufficient statistics and log-probability across choices of η throughout optimization. B.) Distribution of KL divergence across choices of η throughout optimization. C.) Distribution of maximum mean discrepancy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

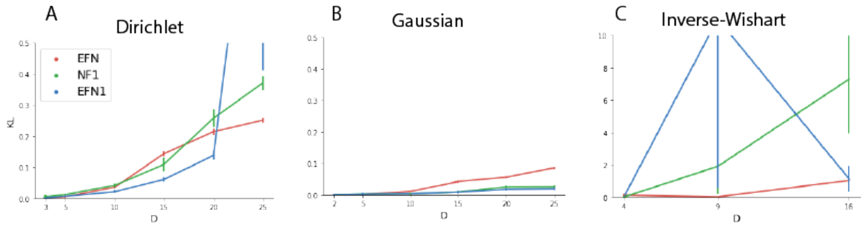

Figure 3: Scaling exponential family networks. A.) Dirichlet. B.) Gaussian C.) Inverse-Wishart

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.

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 $i = 1, ..., N$ (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 $p(z|$ **some fixed number** R **of data points**$)$.
To do this for an EFN, just plug in those R points $x_{i_1}, ..., x_{i_R}$ 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 Uria, 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] Luc Devroye. *Non-uniform random variate generation*. Springer-Verlag, New York, 1986.

[17] 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.

[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.

[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 Wiegerinck. 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] Matthew D Hoffman and Matthew J Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, 2016.

[33] Chris Cremer, Xuechen Li, and David Duvenaud. Inference suboptimality in variational autoencoders. *arXiv preprint arXiv:1801.03558*, 2018.

[34] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

[35] 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.

[36] David JC MacKay and Linda C Bauman Peto. A hierarchical dirichlet language model. *Natural language engineering*, 1(3):289–308, 1995.

[37] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[38] Jonathan K Pritchard, Matthew Stephens, and Peter Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, 2000.

[39] 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.

[40] 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.

[41] 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.

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

5 Appendix

Exponential form of posterior for Dirichlet-Dirichlet
$\boldsymbol{z} \sim Dir(\boldsymbol{\alpha}_0)$
$\boldsymbol{x}_i \sim Dir(\beta \boldsymbol{z})$
$p(\boldsymbol{z}) \propto \exp\,(\boldsymbol{\alpha}_0^T \log(\boldsymbol{z}) - \sum_{d=1}^{D} \log(z_d))$
$p(\boldsymbol{x}_i \mid \boldsymbol{z}) \propto \exp\,(\beta \boldsymbol{z}^T \log(\boldsymbol{x}_i) - \sum_{d=1}^{D} \log(x_{i,d}) - (\sum_{d=1}^{D} \log(\Gamma(\beta z_d)) - \log(\Gamma(\beta \sum_{d=1}^{D} z_d))))$
$p(X \mid \boldsymbol{z}) \propto \exp\,(\beta \boldsymbol{z}^T \left[\sum_{i=1}^{N} \log(\boldsymbol{x}_i)\right] - \sum_{i,d=1}^{N,D} \log(x_{i,d}) - N(\sum_{d=1}^{D} \log(\Gamma(\beta z_d)) - \log(\Gamma(\beta \sum_{d=1}^{D} z_d))))$

$p(\boldsymbol{z} \mid X) \propto p(\boldsymbol{z}) p(X \mid \boldsymbol{z})$
$\propto \exp\,(\boldsymbol{\alpha}_0^T \log(\boldsymbol{z}) - \sum_{d=1}^{D} \log(z_d))$
$\exp\,(\beta \boldsymbol{z}^T \left[\sum_{i=1}^{N} \log(\boldsymbol{x}_i)\right] - \sum_{i,d=1}^{N,D} \log(x_{i,d}) - N(\sum_{d=1}^{D} \log(\Gamma(\beta z_d)) - \log(\Gamma(\beta \sum_{d=1}^{D} z_d))))$

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

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

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

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

Here's a more cleaned up version:

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

or just using the Beta function:

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