Latent Gaussian Processes for Distribution Estimation of Multivariate Categorical Data

Yarin Gal YG279@CAM.AC.UK
Yutian Chen YC373@CAM.AC.UK
Zoubin Ghahramani ZOUBIN@CAM.AC.UK
University of Cambridge

Abstract

Multivariate categorical data occur in many applications of machine learning. One of the main difficulties with these vectors of categorical variables is sparsity. The number of possible observations grows exponentially with vector length, but dataset diversity might be poor in comparison. Recent models have gained significant improvement in *supervised* tasks with this data. These models embed observations in a continuous space to capture similarities between them. Building on these ideas we propose a Bayesian model for the *unsupervised* task of distribution estimation of multivariate categorical data.

We model vectors of categorical variables as generated from a non-linear transformation of a continuous latent space. Non-linearity captures multi-modality in the distribution. The continuous representation addresses sparsity. Our model ties together many existing models, linking the linear categorical latent Gaussian model, the Gaussian process latent variable model, and Gaussian process classification. We derive inference for our model based on recent developments in sampling based variational inference. We show empirically that the model outperforms its linear and discrete counterparts in imputation tasks of sparse data.

1 Introduction

Multivariate categorical data is common in fields ranging from language processing to medical diagnosis. Recently proposed supervised models have gained significant improvement in tasks involving *big labelled data* of this form (see for example Bengio et al. (2006); Collobert & Weston (2008)). These models rely on information that had been largely ignored before: similarity between vectors of categorical variables. But what should we do in the unsupervised setting, when we face *small unlabelled data* of this form?

Medical diagnosis provides good examples of small unlabelled data. Consider a dataset composed of test results of a relatively small number of patients. Each patient has a medical record composed often of dozens of examinations, taking various categorical test results. We might be faced with the task of deciding which tests are necessary for a patient under examination to take, and which examination results could be deduced from the existing tests. This can be achieved with *distribution estimation*.

Several tools in the Bayesian framework could be used for this task of distribution estimation of unlabelled small datasets. Tools such as the Dirichlet-Multinomial distribution and its extensions are an example of such. These rely on relative frequencies of categorical variables appearing with others, with the addition of various smoothing techniques. But when faced with long multivariate sequences, these models run into problems of sparsity. This occurs when the data consists of vectors of categorical variables with most configurations of categorical values not in the dataset. In medical diagnosis this happens when there is a large number of possible examinations compared to a small number of patients.

Building on ideas used for big labelled discrete data, we propose a Bayesian model for distribution estimation of small unlabelled data. Existing supervised models for discrete labelled data embed the observations in a continuous space. This is used to find the similarity between vectors of categorical variables. We extend this idea to the small unlabelled domain by modelling the continuous embedding as a latent variable. A generative model is used to find a distribution over the discrete observations by modelling them as dependent on the continuous latent variables.

Following the medical diagnosis example, patient n would be modelled by a continuous latent variable $x_n \in \mathcal{X}$. For each examination d, the latent x_n induces a vector of probabilities $\mathbf{f} = (f_{nd1}, ..., f_{ndK})$, one probability for each possible test result k. A categorical distribution returns test result y_{nd} based on these probabilities, resulting in a patient's medical assessment $y_n = (y_{n1}, ..., y_{nD})$. We need to de-

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

cide how to model the distribution over the latent space $\mathcal{X}$ and vectors of probabilities $\mathbf{f}$.

We would like to capture sparse multi-modal categorical distributions. A possible approach would be to model the continuous representation with a simple latent space and non-linearly transform the points in the space to obtain probabilities. In this approach we place a standard normal distribution prior on a latent space, and feed the output of a non-linear transformation of the latent space into a Softmax to obtain probabilities. We use sparse Gaussian processes (GPs) to transform the latent space non-linearly. Sparse GPs form a distribution over functions supported on a small number of points with linear time complexity with respect to the number of data points (Quiñonero-Candela & Rasmussen, 2005; Titsias, 2009). We use a covariance function that is able to transform the latent space non-linearly. We name this model the *Categorical Latent Gaussian Process* (CLGP). Using a Gaussian process with a linear covariance function recovers the linear Gaussian model (LGM, Marlin et al., 2011). This model linearly transforms a continuous latent space resulting in discrete observations.

The Softmax likelihood is not conjugate to the our Gaussian prior, and integrating the latent variables with a Softmax distribution is intractable. A similar problem exists with LGMs. Marlin et al. (2011) solved this by using variational inference and various bounds for the likelihood in the binary case, or alternative likelihoods to the Softmax in the categorical case (Khan et al., 2012). Many bounds have been studied in the literature for the binary case: Jaakkola and Jordan's bound (Jaakkola & Jordan, 1997), the tilted bound (Knowles & Minka, 2011), piecewise linear and quadratic bounds (Marlin et al., 2011), and others. But for categorical data fewer bounds exist, since the multivariate Softmax is hard to approximate in high-dimensions. The Bohning bound (Böhning, 1992) and Blei and Lafferty's bound (Blei & Lafferty, 2006) give poor approximation (Khan et al., 2012).

Instead we use recent developments in sampling-based variational inference (Blei et al., 2012) to avoid integrating the latent variables with the Softmax analytically. Our approach takes advantage of this tool to obtain simple yet powerful model and inference. We use Monte Carlo integration to approximate the non-conjugate likelihood obtaining noisy gradients (Kingma & Welling, 2013; Rezende et al., 2014; Titsias & Lázaro-Gredilla, 2014). We then use learning-rate free stochastic optimisation (Tieleman & Hinton, 2012) to optimise the noisy objective. We leverage symbolic differentiation (Theano, Bergstra et al., 2010) to obtain simple and modular code[1].

[1]Python code for the model and inference is given in the appendix, and available at `http://github.com/yaringal/CLGP`

We experimentally show the advantages of using non-linear transformations for the latent space. We follow the ideas brought in Paccanaro & Hinton (2001) and evaluate the models on relation embedding and relation learning. We then demonstrate the utility of the model in the real-world *sparse data* domain. We use a medical diagnosis dataset where data is scarce, comparing our model to discrete frequency based models. We use the estimated distribution for a task similar to the above, where we attempt to infer which test results can be deduced from the others. We compare the models on the task of imputation of raw data studying the effects of government terror warnings on political attitudes. We then evaluate the continuous models on a binary Alphadigits dataset composed of binary images of handwritten digits and letters, where each class contains a small number of images. We inspect the latent space embeddings and separation of the classes. Lastly, we evaluate the robustness of our inference, inspecting the Monte Carlo estimate variance over time.

2 Related Work

Our model (CLGP) relates to some key probabilistic models (fig. 1). It can be seen as a non-linear version of the *latent Gaussian model* (LGM, Khan et al. (2012)) as discussed above. In the LGM we have a standard normal prior placed on a latent space, which is transformed linearly and fed into a Softmax likelihood function. The probability vector output is then used to sample a single categorical value for each categorical variable (e.g. medical test results) in a list of categorical variables (e.g. medical assessment). These categorical variables correspond to elements in a multivariate categorical vector. The parameters of the linear transformation are optimised directly within an EM framework. Khan et al. (2012) avoid the hard task of approximating the Softmax likelihood by using an alternative function (product of sigmoids) which is approximated us-

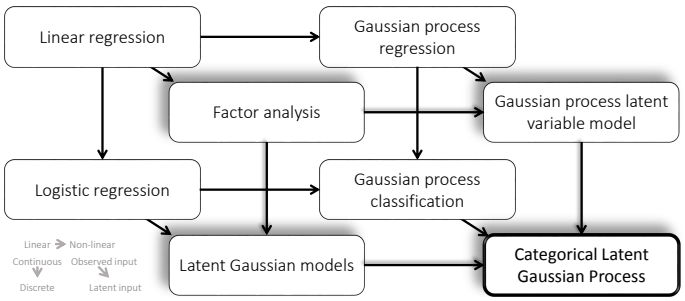


Figure 1. Relations between existing models and the model proposed in this paper (*Categorical Latent Gaussian Process*); the model can be seen as a non-linear version of the *latent Gaussian model* (left to right), it can be seen as a latent counterpart to the *Gaussian process classification* model (back to front), or alternatively as a discrete extension of the *Gaussian process latent variable model* (top to bottom).

ing numerical techniques. Our approach avoids this cumbersome inference.

Our proposed model can also be seen as a discrete extension of the *Gaussian process latent variable model* (GPLVM, Lawrence, 2005). This model has been proposed recently as means of performing non-linear dimensionality reduction (counterpart to the linear principal component analysis (Tipping & Bishop, 1999)) and density estimation in continuous space. Lastly, our model can be seen as a latent counterpart to the *Gaussian process classification* model (Williams & Rasmussen, 2006), in which a Softmax function is again used to discretise the continuous values. The continuous valued outputs are obtained from a Gaussian process, which non-linearly transforms the inputs to the classification problem. Compared to GP classification where the inputs are fully observed, in our case the inputs are latent. However, note that GP classification is a special case of the CLGP model, in the same way that GP regression can be seen as a special case of the Gaussian process latent variable model. In both cases the latent inputs can be modelled with delta functions fixed at the observations which are not optimised.

3 A Latent Gaussian Process Model for Multivariate Categorical Data

We consider a generative model for a dataset $\mathbf{Y}$ with N observations (patients for example) and D categorical variables (different possible examinations). The d-th categorical variable in the n-th observation, y_{nd}, is a categorical variable that can take an integer value from 0 to K_d. For ease of notation, we assume all the categorical variables have the same cardinality, i.e. $K_d \equiv K,\ \forall d = 1, \dots, D$.

In our generative model, each categorical variable y_{nd} follows a categorical distribution with probability given by a Softmax with weights $\mathbf{f}_{nd} = (0, f_{nd1}, \dots, f_{ndK})$. Each weight f_{ndk} is the output of a nonlinear function of a Q dimensional latent variable $\mathbf{x}_n \in \mathbb{R}^Q$: $\mathcal{F}_{dk}(\mathbf{x}_n)$. To complete the generative model, we assign an isotropic Gaussian distribution prior with standard deviation σ_x^2 for the latent variables, and a Gaussian process prior for each of the nonlinear functions. We also consider a set of M auxiliary variables which are often called inducing inputs. These inputs $\mathbf{Z} \in \mathbb{R}^{M \times Q}$ lie in the latent space with their corresponding outputs $\mathbf{U} \in \mathbb{R}^{M \times D \times K}$ lying in the weight space (together with f_{ndk}). The inducing points are used as "support" for the function. Evaluating the covariance function of the GP on these instead of the entire dataset allows us to perform approximate inference in $\mathcal{O}(M^2 N)$ time complexity instead of $\mathcal{O}(N^3)$ (where M is the number of inducing points and N is the number of data points (Quiñonero-Candela & Rasmussen, 2005)).

The model is expressed as:

$$x_{nq} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_x^2) \tag{1}$$

$$\begin{aligned} \mathcal{F}_{dk} &\overset{\text{iid}}{\sim} \mathrm{GP}(0, \mathbf{K}_d) \\ f_{ndk} &= \mathcal{F}_{dk}(\mathbf{x}_n), \quad u_{mdk} = \mathcal{F}_{dk}(\mathbf{z}_m) \\ y_{nd} &\sim \mathrm{Softmax}(\mathbf{f}_{nd}), \end{aligned}$$

for $n \in [N]$ (the set of naturals between 1 and N), $q \in [Q]$, $d \in [D]$, $k \in [K]$, $m \in [M]$, covariance matrices $\mathbf{K}_d$, and where the Softmax distribution is defined as,

$$\mathrm{Softmax}(y = k; \mathbf{f}) = \mathrm{Categorical}\left(\frac{\exp(f_k)}{\exp(\mathrm{lse}(\mathbf{f}))}\right),$$

$$\mathrm{lse}(\mathbf{f}) = \log\left(1 + \sum_{k'=1}^{K} \exp(f_{k'})\right), \tag{2}$$

for $k = 0, \dots, K$ and with $f_0 := 0$.

Following our medical diagnosis example from the introduction, each patient is modelled by latent $\mathbf{x}_n$; for each examination d, $\mathbf{x}_n$ has a sequence of weights $(f_{nd1}, ..., f_{ndK})$, one weight for each possible test result k, that follows a Gaussian process; Softmax returns test result y_{nd} based on these weights, resulting in a patient's medical assessment $\mathbf{y}_n = (y_{n1}, ..., y_{nD})$.

Define $\mathbf{f}_{dk} = (f_{1dk}, ..., f_{Ndk})$ and define $\mathbf{u}_{dk} = (u_{1dk}, ..., u_{Mdk})$. The joint distribution of $(\mathbf{f}_{dk}, \mathbf{u}_{dk})$ with the latent nonlinear function, $\mathcal{F}_{dk}$, marginalized under the GP assumption, is a multi-variate Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{K}_d([\mathbf{X}, \mathbf{Z}], [\mathbf{X}, \mathbf{Z}]))$. It is easy to verify that when we further marginalize the inducing outputs, we end up with a joint distribution of the form $\mathbf{f}_{dk} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_d(\mathbf{X}, \mathbf{X})), \forall d, k$. Therefore, the introduction of inducing outputs does not change the marginal likelihood of the data $\mathbf{Y}$. These are used in the variational inference method in the next section and the inducing inputs $\mathbf{Z}$ are considered as variational parameters.

We use the automatic relevance determination (ARD) radial basis function (RBF) covariance function for our model. ARD RBF is able to select the dimensionality of the latent space automatically and transform it non-linearly.

4 Inference

The marginal log-likelihood, also known as log-evidence, is intractable for our model due to the non-linearity of the covariance function of the GP and the Softmax likelihood function. We first describe a lower bound of the log-evidence (ELBO) by applying Jensen's inequality with a variational distribution of the latent variables following Titsias & Lawrence (2010).

Consider a variational approximation to the posterior distribution of $\mathbf{X}$, $\mathbf{F}$ and $\mathbf{U}$ factorized as follows:

$$q(\mathbf{X}, \mathbf{F}, \mathbf{U}) = q(\mathbf{X}) q(\mathbf{U}) p(\mathbf{F} | \mathbf{X}, \mathbf{U}). \tag{3}$$

We can obtain the ELBO by applying Jensen's inequality

$$\log p(\mathbf{Y}) = \log \int p(\mathbf{X}) p(\mathbf{U}) p(\mathbf{F} | \mathbf{X}, \mathbf{U}) p(\mathbf{Y} | \mathbf{F}) \mathrm{d}\mathbf{X} \mathrm{d}\mathbf{F} \mathrm{d}\mathbf{U}$$

$$
\begin{aligned}
&\geq \int q(\mathbf{X})q(\mathbf{U})p(\mathbf{F}|\mathbf{X},\mathbf{U}) \\
&\quad \cdot \log \frac{p(\mathbf{X})p(\mathbf{U})p(\mathbf{F}|\mathbf{X},\mathbf{U})p(\mathbf{Y}|\mathbf{F})}{q(\mathbf{X})q(\mathbf{U})p(\mathbf{F}|\mathbf{X},\mathbf{U})} \mathrm{d}\mathbf{X}\mathrm{d}\mathbf{F}\mathrm{d}\mathbf{U} \\
&= -\mathrm{KL}(q(\mathbf{X})\|p(\mathbf{X})) - \mathrm{KL}(q(\mathbf{U})\|p(\mathbf{U})) \\
&\quad + \sum_{n=1}^{N}\sum_{d=1}^{D} \int q(\mathbf{x}_n)q(\mathbf{U}_d)p(\mathbf{f}_{nd}|\mathbf{x}_n,\mathbf{U}_d) \\
&\qquad \cdot \log p(\mathbf{y}_{nd}|\mathbf{f}_{nd}) \mathrm{d}\mathbf{x}_n \mathrm{d}\mathbf{f}_{nd} \mathrm{d}\mathbf{U}_d \\
&:= \mathcal{L} \qquad (4)
\end{aligned}
$$

where

$$
p(\mathbf{f}_{nd}|\mathbf{x}_n,\mathbf{U}_d) = \prod_{k=1}^{K} \mathcal{N}(f_{ndk}|\mathbf{a}_{nd}^T\mathbf{u}_{dk}, b_{nd}) \qquad (5)
$$

with

$$
\begin{aligned}
\mathbf{a}_{nd} &= \mathbf{K}_{d,MM}^{-1}\mathbf{K}_{d,Mn}, \\
b_{nd} &= K_{d,nn} - \mathbf{K}_{d,nM}\mathbf{K}_{d,MM}^{-1}\mathbf{K}_{d,Mn}. \qquad (6)
\end{aligned}
$$

Notice however that the integration of $\log p(\mathbf{y}_{nd}|\mathbf{f}_{nd})$ in eq. 4 involves a nonlinear function ($\mathrm{lse}(\mathbf{f})$ from eq. 2) and is still intractable. Consequently, we do not have an analytical form for the optimal variational distribution of $q(\mathbf{U})$ unlike in Titsias & Lawrence (2010). Instead of applying a further approximation/lower bound on $\mathcal{L}$, we want to obtain better accuracy by following a sampling-based approach (Blei et al., 2012; Kingma & Welling, 2013; Rezende et al., 2014; Titsias & Lázaro-Gredilla, 2014) to compute the lower bound $\mathcal{L}$ and its derivatives with the Monte Carlo method.

Specifically, we draw samples of $\mathbf{x}_n$, $\mathbf{U}_d$ and $\mathbf{f}_{nd}$ from $q(\mathbf{x}_n)$, $q(\mathbf{U}_d)$, and $p(\mathbf{f}_{nd}|\mathbf{x}_n,\mathbf{U}_d)$ respectively and estimate $\mathcal{L}$ with the sample average. Another advantage of using the Monte Carlo method is that we are not constrained to a limited choice of covariance functions for the GP that is otherwise required for an analytical solution in standard approaches to GPLVM for continuous data (Titsias & Lawrence, 2010; Hensman et al., 2013).

We consider a mean field approximation for the latent points $q(\mathbf{X})$ as in (Titsias & Lawrence, 2010) and a joint Gaussian distribution with the following factorisation for $q(\mathbf{U})$:

$$
\begin{aligned}
q(\mathbf{U}) &= \prod_{d=1}^{D}\prod_{k=1}^{K} \mathcal{N}(\mathbf{u}_{dk}|\boldsymbol{\mu}_{dk}, \boldsymbol{\Sigma}_d), \\
q(\mathbf{X}) &= \prod_{n=1}^{N}\prod_{i=1}^{Q} \mathcal{N}(x_{ni}|m_{ni}, s_{ni}^2) \qquad (7)
\end{aligned}
$$

where the covariance matrix $\boldsymbol{\Sigma}_d$ is shared for the same categorical variable d (remember that K is the number of values this categorical variable can take). The KL divergence in $\mathcal{L}$ can be computed analytically with the given variational distributions. The parameters we need to optimise over[2] include the hyper-parameters for the GP $\boldsymbol{\theta}_d$, variational parameters for the inducing points $\mathbf{Z}$, $\boldsymbol{\mu}_{dk}$, $\boldsymbol{\Sigma}_d$, and the mean and standard deviation of the latent points m_{ni}, s_{ni}.

[2]Note that the number of parameters to optimise over can be reduced by transforming the latent space non-linearly to a second low-dimensional latent space, which is then transformed linearly to the weight space containing points f_{ndk}.

4.1 Transforming the Random Variables

In order to obtain a Monte Carlo estimate to the gradients of $\mathcal{L}$ with low variance, a useful trick introduced in (Kingma & Welling, 2013) is to transform the random variables to be sampled so that the randomness does not depend on the parameters with which the gradients will be computed. We present the transformation of each variable to be sampled as follows:

Transforming $\mathbf{X}$. For the mean field approximation, the transformation for $\mathbf{X}$ is straightforward as

$$
x_{ni} = m_{ni} + s_{ni}\varepsilon_{ni}^{(x)}, \quad \varepsilon_{ni}^{(x)} \sim \mathcal{N}(0,1) \qquad (8)
$$

Transforming $\mathbf{u}_{dk}$. The variational distribution of $\mathbf{u}_{dk}$ is a joint Gaussian distribution. Denote the Cholesky decomposition of $\boldsymbol{\Sigma}_d$ as $\mathbf{L}_d\mathbf{L}_d^T = \boldsymbol{\Sigma}_d$. We can rewrite $\mathbf{u}_{dk}$ as

$$
\mathbf{u}_{dk} = \boldsymbol{\mu}_{dk} + \mathbf{L}_d\boldsymbol{\varepsilon}_{dk}^{(u)}, \quad \boldsymbol{\varepsilon}_{dk}^{(u)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_M) \qquad (9)
$$

We optimize the lower triangular matrix $\mathbf{L}_d$ instead of $\boldsymbol{\Sigma}_d$.

Transforming $\mathbf{f}_{nd}$. Since the conditional distribution $p(\mathbf{f}_{nd}|\mathbf{x}_n,\mathbf{U}_d)$ in Eq. 5 is factorized over k we can define a new random variable for every f_{ndk}:

$$
f_{ndk} = \mathbf{a}_{nd}^T\mathbf{u}_{dk} + \sqrt{b_{nd}}\varepsilon_{ndk}^{(f)}, \quad \varepsilon_{ndk}^{(f)} \sim \mathcal{N}(0,1) \qquad (10)
$$

Notice that the transformation of the variables does not change the distribution of the original variables and therefore does not change the KL divergence in Eq. 5.

4.2 Lower Bound with Transformed Variables

Given the transformation we just defined, we can represent the lower bound as

$$
\begin{aligned}
\mathcal{L} = &-\sum_{n=1}^{N}\sum_{i=1}^{Q} \mathrm{KL}(q(x_{ni})\|p(x_{ni})) \qquad (11) \\
&- \sum_{d=1}^{D}\sum_{k=1}^{K} \mathrm{KL}(q(\mathbf{u}_{dk})\|p(\mathbf{u}_{dk})) \\
&+ \sum_{n=1}^{N}\sum_{d=1}^{D} \mathbb{E}_{\boldsymbol{\varepsilon}_n^{(x)},\boldsymbol{\varepsilon}_d^{(u)},\boldsymbol{\varepsilon}_{nd}^{(f)}} \Big[\\
&\log \mathrm{Softmax}\left(\mathbf{y}_{nd}\middle|\mathbf{f}_{nd}\left(\boldsymbol{\varepsilon}_{nd}^{(f)}, \mathbf{U}_d(\boldsymbol{\varepsilon}_d^{(u)}), \mathbf{x}_n(\boldsymbol{\varepsilon}_n^{(x)})\right)\right) \Big]
\end{aligned}
$$

where the expectation in the last line is with respect to the fixed distribution defined in Eqs. 8, 9 and 10. Each expectation term that involves the Softmax likelihood, denoted as $\mathcal{L}_s^{nd}$, can be estimated using Monte Carlo integration as

$$
\mathcal{L}_s^{nd} \approx \qquad (12)
$$

$$\frac{1}{T}\sum_{i=1}^{T}\log \text{Softmax}\left(\mathbf{y}_{nd}\middle|\mathbf{f}_{nd}\left(\boldsymbol{\varepsilon}^{(f)}_{nd,i}, \mathbf{U}_d(\boldsymbol{\varepsilon}^{(u)}_{d,i}), \mathbf{x}_n(\boldsymbol{\varepsilon}^{(x)}_{n,i})\right)\right)$$

with $\boldsymbol{\varepsilon}^{(x)}_{n,i}, \boldsymbol{\varepsilon}^{(u)}_{d,i}, \boldsymbol{\varepsilon}^{(f)}_{nd,i}$ drawn from their corresponding distributions. Since these distributions do not depend on the parameters to be optimized, the derivatives of the objective function $\mathcal{L}$ are now straight-forward to compute with the same set of samples using the chain rule.

4.3 Stochastic Gradient Descent

We use gradient descent to find an optimal variational distribution. Gradient descent with noisy gradients is guaranteed to converge to a local optimum given decreasing learning rate with some conditions, but is hard to work with in practice. Initial values set for the learning rate influence the rate at which the algorithm converges, and misspecified values can cause it to diverge. For this reason new techniques have been proposed that handle noisy gradients well. Optimisers such as AdaGrad (Duchi et al., 2011), AdaDelta (Zeiler, 2012), and RMSPROP (Tieleman & Hinton, 2012) have been proposed, each handling the gradients slightly differently, all averaging over past gradients. Schaul et al. (2013) have studied empirically the different techniques, comparing them to one another on a variety of unit tests. They found that RMSPROP works better on many test sets compared to other optimisers evaluated. We thus chose to use RMSPROP for our experiments.

A major advantage of our inference is that it is extremely easy to implement and adapt. The straight-forward computation of derivatives through the expectation makes it possible to use symbolic differentiation. We use Theano (Bergstra et al., 2010) for the inference implementation, where the generative model is implemented as in Eqs. 8, 9 and 10, and the optimisation objective, evaluated on samples from the generative model, is given by Eq. 11.

5 Experimental Results

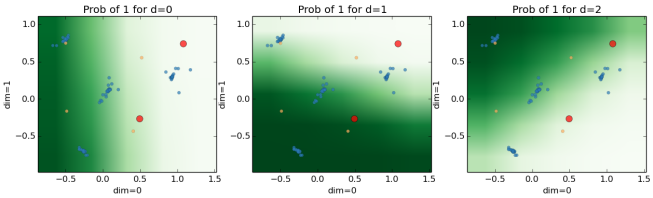
We evaluate the categorical latent Gaussian process (CLGP), comparing it to existing models for multivariate categorical distribution estimation. These include models based on a discrete latent representation (such as the Dirichlet-Multinomial), and continuous latent representation with a linear transformation of the latent space (latent Gaussian model (LGM, Khan et al., 2012)). We demonstrate over-fitting problems with the LGM, and inspect the robustness of our inference.

For the following experiments, both the linear and non-linear models were initialised with a 2D latent space. The mean values of the latent points, m_n, were initialised at random following a standard normal distribution. The mean values of the inducing outputs ($\boldsymbol{\mu}_{dk}$) were initialised with a normal distribution with standard deviation 10^{-2}. This is equivalent to using a uniform initial distribution for all values. We initialise the standard deviation of each latent point (s_n) to 0.1, and initialise the length-scales of the ARD RBF covariance function to 0.1. We then optimise the variational distribution for 500 iterations. At every iteration we optimise the various quantities while holding $\mathbf{u}_{dk}$'s variational parameters fixed, and then optimise $\mathbf{u}_{dk}$'s variational parameters holding the other quantities fixed.

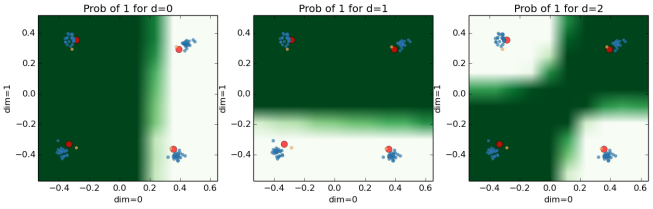
Our setting supports semi-supervised learning with partially observed data. The latents of partially observed points are optimised with the training set, and then used to predict the missing values. We assess the performance of the models using test-set perplexity (a measure of how much the model is surprised by the missing test values). This is defined as the exponent of the negative average log predicted probability.

5.1 Linear Models Have Difficulty with Multi-modal Distributions

We show the advantages of using a non-linear categorical distribution estimation compared to a linear one, evaluating the CLGP against the linear LGM. We implement the latent Gaussian model using a linear covariance function in our model; we remove the KL divergence term in $\mathbf{u}$ following the model specification in (Khan et al., 2012), and use our inference scheme described above. Empirically, the Monte Carlo inference scheme with the linear model results in the same test error on (Inoguchi, 2008) as the piece-wise bound based inference scheme developed in (Khan et al., 2012).

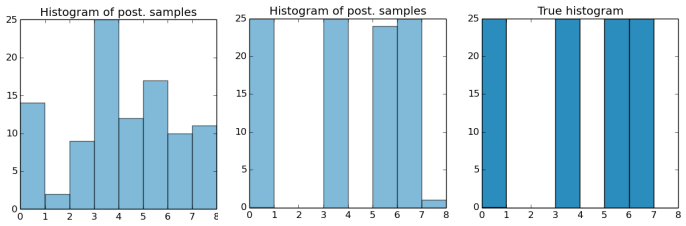


(a) LGM



(b) CLGP

Figure 2. **Density over the *latent* space for XOR as predicted by the linear model (top, LGM), and non-linear model (bottom, CLGP).** Each figure shows the probability $p(\mathbf{y}_d = 1|\mathbf{x})$ as a function of $\mathbf{x}$, for $d = 0, 1, 2$ (first, second, and third digits in the triplet left to right). The darker the shade of green, the higher the probability. In blue are the latents corresponding to the training points, in yellow are the latents corresponding to the four partially observed test points, and in red are the inducing points used to support the function.



(a) LGM (b) CLGP (c) Ground truth

Figure 3. **Histogram of categorical values for XOR** (encoded in binary for the 8 possible values) for samples drawn from the posterior of the latent space of the linear model (left, LGM), the non-linear model (middle, CLGP), and the data used for training (right).

5.1.1 Relation Embedding – Exclusive Or

A simple example of relational learning (following Paccanaro & Hinton (2001)) can be used to demonstrate when linear latent space models fail. In this task we are given a dataset with example relations and the model is to capture the distribution that generated them. A non-linear dataset is constructed using the XOR (exclusive or) relation. We collect 25 positive examples of each assignment of the binary relation (triplets of the form $(0,0,0)$, $(0,1,1)$, $(1,0,1)$, $(1,1,0)$, corresponding to 0 XOR $1 = 1$ and so on). We then maximise the variational lower bound using RMSPROP for both the linear and non-linear models with 20 samples for the Monte Carlo integration. We add four more triplets to the dataset: $(0,0,?)$, $(0,1,?)$, $(1,0,?)$, $(1,1,?)$. We evaluate the probabilities the models assign to each of the missing values (also known as *imputation*) and report the results.

We assessed the test-set perplexity repeating the experiment 3 times and averaging the results. The linear model obtained a test-set perplexity (with standard deviation) of 75.785 ± 13.221, whereas the non-linear model obtained a test-set perplexity of 1.027 ± 0.016. A perplexity of 1 corresponds to always predicting correctly.

During optimisation the linear model jumps between different local modes, trying to capture all four possible triplets (fig. 2). The linear model assigns probabilities to the missing values by capturing some of the triplets well, but cannot assign high probability to the others. In contrast, the CLGP model is able to capture all possible values of the relation. Sampling from probability vectors from the latent variational posterior for both models, we get a histogram of the posterior distribution (fig. 3). As can be seen, the CLGP model is able to fully capture the distribution whereas the linear model is incapable of it.

5.1.2 Relation Learning – Complex Example

We repeat the previous experiment with a more complex relation. We generated 1000 strings from a simple probabilistic context free grammar with non-terminals $\{\alpha, \beta, A, B, C\}$, start symbol α, terminals $\{a, b, c, d, e, f, g\}$, and derivation rules:

$$\alpha \to A\beta\ [1.0]$$
$$\beta \to BA\ [0.5] \mid C\ [0.5]$$
$$A \to a\ [0.5] \mid b\ [0.3] \mid c\ [0.2]$$
$$B \to d\ [0.7] \mid e\ [0.3]$$
$$C \to f\ [0.7] \mid g\ [0.3]$$

where we give the probability of following a derivation rule inside square brackets. Following the start symbol and the derivation rules we generate strings of the form ada, af, ceb, and so on. We add two "start" and "end" symbols $s, \overline{s}$ to each string obtaining strings of the form $ssceb\overline{ss}$. We then extract triplets of consecutive letters from these strings resulting in training examples of the form ssc, sce, ..., $b\overline{ss}$. This is common practice in text preprocessing in the natural language community. It allows us to generate relations from scratch by conditioning on the prefix ss to generate the first non-terminal (say a), then condition on the prefix sa, and continue in this vein iteratively. Note that a simple histogram based approach will do well on this dataset (and the previous dataset) as it is not sparse.

Split	LGM	**CLGP**
1	7.898 ± 3.220	2.769 ± 0.070
2	26.477 ± 23.457	2.958 ± 0.265
3	6.748 ± 3.028	2.797 ± 0.081

Table 1. **Test-set perplexity on relational data**. Compared are the linear LGM and the non-linear CLGP.

We compared the test-set perplexity of the non-linear CLGP (with 50 inducing points) to that of the linear LGM on these training inputs, repeating the same experiment set-up as before. We impute one missing value at random in each test string, using 20% of the strings as a test set with 3 different splits. The results are presented in table 1. The linear model cannot capture this data well, and seems to get very confident about misclassified values (resulting in very high test-set perplexity).

5.2 Sparse Small Data

We assess the model in the real-world domain of small sparse data. We compare the CLGP model to a histogram based approach, demonstrating the difficulty with frequency models for sparse data. We further compare our model to the linear LGM.

5.2.1 Medical Diagnosis

We use the Wisconsin breast cancer dataset for which obtaining samples is a long and expensive process[3]. The dataset is composed of 683 data points, with 9 categorical variables taking values between 1 and 10, and an additional categorical variable taking 0,1 values – 2×10^9 possi-

[3] Obtained from the UCI repository

Split	Baseline	Multinomial	Uni-Dir-Mult	Bi-Dir-Mult	LGM	**CLGP**
1	8.68	4.41	4.41	3.41	3.57 ± 0.208	$\mathbf{2.86 \pm 0.119}$
2	8.68	∞	4.42	**3.49**	$\mathbf{3.47 \pm 0.252}$	$\mathbf{3.36 \pm 0.186}$
3	8.85	4.64	4.64	3.67	12.13 ± 9.705	$\mathbf{3.34 \pm 0.096}$

Table 2. **Test-set perplexity on Breast cancer dataset, predicting randomly missing *categorical test results***. The models compared are: *Baseline* predicting uniform probability for all values, *Multinomial* – predicting the probability for a missing value based on its frequency, *Uni-Dir-Mult* – Unigram Dirichlet Multinomial with concentration parameter $\alpha = 0.01$, *Bi-Dir-Mult* – Bigram Dirichlet Multinomial with concentration parameter $\alpha = 1$, *LGM*, and the proposed model (*CLGP*).

ble configurations. We use three quarters of the dataset for training and leave the rest for testing, averaging the test-set perplexity on three repetitions of the experiment. We use three different random splits of the dataset as the distribution of the data can be fairly different among different splits. In the test set we randomly remove one of the 10 categorical values, and test the models' ability to recover that value. Note that this is a harder task than the usual use of this dataset for binary classification. We use the same model set-up as in the first experiment.

We compare our model (*CLGP*) to a baseline predicting uniform probability for all values (*Baseline*), a multinomial model predicting the probability for a missing value based on its frequency (*Multinomial*), a unigram Dirichlet Multinomial model with concentration parameter $\alpha = 0.01$ (*Uni-DirMult*), a bigram Dirichlet Multinomial with concentration parameter $\alpha = 1$ (*Bi-Dir-Mult*), and the linear continuous space model (*LGM*). More complicated frequency based approaches are possible, performing variable selection and then looking at frequencies of triplets of variables. These will be very difficult to work with in this sparse small data problem.

We evaluate the models' performance using the test-set perplexity metric as before (table 2). As can be seen, the frequency based approaches obtain worse results than the continuous latent space approaches. The frequency model with no smoothing obtains perplexity ∞ for split 2 because one of the test points has a value not observed before in the training set. Using smoothing solves this. The baseline (predicting uniformly) obtains the highest perplexity on average. The linear model exhibits high variance for the last split, and in general has higher perplexity standard deviation than the non-linear model.

5.2.2 Terror Warning Effects on Political Attitude

We further compare the models on raw data collected in an experimental study of the effects of government terror warnings on political attitudes (from the START Terrorism Data Archive Dataverse[4]). This dataset consists of 1282 cases and 49 variables. We used 1000 cases for training and 282 for test. From the 49 variables, we used the 17 corresponding to categorical answers to study questions (the other variables are subjects' geographic info and an answer to an open question). The 17 variables take 5 or 6 possible values for each variable with some of the values missing.

Baseline	Multinomial	LGM	**CLGP**
2.50	2.20	3.07	**2.11**

Table 3. **Test-set perplexity for terror warning effects on political attitude**. Compared are the linear LGM and the non-linear CLGP.

We repeat the same experiment set-up as before, with a 6 dimensional latent space, 100 inducing points, 5 samples to estimate the lower bound, and running the optimiser for 1000 iterations. We compare a *Baseline* using a uniform distribution for all values, a *Multinomial* model using the frequencies of the individual variables, the linear *LGM*, and the *CLGP*. The results are given in table 3. On the training set, the CLGP obtained a training error of 1.56, and the LGM obtained a training error of 1.34. The linear model seems to over-fit to the data.

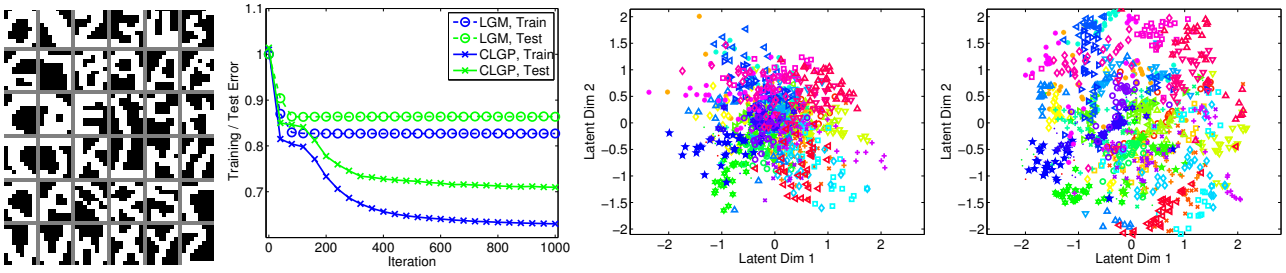
5.2.3 Handwritten Binary Alphadigits

We evaluate the performance of our model on a sparse dataset with a large number of dimensions. The dataset, Binary Alphadigits, is composed of 20×16 binary images of 10 handwritten digits and 26 handwritten capital letters, each class with 39 images[5]. We resize each image to 10×8, and obtain a dataset of 1404 data points each with 80 binary variables. We repeat the same experiment set-up as before with 2 latent dimensions for ease of visual comparison of the latent embeddings. Fig. 4a shows an example of each alphadigit class. Each class is then randomly split to 30 training and 9 test examples. In the test set, we randomly remove 20% of the pixels and evaluate the prediction error.

Fig. 4b shows the training and test error in log perplexity for both models. LGM converges faster than CLGP but ends up with a much higher prediction error due to its limited modeling capacity with 2 dimensional latent variables. This is validated with the latent embedding shown in fig. 4c and 4d. Each color-marker combination denotes one alphadigit class. As can be seen, CLGP has a better separation of the different classes than LGM even though the class labels are not used in the training.

[4]Obtained from the Harvard Dataverse Network thedata.harvard.edu/dvn/dv/start/faces/study/StudyPage.xhtml?studyId=71190

[5]Obtained from http://www.cs.nyu.edu/~roweis/data.html



(a) Example alphadigits (b) Train & Test log perplexity (c) LGM latent space (d) CLGP latent space

Figure 4. **Binary alphadigit dataset.** (a) example of each class (b) Train and test log perplexity for 1000 iterations. (c,d) 2-D latend embedding of the 36 alphadigit classes with LGM (left) and CLGP model (right). Each color-marker combination denotes one class.

5.3 Latent Gaussian Model Over-fitting

It is interesting to note that the latent Gaussian model (LGM) over-fits on the different datasets. It is possible to contribute this to the lack of regularisation over the linear transformation – the weight matrix used to transform the latent space to the Softmax weights is optimised without a prior. In all medical diagnosis experiment repetitions, for example, the model's training log perplexity decreases while the test log perplexity starts increasing (see fig. 5 for the train and test log perplexities of split 1 of the breast cancer dataset). Note that even though the test log perplexity starts increasing, at its lowest point it is still higher than the end test log perplexity of the CLGP model. This is observed for all splits and all repetitions.

It is worth noting that we compared the CLGP to the LGM on the ASES survey dataset (Inoguchi, 2008) used to assess the LGM in (Khan et al., 2012). We obtained a test perplexity of 1.98, compared to LGM's test perplexity of 1.97. We concluded that the dataset was fairly linearly separable.

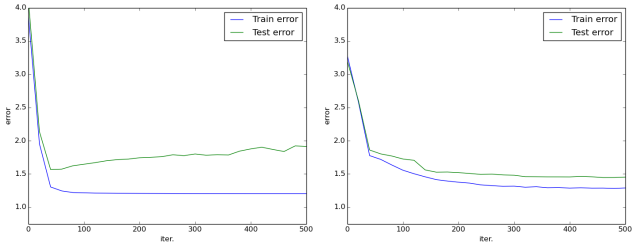
5.4 Inference Robustness

Lastly, we inspect the robustness of our inference, evaluating the Monte Carlo (MC) estimate standard deviation as optimisation progresses. Fig. 6 shows the ELBO and MC standard deviation per iteration (on log scale) for the Alphadigit dataset. It seems that the estimator standard deviation decreases as the approximating variational distribution fits to the posterior. This makes the stochastic optimisation easier, speeding up convergence. A further theoretical study of this behaviour in future research will be of interest.

6 Discussion and Conclusions

We have presented the first Bayesian model capable of capturing sparse multi-modal categorical distributions based on a continuous representation. The model time complexity is $\mathcal{O}(NM^2DK)$ for N data points, M inducing points, D categorical variables, and K values for each variable. When used with a single variable, this is the same as that of the variational inference in sparse GPs (Titsias, 2009). This model ties together many existing models in the field, linking the linear and discrete *latent Gaussian models* to the non-linear continuous space *Gaussian process latent variable model* and to the fully observed discrete *Gaussian process classification*.

In future work we aim to answer short-comings in the current model such as scalability to high dimensionality and robustness. We scale the model following research on GP scalability done in (Hensman et al., 2013; Gal et al., 2014). The robustness of the model depends on the sample variance in the Monte Carlo integration. As discussed in Blei et al. (2012), variance reduction techniques can help in the estimation of the integral, and methods such as the one developed in Wang et al. (2013) can effectively increase inference robustness.



(a) LGM (b) CLGP

Figure 5. **Train and test log perplexities for LGM (left) and the CLGP model (right) for one of the splits of the breast cancer dataset.** The train log perplexity of LGM decreases while the test log perplexity starts increasing at iteration 50.

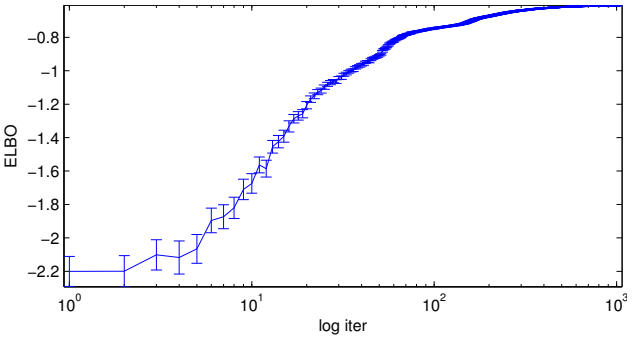


Figure 6. **ELBO and standard deviation per iteration** (on log scale) for the Alphadigits dataset.

Acknowledgements

YG is supported by the Google European fellowship in Machine Learning.

References

Bengio, Yoshua, Schwenk, Holger, Senécal, Jean-Sébastien, Morin, Fréderic, and Gauvain, Jean-Luc. Neural probabilistic language models. In *Innovations in Machine Learning*, pp. 137–186. Springer, 2006.

Bergstra, James, Breuleux, Olivier, Bastien, Frédéric, Lamblin, Pascal, Pascanu, Razvan, Desjardins, Guillaume, Turian, Joseph, Warde-Farley, David, and Bengio, Yoshua. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.

Blei, David and Lafferty, John. Correlated topic models. *Advances in neural information processing systems*, 18: 147, 2006.

Blei, David M, Jordan, Michael I, and Paisley, John W. Variational Bayesian inference with stochastic search. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pp. 1367–1374, 2012.

Böhning, Dankmar. Multinomial logistic regression algorithm. *Annals of the Institute of Statistical Mathematics*, 44(1):197–200, 1992.

Collobert, Ronan and Weston, Jason. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pp. 160–167. ACM, 2008.

Duchi, John, Hazan, Elad, and Singer, Yoram. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

Gal, Yarin, van der Wilk, Mark, and Rasmussen, Carl E. Distributed variational inference in sparse Gaussian process regression and latent variable models. In *Advances in Neural Information Processing Systems 27*. 2014.

Hensman, James, Fusi, Nicolo, and Lawrence, Neil D. Gaussian processes for big data. In Nicholson, Ann and Smyth, Padhraic (eds.), *UAI*. AUAI Press, 2013.

Inoguchi, Takashi. Asia Europe survey (ASES): A multinational comparative study in 18 countries, 2001. In *Inter-university Consortium for Political and Social Research (ICPSR)*, 2008.

Jaakkola, Tommi S. and Jordan, Michael I. A variational approach to Bayesian logistic regression models and their extensions. In *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*, 1997.

Khan, Mohammad E, Mohamed, Shakir, Marlin, Benjamin M, and Murphy, Kevin P. A stick-breaking likelihood for categorical data analysis with latent Gaussian models. In *International conference on Artificial Intelligence and Statistics*, pp. 610–618, 2012.

Kingma, Diederik P and Welling, Max. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Knowles, David A and Minka, Tom. Non-conjugate variational message passing for multinomial and binary regression. In *Advances in Neural Information Processing Systems*, pp. 1701–1709, 2011.

Lawrence, Neil. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *The Journal of Machine Learning Research*, 6:1783–1816, 2005.

Marlin, Benjamin M, Khan, Mohammad Emtiyaz, and Murphy, Kevin P. Piecewise bounds for estimating bernoulli-logistic latent Gaussian models. In *ICML*, pp. 633–640, 2011.

Paccanaro, Alberto and Hinton, Geoffrey E. Learning distributed representations of concepts using linear relational embedding. *Knowledge and Data Engineering, IEEE Transactions on*, 13(2):232–244, 2001.

Quiñonero-Candela, Joaquin and Rasmussen, Carl Edward. A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.

Rezende, Danilo J, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1278–1286, 2014.

Schaul, Tom, Antonoglou, Ioannis, and Silver, David. Unit tests for stochastic optimization. abs/1312.6055, 2013. URL http://arxiv.org/abs/1312.6055.

Tieleman, T. and Hinton, G. Lecture 6.5 - rmsprop, COURSERA: Neural networks for machine learning, 2012.

Tipping, Michael E and Bishop, Christopher M. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

Titsias, Michalis and Lawrence, Neil. Bayesian Gaussian process latent variable model. In *Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

Titsias, Michalis and Lázaro-Gredilla, Miguel. Doubly stochastic variational Bayes for non-conjugate inference. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1971–1979, 2014.

Titsias, Michalis K. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics 12*, pp. 567–574, 2009.

Wang, Chong, Chen, Xi, Smola, Alex, and Xing, Eric. Variance reduction for stochastic gradient optimization. In *Advances in Neural Information Processing Systems*, pp. 181–189, 2013.

Williams, Christopher KI and Rasmussen, Carl Edward. Gaussian processes for machine learning. *the MIT Press*, 2(3):4, 2006.

Zeiler, Matthew D. Adadelta: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.