Deep Exponential Families

1. Intro: Exponential Families

Random variables from the exponential family take the form

$$p(x) = h(x)\exp(\nu^T T(x) - a(\nu))$$

where h is the base measure, ν are natural parameters, T sufficient statistics (a statistic such that no information can be gained by additional statistics), a is the log normalizer.

Some examples include Gaussian, Poisson, gamma, von Mises, Beta, Binomial.

Example 1.1 (Bernoulli)**.**

$$\nu = \frac{\pi}{1-\pi}$$

$$T(x) = x$$

$$a(\nu) = -\log(1-\pi) = \log(1+e^{\nu})$$

$$h(x) = 1$$

Exponential families are completely specified by their sufficient statistics. The expectation of sufficient statistics T is

$$\mathbb{E}[T(x)] = \nabla_\nu a(\nu)$$

the gradient of the log-normalizer. We will use this fact later.

2. Deep Exponential Families: Overview

We want a general method for deep unsupervised feature learning.

Let observations arise from a cascade of layers of latent variables where each layer's variables are drawn from an exponential family governed by the inner product of the previous layer's variables and a set of weights.

You can also change the prior on weights.

You might have seen special cases of these models:

(a) Bernoulli latent variables -> sigmoid belief network
(b) Gamma latent variables -> deep nonnegative matrix factorization
(c) Gaussian latent variables -> deep latent gaussian models (rezende 14)

They are a subset of stochastic feed forward networks (neal paper).

3. Deep Exponential Families: The Model

A deep exponential family chains exponential families in a hierarchy of L layers of hidden variables $z_{n,1}, ..., z_{n,L}$. There are $L-1$ layers of weights $W_1, ..., W_{L-1}$ where each W_l is a collection of K_l vectors and we give each weight a prior distribution $p(W_l)$.

Now we will consider a single data point. Draw the 'top' layer of latent variables

$$p(z_{L,k}) = EXPFAM_L(z_{L,k}, \nu)$$

then draw each subsequent layer conditioned on the previous layer,

$$p(z_{l,k}|z_{l+1}, w_{l,k}) = EXPFAM_l(z_{l,k}, g_l(z_{l+1}^T w_{l,k}))$$

where g_l maps the inner product to the natural parameter ν.

4. Deep Exponential Families: The Likelihood

Data are drawn conditioned on the 'lowest' layer of the DEP $p(x_{n,i}|z_{n,1})$.

Let's take the case of counts data and use Poisson with rate λ,

$$p(x_{n,i} = x) = e^{-\lambda}\frac{\lambda^x}{x!}$$

If we let $x_{n,i}$ be the count of type i associated with observation n then,

$$p(x_{n,i}|z_1, W_0) = Poisson(z_{n,1}^T w_{0,i})$$

and we let W_0 be gamma distributed. Intuitively, W puts mass on groups of terms i.e. topics and we can make hierarchies of these things $p(z_{n,1}|z_{n,2}, W_1)$ gives the distribution of topics given super-topics and so on.

5. Deep Exponential Families: Link Functions

Extending the fact about exponential families above

$$\mathbb{E}[T(x)] = \nabla_\nu a(\nu) = \nabla_\nu a(g_l(z_{l+1}^T w_{l,k}))$$

shows that the moments of the sufficient statistics are completely specified by the previous layer and the link function g_l.

Consider the identity $g_l(x) = x$. Then the log normalizer transforms the expectation of latent variables. This is like a partition function Z (in the gaussian case like a softmax). In the sigmoid belief network the identity recovers sigmoid link.

6. Sparse Gamma Deep Exponential Families

The gamma distribution is an exponential family distribution with support over positive reals and pdf,

$$p(z) = z^{-1}\exp(\alpha\log(z) - \beta z - \log\Gamma(\alpha) - \alpha\log(\beta))$$

Now what is our link function?

$$g_\alpha = \alpha_l, g_\beta = \frac{\alpha_l}{z_{l+1}^T w_{l,k}}$$

and since the expectation of gamma variables needs to be positive we let the weight matrices be gamma distributed.

When α is small the gamma distribution puts most of its mass near 0 i.e. 'sparse gamma'. They perform a lot like the spike and slab prior that you may have seen previously. How does this differ from Poisson? When EV is high the Poisson distribution will give many large valued samples while in sparse gamma the samples will be either close to zero or larger than Poisson draws. When doing topic modeling, this means that an observation doesn't need to have every super topic of the concept it belongs to.

7. Sigmoid Belief Network

Consider a heirarchy of latent Bernoulli layers where the mean of a feature at layer l is given by linear combination of features at layer $l+1$ passed through a sigmoid. We will show that this is a special case of Bernoulli DEF.

$$p(z_{l,k}|z_{l+1}, w_{l,k}) = \exp(z_{l+1}^T w_{l,k} z_{l,k} - \log(1 + \exp(z_{l+1}^T w_{l,k})))$$

where $z_{l,k} \in 0, 1$. I leave it to exercises to show that the derviative of the log normalizer of the Bernoulli is the logistic functions. This gives us sigmoid belief nets.

8. Inference

As an example we will consider the Poisson DEF. We perform posterior inference by extending variational methods to general DEFs. Like usual, introduce variational family and maximize the ELBO

$$L(q) = \mathbb{E}_{q(z,W)}[\log p(x, z, W) - \log q(z, W)]$$

where z are latents associated with each observation, W latents shared across observations. Get the variational distribution q by the mean field variational family (factorize)

$$q(z, W) = q(W_0) \prod_{l=1}^{L} q(W_l) \prod_{n=1}^{N} q(z_{n,l})$$

where each $q(z_{n,l,k})$ is from the exponential family in question and $q(W)$ is of the same family as $p(W)$.

To optimize this objective we need expectations under q which we get by black box variational inference (Blei https://arxiv.org/pdf/1401.0118.pdf) i.e. rewrite the gradient of our objective as the expectation of an easy-to-implement function f of the latent and observed variables where the expectation is taken wrt the variational distribution. Then sample the variational distribution, evaluate f to get the MC estimate of the gradient. Essentially this gives us a generic variational method that requires only that we be able to evaluate the log-likelihood for each new model, then we can use a library of variance-reducing techniques.

Long story short we get,

$$\nabla_{\lambda_{n,l,k}} L = \mathbb{E}_q[\nabla_{\lambda_{n,l,k}} \log q(z_{n,l,k})(\log p_{n,l,k}(x, z, W) - \log q(z_{n,l,k}))$$

which we evaluate at several samples to get MC estimate.

The paper details the Markov blankets for latent variables at all parts of the architecture.

9. Exponential Family Embeddings: Motivation

There are many variants of word embeddings, but they reflect the same core ideas. Each term in a vocabulary is associated with an embedding and a context vector which govern conditional probabilities that relate each word to its context. The conditional probability of a word combines its embedding and the context vectors of its surrounding words. We want to generalize word embeddings to other types of high dimensional data where a data point is governed by its context.

Define an ef-embedding as (1) a context which specifies which other data points each observation depends on, (2) a conditional exponential family that sets the appropriate distribution (Gauss for real-valued, Poisson for counts) and combines embeddings and context vectors to form its natural parameter, and (3) an embedding structure which specifies how embeddings and context vectors are shared across the conditional distributions of each observation.

10. Exponential Family Embeddings: Setup

Consider a matrix x of I observations where each x_i is a D-vector. ex: in language x_i indicator for word at position i and D is vocab size.

We need a context function, a conditional exponential family, and an embedding structure.

11. Context

Let each data point i have a context c_i, a set of indices of other data points. Simply model $x_i|x_{c_i}$.

12. Conditional Exponential Family

$x_i|x_{c_i}$ is drawn from $EXPFAM(\nu_i(x_{c_i}), T(x_i))$ where ν natural parameter, T sufficient statistic. Let the embedding be $\rho[i] \in \mathbb{R}^K$. Let the context be $\alpha[i] \in \mathbb{R}^{K \times D}$.

We write the natural parameter wrt these vectors. Consider the linear embedding

$$\nu_i(x_{c_i}) = f_i\Big(\rho[i]^T \sum_{j \in c_i} \alpha[j] x_j\Big)$$

13. Embedding Structure

How do we share vectors across the data? For language, the ρ and α are shared across all positions. But what if we want to restrict this sharing? We can index ρ_n and α_n.

14. Objective Function

We want to sum the log conditional probability of each data point, including regularizers for embeddings and context vectors (i.e. gaussian probability might lead to l2 regularization). We also use regularizers if we want to constrain the embedding ex. to be nonnegative.

$$L(\rho, \alpha) = \sum_{i=1}^{I} (\nu_i^T T(x_i)) + \log p(\rho) + \log p(\alpha)$$

which is just the conditional likelihood.

15. Examples

See paper for examples.

16. Zero-Inflated Exponential Family Embeddings

http://proceedings.mlr.press/v70/liu17a/liu17a.pdf JMLR 2016

What about sparse data? This is something I happen to care about a lot. Genomics data are sparse vectors in the context of both other genes and other modalities (i.e. single cell RNA in the context of ATAC-seq gives transcriptomics in the context of regulation of transcription).

Zeros dominate the data which leads to bad embeddings. And what is 'true sparsity' vs sparsity introduced by sampling (sequencing method).

Zeroes might occur because they are part of the underlying process we want to model or they are part of a different process.

The idea here is to use exponential family embeddings with conditional distributions that place extra probability mass on zero that captures 'alternative reasons' that an observation might be 0. Effectively we downweight the zeros of the data st the embeddings no longer need to explain all zeros, improving the representation.

17. Exposure Modeling

Add an exposure indicator to EFE. For each value x_{ij} we define an exposure indicator b_{ij} to indicate whether the item j is exposed to the interaction with context items. Each b_{ij} is Bernoulli random variable Bernoulli($u_i j$). Exposure indicators and probabilities are denoted,

$$b_{ij} : j \in s_i, u_i j | j \in s_i$$

Suppose we have information about the exposure probability u_{ij} given by covariates $v_i \in \mathbb{R}^d$,

$$u_{ij} = logistic(w_j^T v_i + w_j^0)$$

If there are no covariates we just use the intercept and the exposure probabilities are shared by all observation-context pairs.

18. Incorporating Exposure Model in Embedding Model

When x_i is an observation j the indicator b_{ij} decides whether x_{ij} is zero or from the embedding distribution with a delta function. This gives us x_i^+ which is a smaller embedding model restricted to s_i^+ the context to which it is exposed.