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

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

- generative probabilistic models are the fundamental object of bayesian modeling [1].
- classic issue has been tractability-expressivity tradeoff. choosing and even defining a statistical model is hard [2, 1]
- However, these models are chosen to be generic and flexible, rather than in the classic sense of instantiating a set of statistical assumptions concerning the process of generating some data. somehow offering an explanation or a structured assumption about data. This is not bad per se, but leaves much to be desired in terms of modeling.
- recently implicit probabilistic models have been used a lot, and for VI in particular [3, 4, 5] (more blei stuff here)
- 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.

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

- 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 expressiity
- 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 intiializer if not as an amortizer.

People use lots of implicit generative models:

Across machine learning, including ABC [9] , GANs [10], VAEs [3, 4], 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. We prefer and use the terminology of [11], calling such a distribution an *implicit generative model*, defined as something like eq 1 and 2 in [11]:

$$q_\theta(z)$$

Also use the proper notation of the density implied by the pushforward measure of the function $f_{\theta\sharp}$ 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.

A note on amortization

Several have pointed out that these IGMs are in fact strictly less expressive than a mean field, at least in the conventional VI setting. See for example http://dustintran.com/blog/variational-auto-encoders-do-not-train-complex-generative-models (here I like the line “The neural network used in the encoder (variational distribution) does not lead to any richer approximating distribution. It is a way to amortize inference such that the number of parameters does not grow with the size of the data (an incredible feat, but not one for expressivity!) (Stuhlmuller et al., 2013)“). You have to optimize for every data point individually, or instead you get to do so in aggregate once in advance (at a much higher cost) and then recover that cost over future data points within that distribution (and hence the term amortization, though perhaps there is shared statistical power as well) Etc etc what we are doing here is *amortized* amortized inference, in the sense that we are amortizing not the data points, but the distribution itself.

...

This should not be confused with "Learning to learn by gradient descent by gradient descent" [12]

...

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).

2 Exponential family networks

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:

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

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.

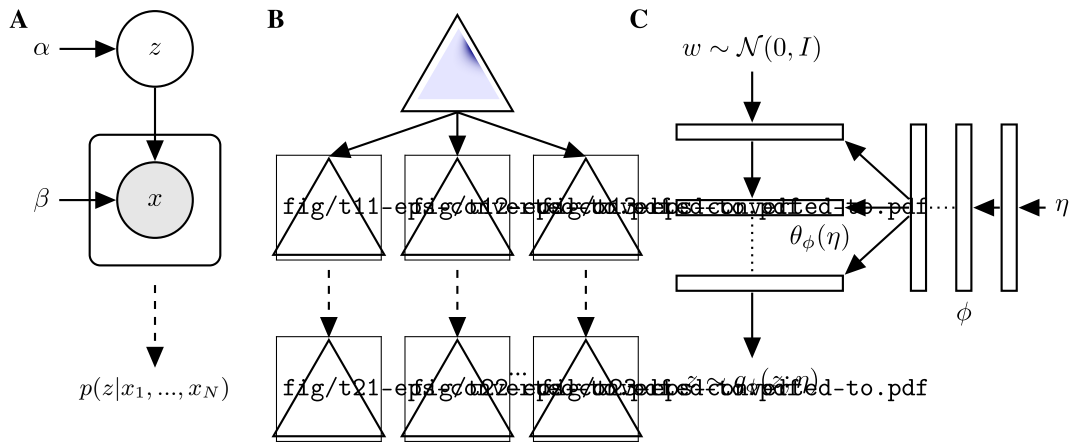

Figure 1: Learning exponential families. A shows the graphical model, emphasizing conditional iid sampling. B shows Dirichlet prior (a density), conditional Dirichlet observations (some observed points in the simplex), and then the posteriors learned by an EFN. SRB to fill in these triangles. C shows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

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.

Why this is important

Exp fams are awesome and fundamental. Also [**?**] 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.

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).

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 [**?**]). 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.

Aside

A neat idea 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.

Why Flow Networks

Density networks are an old idea [13].

We choose flow networks [14]. 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 [15, 16, 14, 17, 18]. Note that what norm flows [14] 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 [19]; same idea in more depth and that argues for the normal prior in [20]. Really the norm flow is not so special as this is a well established classic idea. A nice line from Rezende and

Mohamed is: Thus, an ideal family of variational distributions q(z|x) is one that is highly flexible, preferably flexible enough to contain the true posterior as one solution. One path towards this ideal is based on the principle of nor- malizing flows (Tabak Turner, 2013; Tabak VandenEijnden, 2010).

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 *(Saul and Jordan, 1996; Barber and Wiegerinck, 1999); this is called structured variational inference. Another way to expand the family is to consider mixtures of variational densities, i.e., additional latent variables within the variational family (Bishop et al., 1998).* or newer stuff [] [Tran Copula VI, Hoffman and Blei 2015].

As noted in norm flows paper: "The true posterior distribution will be more com- plex than this assumption allows for, and defining multi- modal and constrained posterior approximations in a scalable manner remains a significant open problem in varia- tional inference."

Related work / How close is this to norm flows or VAE

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 $E_{q_{\phi(\eta)}}\left(\eta^\top t(z)\right)$, 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 related work is that this is somehow the dual of MEFN, or a generalization of the dual problem. In the wainwright and jordan sense of forward and backward mappings.

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

3 Results

Chapter 1, Fig 1

Toy figure that demonstrates what we are doing and a simple example. Note this should probably not be in Results but in the EFN section or similar. Ideas:

- value of a restricted model, see hastie tibshirani fig 7.2, or porbanz's batman version from 4400 slides. ... well that's a bit off topic. At least worth a mention in motivation.
- graphical model. yeah probably needed.
- network model. yeah probably needed.
- cartoon example three sets of natural parameters in, three dirichlet distributions out. Or similar.

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

Figure 2.
Single EFN:
Panel A: r^2 throughout training
Panel B: KL throughout training
Panel C: Distributin of MMD p values

Figure 3:
EFN performance by dimensionality
Panel A: Dir KL for NF1 and EFN
Panel B NIW KL for NF1 and EFN
Panel C: Gaussian KL for NF1 and EFN

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.

Fig 4. [This idea was Fig 5 in disguise; see below. Currently no need for this figure].

Chapter 3: Fig 5 and 6

Fig 5. The intractable posterior inference example. **Key real data result**. Learn the full posterior family for some problem (see ideas below). Then get some data X. Then find the posterior distribution for that data by indexing the natural parameters (as in, just plugging in the correct choice of η, which is after all some function of the prior and X). That gives the EFN posterior $q(z|X)$. (Possible 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).

Fig 5. Heaps of examples with conditional iid exp fams. Math details of that pending. Some cool examples:

- Censored data. normal prior, censored normal observations, what is posterior distribution on mean? Lots of work in that.
- Truncated data. truncated mvn prior, with some observations thereafter, what is posterior? (Does this work?...)
- Poisson/Bern "process" data. Phony process like in neuro, normal prior on log intensity (ooh maybe that's not an exp fam prior), then a "spike train" of bern or poisson count observations
- multivariate t with inverse wishart prior or something like that. That's neat but doesn't have great "oh yeah people do care about that problem" recognition. Seems contrived.
- check MKB book for other cool MV distributions. (Marshall-Olkin)... seems contrived.
- Elliptically contoured prior with some conditionally iid exp fam observations. People in ML like elliptical distributions.
- von Mises-Fisher distribution, eg http://www.jmlr.org/papers/volume6/banerjee05a/banerjee05a.pdf or https://arxiv.org/pdf/1605.00316.pdf, but again not clustering (see below), since it's a local latent variable problem then.s
- Note: a whole heap of models don't quite fit comfortably here.
  - Bayesian Logistic Regression. This is an intractable exp fam in the desired sense, but the natural parameter (when parameterized) depends on x_i. Thus, it grows with every datapoint, or put differently it's a diff exp fam for every dataset. No bueno. This is then true of GLMs, so those are out too.
  - Latent Dirichlet Allocation. Local variational parameters mean that the exp fam grows with datasize. That means that the posterior is already too big for uninteresting sizes of LDA. This is then true of hierarchical models with local latent variables in general.

Fig 6. The Killer real data. Perhaps Gibbs or Markov Random Field. Learn it, then pick some η, then show samples from it. Can this look interesting? Some thoughts...

Criteria:

- Needs to be an exp fam.

Figure 2: Figure 1: possibly Fig 7.2 bias-variance tradeoff and then benefit of a restricted model from Hastie Book, or similar from W4400 (ask PO for batman permission).

Table 1: Sample table title

Part		
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

- Needs to be a forward exp fam. As in, not fit to data, because we don't have μ parameters, we have η parameters.
- "real data" is a misnomer, since we are not doing VI or similar. Really we want an exp fam that is real and somehow useful in its own right, and that people want to sample from.
- Reminder: we will *always* be comparing to "well normally you can do this with learning a *single* distribution in the $\min KL(q||p)$ sense. That's fine. The point is we can learn the whole family, then choose and sample, vs just one by one.
- something hard to sample will be key, since the "toy" results will have used things we already "know" how to sample, like NIW or Dirichlet.

Ideas:

- Fancy Exp Fam like Marshall-Olkin. Yeah but who really cares about this esoteric distribution? It doesn't look cool visually either.
- Ising models: classic, bw images, but gross NP-Hard Cooper 1990.
- Potts model: great because failure of MCMC (Gibbs sampling) here is at least locally well known from Geman and Geman 1984 through Sudderth correcting this (see Gibbs sampler slides from Advanced ML, Peter's part). But that is kind of a failure example, not an interesting one (MRFs are smoothness prior, not segmentation prior). Also both Potts and Ising are NP-hard Cooper 1990 The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks
- Markov Random Fields / Gibbs Random Fields (same, by Hammersley Clifford theorem). Yes this is cool: image distributions, texture distributions. Can show wild diff sets of textures, none of which require any sampling or any such thing. Can we make this super intractable from an MCMC perspective? Need to read on how sampling is done there. Erik Sudderth and his phd thesis are likely good resources.
- Gatys and Simoncelli texture stuff (see for example MEFN paper for refs); those are interesting distributions on textures, or specified moments. Can then just sample from this family.

```
\usepackage[pdftex]{graphicx} ...
\includegraphics[width=0.8\linewidth]{myfile.pdf}
```

4 Appendix

Exponential form of posterior for Dirichlet-Dirichlet

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

$\boldsymbol{x}_i \sim Dir(\beta \boldsymbol{z})$

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

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

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

$p(\boldsymbol{z} \mid X) \propto p(\boldsymbol{z}) p(X \mid \boldsymbol{z})$

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

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

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

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

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

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:

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

or just using the Beta function:

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

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. 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] Shakir Mohamed and Balaji Lakshminarayanan. Learning in implicit generative models. 10 2016.

[12] 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.

[13] D. J. C. MacKay and M. N. Gibbs. Density networks. In *Statistics and Neural Networks*, pages 129–146. Oxford, 1997.

[14] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

[15] 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.

[16] Oren Rippel and Ryan Prescott Adams. High-dimensional probability estimation with deep density models. *arXiv preprint arXiv:1302.5125*, 2013.

[17] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[18] Jörn-Henrik Jacobsen, Arnold Smeulders, and Edouard Oyallon. i-revnet: Deep invertible networks. *arXiv preprint arXiv:1802.07088*, 2018.

[19] Scott Saobing Chen and Ramesh A Gopinath. Gaussianization. In *Advances in neural information processing systems*, pages 423–429, 2001.

[20] 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.

Stuff on wake sleep and the Helmholtz machine

Stuff on sampling from Gibbs distributions (max ent models), and sampling from exp fams generally, with MCMC and such.

Flow networks

Devroye's book.

Hoffman et al 2013 SVI

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

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

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

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

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

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