Gray-box inference for structured Gaussian process models

Pietro Galliani
SMiLe CLiNiC
University of Sussex

Amir Dezfouli
The University of
New South Wales

Edwin V. Bonilla
The University of
New South Wales

Novi Quadrianto
SMiLe CLiNiC
University of Sussex

Abstract

We develop an automated variational inference method for Bayesian structured prediction problems with Gaussian process (GP) priors and linear-chain likelihoods. Our approach does not need to know the details of the structured likelihood model and can scale up to a large number of observations. Furthermore, we show that the required expected likelihood term and its gradients in the variational objective (ELBO) can be estimated efficiently by using expectations over very low-dimensional Gaussian distributions. Optimization of the ELBO is fully parallelizable over sequences and amenable to stochastic optimization, which we use along with control variate techniques to make our framework useful in practice. Results on a set of natural language processing tasks show that our method can be as good as (and sometimes better than, in particular with respect to expected log-likelihood) hard-coded approaches including SVM-struct and CRFs, and overcomes the scalability limitations of previous inference algorithms based on sampling. Overall, this is a fundamental step to developing automated inference methods for Bayesian structured prediction.

1 INTRODUCTION

Developing automated inference methods for complex probabilistic models has become arguably one of the most exciting areas of research in machine learning, with notable examples in the probabilistic programming community given by STAN (Hoffman and Gelman, 2014) and CHURCH (Goodman et al., 2008). One of the main challenges for these types of approaches is to formulate expressive probabilistic models and develop generic yet efficient inference methods for them. From a variational inference perspective, one particular approach that has addressed such a challenge is the black-box variational inference framework of Ranganath et al. (2014).

While the works of Hoffman and Gelman (2014) and Ranganath et al. (2014) have been successful with a wide range of priors and likelihoods, their direct application to models with Gaussian process (GP) priors is cumbersome, mainly due to the large number of highly coupled latent variables in such models. In this regard, very recent work has investigated automated inference methods for general likelihood models when the prior is given by a sparse Gaussian process (Hensman et al., 2015b; Dezfouli and Bonilla, 2015). While these advances have opened up opportunities for applying GP-based models well beyond regression and classification settings, they have focused on models with i.i.d observations and, therefore, are unsuitable for addressing the more challenging task of *structured prediction*.

Structured prediction refers to the problem where there are interdependencies between outputs and it is necessary to model these dependencies explicitly. Common examples are found in natural language processing (NLP) tasks, computer vision and bioinformatics. By definition, observation models in these problems are not i.i.d and standard learning frameworks have been extended to consider the constraints imposed by structured prediction tasks. Popular structured prediction frameworks are conditional random fields (CRFs; Lafferty et al., 2001), maximum margin Markov networks (Taskar et al., 2004) and structured support vector machines (SVM-struct, Tsochantaridis et al., 2005).

Proceedings of the 20^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54.

From a non-parametric Bayesian modeling perspective, in general, and from a GP modeling perspective, in particular, structured prediction problems present very hard inference challenges because of the rapid explosion of the number of latent variables with the size of the problem. Furthermore, structured likelihood functions are usually very expensive to compute. In an attempt to build non-parametric Bayesian approaches to structured prediction, Bratières et al. (2015) have proposed a framework based on a CRF-type modeling approach with GPs, and use elliptical slice sampling (ESS; Murray et al., 2010) as part of their inference method. Unfortunately, although their method can be applied to linear chain structures in a generic way without considering the details of the likelihood model, it is not scalable as it involves sampling from the full GP prior.

In this paper we present an approach for automated inference in structured GP models with linear chain likelihoods that builds upon the structured GP model of Bratières et al. (2015) and the sparse variational framework of Dezfouli and Bonilla (2015). In particular, we show that the model of Bratières et al. (2015) can be mapped onto a generalization of the automated inference framework of Dezfouli and Bonilla (2015). Unlike the work of Bratières et al. (2015), by introducing sparse GP priors in structured prediction models, our approach is scalable to a large number of observations. More importantly, this approach is also generic in that it does not need to know the details of the likelihood model in order to carry out posterior inference. Finally, we show that our inference method is statistically efficient, as it only requires expectations over low-dimensional Gaussian distributions in order to carry out posterior approximation.

Our experiments on a set of NLP tasks, including noun phrase identification, chunking, segmentation, and named entity recognition, show that our method can be as good as (and sometimes better than, in particular with respect to expected log-likelihood) hard-coded approaches including SVM-struct and CRFs, and overcomes the scalability limitations of previous inference algorithms based on sampling.

We refer to our approach as "gray-box" inference since, in principle, for general structured prediction problems it may require some human intervention. Nevertheless, when applied to fixed structures, our proposed inference method is entirely "black box". For example, as we will see, we can replace the exact likelihood with a pseudo-likelihood without needing to make any other modifications to our code.

2 GAUSSIAN PROCESS MODELS FOR STRUCTURED PREDICTION

Here we are interested in structured prediction problems where we observe input-output pairs $\mathcal{D} = \{\mathbf{X}_n, \mathbf{y}_n\}_{n=1}^{N_{\text{seq}}}$, where N_{seq} is the total number of observations, $\mathbf{X}_n \in \mathcal{X}$ is a descriptor of observation n and $\mathbf{y}_n \in \mathcal{Y}$ is a structured object such as a sequence, a tree or a grid that reflects the interdependences between its individual constituents. Our goal is that of, given a new input descriptor $\mathbf{X}_\star$, predicting its corresponding structured labels $\mathbf{y}_\star$, and more generally, a distribution over these labels.

A fairly general approach to address this problem with Gaussian process (GP) priors was proposed by Bratières et al. (2015) based on CRF-type models, where the distribution of the output given the input is defined in terms of cliques, i.e. sets of fully connected nodes. Such a distribution is given by:

$$p(\mathbf{y}|\mathbf{X}, \mathbf{f}) = \frac{\exp\left(\sum_c f(c, \mathbf{X}_c, \mathbf{y}_c)\right)}{\sum_{\mathbf{y}' \in \mathcal{Y}} \exp\left(\sum_c f(c, \mathbf{X}_c, \mathbf{y}'_c)\right)}, \quad (1)$$

where $\mathbf{X}_c$ and $\mathbf{y}_c$ are tuples of nodes belonging to clique c; $f(c, \mathbf{X}_c, \mathbf{y}_c)$ is their corresponding latent variable; and $\mathbf{f}$ is the collection of all these latent variables, which are assumed to be drawn from a zero-mean GP prior with covariance function $\kappa(\cdot, \cdot; \boldsymbol{\theta})$, with $\boldsymbol{\theta}$ being the hyperparameters. It is clear that such a model is a generalization of vanilla CRFs where the potentials are draws from a GP instead of being linear functions of the features.

2.1 Linear chain structures

In this paper we focus on linear chain structures where the output corresponding to datapoint n is a linear chain of length T_n, whose corresponding constituents stem from a common set. In other words, $\mathbf{X}_n$ is a $T_n \times D$ matrix of feature descriptors and $\mathbf{y}_n$ is a sequence of T_n labels drawn from the same vocabulary $\mathcal{V}$. In this case, in order to completely define the prior over the clique-dependent latent functions in Equation (1), it is necessary to specify covariance functions over the cliques. To this end, Bratières et al. (2015) propose a kernel that is non-zero only when two cliques are of the same type, i.e. both are unary cliques or both are pairwise cliques. Furthermore, these kernels are defined as:

$$\kappa_{\text{un}}((t, \mathbf{x}_t, y_t), (t', \mathbf{x}'_t, y_{t'})) = \mathbb{I}[y_t = y_{t'}]\kappa(\mathbf{x}_t, \mathbf{x}'_t)$$
$$\kappa_{\text{bin}}((y_t, y_{t+1}), (y_{t'}, y_{t'+1})) = \mathbb{I}[y_t = y_{t'} \wedge y_{t+1} = y_{t'+1}],$$

where κ_{un} is the covariance on unary functions and κ_{bin} is the covariance on pairwise functions. With a suitable ordering of these latent functions, we obtain a posterior covariance matrix that is block-diagonal, with the first $|\mathcal{V}|$ blocks corresponding to the unary covariances, each of size T_n; and the last block, corresponding to the pairwise covariances, being a diagonal (identity) matrix of size $|\mathcal{V}|^2$, where $|\mathcal{V}|$ denotes the vocabulary size.

To carry out inference in this model, Bratières et al. (2015) propose a sampling scheme based on elliptical slice sampling (ESS; Murray et al., 2010). In the following section, we show an equivalent formulation of this model that leverages the general class of models with i.i.d likelihoods presented by Nguyen and Bonilla (2014). Understanding structured GP models from such a perspective will allow us to generalize the results of Nguyen and Bonilla (2014); Dezfouli and Bonilla (2015) in order to develop an automated variational inference framework. The advantages of such a framework are that of (i) dealing with generic likelihood models; and (ii) enabling stochastic optimization techniques for scalability to large datasets.

3 FULL GAUSSIAN PROCESS PRIORS AND AUTOMATED INFERENCE

Nguyen and Bonilla (2014), building upon the work of Opper and Archambeau (2009), developed an automated variational inference framework for a class models with Gaussian process priors and generic i.i.d likelihoods. Although such an approach is an important step towards black-box inference with GP priors, assuming i.i.d observations is, by definition, unsuitable for structured models.

One way to generalize such an approach to structured models of the types described in §2.1 is to differentiate between GP priors over latent functions on unary nodes and GP priors over latent functions over pairwise nodes. More importantly, rather than considering i.i.d likelihoods over all observations, we assume likelihoods that factorize over sequences, while allowing for statistical dependences within a sequence. Therefore, our prior model $p(\mathbf{f}) = p(\mathbf{f}^{\text{un}})p(\mathbf{f}^{\text{bin}})$ for linear chain structures decomposes as

$$p(\mathbf{f}) = \left(\prod_{j=1}^{|\mathcal{V}|} \mathcal{N}(\mathbf{f}_j^{\text{un}}; \mathbf{0}, \mathbf{K}_j)\right) \mathcal{N}(\mathbf{f}^{\text{bin}}; \mathbf{0}, \mathbf{K}^{\text{bin}}), \quad (2)$$

where $\mathbf{f}$ is the vector of all latent function values of unary nodes $\mathbf{f}^{\text{un}}$ and the function values of pairwise nodes $\mathbf{f}^{\text{bin}}$. Accordingly, $\mathbf{f}_j^{\text{un}}$ is the vector of unary functions of latent process j, corresponding to the jth label in the vocabulary, which is drawn from a zero-mean GP with covariance function $\kappa_j(\cdot, \cdot; \boldsymbol{\theta}_j)$. This covariance function, when evaluated at all the input pairs in $\{\mathbf{X}_n\}$, induces the $N \times N$ covariance matrix $\mathbf{K}_j$, where $N = \sum_{n=1}^{N_{\text{seq}}} T_n$ is the total number of observations. Similarly, $\mathbf{f}^{\text{bin}}$ is a zero-mean $|\mathcal{V}|^2$-dimensional Gaussian random variable with covariance matrix given by $\mathbf{K}^{\text{bin}}$. We note here that while the unary functions are draws from a GP indexed by $\mathbf{X}$, the distribution over pairwise functions is a finite Gaussian (not indexed by $\mathbf{X}$).

Given the latent function values, our conditional likelihood is defined by:

$$p(\mathbf{y}|\mathbf{f}) = \prod_{n=1}^{N_{\text{seq}}} p(\mathbf{y}_n|\mathbf{f}_n), \quad (3)$$

where, omitting the dependency on the input $\mathbf{X}$ for simplicity, each individual conditional likelihood term is computed using a valid likelihood function for sequential data such as that defined by the structured softmax function in Equation (1); $\mathbf{y}_n$ denotes the labels of sequence n; and $\mathbf{f}_n$ is the corresponding vector of latent (unaries and pairwise) function values.

Theorem 1 *The model class defined by the prior in Equation* (2) *and the likelihood in Equation* (3) *contains the structured* GP *model proposed by Bratières et al. (2015).*

The proof of this is trivial and can be done by (i) setting all the covariance functions of the unary latent process (κ_j) to be the same; (ii) making $\mathbf{K}^{\text{bin}} = \mathbf{I}$; and (iii) using the structured softmax function in Equation (1) as each of the individual terms $p(\mathbf{y}_n|\mathbf{f}_n)$ in Equation (3). This yields exactly the same model as specified by Bratières et al. (2015), with prior covariance matrix with block-diagonal structure described in §2.1 above. ■

The practical consequences of the above theorem is that we can now leverage the results of Nguyen and Bonilla (2014) in order to develop a variational inference (VI) framework for structured GP models that can be carried out without knowing the details of the conditional likelihood. Furthermore, as we shall see in the next section, in order to deal with the intractable nonlinear expectations inherent to VI, the proposed method only requires expectations over low-dimensional Gaussian distributions.

3.1 Automated variational inference

In this section we develop a method for estimating the posterior over the latent functions given the prior and likelihood models defined in Equations (2) and (3). Since the posterior is analytically intractable and the prior involves a large number of coupled latent variables, we resort to approximations given by variational inference (VI; Jordan et al., 1998). To this end, we start by defining our variational approximate posterior distribution:

$$q(\mathbf{f}) = q(\mathbf{f}^{\text{un}})q(\mathbf{f}^{\text{bin}}), \quad \text{for} \tag{4}$$

$$\begin{aligned} q(\mathbf{f}^{\text{un}}) &= \sum_{k=1}^{K} \pi_k q_k(\mathbf{f}^{\text{un}}|\mathbf{b}_k, \mathbf{\Sigma}_k) \\ &= \sum_{k=1}^{K} \pi_k \prod_{j=1}^{|\mathcal{V}|} \mathcal{N}(\mathbf{f}_j^{\text{un}}; \mathbf{b}_{kj}, \mathbf{\Sigma}_{kj}) \quad , \end{aligned} \tag{5}$$

$$q(\mathbf{f}^{\text{bin}}) = \mathcal{N}(\mathbf{f}^{\text{bin}}; \mathbf{m}^{\text{bin}}, \mathbf{S}^{\text{bin}}), \tag{6}$$

where $q(\mathbf{f}^{\text{un}})$ and $q(\mathbf{f}^{\text{bin}})$ are the approximate posteriors over the unary and pairwise nodes respectively; each $q_k(\mathbf{f}_j^{\text{un}}) = \mathcal{N}(\mathbf{f}_j^{\text{un}}; \mathbf{b}_{kj}, \mathbf{\Sigma}_{kj})$ is a N-dimensional full Gaussian distribution; and $q(\mathbf{f}^{\text{bin}})$ is a $|\mathcal{V}|^2$-dimensional Gaussian.

In order to estimate the parameters of the above distribution, variational inference entails the optimization of the so-called evidence lower bound ($\mathcal{L}_{\text{elbo}}$), which can be shown to be a lower bound of the true marginal likelihood, and is composed of a KL-divergence term ($\mathcal{L}_{\text{kl}}$), between the approximate posterior and the prior, and an expected log likelihood term ($\mathcal{L}_{\text{ell}}$):

$$\mathcal{L}_{\text{elbo}} = -\text{KL}(q(\mathbf{f})\|p(\mathbf{f})) + \langle \log p(\mathbf{y}|\mathbf{f}) \rangle_{q(\mathbf{f})}, \tag{7}$$

where the angular bracket notation $\langle \cdot \rangle_q$ indicates an expectation over the distribution q. Although the approximate posterior is an N-dimensional distribution, the expected log likelihood term can be estimated efficiently using expectations over much lower-dimensional Gaussians.

Theorem 2 *For the structured* GP *model defined in Equations* (2) *and* (3)*, the expected log likelihood over the variational distribution defined in Equations* (4) *to* (6) *and its gradients can be estimated using expectations over T_n-dimensional Gaussians and $|\mathcal{V}|^2$-dimensional Gaussians, where T_n is the length of each sequence and $|\mathcal{V}|$ is the vocabulary size.*

The proof is constructive and can be found in the supplementary material. Let $\mathcal{L}_{\text{ell}}^{(k,n)} \stackrel{\text{def}}{=} \langle \log p(\mathbf{y}_n|\mathbf{f}_n) \rangle$ be the individual expected log-likelihood terms; and $\boldsymbol{\lambda}^{\text{un}}$ and $\boldsymbol{\lambda}^{\text{bin}}$ be the variational parameters corresponding to unary and binary factors. Hence we have that $\mathcal{L}_{\text{ell}}$ and its gradients are given by

$$\mathcal{L}_{\text{ell}} = \sum_{n=1}^{N_{\text{seq}}} \sum_{k=1}^{K} \pi_k \mathcal{L}_{\text{ell}}^{(k,n)}, \tag{8}$$

$$\nabla_{\boldsymbol{\lambda}_k^{\text{un}}} \mathcal{L}_{\text{ell}}^{(k,n)} = \left\langle \log p(\mathbf{y}_n|\mathbf{f}_n) \nabla_{\boldsymbol{\lambda}_k^{\text{un}}} \log q_{kn}(\mathbf{f}_n^{\text{un}}) \right\rangle, \tag{9}$$

$$\nabla_{\boldsymbol{\lambda}^{\text{bin}}} \mathcal{L}_{\text{ell}}^{(k,n)} = \left\langle \log p(\mathbf{y}_n|\mathbf{f}_n) \nabla_{\boldsymbol{\lambda}^{\text{bin}}} \log q(\mathbf{f}^{\text{bin}}) \right\rangle, \tag{10}$$

where the expectations are computed wrt the approximate marginal posterior $q_{kn} = q_{kn}(\mathbf{f}_n^{\text{un}})q(\mathbf{f}^{\text{bin}})$; and $q_{kn}(\mathbf{f}_n^{\text{un}})$ is a $(T_n \times |\mathcal{V}|)$-dimensional Gaussian with block-diagonal covariance $\mathbf{\Sigma}_{k(n)}$, each block of size $T_n \times T_n$. Therefore, we can estimate the above terms by sampling from T_n-dimensional Gaussians independently. Furthermore, $q(\mathbf{f}^{\text{bin}})$ is a $|\mathcal{V}|^2$-dimensional Gaussian, which can also be sampled independently. In practice, we can assume that the covariance of $q(\mathbf{f}^{\text{bin}})$ is diagonal and we only sample from univariate Gaussians for the pairwise functions.

It is important to emphasize the practical consequences of Theorem 2. Although we have a fully correlated prior and a fully correlated approximate posterior over $N = \sum_{n=1}^{N_{\text{seq}}} T_n$ unary function values, yielding full N-dimensional covariances, we have shown that for these classes of models we can estimate $\mathcal{L}_{\text{ell}}$ by only using expectations over T_n-dimensional Gaussians. We refer to this result as the *statistical efficiency* of the inference algorithm.

Nevertheless, even when having only one latent function and using a single Gaussian approximation ($K = 1$), optimization of the $\mathcal{L}_{\text{elbo}}$ in Equation (7) is completely impractical for any realistic dataset concerned with structured prediction problems, due to its high memory requirements $\mathcal{O}(N^2)$ and time complexity $\mathcal{O}(N^3)$.

In the next section we will use a sparse GP approach within our variational framework in order to develop a practical algorithm for structured prediction.

4 SPARSE APPROXIMATION

In this section we describe a scalable approach to inference in the structured GP model defined in §3 by introducing the so-called sparse GP approximations (Quiñonero-Candela and Rasmussen, 2005) into our variational framework. Variational approaches to sparse GP models were developed by Titsias (2009) for Gaussian i.i.d likelihoods, then made scalable to large datasets and generalized to non-Gaussian

(i.i.d) likelihoods by Hensman et al. (2015a,b); Dezfouli and Bonilla (2015). The main idea of such approaches is to introduce a set of M *inducing variables* $\{\mathbf{u}_m\}_{m=1}^M$ for each latent process, which lie in the same space as $\{\mathbf{f}_m\}$ and are drawn from the same GP prior. These inducing variables are the latent function values of their corresponding set of *inducing inputs* $\{\mathbf{Z}_m\}$. Subsequently, we redefine our prior in terms of these inducing inputs/variables.

In our structured GP model, only the unary latent functions are drawn from GPs indexed by $\mathbf{X}$. Hence we assume a GP prior over the inducing variables and a conditional prior over the unary latent functions, which both factorize over the latent processes. This yields the joint distribution over unary functions, pairwise functions and inducing variables given by:

$$p(\mathbf{f}, \mathbf{u}) = p(\mathbf{u})p(\mathbf{f}^{\text{un}}|\mathbf{u})p(\mathbf{f}^{\text{bin}}), \tag{11}$$

where the marginal prior over the inducing variables is $p(\mathbf{u}) = \prod_{j=1}^{|\mathcal{V}|} p(\mathbf{u}_j)$; the conditional prior is given by $p(\mathbf{f}^{\text{un}}|\mathbf{u}) = \prod_{j=1}^{|\mathcal{V}|} \mathcal{N}(\mathbf{f}_j^{\text{un}}; \tilde{\boldsymbol{\mu}}_j, \widetilde{\mathbf{K}}_j)$; and the prior over the pairwise functions is defined as before, i.e. $p(\mathbf{f}^{\text{bin}}) = \mathcal{N}(\mathbf{f}^{\text{bin}}; \mathbf{0}, \mathbf{K}^{\text{bin}})$. The means and covariances of the individual conditional distributions over the unary functions are given by: $\tilde{\boldsymbol{\mu}}_j = \mathbf{A}_j\mathbf{u}_j$ and $\widetilde{\mathbf{K}}_j = \kappa_j(\mathbf{X}, \mathbf{X}) - \mathbf{A}_j\kappa(\mathbf{Z}_j, \mathbf{X})$ with $\mathbf{A}_j = \kappa(\mathbf{X}, \mathbf{Z}_j)\kappa(\mathbf{Z}_j, \mathbf{Z}_j)^{-1}$.

By keeping an explicit representation of the inducing variables, our goal is to estimate the joint posterior over the unary functions, pairwise functions and inducing variables given the observed data. To this end, we assume that our variational approximate posterior is given by:

$$q(\mathbf{f}, \mathbf{u}|\boldsymbol{\lambda}) = q(\mathbf{u}|\boldsymbol{\lambda}^{\text{un}})p(\mathbf{f}^{\text{un}}|\mathbf{u})q(\mathbf{f}^{\text{bin}}|\boldsymbol{\lambda}^{\text{bin}}), \tag{12}$$

where $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}^{\text{un}}, \boldsymbol{\lambda}^{\text{bin}}\}$ are the variational parameters; $p(\mathbf{f}^{\text{un}}|\mathbf{u})$ is defined as above; $q(\mathbf{f}^{\text{bin}}|\boldsymbol{\lambda}^{\text{bin}})$ is defined as in Equation (6), i.e. a Gaussian with parameters $\boldsymbol{\lambda}^{\text{bin}} = \{\mathbf{m}^{\text{bin}}, \mathbf{S}^{\text{bin}}\}$; and

$$q(\mathbf{u}|\boldsymbol{\lambda}^{\text{un}}) = \sum_{k=1}^{K} \pi_k q_k(\mathbf{u}|\mathbf{m}_k, \mathbf{S}_k), \tag{13}$$

with $q_k(\mathbf{u}|\mathbf{m}_k, \mathbf{S}_k) = \prod_{j=1}^{|\mathcal{V}|} \mathcal{N}(\mathbf{u}_j; \mathbf{m}_{kj}, \mathbf{S}_{kj})$, $\boldsymbol{\lambda}^{\text{un}} = \{\pi_k, \mathbf{m}_k, \mathbf{S}_k\}$, and $\mathbf{m}_{kj}, \mathbf{S}_{kj}$ denote the posterior mean and covariance of the inducing variables for mixture component k and latent function j.

4.1 Evidence lower bound

The KL term in the evidence lower bound now considers a KL divergence between the joint approximate posterior in Equation (12) and the joint prior in Equation (11). Because of the structure of the approximate posterior, it is easy to show that the term $p(\mathbf{f}^{\text{un}}|\mathbf{u})$ vanishes from the KL (see e.g. Titsias, 2009), yielding an objective function that is composed of a KL between the distributions over the inducing variables; a KL between the distributions over the pairwise functions, and the expected log likelihood over the joint approximate posterior:

$$\begin{aligned}\mathcal{L}_{\text{elbo}}(\boldsymbol{\lambda}) = &- \text{KL}(q(\mathbf{u})\|p(\mathbf{u})) - \text{KL}(q(\mathbf{f}^{\text{bin}})\|p(\mathbf{f}^{\text{bin}})) \\ &+ \left\langle \sum_{n=1}^{N_{\text{seq}}} \log p(\mathbf{y}_n|\mathbf{f}_n) \right\rangle_{q(\mathbf{f},\mathbf{u}|\boldsymbol{\lambda})},\end{aligned} \tag{14}$$

where $\text{KL}(q(\mathbf{f}^{\text{bin}})\|p(\mathbf{f}^{\text{bin}}))$ is a straightforward KL divergence between two Gaussians and $\text{KL}(q(\mathbf{u})\|p(\mathbf{u}))$ is a KL divergence between a mixture of Gaussians and a Gaussian, which we bound using Jensen's inequality. The expressions for these terms are given in the supplementary material.

Let us now consider the expected log-likelihood term in Equation (14), which is an expectation of the conditional likelihood over the joint posterior $q(\mathbf{f}, \mathbf{u}|\boldsymbol{\lambda})$. The following result tells us that, as in the non-sparse case, this term can still be estimated efficiently using expectations over low-dimensional Gaussians.

Theorem 3 *The expected log likelihood term in Equation* (14)*, with a generic structured conditional likelihood* $p(\mathbf{y}_n|\mathbf{f}_n)$ *and variational distribution* $q(\mathbf{f}, \mathbf{u}|\boldsymbol{\lambda})$ *defined in Equation* (12)*, and its gradients can be estimated using expectations over* T_n*-dimensional Gaussians and* $|\mathcal{V}|^2$*-dimensional Gaussians, where* T_n *is the length of each sequence and* $|\mathcal{V}|$ *is the vocabulary size.*

As in the full (non-sparse) case, the proof is constructive and can be found in the supplementary material. This means that, in the sparse case, the expected log likelihood and its gradients can also be computed using Equations (8) to (10), where the mean and covariances of each $q_{kn}(\mathbf{f}_n^{\text{un}})$ are determined by the means and covariances of the posterior over the inducing variables. Thus, as before, $q_{kn}(\mathbf{f}_n^{\text{un}})$ is a $(T_n \times |\mathcal{V}|)$-dimensional Gaussian with block-diagonal structure, where each of the $j = 1, \ldots, |\mathcal{V}|$ blocks has mean and covariance given by

$$\mathbf{b}_{kjn} = \mathbf{A}_{jn}\mathbf{m}_{kj}, \tag{15}$$

$$\boldsymbol{\Sigma}_{kjn} = \widetilde{\mathbf{K}}_j^n + \mathbf{A}_{jn}\mathbf{S}_{kj}\mathbf{A}_{jn}^T \tag{16}$$

where

$$\mathbf{A}_{jn} \stackrel{\text{def}}{=} \kappa(\mathbf{X}_n, \mathbf{Z}_j)\kappa(\mathbf{Z}_j, \mathbf{Z}_j)^{-1} \quad , \tag{17}$$

$$\widetilde{\mathbf{K}}_j^n \stackrel{\text{def}}{=} \kappa_j(\mathbf{X}_n, \mathbf{X}_n) - \mathbf{A}_{jn}\kappa(\mathbf{Z}_j, \mathbf{X}_n) \tag{18}$$

and as mentioned in §2.1, $\mathbf{X}_n$ is the $T_n \times D$ matrix of feature descriptors corresponding to sequence n.

4.2 Expectation estimates

In order to estimate the expectations in Equations (8) to (10), we use a simple Monte Carlo approach where we draw samples from our approximate distributions and compute the empirical expectations. For example, for the $\mathcal{L}_{\text{ell}}$ we have:

$$\widehat{\mathcal{L}}_{\text{ell}} = \frac{1}{S}\sum_{n=1}^{N_{\text{seq}}}\sum_{k=1}^{K}\pi_k\sum_{i=1}^{S}\log p(\mathbf{y}_n|\mathbf{f}_{nki}^{\text{un}}, \mathbf{f}_i^{\text{bin}}), \tag{19}$$

with $\mathbf{f}_{nki}^{\text{un}} \sim \mathcal{N}(\mathbf{b}_{k(n)}, \mathbf{\Sigma}_{k(n)})$ and $\mathbf{f}_i^{\text{bin}} \sim \mathcal{N}(\mathbf{m}^{\text{bin}}, \mathbf{S}^{\text{bin}})$, for $i = 1, \ldots, S$, where S is the number of samples, and each of the individual blocks of $\mathbf{b}_{k(n)}$ and $\mathbf{\Sigma}_{k(n)}$ are given in Equations (15) and (16), respectively. We use a similar approach for estimating the gradients of the $\mathcal{L}_{\text{ell}}$ and they are given in the supplementary material.

5 LEARNING

We learn the parameters of our model, i.e. the parameters of our approximate variational posterior and the hyperparameters ($\{\boldsymbol{\lambda}, \boldsymbol{\theta}\}$) through gradient-based optimization of the variational objective ($\mathcal{L}_{\text{elbo}}$). One of the main advantages of our method is the decomposition of $\mathcal{L}_{\text{ell}}$ in Equation (19) and its gradients as a sum of expectations of the individual likelihood terms for each sequence. This result enables us to use parallel computation and stochastic optimization in order to make our algorithms useful in practice.

In our experiments, we use 500 inducing inputs $\{\mathbf{Z}_j\}$ and select them via K-means clustering. As discussed in the supplementary material, the step sizes for stochastic gradient descent were chosen automatically and adaptively by our code.

5.1 Computational complexity

The time-complexity of our stochastic optimization is dominated by the computation of the posterior's entropy, Gaussian sampling, and running the forward-backward algorithm, which yields an overall cost of $O(M^3 + T_n^3 + ST_n|\mathcal{V}|^2)$ for each sequence n. The space complexity is dominated by storing inducing-point covariances, which is $O(M^2)$. To put this in the perspective of other available methods, the existing Bayesian structured model with ESS sampling (Bratières et al., 2015) has time and memory complexity of $O(N^3)$ and $O(N^2)$ respectively, where N is the total number of observations (e.g. words). CRF's time and space complexity with stochastic optimization depends on the feature dimensionality, i.e. it is $O(D)$. The actual running time of CRF also depends on the cost of model selection via a cross-validation procedure. ESS sampling makes the method of Bratières et al. (2015) completely unfeasible for large datasets and CRF has high running times for problems with high dimensions and many hyperparameters. Our work aims to make Bayesian structured prediction practical for large datasets, while being able to use infinite-dimensional feature spaces as well as sidestepping a costly cross-validation procedure.

5.2 Variance reduction

Our goal is to approximate an expectation of a function $g(\mathbf{f})$ over the random variable $\mathbf{f}$ that follows a distribution $q(\mathbf{f})$, i.e. $\mathbb{E}_q[g(\mathbf{f})]$ via Monte Carlo samples. The simplest way to reduce the variance of the empirical estimator $\bar{g}$ is to subtract from $g(\mathbf{f})$ another function $h(\mathbf{f})$ that is highly correlated with $g(\mathbf{f})$. We note that, in the case of variational inference, this technique was introduced in Blei et al. (2012). In more detail, for any value of $\hat{a}$, the function $\tilde{g}(\mathbf{f}) := g(\mathbf{f}) - \hat{a}h(\mathbf{f})$ will have the same expectation as $g(\mathbf{f})$, i.e. $\mathbb{E}_q[\tilde{g}] = \mathbb{E}_q[g]$, provided that $\mathbb{E}_q[h] = 0$. In general, to ensure unbiasedness, $\mathbb{E}_q[h]$, if easily and efficiently computable, can be subtracted from h to form an estimator $\tilde{g} := g - h + \mathbb{E}_q[h]$. More importantly, as the variance of the new function is $\text{Var}[\tilde{g}] = \text{Var}[g] + \hat{a}^2\text{Var}[h] - 2\hat{a}\text{Cov}[g, h]$, our problem boils down to finding suitable $\hat{a}$ and h so as to minimize $\text{Var}[\tilde{g}]$.

In our case, $q(\mathbf{f})$ is the variational distribution and $g(\mathbf{f}) = \log p(\mathbf{y}_n|\mathbf{f}_n)\nabla_\lambda \log q(\mathbf{f})$ (see supplementary material). Previous work (Ranganath et al., 2014; Dezfouli and Bonilla, 2015) has found that a suitable correction term is given by $h(\mathbf{f}) = \nabla_\lambda \log q(\mathbf{f})$, which has expectation zero. Given this, the optimal $\hat{a}$ can be computed as $\hat{a} = \text{Cov}[g, h]/\text{Var}[h]$. The use of control variates is essential to achieve good performance in our framework.

5.2.1 Piecewise pseudo-likelihood

In order to demonstrate the flexibility of our approach, we also tested the performance of our framework when the true likelihood is approximated by a piecewise pseudo-likelihood (Sutton and McCallum,

Table 1: Datasets used in our experiments. For each dataset we see the number of categories (or vocabulary $|\mathcal{V}|$), the number of features (D), the number of training sequences (N_{seq}), and the average (across folds) number of training words ($\bar{N}$). All numbers refer to the small-scale experiments.

Dataset	$\lvert\mathcal{V}\rvert$	D	N_{seq}	$\bar{N}$
BASE NP	3	6,438	150	3,739.8
CHUNKING	14	29,764	50	1,155.8
SEGMENTATION	2	1,386	20	942
JAPANESE NE	17	102,799	50	1,315.4

2007) that only takes in consideration the local interactions within a single factor between the variables in our model. We emphasize that this change did not require any modification to our inference engine and we simply used this pseudo-likelihood as a drop-in replacement for the exact likelihood.

6 EXPERIMENTS

In this section we evaluate our approach using small-scale experiments on the benchmarks used by Bratières et al. (2015), which target several standard NLP problems and are summarized in Table 1. These include noun phrase identification (BASE NP); chunking, i.e. shallow parsing labels sentence constituents (CHUNKING); identification of word segments in sequences of Chinese ideograms (SEGMENTATION); and Japanese named entity recognition (JAPANESE NE). We also consider larger-scale experiments on BASE NP and CHUNKING, which have significantly more training data available.

6.1 Small-scale experiments

When comparing the error rates on Table 2 we see that our approach is on par with competitive benchmarks which, unlike our method, exploit the structure of the likelihood. More importantly, when analyzing the test likelihoods on Table 3 we see that our method with true likelihood (GP-VAR-T) is significantly better than CRF for all benchmarks except SEGMENTATION, where it has a similar performance. Finally, the log-likelihood results of GP-ESS (Bratières et al., 2015) are also consistently worse than ours, owing largely to the higher computational cost of sampling.

6.2 Larger-scale experiments

Here we evaluate our approach on BASE NP and CHUNKING using $N_{\text{seq}} = 500$ training sequences and the remaining (323) sequences for testing, with a five-fold cross-validation setting. This amounts to roughly $11,611$ words on average. We compare with CRF, as this was the best baseline in our previous experiments. We also note that that the original GP-ESS method is completely impractical in this setting.

On BASE NP, our method has a lower average test log-likelihood (1265.52 vs. 1355.63) but a higher error rate (5.15% vs 4.50%) than CRF. This reinforces our previous message that our method can provide better predictive probabilities than its competitors. However, our results on CHUNKING, 2511.48 vs 1862.96 for test log-likelihoods and 8.60% vs 7.20% for error rates, indicate that our method lags behind CRF on this dataset. We attribute this result to CHUNKING having a much higher dimensionality than BASE NP, which is a more critical issue with large datasets.

7 RELATED WORK

Recent advances in sparse GP models for regression (Titsias, 2009; Hensman et al., 2013) have allowed the applicability of such models to very large datasets, opening opportunities for the extension of these ideas to classification and to problems with generic i.i.d likelihoods (Hensman et al., 2015a; Nguyen and Bonilla, 2014; Dezfouli and Bonilla, 2015; Hensman et al., 2015b). However, none of these approaches is actually applicable to structured prediction problems, which inherently deal with non-i.i.d likelihoods.

Twin Gaussian processes (Bo and Sminchisescu, 2010) address structured continuous-output problems by forcing input kernels to be similar to output kernels. In contrast, here we deal with the harder problem of structured *discrete*-output problems, where one usually requires computing expensive likelihoods during training. The structured continuous-output problem is somewhat related to the area of multi-output regression with GPs for which, unlike discrete structured prediction with GPs, the literature is relatively mature (Álvarez et al., 2010; Álvarez and Lawrence, 2011, 2009; Bonilla et al., 2008).

The original structured Gaussian process model, (GPSTRUCT, Bratières et al., 2015) uses Markov Chain Monte Carlo (MCMC) sampling as the inference method and is not equipped with sparsification techniques that are crucial for scaling to large data. Bratières et al. (2014) have explored a distributed version of GPSTRUCT based on the pseudo-likelihood approximation (Besag, 1975) where several weak learners are trained on subsets of GPSTRUCT's latent variables and bootstrap data. However, within

Table 2: Mean error rates and standard deviations in brackets on small-scale experiments using 5-fold cross-validation. The average number of observed words ($\bar{N}$) on these problems range from 942 to 3740. SVM corresponds to structured support vector machines; CRF to conditional random fields; GP-ESS corresponds to GPSTRUCT with ESS for inference (Bratières et al., 2015); GP-VAR-T corresponds to our method with true likelihood; and GP-VAR-P corresponds to our our method with piecewise pseudo-likelihood.

Dataset	Method				
	SVM	CRF	GP-ESS	GP-VAR-T	GP-VAR-P
BASE NP	5.9 (0.4)	5.3 (0.5)	**5.1 (0.4)**	5.6 (0.5)	5.2 (0.3)
CHUNKING	9.8 (1.0)	**8.5 (1.0)**	**8.5 (1.0)**	9.4 (1.6)	9.0 (1.0)
SEGMENTATION	16.2 (2.2)	15.4 (1.1)	14.9 (1.8)	**14.5 (1.5)**	15.3 (2.2)
JAPANESE NE	5.6 (0.8)	**5.2 (0.7)**	5.6 (0.7)	5.4 (0.6)	5.6 (0.6)

Table 3: Negative expected log-likelihoods and standard deviations in brackets on small-scale experiments using 5-fold cross-validation. As before, CRF refers to conditional random fields; GP-ESS to GPSTRUCT with ESS for inference (Bratières et al., 2015); GP-VAR-T to our method with true likelihood; and GP-VAR-P to our method with piecewise pseudo-likelihood.

Dataset	Method			
	CRF	GP-ESS	GP-VAR-T	GP-VAR-P
BASE NP	944 (835)	887 (57)	622 (34)	**603 (21)**
CHUNKING	517 (113)	704 (116)	**407 (43)**	587 (100)
SEGMENTATION	**253 (41)**	316 (52)	255 (45)	298 (53)
JAPANESE NE	592 (131)	806 (135)	**339 (38)**	411 (94)

each weak learner, inference is still done via MCMC. A variational alternative for GPSTRUCT inference (Srijith et al., 2014, 2016) is also available. However, it relies on pseudo-likelihood approximations and was only evaluated on small-scale problems. Unlike this work, our approach can deal with both pseudo-likelihoods and generic (linear-chain) structured likelihoods, and we rely on our sparse approximation procedure and our automated variational inference technique – rather than on bootstrap aggregation – to achieve good performance on larger datasets.

8 CONCLUSION & DISCUSSION

We studied a Bayesian structured prediction model with GP priors and linear-chain likelihoods. We developed an automated variational inference algorithm that is statistically efficient in that only requires expectations over very low-dimensional Gaussians in order to estimate the expected likelihood term in the variational objective. We exploited these types of theoretical insights as well as practical statistical and optimization tricks to make our inference framework scalable and effective. Our model generalizes recent advances in CRFs (Koltun, 2011) by allowing general positive definite kernels defining their energy functions and opens new directions for combining deep learning with structure models (Zheng et al., 2015).

As mentioned in the introduction, for general structured prediction problems one may need to set up the configuration of the latent functions (e.g. the unary and pairwise functions in the linear-chain case). Thus, the process of developing an inference procedure for a different structure (e.g. when considering higher-order interactions) requires some human intervention. Nevertheless, when applied to fixed structures our approach is "black box" with respect to the choice of likelihood, inasmuch as different likelihoods can be used without any change to the inference engine.

Furthermore, we have already seen in our small-scale experiments a possible way to extend our method to more general structured likelihoods, where the exact likelihood is replaced by a piecewise pseudo-likelihood. Such an approach might be considered for using our framework in models such as grids or skip-chains, for which the evaluation of the true structured likelihood would be intractable.

Overall, we believe our approach is a fundamental step to developing automated inference methods for general structured prediction problems.

References

Mauricio Álvarez and Neil D Lawrence. Sparse convolved Gaussian processes for multi-output regression. In *NIPS*, pages 57–64. 2009.

Mauricio A Álvarez and Neil D Lawrence. Computationally efficient convolved multiple output Gaussian processes. *JMLR*, 12(5):1459–1500, 2011.

Mauricio A. Álvarez, David Luengo, Michalis K. Titsias, and Neil D. Lawrence. Efficient multioutput Gaussian processes through variational inducing kernels. In *AISTATS*, 2010.

Julian Besag. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 24:179–195, 1975.

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.

Liefeng Bo and Cristian Sminchisescu. Twin Gaussian processes for structured prediction. *International Journal of Computer Vision*, 87(1-2):28–52, 2010.

Edwin V. Bonilla, Kian Ming A. Chai, and Christopher K. I. Williams. Multi-task Gaussian process prediction. In *NIPS*. 2008.

Sébastien Bratières, Novi Quadrianto, Sebastian Nowozin, and Zoubin Ghahramani. Scalable Gaussian process structured prediction for grid factor graph applications. In *ICML*, 2014.

Sébastien Bratières, Novi Quadrianto, and Zoubin Ghahramani. GPstruct: Bayesian structured prediction using Gaussian processes. *IEEE TPAMI*, 37:1514–1520, 2015.

Amir Dezfouli and Edwin V Bonilla. Scalable inference for Gaussian process models with black-box likelihoods. In *NIPS*. 2015.

Noah D. Goodman, Vikash K. Mansinghka, Daniel M. Roy, Keith Bonawitz, and Joshua B. Tenenbaum. Church: A language for generative models. In *UAI*, 2008.

James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. In *UAI*, 2013.

James Hensman, Alexander Matthews, and Zoubin Ghahramani. Scalable variational Gaussian process classification. In *AISTATS*, 2015a.

James Hensman, Alexander G Matthews, Maurizio Filippone, and Zoubin Ghahramani. MCMC for variationally sparse Gaussian processes. In *NIPS*. 2015b.

Matthew D. Hoffman and Andrew Gelman. The no-u-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *JMLR*, 15(1):1593–1623, 2014.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. *An introduction to variational methods for graphical models*. Springer, 1998.

Vladlen Koltun. Efficient inference in fully connected CRFs with Gaussian edge potentials. In *NIPS*, 2011.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

Iain Murray, Ryan Prescott Adams, and David J.C. MacKay. Elliptical slice sampling. In *AISTATS*, 2010.

Trung V. Nguyen and Edwin V. Bonilla. Automated variational inference for Gaussian process models. In *NIPS*. 2014.

Manfred Opper and Cédric Archambeau. The variational Gaussian approximation revisited. *Neural computation*, 21(3):786–792, 2009.

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

Rajesh Ranganath, Sean Gerrish, and David M. Blei. Black box variational inference. In *AISTATS*, 2014.

P. K. Srijith, P. Balamurugan, and Shirish Shevade. Efficient variational inference for Gaussian process structured prediction. In *NIPS Workshop on Advances in Variational Inference*, 2014.

P.K. Srijith, P. Balamurugan, and Shirish Shevade. Gaussian process pseudo-likelihood models for sequence labeling. In *ECML-PKDD 2016*, 2016.

Charles Sutton and Andrew McCallum. Piecewise pseudolikelihood for efficient training of conditional random fields. In *ICML*, 2007.

Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-margin Markov networks. In *NIPS*. 2004.

Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *AISTATS*, 2009.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453–1484, December 2005.

Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.