Latent Gaussian Processes for Distribution Estimation of Multivariate Categorical Data

Yarin Gal **Yutian Chen** **Zoubin Ghahramani**

University of Cambridge
{ yg279, yc373, zoubin }@cam.ac.uk

Abstract

Multivariate categorical data occur in many applications of machine learning, such as data analysis and language processing. Here we develop a flexible class of models for distribution estimation in such multivariate (i.e. vectors of) categorical data. Multivariate categorical data is challenging because the number of possible discrete observation vectors grows exponentially with the number of categorical variables in the vector. In particular, we address the problem of estimating the distribution when the data is sparsely sampled—i.e. in the typical case when the diversity of the data points is poor compared to the exponentially many possible observations. We make use of a continuous latent Gaussian space, but unlike previous linear approaches, we learn a non-linear transformation between this latent space and the multivariate categorical observation space. Non-linearity is essential for capturing multi-modality in the distribution. Our model ties together many existing models, linking the categorical linear latent Gaussian model, the Gaussian process latent variable model, and Gaussian process classification. We derive effective inference for our model based on recent developments in sampling-based variational inference and stochastic optimisation.

1 Introduction

Categorical distribution estimation (CDE) forms one of the core problems in machine learning, and can be used to perform tasks ranging from survey analysis (Inoguchi, 2008) to cancer prediction (Zwitter and Soklic, 1988). One of the major challenges in CDE is sparsity. Sparsity can occur either when there is a single categorical variable with many possible values, some appearing scarcely, or when the data consists of vectors of categorical variables, with most configurations of categorical values not in the dataset. We focus on Bayesian approaches to the latter, multivariate, case. Existing approaches to Bayesian-CDE concentrate on either discrete or continuous latent representations (Agresti and Hitchcock, 2005). Discrete representations are based on frequencies of observations but cannot handle sparse samples well. Existing continuous representations linearly transform a latent space before discretisation, but cannot capture multi-modality in the data.

We would like to capture sparse multi-modal categorical distributions. A possible approach is to model the continuous representation with a non-linear transformation. 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 (instead of using a linear transformation). However, the Softmax likelihood is not conjugate to the Gaussian prior. A similar problem exists developing *linear* Gaussian models (LGMs) in a variational setting. Marlin et al. (2011) used various approximations 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 and Jordan, 1997), the tilted bound (Knowles and Minka, 2011), piecewise linear and quadratic bounds (Marlin et al., 2011), and others. For categorical data

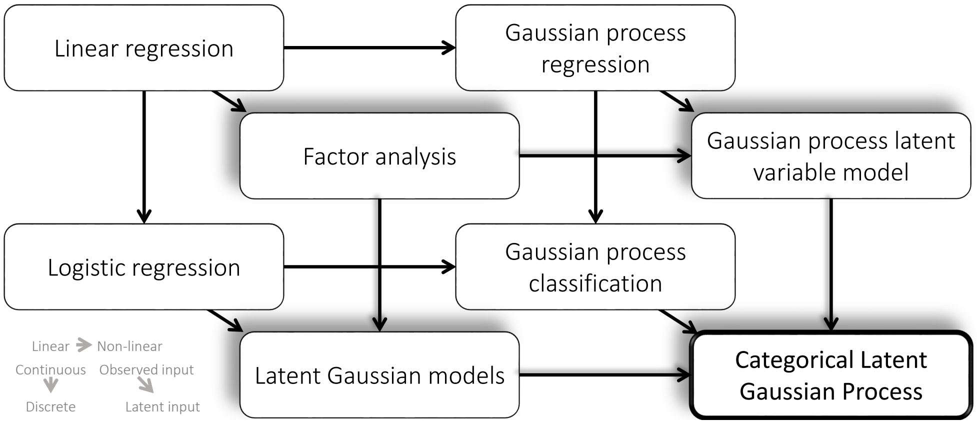

Figure 1: Relations between existing models and the model proposed in this paper (*Categorical Latent Gaussian Process*); the model can be seen as a non-linear version of the *latent Gaussian model* (left to right, Khan et al. (2012)), it can be seen as a latent counterpart to the *Gaussian process classification* model (back to front, Williams and Rasmussen (2006)), or alternatively as a discrete extension of the *Gaussian process latent variable model* (top to bottom, Lawrence (2005)).

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 and Lafferty, 2006) give poor approximation (Khan et al., 2012).

In this paper we propose to use recent developments in stochastic variational inference (SVI, Hoffman et al., 2013) to avoid computing the Softmax bound explicitly, while using 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 (Quiñonero-Candela and Rasmussen, 2005; Titsias, 2009). Our approach takes advantage of these tools to obtain simple yet powerful model and inference. We use Monte Carlo integration to approximate the non-conjugate likelihood obtaining noisy gradients (Blei et al., 2012; Kingma and Welling, 2013; Rezende et al., 2014; Titsias and Lázaro-Gredilla, 2014). We then use learning-rate free stochastic optimisation (Tieleman and Hinton, 2012) to optimise the noisy objective. We leverage symbolic differentiation (Theano, Bergstra et al., 2010) to obtain simple and modular code[1]. We develop the inference for the linear case (using a Gaussian process with a linear covariance function) resulting in trivial implementation of the LGM with performance identical to (Khan et al., 2012). We then extend the model to non-linear covariance functions that are able to transform the latent space non-linearly. We name this model *Categorical Latent Gaussian Process* (CLGP).

We experimentally show the advantages of using non-linear transformations for the latent space. We follow the ideas brought in Paccanaro and Hinton (2001) and study the models with the task of *relational learning*. We use the simple (and forgotten) XOR dataset for this, capturing the non-linear XOR relation based on observations of triplets such as $(1, 1, 0)$. We further evaluate our model and inference in the semi-supervised setting, training the model with partially observed relations for imputation. We then demonstrate the utility of the model in the real-world *small data* domain when data is scarce, comparing our model to discrete frequency based models. For this we use the ubiquitous Wisconsin breast cancer dataset, where the number of observations is small and costly to obtain. However we replace the simple supervised classification task of predicting the development of breast cancer in patients. Instead we use the estimated distribution for the much more difficult task of *deciding which tests are needed or can be deduced from the others*. Lastly, we evaluate the robustness of our inference, inspecting the Monte Carlo estimate variance over time for different number of samples. These experiments are given in the appendix.

2 Related Work

Our model (CLGP) relates to some key probabilistic models in the field (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

[1] The entire code, consisting of 80 lines of Python, is available at `github.com/yaringal/CLGP`

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. question) in a list of categorical variables (e.g. survey). 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 using numerical techniques. Our approach avoids this cumbersome inference.

Our proposed model can also be seen as a latent counterpart to the *Gaussian process classification* model (Williams and 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. Lastly, our model can 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 and Bishop, 1999)) and density estimation in continuous space.

3 A Latent Gaussian Process Model for Multivariate Categorical Data

We consider a generative model for a dataset $\mathbf{Y}$ with N observations (people taking part in a survey for example) and D categorical variables (different questions in the survey). 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 $(0, f_{nd1}, \dots, f_{ndK})$. 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 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 $\mathcal{O}(M^2N)$ 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 and Rasmussen, 2005)).

The model is expressed as:

$$\begin{aligned}
x_{ni} &\overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_x^2) & n &= 1, \dots, N, i = 1, \dots, Q \\
\mathcal{F}_{dk} &\overset{\text{iid}}{\sim} \text{GP}(\mathbf{K}_d) & d &= 1, \dots, D, k = 1, \dots, K \\
f_{ndk} &= \mathcal{F}_{dk}(\mathbf{x}_n), \quad u_{mdk} = \mathcal{F}_{dk}(\mathbf{z}_m) & n &= 1, \dots, N, m = 1, \dots, M \\
y_{nd} &\sim \text{Softmax}(\mathbf{f}_{nd}), & n &= 1, \dots, N, d = 1, \dots, D
\end{aligned} \tag{1}$$

where the Softmax function is computed as (we define $f_0 := 0$)

$$\text{Softmax}(y = k; \mathbf{f}) = \frac{\exp(f_k)}{\exp(\text{lse}(\mathbf{f}))}, k = 0, \dots, K, \quad \text{lse}(\mathbf{f}) = \log(1 + \sum_{k'=1}^{K} \exp(f_{k'})). \tag{2}$$

We tie the covariance matrices of the GPs, $\mathbf{K}_d$, to be the same for all categorical values in a given categorical variable and allow them to be different across categorical variables. 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) 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, aka 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 and 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

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

where

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

with

$$\mathbf{a}_{nd} = \mathbf{K}_{d,MM}^{-1}\mathbf{K}_{d,Mn}, \quad b_{nd} = K_{d,nn} - \mathbf{K}_{d,nM}\mathbf{K}_{d,MM}^{-1}\mathbf{K}_{d,Mn}. \tag{6}$$

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 and 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 and Welling, 2013; Rezende et al., 2014; Titsias and Lázaro-Gredilla, 2014) to compute the 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 and Lawrence, 2010; Hensman et al., 2013).

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

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

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 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}.

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

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

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

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

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

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}:

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

Notice that the transformation of the variables does not change the distribution of the original variables and therefore does not change the value of 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

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

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

$$\mathcal{L}_s^{nd} \approx \frac{1}{T}\sum_{i=1}^{T}\log\mathrm{Softmax}\left(\mathbf{y}_{nd}\middle|\mathbf{f}_{nd}\left(\boldsymbol{\varepsilon}_{nd}^{(f)}, \mathbf{U}_d(\boldsymbol{\varepsilon}_d^{(u)}), \mathbf{x}_n(\boldsymbol{\varepsilon}_n^{(x)})\right)\right) \tag{12}$$

with $\boldsymbol{\varepsilon}_n^{(x)}, \boldsymbol{\varepsilon}_d^{(u)}, \boldsymbol{\varepsilon}_{nd}^{(f)}$ 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 and 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 straightforward 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 Discussion and Conclusions

We have presented the first Bayesian model capable of capturing sparse multi-modal categorical distributions based on a continuous representation. 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 and robustness. We will scale the model following research on GP scalability done in (Hensman et al., 2013; Gal et al., 2014). Robustness of the model depends critically 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.

Acknowledgments

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

References

Alan Agresti and David B Hitchcock. Bayesian inference for categorical data analysis. *Statistical Methods and Applications*, 14(3):297–330, 2005.

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

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

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

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

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

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

James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

M. Zwitter and M. Soklic. Breast cancer dataset. In *University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia*, 1988.

A Experimental Results

We show the advantages of using a non-linear CDE compared to a linear one. For this we evaluate our model (CLGP) against the linear LGM (Khan et al., 2012). 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).

We demonstrate that linear models have difficulty with multi-modal distributions. We compare the linear and non-linear models on the simple task of *relational learning* (Paccanaro and Hinton, 2001), capturing the non-linear XOR relation based on observations of triplets such as $(1, 1, 0)$. The models are evaluated on their performance predicting missing values within partially observed relations.

We then assess the model in the real-world domain of sparse data, using the Wisconsin Breast Cancer dataset for which obtaining samples is a long and expensive process (Zwitter and Soklic, 1988). We compare the CLGP model to a histogram based approach, demonstrating the difficulty with frequency based approaches for sparse data. We further compare our model to the linear LGM on this dataset, demonstrating over-fitting problems with the model proposed in (Khan et al., 2012). Finally, we inspect the robustness of our inference, evaluating the Monte Carlo estimate variance.

For the following experiments, both the linear and non-linear models were initialise with a 2D latent space. The mean values of the latent points, m_n, were initialised at random following a standard normal distribution, as well as the mean values of the inducing outputs ($\boldsymbol{\mu}_{dk}$). 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 0.2. 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. We hold the covariance function hyper-parameters and s fixed for the first 10 iterations, as optimising these before the latent means move and an initial function is estimated results in divergence of the optimisation algorithm. Furthermore, we use RMSPROP for all quantities apart from s, for which we use gradient descent for optimisation as RMSPROP seems to change these quantities too fast. For the non-linear experiment we used 4 inducing points, and for all experiments we use the same kernel across all categorical variables. Lastly, our optimisation for the latents is harder than that of the linear model (in the linear model this optimisation is done over a convex surface whereas in the non-linear model it is not). We thus find all data points with predicted probability less than a predefined value for at least one of the categorical variables, and randomly sample a new latent mean value for each of these data points, assigning them to one of the inducing inputs. We do this every 20 iterations starting after the 100'th iteration, and use the predicted probability of the training set alone, ignoring the test variables. Our setting supports semi-supervised learning were the latents of partially observed data points are optimised with the training set, and then used to predict the missing values.

We assess the performance of the models using the same metric brought in (Khan et al., 2012). For each data point and each categorical variable, we take the probability the model assigns to the true categorical value for that variable. These are then passed through $-\log_2(p)$ and averaged. Thus, for the binary case, a random guess of each one of the binary values for each one of the data points (assigning probability 0.5) results in an error of 1. Correct assignment of all values results in error 0, and assignment of probability zero to at least one correct categorical values results in error ∞.

A.1 Linear Models Have Difficulty with Multi-modal Distributions

A simple example of relational learning (following Paccanaro and 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. For testing we introduce new $x_{n'}$ with missing categorical values for some of the variables. We add four more

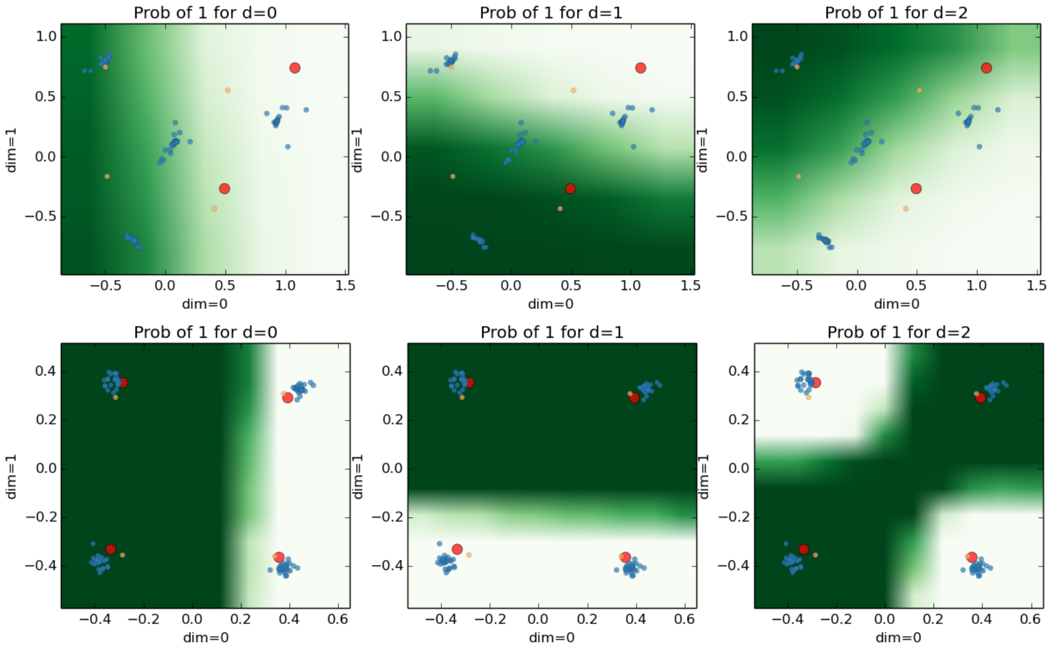

Figure 2: **Density over the *latent* space as predicted by the linear model (top, LGM), and non-linear model (bottom, CLGP).** Each figure, left to right, shows the density over the same latent space for a different single categorical variable (left to right: first digit in the triplet, second digit, and third digit in the triplet). The shade of green indicates the probability of a point in the latent space to take value 1 for that categorical variable. The darker the shade, 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.

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 and report the results.

We assessed the error of both linear and non-linear models on the task of predicting the 4 missing values (also known as *imputation*), repeating the experiment 3 times and averaging the results. The linear model obtained an error (and standard deviation) of 6.22 ± 0.24, whereas the non-linear model obtained an error of 0.03 ± 0.02. Note that a simple histogram based approach will do well on this dataset as it is not sparse.

During optimisation the linear model consistently jumps between different local modes, trying to capture all four possible triplets (fig. 2). The model assigns probabilities to the the missing values by capturing some of the triplets well, but cannot assign high probability to the others. An example assignment of probabilities to the correct answers (for the missing values of the four triplets) would be $(0.01,\ 0.43,\ 1,\ 0.01)$. In contrast, the CLGP model is able to capture all possible values of the relation. An example assignment of probabilities to the correct answers would be $(0.99, 0.99, 0.98, 1)$. Sampling from probability vectors sampled 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.

A.2 Sparse Datasets

We now assess our model in the real-world domain of sparse data, comparing our continuous latent approach to frequency based approaches. We use the Wisconsin Breast Cancer dataset (Zwitter and Soklic, 1988). 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 – indicating whether a tumour is benign or malignant. We use three quarters of the dataset for training and leave the rest

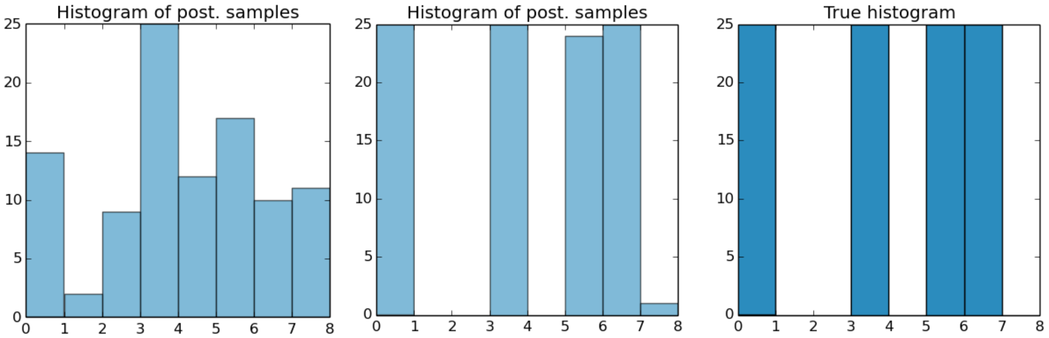

Figure 3: **Histogram of categorical values** (encoded in binary for the 8 possible values) for samples drawn from the posterior of the latent space of the linear model (left, LGM), the non-linear model (middle, CLGP), and the data used for training (right).

for testing, averaging the error on three repetitions of the experiment. We use three different random splits of the dataset. 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 before.

We compare our model (*CLGP*) to a baseline model predicting uniform probability for all values (*Baseline*), the linear *LGM* model, and a frequency based model predicting the probability for a missing value based on its frequency in the training set (*Multinomial*). The last model can be interpreted as the maximum likelihood estimate (MLE) of a multinomial likelihood over the observations for each single categorical variable. Lastly, we use a smoothed frequency model using Laplace smoothing (with a smoothing coefficient of 1, commonly known as "add one smoothing"), and a frequency model using a smaller smoothing coefficient of 0.01. The smoothed frequency models are equivalent to the maximum a posteriori (MAP) estimate of the MLE model with a Dirichlet distribution prior (referred to as *Dirichlet Multinomial* in the experiments). Small parameters for the Dirichlet distribution are often used in sparse settings. More complicated frequency based approaches are possible, performing variable selection and then looking at frequencies of pairs or triplets of variables. These will be very difficult in this sparse small data problem.

We evaluate the models' performance using imputation error as before on the Breast cancer dataset (Table 1). As can be seen in the results, the frequency based approaches obtain worse results than the continuous latent space approaches. The frequency model with no smoothing obtains error ∞ for split 2 because one of the test points has a value not observed before in the training set. Using smoothing solves this but results in higher error values for the other splits. The baseline (predicting uniformly) obtains the highest error on average. The linear model exhibits high variance for the last

Split	Baseline	Multinomial	Dirichlet Multinomial ($\alpha = 1$)	Dirichlet Multinomial ($\alpha = 0.01$)	LGM	**CLGP**
1	3.118	2.130	2.146	2.131	1.835 ± 0.085	$\mathbf{1.517 \pm 0.060}$
2	3.118	∞	2.152	2.174	$\mathbf{1.797 \pm 0.103}$	$\mathbf{1.748 \pm 0.078}$
3	3.145	2.224	2.229	2.224	3.601 ± 1.042	$\mathbf{1.742 \pm 0.042}$

Table 1: **Error (and standard deviation) on the Breast cancer dataset,** predicting the randomly missing *categorical* values. The error values are calculated for three different random splits of the dataset. The models compared are (left to right): a baseline model predicting uniform probability for all values (*Baseline*), a frequency model predicting the probability for a missing value based on its frequency in the training set (*Multinomial*), *Dirichlet-Multinomial* models with concentration parameters $\alpha = 1$ and $\alpha = 0.01$ (also known as Laplace smoothing), the *LGM* model, and lastly the proposed model (*CLGP*).

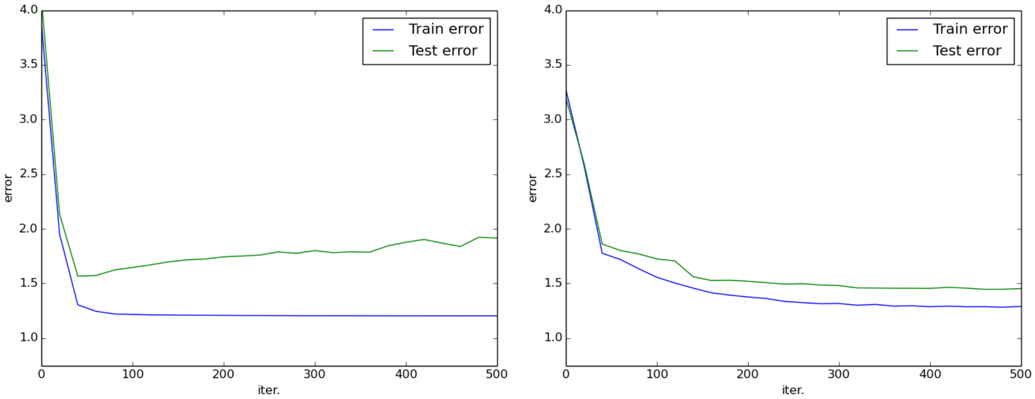

Figure 4: **Train and test error for LGM (left) and the CLGP model (right) for one of the splits of the breast cancer dataset.** The train error of LGM decreases while the test error starts increasing at iteration 50.

split (with error values 2.3, 3.7, and 4.8), and in general has higher error standard deviation than the non-linear model.

A.2.1 LGM Over-fitting

It is interesting to note that the latent Gaussian model (LGM) exhibits over-fitting on the last dataset. 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 repetitions the model's training error decreases while the test error starts increasing (see fig. 4 for the train and test error of split 1). It is interesting to note that even though the test error starts increasing, at its lowest point it is still higher than the end test error of the CLGP model. This is observed for all splits and all repetitions.

A.3 Inference Robustness

Lastly, we inspect the robustness of our inference, evaluating the Monte Carlo estimate standard deviation. Fig. 5 shows the average ELBO standard deviation per iteration (averaged over 3 repetitions)

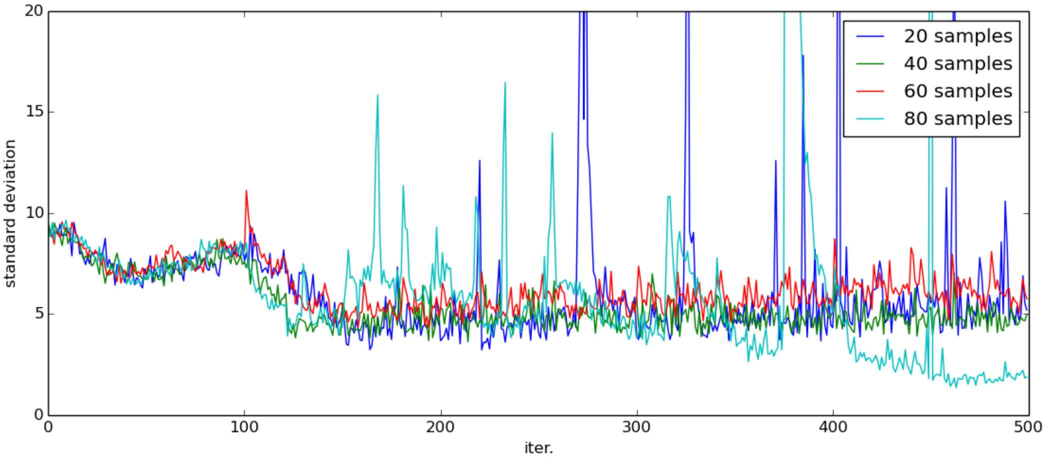

Figure 5: **Standard deviation (averaged over 3 repetitions) per iteration on the XOR dataset.** Shown are standard deviation using Monte Carlo integration with 20, 40, 60, and 80 samples.

on the XOR dataset. We used Monte Carlo integration with 20 samples (used for the experiments above), 40, 60, and 80 samples. The standard deviation for 20 and 80 samples fluctuates considerably, with erratic peaks as high as 100. There is a general decreasing trend in the standard deviation's magnitude, with a slight increase after 40 iterations. After 100 iterations (when poorly performing latents are shuffled) there is a sharp decrease in standard deviation. From this plot we see a slight decrease in variance with more samples. It is interesting to note that as the approximating variational distribution gets closer to the true posterior, the variance decreases.