Latent Gaussian Processes for Distribution Estimation of Multivariate Categorical Data – Appendix

A Appendix

A.1 Code

The basic model and inference (without covariance matrix caching) can be implemented in 20 lines of Python and Theano for each categorical variable:

```
import theano.tensor as T
m = T.dmatrix('m') # ..and other variables
X = m + s * randn(N, Q)
U = mu + L.dot(randn(M, K))
Kmm = RBF(sf2, l, Z)
Kmn = RBF(sf2, l, Z, X)
Knn = RBFnn(sf2, l, X)
KmmInv = T.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 optimise LS with RMS-PROP
```