Supplementary Material – Scalable Gaussian Process Structured Prediction for Grid Factor Graph Applications

1 Algorithm complexity

Based on sections 3 and 4 of the main paper, we are in a position to outline the complete algorithm's complexity.

Learning occurs in each weak learner in parallel. For each weak learner t, a one-time setup phase involves computing the kernel matrix from the features, and taking its Cholesky: with $|\mathcal{V}^t|$ subsampled pixels and $|\mathcal{D}_t|$ subsampled images, the kernel computation takes $O(M^2)$, and a Cholesky factorisation takes $O(M^3)$, where M is $|\mathcal{V}^t| \times |\mathcal{D}_t| + |\mathcal{Y}|^2$. Space requirements here are dominated by the storage of the Cholesky matrix. The ESS runtime is dominated by likelihood computations. Each PL computations takes $O(|\mathcal{V}^t||\mathcal{D}_t||\mathcal{Y}|^2)$. In practice the number of required MCMC iterations seems adequate when each WL runs for 12 hours.

Partial prediction (1c in section 4) is carried out on each weak learner, for each image in sequence. Runtime is dominated by the TRW computation for each image, for each sample $\tilde{\mathbf{E}}_t$ obtained by ESS. Space requirements are dominated by storing the marginals for each such sample, and by the train-test and test-test kernel matrices: for a weak learner t with $|\mathcal{V}_{\text{test}}|$ pixels in each test image, these matrices contain $|\mathcal{V}^t||\mathcal{V}_{\text{test}}|$ and $|\mathcal{V}_{\text{test}}|^2$ elements respectively, but they need not be present in memory simultaneously since partial predictions are independent for different images. GPstruct has this in common with other kernel methods that at test time, it needs to evaluate, store and take the Cholesky of these matrices.

Aggregate prediction (stage 2 in section 4) is carried out for each image independently and is the fastest stage, consisting of averaging the marginals over weak learners.

2 Image segmentation: quality of the predicted class posteriors

The models assessed in the main paper produce probabilistic output, therefore not only predictive accuracy, but also the quality of predictions is relevant.

We find that `GPstruct` produces better-calibrated predictions than `CRF LBMO`, which is a state-of-the-art method for semantic segmentation. To assess whether pre-

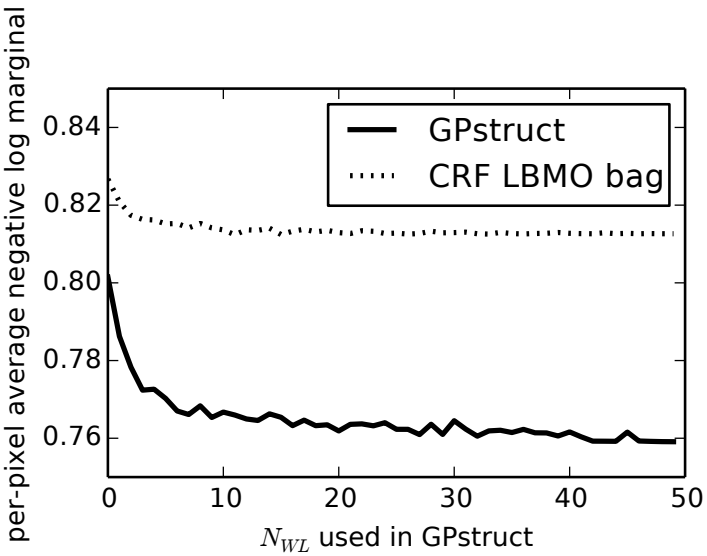

Figure 1: Quality of the predicted class posteriors as measured by the negative log marginal posteriors $-\sum_{\text{pixel positions } i} \log \Pr(\mathbf{y}_i^*|\mathbf{x}^*)$ (smaller is better). The experimental setup is that of the Stanford Background Dataset, with a training set of 50 images. We report per-pixel averages over 5 folds. The size of the weak learner set used for prediction (noted N_{WL}) varies between 1 (a single weak learner is used for prediction) and 50 (the predictions of all the available weak learners are aggregated). For each N_{WL}, the number of weak learner sets which are evaluated and averaged is $\max(5, 50/N_{WL})$.

dictions are well-calibrated, we evaluate a loss function for each prediction, taking into account the marginal probability associated with each pixel's labels. In our present task, no label-dependent loss function is given; therefore we can simply use log-loss and the marginal for the correct prediction, given by the test data labels:

$$-\sum_{\text{pixel positions } i} \log \Pr(\mathbf{y}_i^*|\mathbf{x}^*) \tag{1}$$

The good calibration is due to the "Bayesian" property of the GPstruct model (cf. [1], section 2.3): i.e., uncertainty in the parameters is preserved during training and carried over to the prediction stage. This gives GPstruct an advantage in real-life tasks requiring well-calibrated probabilistic predictions.

3 Image Denoising

In our third experiment, we adapt the experimental protocol of [2]. We create a binary denoising problem using the Berkeley Segmentation Dataset[1]. We use 100 images of

[1] `http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/`

Image Denoising Results

	$\sigma = 2.5$	$\sigma = 1.5$
`independent`	24.4	39.9
`CRF PL`	6.7	12.9
`CRF LBMO`	6.4	11.7
`GPstruct`	6.8	11.9

Table 1: Per-pixel accuracy on the image denoising task. `GPstruct` is comparable to the state-of-the-art method `CRF LBMO`.

size 50×100, binarised pixel-wise with a threshold at the image mean gray level. We use 30 images for training, and 70 for testing. The noisy images are generated as $x_i = y_i(1-t_i^\sigma)+(1-y_i)t_i^\sigma$, where y_i is the true binary label, and $t_i \in [0, 1]$ is a uniform random variable. The variable $\sigma \in (1, \infty)$ is the noise level, where lower values correspond to more noise. We experiment with $\sigma = 2.5$ and $\sigma = 1.5$. Even though this is a toy problem, the ability to systematically vary the noise level is illustrative to highlight the model mis-specification issue [2]. Unary features are $(1, x_i)$. Pairwise features are data independent pairwise Potts factors [3].

`GPstruct` is used with 50 weak learners, and the kernel used for the unary features is the squared exponential kernel described in the paper, while the kernel for the pairwise features is an identity kernel with a scale factor of 0.01. We retain 50 subsampled pixels and their corresponding four neighbours per image in the PL computation.

Results are presented in table 1.

At low noise level ($\sigma = 2.5$), the three methods which incorporate pairwise smoothness terms perform equally well. At lower noise level ($\sigma = 1.5$), the performance of `GPstruct` remains comparable to `CRF LBMO`.

These experimental results confirm that GPstruct, a likelihood-based method, is on par with state-of-the-art marginal optimisation methods, and copes for model mis-specification.

References

[1] Sébastien Bratières, Novi Quadrianto, and Zoubin Ghahramani. Bayesian structured prediction using Gaussian processes. 2013. http://arxiv.org/abs/1307.3846.

[2] Justin Domke. Learning graphical model parameters with approximate marginal inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(10):2454–2467, 2013.

[3] Andrew Blake, Pushmeet Kohli, and Carsten Rother. *Markov Random Fields for Vision and Image Processing*. MIT Press, 2011.