Deep Exponential Families

Rajesh Ranganath Princeton University

Linpeng Tang Princeton University

Laurent Charlin Columbia University

David M. Blei Columbia University

{rajeshr,linpengt}@cs.princeton.edu
{lcharlin,blei}@cs.columbia.edu

Abstract

We describe *deep exponential families* (DEFs), a class of latent variable models that are inspired by the hidden structures used in deep neural networks. DEFs capture a hierarchy of dependencies between latent variables, and are easily generalized to many settings through exponential families. We perform inference using recent "black box" variational inference techniques. We then evaluate various DEFs on text and combine multiple DEFs into a model for pairwise recommendation data. In an extensive study, we show going beyond one layer improves predictions for DEFs. We demonstrate that DEFs find interesting exploratory structure in large data sets, and give better predictive performance than state-of-the-art models.

1 Introduction

In this paper we develop deep exponential families (DEFs), a flexible family of probability distributions that reflect the intuitions behind deep unsupervised feature learning. In a DEF, observations arise from a cascade of layers of latent variables. Each layer's variables are drawn from an exponential family that is governed by the inner product of the previous layer's variables and a set of weights.

As in deep unsupervised feature learning, a DEF represents hidden patterns, from coarse to fine grained, that compose with each other to form the observations. DEFs also enjoy the advantages of probabilistic modeling. Through their connection to exponential families [7], they support many kinds of data. Overall DEFs combine the powerful representations of deep networks with the flexibility of graphical models.

Consider the problem of modeling documents. We can represent a document as a vector of term counts modeled with Poisson random variables [9]. In one type of DEF, the rate of each term's Poisson count is an inner product of a layer of latent variables (one level up from the terms) and a set of weights that are shared across documents. Loosely, we can think of the latent layer the observations as per-document "topic" activations, each of which ignites a set of related terms via their inner product with the weights. These latent topics are, in turn, modeled in a similar way, conditioned on a layer above of "super topics." Just as the topics group related terms, the super topics group related topics, again via the inner product.

Figure 1 illustrates an example of a three level DEF uncovered from a large set of articles in The New York Times. (This style of model, though with different details, has been previously studied in the topic modeling literature [21].) Conditional on the word counts of the articles, the DEF defines a posterior distribution of the per-document cascades of latent variables and the layers of weights. Here we have visualized two third-layer topics which correspond to the *concepts* of "Government" and "Politics". We focus on "Government" and notice that the model has discovered, through its second-layer *super-topics*, the three branches of government: judiciary (left), legislative (center) and executive (right).

This is just one example. In a DEF, the latent variables can be from any exponential family: Bernoulli latent variables recover the classical sigmoid belief network [25]; Gamma latent variables give something akin to deep version of nonnegative matrix factorization [20]; Gaussian latent variables lead to the types of models that have recently been explored in the context

Appearing in Proceedings of the 18^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2015, San Diego, CA, USA. JMLR: W&CP volume 38.

of computer vision [28]. DEFs fall into the broad class of stochastic feed forward networks defined by Neal [25]. These networks differ from the undirected deep probabilistic models [30, 37] in that they allow for explaining away, where latent variables compete to explain the observations.

In addition to varying types of latent variables, we can further change the prior on the weights and the observation model. Observations can be real valued, such as those from music and images, binary, such as those in the sigmoid belief network, or multinomial, such as when modeling text. In the language of neural networks, the prior on the weights amounts to choosing a type of regularization; the observation model amounts to choosing a type of loss.

Finally, we can embed the DEF in a more complex model, building "deep" versions of traditional models from the statistics and machine learning research literature. As examples, the DEF can be made part of a multi-level model of grouped data [10], time-series model of sequential data [3], or a factorization model of pairwise data [31]. As am concrete example, we will develop and study the *double DEF*. The double DEF models a matrix of pairwise observations, such as users rating items. It uses two DEFs, one for the latent representation of users and the other for items. The observation of each user/item interaction combines the lowest layer of their individual DEF representations.

In the rest of this paper, we will define, develop, and study deep exponential families. We will explain some of their properties and situate them in the larger contexts of probabilistic models and deep neural networks. We will then develop generic variational inference algorithms for using DEFs. We will show how to use them to solve real-world problems with large data sets, and we will extensively study many DEFs on the problems of document modeling and collaborative filtering. We show that DEF-variants of existing "shallow" models give more interesting exploratory structure and better predictive performance. More generally, DEFs are a flexible class of models which, along with our algorithms for computing with them, let us easily explore a rich landscape of solutions for modern data analysis problems.

In this section we review exponential families and present deep exponential families.

Exponential families. Exponential families [7] are an important class of distributions with convenient mathematical properties. Their form is

$$p(x) = h(x) \exp(\eta^\top T(x) - a(\eta)),$$

where h is the base measure, η are the natural parameters, T are the sufficient statistics, and a is the log-normalizer. The expectation of the sufficient statistics of an exponential family is the gradient of the log-normalizer $\mathrm{E}[T(x)] = \nabla_\eta a(\eta)$. Exponential families are completely specified by their sufficient statistics and base measure; different choices of h and T lead to different distributions. For example, in the normal distribution the base measure is $h = \sqrt{(2\pi)}$ and the sufficient statistics are $T(x) = [x, x^2]$; and for the Beta distribution, a distribution with support over $(0, 1)$, the base measure is $h = 1$ and sufficient statistics are and $T(x) = [\log x, \log 1 - x]$.

Deep exponential families. To construct deep exponential families, we chain exponential families together in a hierarchy, where the draw from one layer controls the natural parameters of the next.

For each data point x_n, the model has L layers of hidden variables $\{\mathbf{z}_{n,1}, ..., \mathbf{z}_{n,L}\}$, where each $\mathbf{z}_{n,\ell} = \{z_{n,\ell,1}, ..., z_{n,\ell,K_\ell}\}$. We assume that $z_{n,\ell,k}$ is a scalar, but the model generalizes beyond this. Shared across data, the model has $L - 1$ layers of weights $\{\mathbf{W}_1, ...\mathbf{W}_{L-1}\}$, where each $\mathbf{W}_\ell$ is a collection of K_ℓ vectors, each one with dimension $K_{\ell+1}$: $\mathbf{W}_\ell = \{\mathbf{w}_{\ell,1}, ...\mathbf{w}_{\ell,K_\ell}\}$. We assume the weights have a prior distribution $p(\mathbf{W}_\ell)$.

For simplicity, we omit the data index n and describe the distribution of a single data point x. First, the top layer of latent variables are drawn given a hyper-parameter η

$$p(z_{L,k}) = \textsc{expfam}_L(z_{L,k}, \eta),$$

where the notation $\textsc{expfam}(x, \eta)$ denotes x is drawn from an exponential family with natural parameter η.[1]

Next, each latent variable is drawn conditional on the previous layer,

$$p(z_{\ell,k} \,|\, \mathbf{z}_{\ell+1}, \mathbf{w}_{\ell,k}) = \textsc{expfam}_\ell(z_{\ell,k}, g_\ell(\mathbf{z}_{\ell+1}^\top \mathbf{w}_{\ell,k})). \tag{1}$$

The function g_ℓ maps the inner product to the natural parameter. Similar to the literature on generalized linear models [26], we call it the *link function*. Figure 2 depicts this conditional structure in a graphical model.

Confirm that the dimensions work: $z_{\ell,k}$ is a scalar; $\mathbf{z}_{\ell+1}$ is a $K_{\ell+1}$ vector and $\mathbf{w}_{\ell,\mathbf{k}}$ is a column from a $K_{\ell+1} \times K_\ell$ dimension matrix. Note each of the K_ℓ variables in layer ℓ depends on all the variables of the higher layer. This gives the model the flavor of a neural network. The subscript ℓ on EXPFAM indicates the type of exponential family can change across layers. This hierarchy of latent variables defines the DEF.

[1] We are loose with the base measure h as it can be absorbed into the dominating measure.

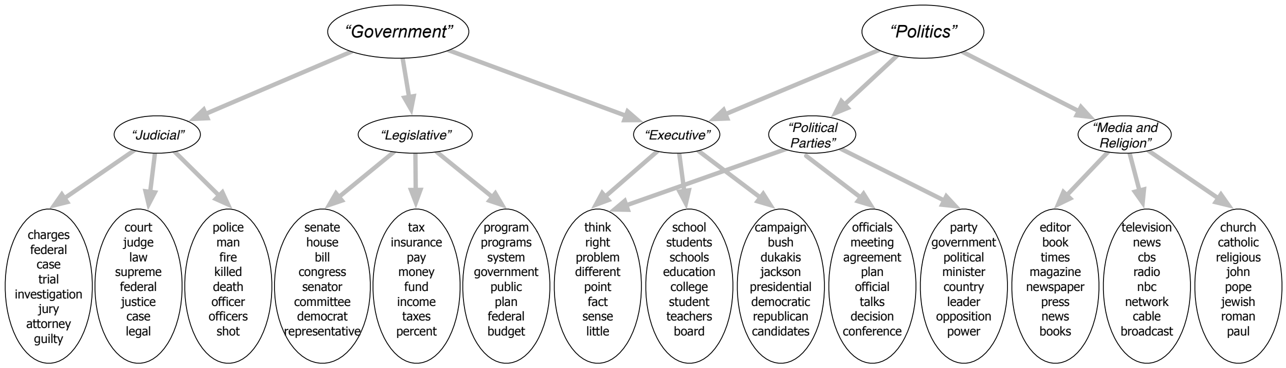

Figure 1: A fraction of the three layer topic hierarchy on 166K *The New York Times* articles. The top words are shown for each topic. The arrows represent hierarchical groupings.

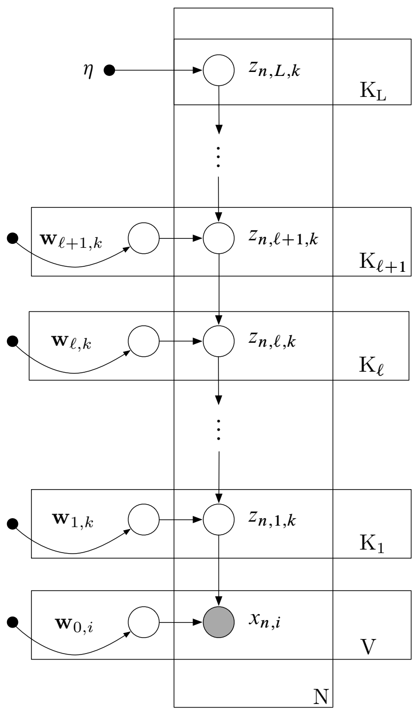

Figure 2: The deep exponential family with V observations.

DEFs can also be understood as random effects models [11] where the variables are controlled by the product of a weight vector and a set of latent covariates.

Likelihood. The data are drawn conditioned on the lowest layer of the DEF, $p(x_{n,i} \,|\, \mathbf{z_{n,1}})$. Separating the likelihood from the DEF will allow us to compose and embed DEFs in other models. Later, we provide an example where we combine two DEFs to form a model for pairwise data.

In this paper we focus on count data, thus we use the Poisson distribution as the observation likelihood. The Poisson distribution with mean λ is

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

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

$$p(x_{n,i} \,|\, \mathbf{z}_1, \mathbf{W_0}) = \text{Poisson}(\mathbf{z}_{n,1}^\top \mathbf{w}_{0,i}),$$

The observation weights $\mathbf{W_0}$ is matrix where each entry is gamma distributed. We will discuss gamma distribution further in the next section.

Returning to the example from the introduction of modeling documents, the x_n are a vector of term counts. The observation weights $\mathbf{W_0}$ put positive mass on groups of terms and thus form "topics." Similarly, the weights on the second layer represents "super topics," and the weights on the third layer represent "concepts." The distribution $p(\mathbf{z}_{n,1} \,|\, \mathbf{z}_{n,2}, \mathbf{W}_1)$ represents the distribution of "topics" given the "super topics" of a document. Figure 1 depicts the compositional and sharing semantics of DEFs.

The link function. Here we explore some of the connections between neural networks and deep exponential families. As we discussed, the latent variable layers in deep exponential families are connected together via a link function, g_ℓ. This link function specifies the natural parameters for $z_{\ell,k}$ from $\mathbf{z}_{\ell+1}^\top \mathbf{w}_{\ell,k}$.

Using properties of exponential families we can determine how the link function alters the distribution of the ℓth layer. The moments of the sufficient statistics of an exponential family are given by the gradient of the log-normalizer $\nabla_\eta a(\eta)$. These moments completely specify the exponential family [7]. Thus in DEFs, the mean of the next layer is controlled by the link function g_ℓ via the gradient of the log-normalizer,

$$\mathrm{E}[T(z_{\ell,k})] = \nabla_\eta a(g_l(\mathbf{z}_{\ell+1}^\top \mathbf{w}_{\ell,k})). \qquad (2)$$

Consider the case of the identity link function, where $g_l(x) = x$. In this case, the expectation of z_ℓ in deep exponential families is a linear function of the weights and previous layer transformed by the gradient of the log-normalizer. This transformation of the expectation is one source of non-linearity in DEFs. It parallels the non-linearities used in neural networks.

To be clear, here is a representation of how the values and weights of one layer control the expected sufficient statistics at the next:

$$\begin{array}{l} \mathbf{w}_{\ell,k} \searrow \\ \qquad\quad g_\ell \xrightarrow{\nabla a} \mathrm{E}[T(z_{n,\ell,k})] \\ \mathbf{z}_{n,\ell+1} \nearrow \end{array}$$

For example, in the sigmoid belief network [25], we will see that the identity link function recovers the sigmoid transformation used in neural networks.

2 Examples

To illustrate the potential of deep exponential families, we present three examples: the sparse gamma DEF, the sigmoid belief network, and a latent Poisson DEF.

Sparse gamma DEF. The sparse gamma DEF is a DEF with gamma distributed layers. The gamma distribution is an exponential family distribution with support over the positive reals. The probability density of the gamma distribution with natural parameters, α and β, is

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

where Γ is the gamma function. The expectation of the gamma distribution is $\mathrm{E}[z] = \alpha\beta^{-1}$.

Through the link function in DEFs, the inner product of the previous layer and the weights control the natural parameters of the next layer. For sparse gamma models, we let control the expected activation of the next layer, while the shape at each layer remains fixed for each layer at α_ℓ. From the expectation of the gamma distribution, these conditions mean the link function can be given as

$$g_\alpha = \alpha_\ell, \quad g_\beta = \frac{\alpha_\ell}{\mathbf{z}_{\ell+1}^\top \mathbf{w}_{\ell,k}}.$$

As the expectation of gamma variables needs to be positive, we let the weight matrices be gamma distributed as well.

We set the shape parameters for the weights and layers to be less than 1. When the shape is less than 1, gamma distributions put most of their mass near zero; we call this type of distribution *sparse gamma*. This type of distribution is akin to a soft spike-slab prior [16]. Spike and slab priors have shown to perform well on feature selection and unsupervised feature discovery [12, 14].

The sparse gamma distribution differs from distributions such as the normal and Poisson in how the probability mass moves given a change in the mean. For example, when the expected value is high, draws from the Poisson distribution are likely to be much larger than zero, while in the sparse gamma distribution draws will either be close to zero or very large. This is like growing the slab of our soft spike and slab prior. We visually demonstrate this in the appendix.

We estimate the posterior on this DEF using one to three layers for two large text copora: *Science* and *The New York Times* (NYT). We defer the discussion of the details of the corpora to Section 5. The topic hierarchy shown earlier in Figure 1 is from a three layer sparse gamma DEF. In the appendix we present a portion of the *Science* hierarchy.

Sigmoid belief network. The sigmoid belief network [23, 25] is a widely used deep latent variable model. It consists of latent Bernoulli layers where the mean of a feature at layer ℓ is given by a linear combination of the features at layer $\ell + 1$ with the weights passed through the sigmoid function.

This is a special case of a deep exponential family with Bernoulli latent layers and the identity link function. To see this, use Eq. 1 to form the Bernoulli conditional of a hidden layer,

$$\begin{aligned} &p(z_{\ell,k} \mid \mathbf{z}_{\ell+1}, \mathbf{w}_{\ell,k}) \\ &\quad = \exp(\mathbf{z}_{\ell+1}^\top \mathbf{w}_{\ell,k} z_{\ell,k} - \log(1 + \exp(\mathbf{z}_{\ell+1}^\top \mathbf{w}_{\ell,k}))) \end{aligned}$$

where $z_{\ell,k} \in \{0, 1\}$.

Using Eq. 2, the expectation of $z_{\ell,k}$ is the derivative of the log-normalizer of the Bernoulli. This derivative is the logistic function. Thus, a DEF with Bernoulli conditionals and identity link function recovers the sigmoid belief network. The weights are real valued, so we set $p(\mathbf{W}_\ell)$ to be a factorized normal distribution.

In the sigmoid belief network, we allow for the natural parameters to have intercepts. The intercepts provide a baseline activation for each feature independent of the shared weights.

Poisson DEF. The Poisson distribution is a distribution over counts that models the number of events occurring in an interval. Its exponential family form with parameter η is

$$p(z) = z!^{-1} \exp(\eta z - \exp(\eta)),$$

where the mean of this distribution is e^η.

We define the Poisson deep exponential family as a DEF with latent Poisson levels and log-link function. [2] This corresponds to the following conditional distribution for the layers

$$p(z_{\ell,k} \mid \mathbf{z}_{\ell+1}, \mathbf{w}_{\ell,k}) \\ = (z_{\ell,k}!)^{-1} \exp(\log(\mathbf{z}_{\ell+1}^\top \mathbf{w}_{\ell,k}) z_{\ell,k} - \mathbf{z}_{\ell+1}^\top \mathbf{w}_{\ell,k}).$$

In the case of document modeling, the value of $z_{2,k}$ represents how many times "super topic" k is represented in this example.

Using the link function property of DEFs described earlier, the mean activation at the next layer is given by the gradient of the log-normalizer

$$\mathrm{E}[z_{lk}] = \nabla_\eta a(\log(\mathbf{z}_{\ell+1}^\top \mathbf{w}_{\ell,k})).$$

For the Poisson, a is the exponential function. Thus its derivative is the exponential function. This means the mean of the next layer is equal to a linear combination of the weights of the previous layer. Our choice of link function requires positivity on the weights, so we let $p(\mathbf{W}_\ell)$ be a factorized gamma distribution.

We also consider Poisson DEFs with real valued weights to allow negative relations between lower layers and higher layers. We set the prior on the weights to be Gaussian in this case. We use log-softmax $\eta = \log(\log(1 + \exp(-\mathbf{z}_{\ell+1}^T \mathbf{w}_{\ell,k})))$ as the link function, where the function inside the first log is the softmax function. It preserves approximate linear relations between mean activation and the inner product $\mathbf{z}_{\ell+1}^T \mathbf{w}_{\ell,k}$ when it is large while allowing for the inner product to take negative values as well.

Similar to the sigmoid belief network, we allow the natural parameter to have an intercept. This type of Poisson DEF can be seen as an extension of the sigmoid belief network, where each observation expresses an integer count number of a feature rather than a binary feature. Table 1 summarizes the DEFs we have described and will study in our experiments.

3 Related Work

Graphical models and neural nets have a long and distinguished history. A full review is outside of the scope of this article, however we highlight some key results as they relate to DEFs. More generally, deep exponential families fall into the broad class of stochastic feed forward belief networks [25], but Neal [25] focuses mainly on one example in this class, the sigmoid belief network, which is a binary latent variable model. Several existing stochastic feed forward networks are DEFs, such as latent Gaussian models [28] and the sigmoid belief network with layerwise dependencies [23].

Undirected graphical models have also been used in inferring compositional hierarchies. Salakhutdinov and Hinton [30] propose deep probabilistic models based on Restricted Boltzmann Machines (RBMs) [36]. RBMs are a two layer undirected probabilistic model with one layer of latent variables and one layer of observations tied together by a weight matrix. Directed models such as DEFs have the property of explaining away, where independent latent variables under the prior become dependent conditioned on the observations. This property makes inference harder than in RBMs, but forces a more parsimonious representation where similar features compete to explain the data rather than work in tandem [12, 1].

RBMs have been extended to general exponential family conditionals in a model called exponential family harmoniums (EFH) [39]. A certain infinite DEF with tied weights is equivalent to an EFH [15], but as our weights are not tied, deep exponential families represent a broader class of models than exponential family harmoniums (and RBMs).

The literature of latent variable models relates to DEFs through hierarchical models and Bayesian factor analysis. Latent tree hierarchies have been constructed with specific distributions (Dirchlet) [21], while Bayesian factor analysis methods such as exponential family PCA [24] and multinomial PCA [8] can be seen as a single layer deep exponential family.

4 Inference

The central computational problem for working with DEFs is posterior inference. The intractability of the partition function means posterior computations require approximations. Past work on sigmoid belief networks has proposed doing greedy layer-wise learning (for a specific kind of network) [15]. Here, instead, we develop variational methods [35] that are applicable to general DEFs.

Variational inference [18] casts the posterior inference problem as an optimization problem. Variational algorithms seek to minimize the KL divergence to the posterior from an approximating distribution q. This is equivalent to maximizing the following [2],

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

where z denotes all latent variables associated with the observations and W all latent variables shared across observations. This objective function is called the Evidence Lower BOund (ELBO) because it is a lower bound on $\log p(x)$.

[2] We add an intercept to ensure positivity of the rate.

z-Dist	$\mathbf{z}_{\ell+1}$	W-dist	$\mathbf{w}_{\ell,k}$	g_ℓ	$\mathrm{E}[T(z_{\ell,k})]$
Gamma	$R_+^{K_{\ell+1}}$	Gamma	$R_+^{K_{\ell+1}}$	[constant; inverse]	$[z_{\ell+1}^\top \mathbf{w}_{\ell,k};\ \Psi(\alpha_\ell) - \log(\alpha) + \log(z_{\ell+1}^\top \mathbf{w}_{\ell,k})]$
Bernoulli	$\{0,1\}^{K_{\ell+1}}$	Normal	$R^{K_{\ell+1}}$	identity	$\sigma(z_{\ell+1}^\top \mathbf{w}_{\ell,k})$
Poisson	$N^{K_{\ell+1}}$	Gamma	$R_+^{K_{\ell+1}}$	log	$z_{\ell+1}^\top \mathbf{w}_{\ell,k}$
Poisson	$N^{K_{\ell+1}}$	Normal	$R^{K_{\ell+1}}$	log-softmax	$\log(1+\exp(z_{\ell+1}^\top \mathbf{w}_{\ell,k}))$

Table 1: A summary of all the DEFs we present in terms of their layer distributions, weight distributions, and link functions.

For the approximating distribution, q, we use the mean field variational family. In the mean field approximating family, the distribution over the latent variables factorizes. Let N be the number of observations, then the variational family is

$$q(z, W) = q(\mathbf{W}_0) \prod_{\ell=1}^{L} q(\mathbf{W}_\ell) \prod_{n=1}^{N} q(\mathbf{z}_{n,\ell}),$$

where $q(z_{n,\ell})$ and $q(W_\ell)$ are fully factorized. Each component in $q(\mathbf{z}_{n,\ell})$ is

$$q(z_{n,\ell,k}) = \text{EXPFAM}_\ell(z_{n,\ell,k}, \lambda_{n,\ell,k}),$$

where the exponential family is the same one as the model distribution p. Similarly, we choose $q(W)$ to be in the same family as $p(W)$ with parameters ξ.

To maximize the ELBO, we need to compute expectations under the approximation q. These expectations for general DEFs will not have a simple analytic form. Thus we use more recent "black box" variational inference techniques that step around computing this expectation [40, 34, 27].

Black box variational inference methods use stochastic optimization[29] to avoid the analytic intractability of computing the objective function. Stochastic optimization works by following noisy unbiased gradients. In black box variational inference [27], the gradient of the ELBO with respect to the parameters of a latent variable can be written as an expectation with respect to the variational approximation.

More formally, let $p_{n,\ell,k}(x, z, W)$ be the terms in the log-joint that contains $z_{n,\ell,k}$ (its Markov blanket), then the gradient for the approximation of $z_{n,\ell,k}$ is

$$\nabla_{\lambda_{n,l,k}} \mathcal{L} = E_q[\nabla_{\lambda_{n,\ell,k}} \log q(z_{n,\ell,k}) \\ (\log p_{n,\ell,k}(x, z, W) - \log q(z_{n,\ell,k}))].$$

We compute Monte Carlo estimates of this gradient by averaging the evaluation of the gradient at several samples. To compute the Monte Carlo estimate of the gradient, we need to be able to sample from the approximation to evaluate the Markov blanket for each latent variable, the approximating distribution, and the gradient of the log of the approximating distribution (score function). We detail the score functions in the appendix. From this equation, we can see that the primary cost in computing the gradients is in evaluating the likelihood and score function on a sample. To speed up our algorithm, we parallelize the likelihood computation across samples.

The Markov blanket for a latent variable in the first layer of a DEF is

$$\log p_{n,1,k}(x, z, W) = \log p(z_{n,1,k} | \mathbf{z}_{n,2}, \mathbf{w}_{1,k}) \\ + \log p(x_n | \mathbf{z}_{n,1}, \mathbf{W}_0). \quad (3)$$

The Markov blank for a latent variable in the intermediate layer is

$$\log p_{n,\ell,k}(x, z, W) = \log p(z_{n,\ell,k} | \mathbf{z}_{n,\ell+1}, \mathbf{w}_{\ell,k}) \\ + \log p(\mathbf{z}_{n,\ell-1} | \mathbf{z}_{n,\ell}, \mathbf{W}_{\ell-1}). \quad (4)$$

The Markov blanket for the top layer is

$$\log p_{n,L,k}(x, z, W) = \log p(z_{n,L,k}) \\ + \log p(\mathbf{z}_{n,L-1} | \mathbf{z}_{n,L}, \mathbf{W}_{L-1}). \quad (5)$$

The gradients for W can be written similarly.

Stochastic optimization requires a learning rate to scale the noisy gradients before applying them. We use RMSProp which scales the gradient by the square root of the online average of the squared gradient.[3] RMSProp captures the varying length scales and noise through the sum of squares term used to normalize the gradient step. We present a sketch of the algorithm in Alg. 1, and present the full algorithm in the appendix.

5 Experiments

We have introduced DEFs and detailed a procedure for posterior inference in DEFs. We now provide an extensive evaluation of DEFs. We report predictive results from 28 different DEF instances where we explore the number of layers (1, 2 or 3), the latent variable distributions (gamma, Poisson, Bernoulli) and the weight distributions (normal, gamma) using a Poisson observational model. Furthermore, we instantiate and

[3] www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

Algorithm 1 BBVI for DEFs

Input: data X, model p, L layers.
Initialize λ, ξ randomly, $t = 1$.
repeat
  Sample a datapoint x
  for s = 1 to S **do**
    $z_x[s], W[s] \sim q$
    $p[s] = \log p(z_x[s], W[s], x)$
    $q[s] = \log q(z_x[s], W[s])$
    $g[s] = \nabla \log q(z_x[s], W[s])$
  end for
  Compute gradient using BBVI
  Update variational parameters for z and W
until change in validation likelihood is small

report results using a combination of two DEFs for pairwise data.

Our results:

- Show improvements over strong baselines for both topic modeling and collaborative filtering on a total of four corpora.
- Lead us to conclude that deeper DEFs and sparse gamma DEFs display the strongest performance overall.

5.1 Text Modeling

We consider two large text corpora *Science* and *The New York Times*. *Science* consists of 133K documents and 5.9K terms. *The New York Times* consists of 166K documents and 8K terms.

Baselines. As a baseline we consider Latent Dirichlet Allocation [5] a popular topic model, and state-of-the-art DocNADE [19]. DocNADE estimates the probability of a given word in a document given the previously observed words in that document. In DocNADE, the connections between each observation and the latent variables used to generate the observations are shared.

We note that the one layer sparse gamma DEF is equivalent to Poisson matrix factorization [9, 13] but our model is fully Bayesian and our variational distribution is collapsed.

Evaluation. We compute perplexity on a held out set of 1,000 documents. Held out perplexity is given by

$$\exp\left(\frac{-\sum_{d \in \text{docs}} \sum_{w \in d} \log p(w \mid \#\text{ held out in } d)}{N_{\text{held out words}}}\right)$$

Conditional on the total number of held out words, the distribution of the held out words becomes multinomial. The mean of the conditional multinomial is given by the normalized Poisson rate in each document. We set the rates to the expected value under the variational distribution. Additionally, we let all methods see ten percent of the words in each document; the other ninety percent form the held out set. This is similar to the document completion evaluation metric [38] except we query the test words independently. We use the observed ten percent to compute the variational distribution for the document specific latent variables, the DEF for the document, while keeping the approximation on the shared weights fixed. In DocNADE, this corresponds to always seeing a fixed set of words first, then evaluating each new word given the first ten percent of the document.

Model	**W**	*NYT*	*Science*
LDA [6]		2717	1711
DocNADE [19]		2496	1725
Sparse Gamma 100	∅	2525	1652
Sparse Gamma 100-30	Γ	2303	1539
Sparse Gamma 100-30-15	Γ	2251	1542
Sigmoid 100	∅	2343	1633
Sigmoid 100-30	$\mathcal{N}$	2653	1665
Sigmoid 100-30-15	$\mathcal{N}$	2507	1653
Poisson 100	∅	2590	1620
Poisson 100-30	$\mathcal{N}$	2423	1560
Poisson 100-30-15	$\mathcal{N}$	2416	1576
Poisson log-link 100-30	Γ	2288	1523
Poisson log-link 100-30-15	Γ	2366	1545

Table 2: Perplexity on a held out collection of 1K *Science* and *NYT* documents. Lower values are better. The DEF **W** column indicates the type of prior distribution over the DEF weights, Γ for the gamma prior and $\mathcal{N}$ for normal (recall that one layer DEFs consist only of a layer of latent variables, thus we represent their prior with the ∅).

Held out perplexity differs from perplexity computed from the predictive distribution $p(x^* \mid x)$. The former can be a more difficult problem as we only ever condition on a fraction of the document. Additionally computing perplexity from the predictive distribution requires computationally demanding sampling procedures which for most models like LDA only allow testing of only a small number (50) of documents [38, 33]. In contrast our held-out test metric can be quickly computed for 1,000 test documents.

Architectures and hyperparameters. We build one, two and three layer hierarchies of the sparse gamma DEF, sigmoid belief network, Poisson DEF, and log-link Poisson DEF. The sizes of the layers are 100, 30, and 15, respectively. We note that while different DEFs may have better predictive performance

Model	Netflix Perplexity	Netflix NDCG	ArXiv Perplexity	ArXiv NDCG
Gaussian MF [32]	–	0.008	–	0.013
1 layer Double DEF	2319	0.031	2138	0.049
2 layer Double DEF	2299	0.022	1893	0.050
3 layer Double DEF	2296	0.037	1940	0.053

Table 3: A comparison of a matrix factorization methods on Netflix and the ArXiv. We find that the Double DEFs outperform the shallow ones on perplexity. We also find that the NDCG of around 100 low-activity users (users with less than 5 and 10 observations in the observed 10% of the held-out set respectively for Netflix and ArXiv). We use Vowpal Wabbit's MF implementation which does not readily provide held-out likelihoods and thus we do not report the perplexity associated with MF.

at different sizes, we consider DEFs of a fixed size as we also seek a compact explorable representation of our corpus. One hundred topics fall into the range of topics searched in the topic modeling literature [4]. We detail the hyperparameters in the appendix.

We observe two phases to DEF convergence; it converges quickly to a good held-out perplexity (around 2,000 iterations) and then slowly improves until final convergence (around 10,000 iterations). Each iteration takes approximately 15 seconds on a modern 32-core machine (from Amazon's AWS).

Results. Table 2 summarizes the predictive results on both corpora. We note that DEFs outperform the baselines on both datasets. Furthermore moving beyond one layer models generally improves performance as expected. The table also reveals that stacking layers of gamma latent variables always leads to similar or better performance. Finally, as shown by the Poisson DEFs with different link functions, we find gamma-distributed weights to outperform normally-distributed weights. Somewhat related, we find sigmoid DEFs (with normal weights) to be more difficult to train and deeper version perform poorly.

5.2 Matrix Factorization

Previously, we constructed models out of a single DEF, but DEFs can be embedded and composed in more complex models. We now present *double DEF*, a factorization model for pairwise data where both the rows and columns are determined by DEFs. The graphical model of the double DEF corresponds to replacing $\mathbf{W_0}$ in Figure 2 with another DEF.

We focus on factorization of counts (ratings, clicks). The observed data are generated with a Poisson. The observation likelihood for this double DEF is $p(x_{n,i} \,|\, \mathbf{z}^c_{n,1}, \mathbf{z}^r_{i,1}) = \text{Poisson}(\mathbf{z}^{c\top}_{n,1} \mathbf{z}^r_{i,1})$, where $\mathbf{z}^c_{n,1}$ is the lowest layer of a DEF for the nth observation and $\mathbf{z}^r_{i,1}$ is the lowest layer of a DEF for the ith item. The double DEF has hierarchies on both users and items.

We infer double DEFs on *Netflix* movie ratings and click data from the *ArXiv* (`www.arXiv.org`) which indicates how many times a user clicked on a paper. Our *Netflix* collection consists of 50K users and 17.7K movies. The movie ratings range from zero to five stars, where zero means the movie was unrated by the user. The *ArXiv* collection consists of 18K users and 20K documents. We fit a one, two, and three layer double DEF where the sizes of the row DEF match the sizes of the column DEF at each layer. The sizes of the layers are 100, 30, and 15. We compare double DEFs to $l2$-regularized (Gaussian) matrix factorization (MF) [32]. We re-use the testing procedure introduced in the previous section (this is referred to as *strong-generalization* in the recommendation literature [22]) where the held-out test set contains one thousand users. For performance and computational reasons we subsample zero-observations for MF as is standard [13]. Further, we also report the commonly-used multi-level ranking measure (untruncated) NDCG [17] for all methods.

Table 3 shows that two-layer DEFs improve performance over the shallow DEF and that all DEFs outperform Gaussian MF. On perplexity the three layer model performs similarly on Netflix and slightly worse on the ArXiv. The table further highlights that when comparing ranking performance, the advantage of deeper models is especially clear on low-activity users (NDCG across all test users is comparable within the three DEFs architectures and is not reported here). This data regime is of particular importance for practical recommender systems. We postulate that this due to the hierarchy in deeper models acting as a more structured prior compared to single-layer models.

6 Discussion

We develop deep exponential families as a way to describe hierarchical relationships of latent variables to capture compositional semantics of data. We present several instantiations of deep exponential families and achieve improved predictive power and interpretable semantic structures for both problems in text modeling and collaborative filtering.

Acknowledgements. We thank Prem Gopalan, Matt Hoffman, and the reviewers for their helpful comments. RR is supported by an NDSEG fellowship. DMB is supported by NSF IIS-0745520, NSF IIS-1247664, ONR N00014-11-1-0651, and DARPA FA8750-14-2-0009.

References

[1] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence*, 35(8), 2013.

[2] C. Bishop. *Pattern Recognition and Machine Learning*. Springer New York., 2006.

[3] D. Blei and J. Lafferty. Dynamic topic models. In *International Conference on Machine Learning*, 2006.

[4] D. Blei and J. Lafferty. A correlated topic model of Science. *Annals of Applied Stat.*, 1(1):17–35, 2007.

[5] D. Blei, T. Griffiths, M. Jordan, and J. Tenenbaum. Hierarchical topic models and the nested Chinese restaurant process. In *NIPS*, 2003.

[6] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3: 993–1022, January 2003.

[7] L. Brown. *Fundamentals of Statistical Exponential Families*. Institute of Mathematical Statistics, 1986.

[8] W. Buntine and A. Jakulin. Applying discrete PCA in data analysis. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 59–66. AUAI Press, 2004.

[9] J. Canny. GaP: A factor model for discrete data. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2004.

[10] A. Gelman and J. Hill. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge Univ. Press, 2007.

[11] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. CRC press, 2013.

[12] I. Goodfellow, A. Courville, and Y. Bengio. Large-scale feature learning with spike-and-slab sparse coding. In *International Conference on Machine Learning (ICML)*, 2012.

[13] P. Gopalan, J. Hoffman, and D. Blei. Scalable recommendation with poisson factorization. *arXiv*, (1311.1704), 2013.

[14] Daniel Hernández-Lobato, José Miguel Hernández-Lobato, and Pierre Dupont. Generalized spike-and-slab priors for bayesian group feature selection using expectation propagation. *The Journal of Machine Learning Research*, 14(1):1891–1945, 2013.

[15] G. Hinton, S. Osindero, and Y. Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7): 1527–1554, July 2006. ISSN 0899-7667.

[16] H. Ishwaran and S. Rao. Spike and slab variable selection: Frequentist and Bayesian strategies. *The Annals of Statistics*, 33(2):730–773, 2005.

[17] Kalervo Järvelin and Jaana Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *In International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pages 41–48. ACM, 2000. ISBN 1-58113-226-3.

[18] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul. Introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

[19] Hugo Larochelle and Stanislas Lauly. A neural autoregressive topic model. In *Neural Information Processing Systems*, 2012.

[20] D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755): 788–791, October 1999.

[21] W. Li and A. McCallum. Pachinko allocation:DAG-stuctured mixture models of topic correlations. In *ICML*, 2006.

[22] Benjamin Marlin. Collaborative filtering: A machine learning perspective. Technical report, University of Toronto, 2004.

[23] A. Mnih and K. Gregor. Neural variational inference and learning in belief networks. In *ICML*, 2014.

[24] S. Mohamed, K. Heller, and Z. Ghahramani. Bayesian exponential family PCA. In *NIPS*, 2008.

[25] R. Neal. Learning stochastic feedforward networks. *Tech. Rep. CRG-TR-90-7: Department of Computer Science, University of Toronto*, 1990.

[26] J. A. Nelder and R. W. M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135:370–384, 1972.

[27] R. Ranganath, S. Gerrish, and D. Blei. Black box variational inference. In *International Conference on Artifical Intelligence and Statistics*, 2014.

[28] D. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *ArXiv e-prints*, January 2014.

[29] H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3): pp. 400–407, 1951.

[30] R. Salakhutdinov and G. Hinton. Deep boltzmann machines. In *AISTATS*, pages 448–455, 2009.

[31] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *International Conference on Machine learning*, 2008.

[32] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Neural Information Processing Systems*, 2008.

[33] Ruslan Salakhutdinov and Geoffrey Hinton. Replicated softmax: an undirected topic model. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1607–1614. 2009.

[34] T. Salimans and D. Knowles. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882, 2013.

[35] L. Saul, T. Jaakkola, and M. Jordan. Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research*, 4:61–76, 1996.

[36] P. Smolenksy. Information processing in dynamical systems: Foundations of harmony theory. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, 1, 1986.

[37] Nitish Srivastava, Ruslan Salakhutdinov, and Geoffrey Hinton. Modeling documents with a deep boltzmann machine. In *UAI*, 2013.

[38] H. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In *International Conference on Machine Learning (ICML)*, 2009.

[39] M. Welling, M. Rosen-Zvi, and G. Hinton. Exponential family harmoniums with an application to information retrieval. In *Neural Information Processing Systems (NIPS) 17*, 2004.

[40] D. Wingate and T Weber. Automated variational inference in probabilistic programming. *ArXiv e-prints*, January 2013.

Deep Exponential Families (Appendix)

Rajesh Ranganath Princeton University
Linpeng Tang Princeton University
Laurent Charlin Columbia University
David M. Blei Columbia University

{rajeshr,linpengt}@cs.princeton.edu
{lcharlin,blei}@cs.columbia.edu

Appendix

Gamma Distribution Figure 1 visually demonstrates how the sparse gamma distrubution and Poisson distribution change when moving from low to high mean. We plot both the Poisson and sparse gamma distribution in both settings. The mode of the Poisson distribution moves, while the slab grows for the gamma.

General Algorithm. Following the notation from the main paper, the general algorithm for mean field variational inference in deep exponential families is given in (Alg. 1).

Properties of q. In our experiments, we use four variational families (Poisson, gamma, Bernoulli, and normal). We detail the necessary score functions here. For the Poisson, the distribution is given by:

$$q(z) = e^{-\lambda} \frac{\lambda^z}{z!}.$$

The score function is

$$\frac{\partial \log q(z)}{\partial \lambda} = -1 + \frac{z}{\lambda}.$$

For the gamma, we use the shape α and scale θ as variational parameters. The distribution is given by

$$q(z) = \frac{1}{\Gamma(\alpha)\theta^\alpha} z^{\alpha-1} e^{-z/\theta}.$$

The score function is

$$\frac{\partial \log q(z)}{\partial \alpha} = -\Psi(\alpha) - \log\theta, + \log z$$
$$\frac{\partial \log q(z)}{\partial \theta} = -\alpha/\theta + z/\theta^2,$$

where Ψ is the digamma function.

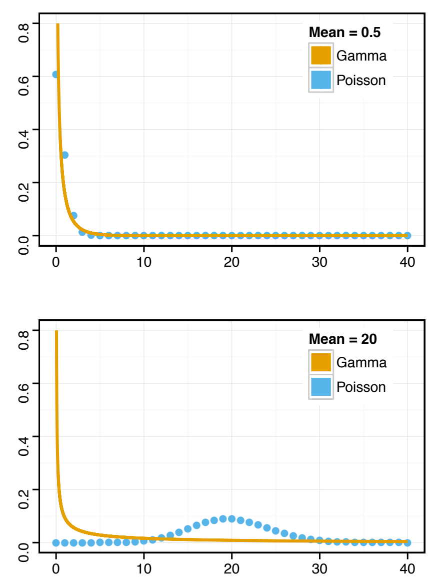

Figure 1: Draws from the Poisson (blue) and sparse gamma distribution (orange) with low and high mean. The shape of the sparse gamma is held fixed. Note the high mean shifts the Poisson, while does not shift the sparse gamma. Notice the spike-slab appearance of the sparse gamma distribution.

For the Bernoulli distribution, we use the natural parameterization with parameter η to form the variational approximation. The distribution is

$$q(z) = \frac{1}{1+e^{-(2z-1)\eta}}.$$

The score function is

$$\frac{\partial \log q(z)}{\partial \eta} = (2z-1)\frac{e^{-(2z-1)\eta}}{1+e^{-(2z-1)\eta}}.$$

For the normal variational approximation, we use the standard parameterization of mean μ and variance σ^2. The distribution is

$$q(z) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

The score function is

$$\frac{\partial \log q(z)}{\partial \mu} = \frac{x-\mu}{\sigma^2}$$
$$\frac{\partial \log q(z)}{\partial \sigma^2} = -\frac{1}{2\sigma^2} + \frac{(x-\mu)^2}{2\sigma^4}.$$

Parameterizations of Variational Distributions. Several of our variational parameters like the variance of the normal have positive constraints To enforce positivity constraints, we transform an unconstrained variable by $\log(1+\exp(x))$. To avoid numerical issues when sampling, we truncate values when appropriate. Gradients of the unconstrained parameters are obtained with the chain rule from the score function and the derivative of softmax: $\exp(x)/(1+\exp(x))$.

Optimization We perform gradient ascent step on the ELBO using

$$\Delta\theta = \rho\Gamma\nabla_\theta \mathrm{ELBO} \quad (1)$$

ρ is a fixed scalar set to 0.2 in our experiments. ∇_θELBO is a noisy gradient estimated using BBVI. Γ is a diagonal preconditioning matrix estimated using the RMSProp heuristic. A diagonal element of Γ is the reciprocal of the squared root of a running average of the squares of historical gradients of that component. We used a window size of 10 in our experiments.

Hyperparameters and Convergence We use the same hyperparameters on Gamma distributions on each layer with shape and rate 0.3. For the sigmoid belief network we use a prior of 0.1 to achieve some sparsity as well. We fix the Poisson prior rate to be 0.1. For gamma W's we use shape 0.1 and rate 0.3. For Gaussian W's we use a prior mean of 0 and variance of 1. We let the experiments run for 10,000 iterations at which point the validation likelihood is stable.

Algorithm 1 BBVI for DEFs

```
Input: data X, model p, L layers .
Initialize λ, ξ randomly, t = 1.
repeat
  // Draw a single data point from X
  n =Unif(D)
  // Get S samples in parallel
  for s = 1 to S do
    z_1[s] ~ q(z_1; λ_{n,1})
    W_0[s] ~ q(W_0 | ξ_0)
    p_0[s] = log p(x_n | z_1[s], W_0[s])
    q_1[s] = log q(z_1[s]; λ_{n,1})
    g_1[s] = ∇_{λ_{n,1}} log q(z_1[s]; λ_{n,1})
    g_{W_0}[s] = ∇_{ξ_1} log q(W_0)
    p_{W_0}[s] = log p(W_0; ξ_1)
    q_{W_0}[s] = log q(W_0; ξ_1)
    for l = 2 to L do
      z_l[s] ~ q(z_l; λ_{n,l})
      W_{l-1}[s] ~ q(W_{l-1} | ξ_{l-1})
      p_l[s] = log p(z_{l-1} | z_l, W_{l-1}[s])
      q_l[s] = log q(z_l; λ_{n,l})
      g_l[s] = ∇_{λ_{n,l}} log q(z_l; λ_{n,l})
      g_{W_{l-1}}[s] = ∇_{ξ_{l-1}} log q(W_{l-1})
      p_{W_{l-1}}[s] = log p(W_{l-1}; ξ_{l-1})
      q_{W_{l-1}}[s] = log q(W_{l-1}; ξ_{l-1})
    end for
    p_L[s] = log p(z_L)
  end for
  // Update parameters
  for l = 1 to L do
    for k = 1 to K_l do
      S = g_{l,k}(p_{l-1} + p_{l,k} - q_{l,k})
      λ_{n,1,k} = λ_{n,1,k} + ρ mean(S)
    end for
    T = g_{W_{l-1}}(p_{W_{l-1}} - q_{W_{l-l}} + p_{l-1})
    ξ_{l-1} = ξ_{l-1} + ρ mean(T)
  end for
until change of val likelihood is less than ε.
```

For the double DEF, we set all shapes to 0.1 and rates to 0.3. We let the Double DEF experiment run for about 10,000 iterations. The validation likelihood had converged for all models by this point.

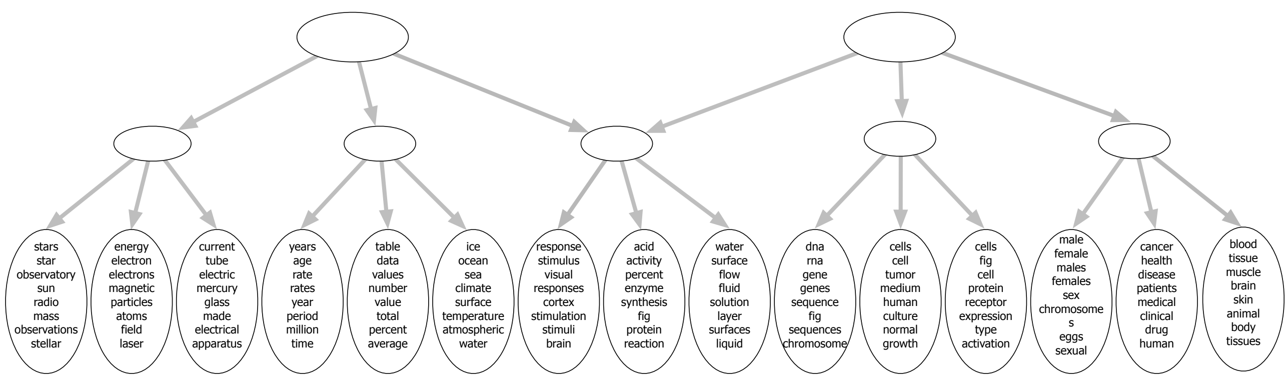

Figure 2: A fraction of the three layer topic hierarchy of the *Science* corpus. The top words are shown for each "topic." The arrows represent hierarchical groupings. We choose top three components at each layer. Similar "topics" are grouped into "super topics." The two "concepts" share a "super topic."