A Nonparametric Model of Censoring

Anonymous Author(s)
Affiliation
Address
email

Abstract

Data-dependent non-response is one of the central problems of statistical inference. We introduce a nonparametric density model capabale of inferring both censored distributions, and a model of the censoring process. This is acheived through the use of a novel inference scheme: "exact-approximate" Hamiltonian Monte Carlo. We demonstrate our model on toy datasets as well as cool real problems.

1 Introduction

[Discuss censoring, truncation, data-dependent non-response]

[Give several examples: surveys, government censorship, faulty sensors]

[Briefly discuss previous approaches]

We introduce a generative non-parametric model to address this problem. Our approach is based on the GP-LVM [**?**, **?**, **?**], a flexible nonparametric density model.

This model is a natural problem with which to demonstrate recent advances in "exact-approximate" Monte Carlo methods, which allow one to construct valid Markov chains with only approximate evaluations of the likelihood function.

2 Censoring problems

In this section, we give a formal definition of censoring and truncation.

Food for thought: If data is truncated, but we know its marginal distribution, would that be more accurately described as censoring, since we do know that the data is there?

[Cite Ben Marlin's thesis. Discuss how people usually assume that the non-response is independent of the data to allow simple inference schemes, but that this is usually unrealistic]

Examples:

3 The Censored GP-LVM

In this section, we define in detail the actual model.

3.1 Gaussian Process Latent Variable Model

The GP-LVM specifies a model wherein latent variables $\mathbf{X}$ are warped by an unknown smooth, function f to produce the observed data $\mathbf{Y}$. The prior used over functions in the GP-LVM is the Gaussian process [**?**].

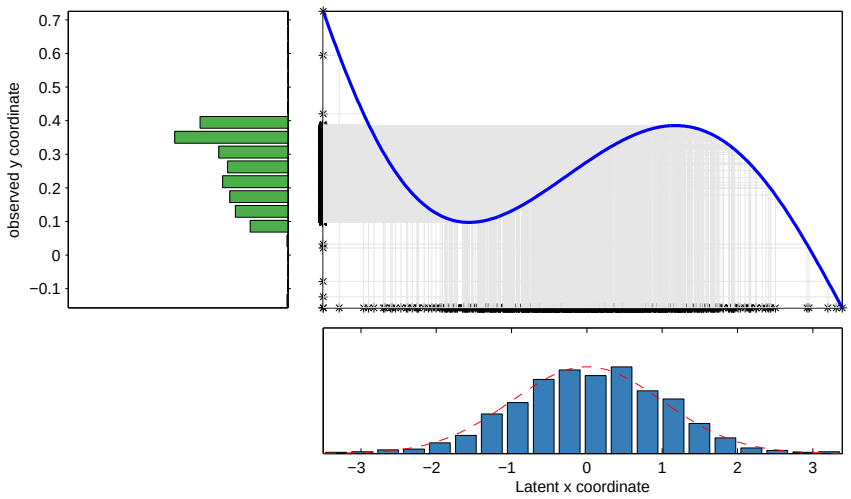

Figure 1: A draw from a Gaussian process latent variable model. Bottom: The latent datapoints $\mathbf{X}$ are distributed according to a parametric base distribution (a Gaussian). Top right: A smooth function f drawn from a Gaussian process prior is applied to obtain $\mathbf{Y} = f(\mathbf{X})$. Left: The observed data $\mathbf{Y}$ is distributed according to a non-Gaussian density.

While not typically thought of as a density model, the GPLVM does define a nonparametric density over observations [**?**]. Figure 1 demonstrates how a Gaussian latent density, when warped by a random smooth function, can give rise to a non-Gaussian density in the observed space.

The dimension of the observed data (D) doesn't need to match the dimension of the latent space (Q). When Q is 2 or 3, the GP-LVM can also be used for visualization of high-dimensional data. The mapping from $\mathbf{X}$ to each dimension of the observed data is assumed to be independent, so the likelihood has a simple form which implicitly integrates over f:

$$p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta}) = (2\pi)^{-\frac{DN}{2}} |\mathbf{K}|^{-\frac{D}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{Y}^\top \mathbf{K}^{-1}\mathbf{Y})\right), \tag{1}$$

where $\mathbf{K}$ is the $N \times N$ covariance matrix defined by the kernel function $k(\mathbf{x}_n, \mathbf{x}_m)$, and $\boldsymbol{\theta}$ is the kernel hyperparameter vector. In this paper, we use an RBF kernel with an additive noise term:

$$k(\mathbf{x}_n, \mathbf{x}_m) = \alpha \exp\left(-\frac{1}{2\ell^2}(\mathbf{x}_n - \mathbf{x}_m)^\top(\mathbf{x}_n - \mathbf{x}_m)\right) + \delta_{nm}\beta^{-1}. \tag{2}$$

3.2 Censoring model

3.2.1 Parametric Model

[Note: We can probably think of a better notation]

Our model assumes that there is an underlying density model $P(\mathbf{Y})$, and further that the probability of censoring depends on the data. We denote the probability that a datapoint from the non-censored population was censored by $P(c|\mathbf{y})$. The probability of observing the same datapoint is simply $P(o|\mathbf{y}) = 1 - P(c|\mathbf{y})$. Conditioned on $\mathbf{Y}$, all censorings are independent: $P(\mathbf{c}|\mathbf{Y}) = \prod_i P(c_i|\mathbf{y}_i)$

3.3 Computing the Likelihood

Explicitly modeling data-dependent non-response is not typically done, since computing the likelihood requires one to integrate over possible responses that one could have seen, but didn't. To define this integral also requires a density model over responses.

In the following we use that the prior on the latent space is i.i.d.:

$$P(\mathbf{X}_c|\mathbf{X}_o) = P(\mathbf{X}_c)$$

and that conditioned on the latent variables, the observations in a GPLVM are independent:

$$P(\mathbf{Y}_o, \mathbf{Y}_c|\mathbf{X}_o, \mathbf{X}_c) = P(\mathbf{Y}_o|\mathbf{X}_o, \mathbf{X}_c)P(\mathbf{Y}_c|\mathbf{X}_o, \mathbf{X}_c)$$

and that GPs satisfy the marginalization property (they are projective):

$$P(\mathbf{Y}_o, |\mathbf{X}_o, \mathbf{X}_c) = P(\mathbf{Y}_o|\mathbf{X}_o)$$

Now, we must attempt to integrate over all possible data that we could have seen. We want to run a chain drawing samples from $P(\mathbf{X}_o|\mathbf{Y}_o)$:

$$P(\mathbf{X}_o|\mathbf{Y}_o) = \frac{P(\mathbf{Y}_o|\mathbf{X}_o)P(\mathbf{X}_o)}{\int P(\mathbf{Y}_o|\mathbf{X}'_o)P(\mathbf{X}'_o)d\mathbf{X}'_o} = \frac{P(\mathbf{Y}_o|\mathbf{X}_o)P(\mathbf{X}_o)}{P(\mathbf{Y}_o)'} \tag{3}$$

So, we actually just need to evaluate $P(\mathbf{Y}_o|\mathbf{X}_o)$:

$$P(\mathbf{Y}_o, \mathbf{c}_o = 0|\mathbf{X}_o) = \iint P(\mathbf{c}, \mathbf{Y}_o, \mathbf{Y}_c, \mathbf{X}_c|\mathbf{X}_o)d\mathbf{Y}_c d\mathbf{X}_c \tag{4}$$

$$= \iint P(\mathbf{c}, \mathbf{Y}_o, \mathbf{Y}_c|\mathbf{X}_o, \mathbf{X}_c)P(\mathbf{X}_c|\mathbf{X}_o)d\mathbf{Y}_c d\mathbf{X}_c \tag{5}$$

$$= \iint P(\mathbf{c}|\mathbf{Y}_o, \mathbf{Y}_c, \mathbf{X}_o, \mathbf{X}_c)P(\mathbf{Y}_o, \mathbf{Y}_c|\mathbf{X}_o, \mathbf{X}_c)P(\mathbf{X}_c)d\mathbf{Y}_c d\mathbf{X}_c \tag{6}$$

$$= \iint P(\mathbf{c}|\mathbf{Y}_o, \mathbf{Y}_c)P(\mathbf{Y}_o, \mathbf{Y}_c|\mathbf{X}_o, \mathbf{X}_c)P(\mathbf{X}_c)d\mathbf{Y}_c d\mathbf{X}_c \tag{7}$$

$$= \iint P(\mathbf{c}_o = 0|\mathbf{Y}_o)P(\mathbf{c}_c = 1|\mathbf{Y}_c)P(\mathbf{Y}_o, \mathbf{Y}_c|\mathbf{X}_o, \mathbf{X}_c)P(\mathbf{X}_c)d\mathbf{Y}_c d\mathbf{X}_c \tag{8}$$

$$= P(\mathbf{c}_o = 0|\mathbf{Y}_o) \iint P(\mathbf{c}_c = 1|\mathbf{Y}_c)P(\mathbf{Y}_c, \mathbf{Y}_o|\mathbf{X}_o, \mathbf{X}_c)P(\mathbf{X}_c)d\mathbf{Y}_c d\mathbf{X}_c \tag{9}$$

$$= P(\mathbf{c}_o = 0|\mathbf{Y}_o) \iint P(\mathbf{c}_c = 1|\mathbf{Y}_c)P(\mathbf{Y}_c|\mathbf{Y}_o, \mathbf{X}_o, \mathbf{X}_c)P(\mathbf{Y}_o|\mathbf{X}_o, \mathbf{X}_c)P(\mathbf{X}_c)d\mathbf{Y}_c d\mathbf{X}_c \tag{10}$$

$$= P(\mathbf{c}_o = 0|\mathbf{Y}_o) \underbrace{P(\mathbf{Y}_o|\mathbf{X}_o)}_{\text{GPs}} \underbrace{\iint \sum_{N_c=0}^{\infty} P(N_c|\mathbf{Y}_o, \mathbf{X}_o) \prod_{i=1}^{Nc} P(\mathbf{c}_c^{(i)} = 1|\mathbf{Y}_c^{(i)})P(\mathbf{Y}_c|\mathbf{Y}_o, \mathbf{X}_o, \mathbf{X}_c)P(\mathbf{X}_c|N_c)d}_{\text{Probability of censoring}} \tag{11}$$

3.4 Estimating the probability of censoring

The big question is, how to estimate the probabiltiy of censoring:

$$\iint \sum_{N_c=0}^{\infty} P(N_c|\mathbf{Y}_o, \mathbf{X}_o) \prod_{i=1}^{Nc} P(\mathbf{c}_c^{(i)} = 1|\mathbf{Y}_c^{(i)})P(\mathbf{Y}_c|\mathbf{Y}_o, \mathbf{X}_o, \mathbf{X}_c)P(\mathbf{X}_c|N_c)d\mathbf{Y}_c d\mathbf{X}_c \tag{12}$$

where

$$\mathbf{x}_1 \dots \mathbf{x}_S \sim_{\text{iid}} P(\mathbf{X}_C) = \mathcal{N}(\mathbf{x}|\mathbf{0}, zvI) \tag{13}$$

and

$$\mathbf{y}_1 \dots \mathbf{y}_S \sim_{\text{iid}} P(\mathbf{y}|\mathbf{Y}_o, \mathbf{X}_o, \mathbf{x}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{y}_d|k(\mathbf{x}, \mathbf{X}_o)\mathbf{K}^{-1}\mathbf{Y}_d, k(\mathbf{x}, \mathbf{x}) - k(\mathbf{x}, \mathbf{X}_o)\mathbf{K}^{-1}k(\mathbf{X}_o, \mathbf{x})) \tag{14}$$

In our experiments, we used $S = 100$. Note that (**??**) is a non-negative, unbiased estimator of the probability that a randomly chosen datapoint will be censored. Since this unbiased estimate is multiplied by the rest of the likelihood, our joint likelihood evaluation is also unbiased.

4 Incorporating Side Information

Often, the reason we suspect that truncation has occured is that the sample distribution along some dimension grossly mismatches the distribution in the population we are attempting to measure. For example, we may notice that a survey had no respondents in a certain income range, or we may know that the joint distribution of survey respondent's age and employment

[Describe how we can use information about the marginals of some of the observed variables to inform where mass might exist, and where censoring is likely to have occurred.]

[Include figures to demonstrate the use of side information in a one-or-two dimensional example]

A derivation with side information:

$$P(\mathbf{Y}_o, \mathbf{c}_o = 1|\mathbf{X}_o) = \iint P_{\text{side}}(\mathbf{Y}_c) P_{\text{side}}(\mathbf{Y}_o) P(\mathbf{c}, \mathbf{Y}_o, \mathbf{Y}_c, \mathbf{X}_c|\mathbf{X}_o) d\mathbf{Y}_c d\mathbf{X}_c \tag{15}$$

$$= P_{\text{side}}(\mathbf{Y}_o) P(\mathbf{c}_o = 0|\mathbf{Y}_o) \underbrace{P(\mathbf{Y}_o|\mathbf{X}_o)}_{\text{GPs}} \underbrace{\iint P(\mathbf{c}_c = 1|\mathbf{Y}_c) P_{\text{side}}(\mathbf{Y}_c) P(\mathbf{Y}_c|\mathbf{Y}_o, \mathbf{X}_o, \mathbf{X}_c) P(\mathbf{X}_c) d\mathbf{Y}_c d\mathbf{X}_c}_{\text{Probability of censoring}} \tag{16}$$

The likelihood gets a constant term for the observed data, and the probability of sensoring gets an importance weight term:

$$\iint P(\mathbf{c}_c = 1|\mathbf{Y}_c) P_{\text{side}}(\mathbf{Y}_c) P(\mathbf{Y}_c|\mathbf{Y}_o, \mathbf{X}_o, \mathbf{X}_c) P(\mathbf{X}_c) d\mathbf{Y}_c d\mathbf{X}_c = \tag{17}$$

$$\approxeq \frac{\sum_{i=1}^{S} P(c_i = 1|\mathbf{y}_i) P_{\text{side}}(\mathbf{y}_i)}{\sum_{i=1}^{S} P_{\text{side}}(\mathbf{y}_i)} \tag{18}$$

In words, we can say that the model won't penalize putting mass in regions that are both compatible with the censoring and with the side information.

4.1 Predictive Density

When computing predictive density of observed points, we must condition on the fact that we will see them:

$$P(y* | c* = 0, x*, \mathbf{X}_o, \mathbf{Y}_o) = \frac{\tilde{P}(y*, c* = 0|x*, \mathbf{X}_o, \mathbf{Y}_o)}{P(c* = 0|\mathbf{X}_o, \mathbf{Y}_o)} = \frac{\tilde{P}(c* = 0|y*) P(y* | x*, \mathbf{X}_o, \mathbf{Y}_o)}{P(c* = 0|\mathbf{X}_o, \mathbf{Y}_o)} \tag{19}$$

and we can estimate $P(c* = 0|\mathbf{X}_o, \mathbf{Y}_o)$ by simple monte carlo:

$$P(c* = 0|y*) \approxeq \frac{1}{S} \sum_{i=1}^{S} P(c_i = 0|\mathbf{y}_i) \tag{20}$$

where

$$\mathbf{x}_1 \dots \mathbf{x}_S \sim_{\text{iid}} P(\mathbf{X}_C) = \mathcal{N}(\mathbf{x}|\mathbf{0}, zvI) \tag{21}$$

and

$$\mathbf{y}_1 \dots \mathbf{y}_S \sim_{\text{iid}} P(\mathbf{y}|\mathbf{Y}_o, \mathbf{X}_o, \mathbf{x}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{y}_d | k(\mathbf{x}, \mathbf{X}_o)\mathbf{K}^{-1}\mathbf{Y}_d, k(\mathbf{x}, \mathbf{x}) - k(\mathbf{x}, \mathbf{X}_o)\mathbf{K}^{-1}k(\mathbf{X}_o, \mathbf{x})) \tag{22}$$

5 Inference

Inference in the GP-LVM requires integrating over the latent coordinates of each datapoint. Thus, the model typically has hundreds or thousands of continuous parameters. This means that Hamiltonian Monte Carlo [HMC] [cite] is an appropriate inference method [**?**].

Evaluating our model's likelihood requires an intractable integral over all datasets which could have been generated, but were censored. This is true even when the warping and censoring functions are given.

Until recently, it was not known if Metropolis-Hasting would still converge to the correct stationary distribution if the evaluation of the likelihood ratio was noisy.

5.1 Exact-Approximate Metropolis-Hastings

Recently, it was shown by [cite] that Metropolis-Hastings can still converge to the correct posterior distribution even when the estimated likelihoods are approximate. The conditions necessary are: ???

Here we give a derivation of exact-approximate MH. Following [**?**], we denote the likelihood $p(x)$ and the approximate likelihood by $r(x)$. The acceptance ratio has the form:

$$A = \frac{p(x')q(x'|x)}{p(x)q(x|x')} \tag{23}$$

The approximate acceptance ratio has the form:

$$A = \frac{r(x')q(x'|x)}{r(x)q(x|x')} \tag{24}$$

If we denote $w = \frac{r(x)}{p(x)}$, then we can rewrite (**??**) as:

$$A = \frac{w'p(w'|x')p(x')q(x'|x)}{wp(w|x)p(x)q(x|x')} \tag{25}$$

or, we can say that we are proposing the pair (w, x) from the proposal distribution. If we know that $\mathbb{E}\left[r(x)\right] = p(x)$, then the target of the chain (**??**) must be proportional to $wp(x)p(w|x)$, which has a marginal $\int wp(x)p(w|x)dw = p(x)$, which is what we wanted.

5.2 Exact-Approximate Hamiltonian Monte Carlo

Hamiltonian Monte-Carlo can be understood as a special case of Metropolis-Hastings with a complicated proposal distribution. Specifically, proposals are generated by approximately following isocontours of the Hamiltonian, defined by the log-likelihood surface and a (randomly sampled) momentum term.

We can show that (**??**) is symmetric. First, we must note that the leapfrog dynamics are symmetric when given opposite momentum. leapfrog(x, m) is a deterministic function of (x, m).

If

$$\text{leapfrog}(\mathbf{x}, m) = \mathbf{x}' \tag{26}$$

then

$$\text{leapfrog}(\mathbf{x}', -m) = \mathbf{x} \tag{27}$$

This fact, combined with a symmetric proposal for $m \sim \mathcal{N}(w|0, 1)$ where $p(m) = p(-m)$ means that

$$\begin{aligned}
Q(x', m'|x, m) =& \text{I}(\text{leapfrog}(\mathbf{x}, m') = \mathbf{x}')p(m'|m) && (28)\\
=& \text{I}(\text{leapfrog}(\mathbf{x}, m') = \mathbf{x}')p(m') && (29)\\
Q(x, m|x', m') =& \text{I}(\text{leapfrog}(\mathbf{x}', -m') = \mathbf{x})p(m|m') && (30)\\
=& \text{I}(\text{leapfrog}(\mathbf{x}', -m') = \mathbf{x})p(m) && (31)\\
& && (32)
\end{aligned}$$

Because we chose $\mathbf{x}'$ by the leapfrog, the indicator part is one. so

$$\begin{aligned}
Q(x', m'|x, m) =& p(m') && (33)\\
Q(x, m|x', m') =& p(-m') && (34)\\
& && (35)
\end{aligned}$$

since the proposal distribution is symmetric, $p(m') = p(-m')$, so

$$\frac{Q('x|x)}{Q(x|x')} = 1 \tag{36}$$

Note that $p(m)$ does not depend on x - if it did, it might be harder to show that we satisfy detailed balance.

Thus, as long as the gradient is exact, HMC can be run without modification with an approximate likelihood, as long as the approximate likelihood $r(x)$ is not recomputed during each iteration.

5.2.1 Exact-Approximate HMC with stochastic gradients

Now, we will show that, even whe the gradients of HMC are stochastic, we can still construct a chain that samples from $p(x)$.

We will now write grad($\mathbf{x}$, $\mathbf{s}$) as the deterministic gradient of the loglikelihood used by the leapfrog dynamics. Note that it is also a function of $\mathbf{s}$, our 'random seed'. As long as $\mathbf{s}$ is constant, leapfrog$(\text{grad}(\mathbf{x}, \mathbf{s}), m)$ is a valid proposal distribution.

We can also put a distribution on $\mathbf{s}$, and as long as its distribution does not depend on x or m,

In our example, we will sample the censored $\mathbf{y}$ by multiplying Normally-distributed s by a cholesky decomposition as in (**??**). These $\mathbf{y}$ depend on $\mathbf{x}$, but the chain is still valid???

$$\begin{aligned}
Q(x', m', s'|x, m, s) =& \text{I}(\text{leapfrog}(\text{grad}(\mathbf{x}, \mathbf{s}), m') = \mathbf{x}')p(m'|m)p(s'|s) && (37)\\
=& \text{I}(\text{leapfrog}(\text{grad}(\mathbf{x}, \mathbf{s}), m') = \mathbf{x}')p(m')p(s') && (38)\\
Q(x, m|x', m') =& \text{I}(\text{leapfrog}(\text{grad}(\mathbf{x}', \mathbf{s}), m') = \mathbf{x})p(m|m')p(s|s') && (39)\\
=& \text{I}(\text{leapfrog}(\text{grad}(\mathbf{x}', \mathbf{s}), m') = \mathbf{x})p(m)p(s) && (40)\\
& && (41)
\end{aligned}$$

6 Related Work

6.1 Multiple-output Regression

Modeling the joint density of all dimensions of the observed data $\mathbf{Y}$ allows us to answer any question we may care to ask about predictive densities. However, if we only wish to predict the conditional density of some dimensions of $\mathbf{y}_{\text{query}}$ conditioned on others $\mathbf{y}_{\text{known}}$, then we may only need to model the conditional density $P(\mathbf{y}_{\text{query}}|\mathbf{y}_{\text{known}})$. The standard regression framework assumes that $\mathbf{y}_{\text{query}} = f(\mathbf{y}_{\text{known}})$. This model can be simpler to use than a general density estimation procedure. Multiple-output regression techniques [cite a bunch] can also "borrow statistical strength" from the different densities it must model.

Regression is also to extrapolate into censored regions of $\mathbf{y}_{\text{known}}$ if given a rich enough model [cite GPSS?], but it is not clear how a purely conditional $P(\mathbf{y}_{\text{query}}|\mathbf{y}_{\text{known}})$ can correctly handle censoring of regions of $\mathbf{y}_{\text{query}}$. It is also not clear how such a conditional model could infer the censoring function.

Thus, our model can be expected to be more appropriate than multiple-output regression under any of the following conditions:

- The data is possibly censored in some of the dimensions we wish to predict;
- We are interested in inferring censoring in any of the data dimensions;
- We wish to use side information about the population densities of any of the dimensions;
- We wish to answer questions about the joint density of the data.
- We wish to estimate how many responses were censored.
- We wish to quantify our uncertainty about the population of interest.

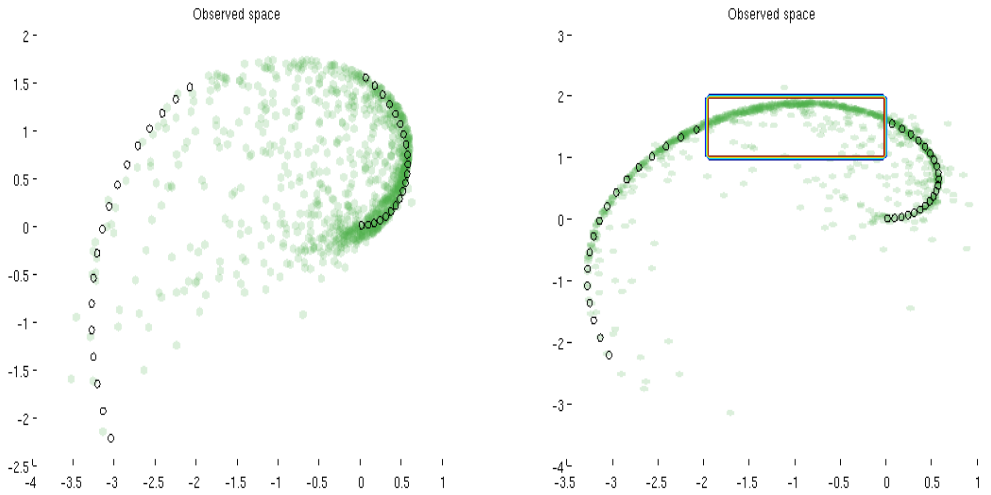

Figure 2: Modeling a truncated dataset. Left: Naively fitting a density manifold to the data. Right: When truncation is modeled explicitly, the model is not penalized for placing mass in censored regions, even though there is no mass there.

6.2 Max Welling and Yee Whye Teh's recent approximate Langevin work

Max Welling proposed using an approximate likelihood for the proposal and an exact likelihood-ratio evaluation. We showed that you can do both.

6.3 Adams and Murray GP density sampler

Can be viewed as a nonparametric sensoring of a Gaussian distribution.

[Massive lit search required]

7 Experiments

7.1 Source code

Code to reproduce all the above experiments will be made available upon publication.

7.2 Synthetic data

[Idea for a figure: plot the censored region in the latent space]

7.3 Real data

8 Conclusions

Acknowledgments

We would like to thank Pushmeet Kohli for helpful discussions.

References

[1] C.E. Rasmussen and CKI Williams. Gaussian Processes for Machine Learning. *The MIT Press, Cambridge, MA, USA*, 2006.

[2] H. Nickisch and C. Rasmussen. Gaussian mixture modeling with Gaussian process latent variable models. *Pattern Recognition*, pages 272–282, 2010.

[3] Zoubin Ghahramani Tomoharu Iwata, David Duvenaud. Warped mixtures for nonparametric cluster shapes. *Arxiv preprint arXiv:1206.1846*, 2012.

[4] Darren Wilkinson. The pseudo-marginal approach to exact approximate mcmc algorithms. http://darrenjw.wordpress.com/2010/09/20/the-pseudo-marginal-approach-to-exact-approximate-mcmc-algorithms/, 2010.

[5] N.D. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. *Advances in Neural Information Processing Systems*, 16:329–336, 2004.

[6] M. Salzmann, R. Urtasun, and P. Fua. Local deformation models for monocular 3D shape recovery. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 1–8, 2008.

[7] N.D. Lawrence and R. Urtasun. Non-linear matrix factorization with Gaussian processes. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 601–608. ACM, 2009.