Latent Gaussian Processes for Distribution Estimation of Multivariate Categorical Data

Yarin Gal, Yutian Chen, Zoubin Ghahramani

University of Cambridge
{yg279, yc373, zg201}@cam.ac.uk

In short:

Multivariate categorical data –

- in data analysis, language processing, medical diagnosis...
- the number of **possible vectors of observations** N **grows exponentially** with the number of discrete variables D in the vector.
- the **diversity of data points is poor** compared to the exponentially many possible observations.

Example: Wisconsin breast cancer (Institute of Oncology, Ljubljana)

Sample code	Thickness	Unif. Cell Size	Unif. Cell Shape	Marginal Adhesion	Epithelial Cell Size	Bare Nuclei	Bland Chromatin	Normal Nucleoli	Mitoses	Class
1000025	5	1	1	1	2	1	3	1	1	Benign
1002945	5	4	4	5	7	10	3	2	1	Benign
1015425	3	1	1	1	2	2	3	1	1	Benign
1016277	6	8	8	1	3	4	3	7	1	Benign
1017023	4	1	1	3	2	1	3	1	1	Benign
1017122	8	10	10	8	7	10	9	7	1	Malignant
1018099	1	1	1	1	2	10	3	1	1	Benign
1018561	2	1	2	1	2	1	3	1	1	Benign
...	...	...	...	...	...	...	...	...	...	...

683 patients, 2×10^9 **possible configurations.**

- We develop a model for **distribution estimation** of multivariate categorical data:

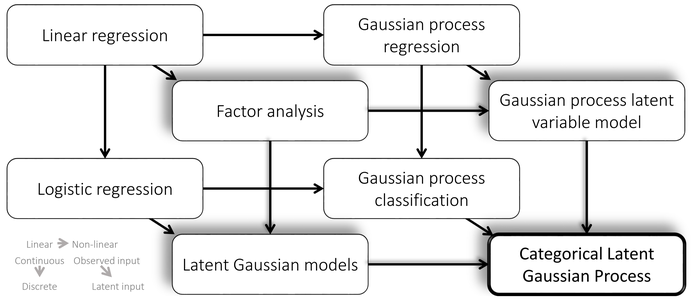
$$P(\mathbf{y} \mid \{\mathbf{y}_n = (y_{n1}, ..., y_{nD}) \mid n = 1, ..., N\})$$

- We use a **continuous** latent Gaussian space and learn a non-linear transformation between it and the multivariate categorical observation space.
- We derive inference for our model based on recent developments in **sampling-based variational inference and stochastic optimisation**.

Relation to other models

Existing approaches use –

- **Discrete representations**: based on frequencies of observations, but cannot handle sparse samples well (e.g. Dirichlet-Multinomial).
- **Continuous representations**: linearly transform a latent space before discretisation, but cannot capture multi-modality in the data (e.g. latent Gaussian model).



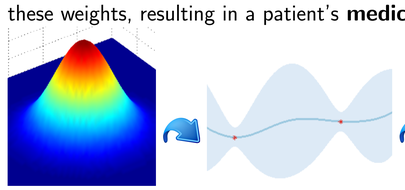
Our model can be seen as a non-linear version of the *latent Gaussian model* (left to right, Khan et al. (2012)), as a latent counterpart to the *Gaussian process (GP) classification* model (back to front, Williams and Rasmussen (2006)), or as a discrete extension of the *Gaussian process latent variable model* (top to bottom, Lawrence (2005)).

The Categorical Latent Gaussian Process (CLGP)

We define the generative model, with kernel $\mathbf{K}(\cdot, \cdot)$, as

$$\mathbf{x}_n \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_x^2 I), \quad (f_{ndk})_{n=1}^N \sim \mathcal{N}(0, \mathbf{K}((\mathbf{x}_n)_{n=1}^N)), \quad y_{nd} \overset{\text{iid}}{\sim} \text{Softmax}(f_{nd1}, ..., f_{ndK}).$$

Following a breast cancer diagnosis example, 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; Softmax returns test result y_{nd} based on these weights, resulting in a patient's **medical assessment** $\mathbf{y}_n = (y_{n1}, ..., y_{nD})$.



Continuous *latent* space of "patients"

Distribution over functions f_{ndk}

$y_{nd} \overset{\text{iid}}{\sim} \text{Softmax}(f_{nd1}, ..., f_{ndK})$
Test result
$\mathbf{y}_n = (y_{n1}, ..., y_{nD})$
Medical diagnosis

Inference

- We use **Sparse GPs** to get linear time complexity – we condition the observations on M inducing inputs $\mathbf{Z}$ with inducing outputs $\mathbf{U}$ with a Gaussian prior.
- Our marginal log-likelihood is intractable. We lower bound the log evidence with a variational approximate posterior $q(\mathbf{X}, \mathbf{F}, \mathbf{U}) = q(\mathbf{X})q(\mathbf{U})p(\mathbf{F}|\mathbf{X}, \mathbf{U})$, with

$$\begin{aligned} x_{ni} &= m_{ni} + s_{ni}\varepsilon_{ni}^{(x)} & \varepsilon_{ni}^{(x)} &\sim \mathcal{N}(0, 1) \\ \mathbf{u}_{dk} &= \boldsymbol{\mu}_{dk} + \mathbf{L}_d \boldsymbol{\varepsilon}_{dk}^{(u)} & \boldsymbol{\varepsilon}_{dk}^{(u)} &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_M) \\ f_{ndk} &= \mathbf{a}_n^T \mathbf{u}_{dk} + \sqrt{b_n}\varepsilon_{ndk}^{(f)} & \varepsilon_{ndk}^{(f)} &\sim \mathcal{N}(0, 1) \end{aligned}$$

and

$$\mathbf{a}_n = \mathbf{K}_{MM}^{-1}\mathbf{K}_{Mn}, \quad b_n = K_{nn} - \mathbf{K}_{nM}\mathbf{K}_{MM}^{-1}\mathbf{K}_{Mn}.$$

Then,

$$\begin{aligned} \log p(\mathbf{Y}) &= \log \int p(\mathbf{X})p(\mathbf{U})p(\mathbf{F}|\mathbf{X}, \mathbf{U})p(\mathbf{Y}|\mathbf{F}) \mathrm{d}\mathbf{X} \mathrm{d}\mathbf{F} \mathrm{d}\mathbf{U} \\ &\geq -\text{KL}(q(\mathbf{X}) \| p(\mathbf{X})) - \text{KL}(q(\mathbf{U}) \| p(\mathbf{U})) \\ &\quad + \sum_{n=1}^N \sum_{d=1}^D \mathbb{E}_{\boldsymbol{\varepsilon}_n^{(x)}, \boldsymbol{\varepsilon}_d^{(u)}, \boldsymbol{\varepsilon}_{nd}^{(f)}} \log \text{Softmax}\left(\mathbf{y}_{nd} \middle| \mathbf{f}_{nd}\left(\boldsymbol{\varepsilon}_{nd}^{(f)}, \mathbf{U}_d(\boldsymbol{\varepsilon}_d^{(u)}), \mathbf{x}_n(\boldsymbol{\varepsilon}_n^{(x)})\right)\right). \end{aligned}$$

Method

1. Monte Carlo integration approximates the likelihood obtaining noisy gradients:

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

2. Learning-rate free stochastic optimisation is used to optimise the noisy objective.
3. Symbolic differentiation is used to get simple and modular code:

```
import theano.tensor as T
X = m + s * randn(N, Q)
U = mu + L.dot(randn(M, K))
Kmm, Kmn, Knn = RBF(sf2, l, Z), RBF(sf2, l, Z, X), RBFnn(sf2, l, X)
KmmInv = sT.matrix_inverse(Kmm)
A = KmmInv.dot(Kmn)
B = Knn - T.sum(Kmn * KmmInv.dot(Kmn), 0)
F = A.T.dot(U) + B[:,None]**0.5 * randn(N, K)
S = T.nnet.softmax(F)
KL_U, KL_X = get_KL_U(), get_KL_X()
LS = T.sum(T.log(T.sum(Y * S, 1))) - KL_U - KL_X
LS_func = theano.function(['inputs'], LS)
dLS_dm = theano.function(['inputs'], T.grad(LS, m)) # and others
# ... and run RMS-PROP
```

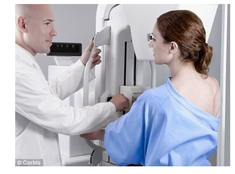
That's all.

Experiments

Test perplexity predicting randomly missing values

Uniform	Multinomial	Bi-Dir-Mult	LGM	**CLGP**
8.68	4.41	3.57	3.41	**2.86**

Wisconsin breast cancer

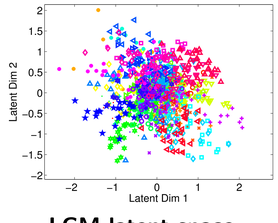


Uniform	Multinomial	LGM	**CLGP**
2.50	2.20	3.07	**2.11**

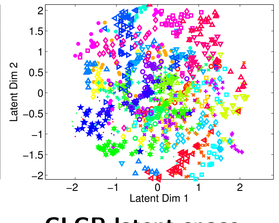
Terror Warning Effects on Political Attitude
(START Terrorism Data Archive)

Data Visualisation

Example alphadigits



LGM latent space



CLGP latent space

Code available at `http://github.com/yaringal/CLGP`.