Gaussian Process Density Counting from Weak Supervision

Matthias von Borstel[1], Melih Kandemir[1], Philip Schmidt[1], Madhavi K. Rao[2], Kumar Rajamani[2], Fred A. Hamprecht[1]

1 Heidelberg University, HCI, Heidelberg, Germany
2 Robert Bosch Engineering, Bangalore, India

Abstract. As a novel learning setup, we introduce learning to count objects within an image from only region-level count information. This level of supervision is weaker than earlier approaches that require segmenting, drawing bounding boxes, or putting dots on centroids of all objects within training images. We devise a weakly supervised kernel learner that achieves higher count accuracies than previous counting models. We achieve this by placing a Gaussian process prior on a latent function the square of which is the count density. We impose non-negativeness and smooth the GP response as an intermediary step in model inference. We illustrate the effectiveness of our model on two benchmark applications: i) synthetic cell and ii) pedestrian counting, and one novel application: iii) erythrocyte counting on blood samples of malaria patients.

1 Introduction

Counting objects of interest within an image is a fundamental requirement of many applications. Biologists gain insights on cell population dynamics from such counts, pedestrian counting helps urban planners, and counting cars is crucial for detecting or foreseeing traffic jams.

Traditional approaches to counting proceed by first detecting all targets and then counting them. The transductive principle [1], instead, suggests never to solve a harder problem than the target application necessitates. As a consequence, recent models [2–4] exploit the fact that estimating the object count does not necessarily require accurate detection of individual objects, let alone their segmentation. They focus exclusively on the easier task of assigning each pixel a density in such a way that when the densities within any image region are integrated, a good prediction of the true object count in that region is obtained. This approach is called *density counting* [2].

The main disadvantage of density counting is that it requires a sufficiently large number of per-object annotations. For instance, a common practice in cell counting is to densely annotate a few tens of images by marking the centroids of *all* cells with a dot. This task demands a considerable effort from the annotator. Given the dots, the ground-truth density counts of individual pixels are approximated by placing a Gaussian kernel on top of each dot within an image.

We propose an object counting model that reconciles the density counting approach with weaker supervision. Our model *learns to predict density counts from a set of image regions for each of which only the number of contained objects is known*. Differently from earlier approaches, our model does not require the user to mark where the target objects are within these regions. Figure 1 gives a visual comparison of annotation requirements of our approach and previous work.

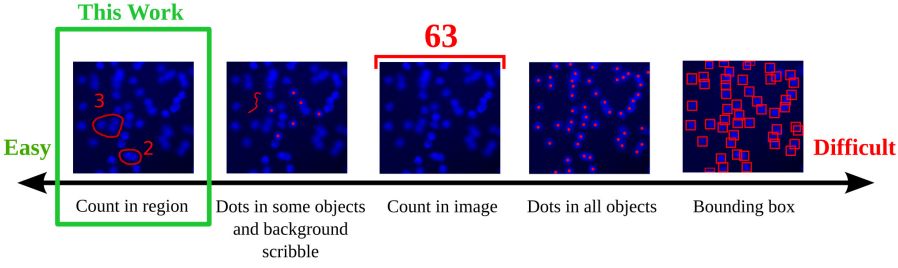

Fig. 1. Comparison of different annotation methods with respect to their difficulty for the annotator. Bounding box annotation is used by [5], dots in *all* objects by [2], dots in some objects by [3, 4], and image-level annotations by [6, 7].

As the learner, we devise a Bayesian model that places a Gaussian process (GP) prior [8] on a latent function whose square is the count density. We impose a smoothness prior on this latent function and assume a Gaussian likelihood that relates the integral over the squared smoothed latent function over a region with the ground-truth count. In addition to facilitating intuitive modeling, the GP prior enables us to employ non-linear kernels on image features, resulting in a model with enhanced expressive power. A welcome feature of our Bayesian approach is that our model produces uncertainty estimates. Finally, we achieve fast and scalable training by sparsify the GP prior and applying stochastic variational inference [9]. Thanks to this scalability, our model is able to operate on individual pixels, rather than superpixels, keeping the model depend loosely on preprocessing.

We evaluate our model on two benchmark data sets: i) cell counting in synthetic fluorescence microscopy images, and ii) pedestrian counting from outdoor video sequences. Additionally, we introduce a novel application for density counting: counting of erythrocytes in blood samples of malaria patients which is useful for diagnostic purposes. In all of these experiments, we observe that the proposed model achieves higher counting accuracies than a large selection of models that are also trained with weak annotations. Our contributions can be summarized as follows. We introduce:

- A new learning setup: Density counting from weak supervision (i.e. region-level counts).
- A novel Bayesian model for weakly supervised density counting with a GP prior on a latent function the square of which is the count density.

– A fast inference algorithm that makes our model usable for pixel-level processing of the input image.
– The first application of density counting to malaria blood cell images.

2 Background

Counting. Approaches to object counting from images can be grouped into two categories: i) counting by detection, and ii) counting by regression. Counting by detection works by first detecting each individual object in an image and then counting the number of detections. In some cases this method is combined with a foregoing segmentation step where each segment is expected to contain one object. This method relies heavily on a good object detector or some other heuristic that identifies regions containing a individual object. These methods work best when the individual objects are clearly distinguishable [10–13].

Counting with regression skips the detection step and infers the count of objects in the image by regression over features associated with individual pixels, an entire image or regions found by a foregoing segmentation step. Segmented regions are allowed to contain multiple objects in this case. This approach is very suitable for cases where the objects are partly occluded or hard to detect individually [6, 7, 14–16]. Lempitsky and Zisserman [2] introduced a third alternative approach for counting objects in images: *density counting*. This method predicts not only an object count for the whole image but a count density for each pixel. Integrating over these pixel count densities in an arbitrary region yields the count of the region. This method has later on been adapted to sparse annotations [3] and has also been used within an interactive model where users could annotate according to feedbacks from the model [4], and finally, has been adapted to deep learning by [17]. These methods have in common that the regression model needs pixel-level density annotations. Providing pixel-level annotations is a tedious task and these methods circumvent this task by applying a density shape assumption around the center of each object specified by the user.

MIR. Our method does not need pixel-level annotations. Instead, it builds on a Multiple Instance Regression (MIR) formulation.[1] In MIR, several regions in the image are annotated with their corresponding counts. The model learns how to assign the pixel-level count densities to obtain the right region counts. The MIR formulation has different modes depending on fundamental assumptions about the structure of the data [19, 18, 20–22]. It either seeks for a prime instance (pixel) in each bag (region) that is responsible for the bag label (region count) or assumes that all instances (pixels) contribute to the bag label (region count). These two modes are called the *prime instance* assumption and the *instance relevance* assumption, respectively. In this work, we adopt the instance relevance assumption, hence, allow each instance label to contribute to the bag label. We then treat the sum of all instance labels as the bag label.

[1] Our definition of MIR differs from that put forward in [18], where a single instance in a bag determines the entire count for the bag.

GP. Gaussian processes are probabilistic kernel learners [23] which apply a prior on the space of functions mapping an input to a continuous output. This prior follows a multivariate normal distribution with a mean $\mu(\mathbf{x})$ and a covariance $k(\mathbf{x}, \mathbf{x}')$ function. It is customary to assume $\mu(\mathbf{x}) = 0$. As for $k(\mathbf{x}, \mathbf{x}')$, any positive definite function can be used. For a data matrix $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]$ with N data points in rows and the corresponding noise-free outputs $\mathbf{f}$, a GP imposes an N-variate normal prior on the mapping function: $\mathbf{f}|\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{XX}})$, where $\mathbf{K}_{\mathbf{XX}}$ is a covariance matrix with entries is $\mathbf{K}_{\mathbf{XX}}[ij] = k(\mathbf{x}_i, \mathbf{x}_j)$ calculated by a positive definite kernel function $k(\cdot, \cdot)$ applied on each pair of feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$.

GPs have been proven useful in many learning setups. GPLVM [28, 33], the generative extension of GPs, attracted widespread attention as an effective non-linear dimensionality reduction tool. Its inference scheme has inspired techniques to scale GPs up to millions of data points [31] and to build alternative deep learning approaches that contain GPs as perceptrons [29]. Finally, GPs improved the state-of-the-art in binary classification from weak labels [24, 25]. We show in this paper that they can be trained by weak supervision for density counting as well. We achieve this by placing a sparse approximation of the GP [26] as a prior on a latent value, the square of which gives the count density of a pixel.

3 Density Counting Setup and Notation

Following the seminal work by Lempitsky and Zisserman [2], we build on the density counting setup, which can formally be defined as follows. Let $\mathbf{I} \in \mathbb{R}^{N_x \times N_y \times N_c}$ be an image of $N_x \times N_y = N$ pixels and N_c channels. We look for a function $g : \mathbf{I} \rightarrow \boldsymbol{\rho}$ that maps this image onto its density map $\boldsymbol{\rho} \in \mathbb{R}_+^N$, such that the sum taken over any region b inside $\boldsymbol{\rho}$ gives the count c_b of target objects in that region: $\sum_{i \in \mathcal{B}_b} \rho_i = c_b$. The task here is to learn a function g that predicts density maps that lead to accurate object counts on all regions. For each pixel i in image $\mathbf{I}$, we extract a feature vector $\mathbf{x}_i \in \mathbb{R}^D$ from the neighbourhood of i and store it as a row in matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$. We use B arbitrarly shaped regions in the image for annotations. One annotation b consists of the set of pixels $\mathcal{B}_b$ that belong to the region b and the count of target objects c_b that reside within that region. The feature vectors of all pixels i that belong to region b are stored as rows in matrix $\mathbf{X}_b \in \mathbb{R}^{N_b \times D}$ and the corresponding object count is element c_b of count vector $\mathbf{c} \in \mathbb{R}_+^B$.

4 Baseline: Counting with Linear Models

A simple model for function g that maps an image $\mathbf{I}$ to its density map $\boldsymbol{\rho}$ would be the linear mapping where the count c_b for each region b is given by

$$c_b = \sum_{i \in \mathcal{B}_b} \rho_i = \sum_{i \in \mathcal{B}_b} \boldsymbol{\omega}^T \mathbf{x}_i = \boldsymbol{\omega}^T \mathbf{X}_b^T \mathbf{1},$$

where $\mathbf{1} = [1, \ldots, 1]^T$ and $\boldsymbol{\omega} \in \mathbb{R}^D$. Several methods exist for learning a parameter vector $\boldsymbol{\omega}$. In [4], a L-2 regularizer with the following objective function is minimized

$$L(\boldsymbol{\omega}; \theta, \varepsilon) = \frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{\omega} + \theta \sum_b \max(0, |\boldsymbol{\omega}^T \mathbf{X}_b^T \mathbf{1} - c_b| - \varepsilon)^2$$

where ε is the allowed divergence from the true count and θ is a regularization parameter. We enhance this model by an additional regularization term to encourage smooth density maps and use as a baseline to motivate the core model proposed in the next section. The resultant objective function is

$$L(\boldsymbol{\omega}, \xi_b; \theta, \varepsilon) = \min_{\boldsymbol{\omega}, \xi_b} \left(\frac{1}{2}\boldsymbol{\omega}^T (\mathbf{X}^T \mathbf{D}^T \mathbf{D} \mathbf{X}) \boldsymbol{\omega} + \theta \sum_b \xi_b \right)$$

subject to:

$$\boldsymbol{\omega}^T \mathbf{X}_b^T \mathbf{1} - c_b - \varepsilon \leq \xi_b, \qquad \boldsymbol{\omega}^T \mathbf{X}_b^T \mathbf{1} - c_b + \varepsilon \geq \xi_b, \qquad \xi_b \geq 0,$$

where $\mathbf{D}\mathbf{X}\boldsymbol{\omega} = \mathbf{D}\boldsymbol{\rho}$ is the first spatial derivative of the density map, and ξ_b are slack variables. The slack variables prevent the model from overfitting to data by allowing a small error on individual data points. This model gives a family of linear count regressors and includes the ridge regression model of [2] as a special case. Furthermore, it improves that model with large margin regularization and density smoothing. As there does not exist any earlier work tailored specifically for weakly supervised density counting, we take as a baseline the weakly-supervised version of a state-of-the-art global count regressor with an added smoothing term.

5 Gaussian Process Multiple Instance Counting

In this section, we describe the proposed GP-based weakly supervised density counting model.

5.1 Core Model

We introduce a novel probabilistic and non-linear model for object counting to address some severe limitations of linear models such as limited flexibility and the possibility of obtaining negative densities in some pixels. Our main contribution is that we place a Gaussian process prior on the latent function $\mathbf{f} \in \mathbb{R}^N$ whose square is the count density $\boldsymbol{\rho}$: $\mathbf{f}|\mathbf{X} \sim \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}_{\mathbf{X}\mathbf{X}})$. We impose the fact that a count is non-negative by assigning a region not the sum of $\mathbf{f}$ but the sum of its element-wise square $c_b = \sum_{i \in \mathcal{B}_b} \rho_i = \sum_{i \in \mathcal{B}_b} f_i^2 = \mathbf{f}_b^T \mathbf{f}_b$, where the index b indicates the part of latent vector $\mathbf{f}$ that belongs to bag b. We make the central assumption that *we only know the counts for image regions* (i.e. a group of pixels).

Following the multiple instance learning terminology, we denote each annotated pixel group as a *bag*. Hence, during training, we are given a group of observations partitioned into bags $\mathbf{X} = \{\mathbf{X}_1 \cup \mathbf{X}_2 \cup \cdots \cup \mathbf{X}_B\}$ with the corresponding bag labels $\mathbf{c} = \{c_1, \cdots, c_B\}$. Note that a bag $\mathcal{B}_b$ is a set of pixels from one or multiple image regions. Put together, the GP prior and the c_b formula above lead to the generative process

$$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}_{\mathbf{XX}}), \qquad p(\mathbf{c}|\boldsymbol{\rho}) = \prod_{b=1}^{N_b} \mathcal{N}(c_b \big| \mathbf{f}_b^T \mathbf{f}_b, \beta^{-1}).$$

The first density is a GP prior on the latent function whose square is the count density map, and the latter performs density counting on the squared latent function $\mathbf{f}$ subject to a small additive measurement noise with precision β. We refer to this novel model as *Gaussian Process Multiple Instance Counting (GPMIC)*.

5.2 Sparsifying the GP

While the GP prior brings the model high expressive power by kernelizing the input patterns, it suffers from the fact that both storage and time complexities of the covariance matrix $\mathbf{K}$ grow quadratically with the total number of unique pixels in the annotated bags. Furthermore, the probability density function of the normal distribution requires inversion of this potentially large matrix. This prevents the above model from generalizing to even modest data set sizes (e.g. a few tens of thousands of instances). We overcome this problem using *Fully Independent Training Conditional (FITC)* [26], a well-known technique to approximate the full GP on vector $\mathbf{f}$ by

$$p(\mathbf{u}|\mathbf{Z}) = \mathcal{N}(\mathbf{u}|\mathbf{0}, \mathbf{K}_{\mathbf{ZZ}}), \qquad p(\mathbf{f}|\mathbf{u}, \mathbf{X}, \mathbf{Z}) = \mathcal{N}(\mathbf{f}|\mathbf{A}\mathbf{u}, \mathbf{B}), \tag{1}$$

where $\mathbf{A} = \mathbf{K}_{\mathbf{ZX}}^T \mathbf{K}_{\mathbf{ZZ}}^{-1}$ and $\mathbf{B} = diag(\mathbf{K}_{\mathbf{XX}} - \mathbf{K}_{\mathbf{ZX}}^T \mathbf{K}_{\mathbf{ZZ}}^{-1} \mathbf{K}_{\mathbf{ZX}})$. Note here that we no longer have to invert the full $\mathbf{K}_{\mathbf{XX}}$ matrix as it never determines the covariance of a normal distribution. Instead, we define a so-called *inducing point* set $\mathbf{Z} \in \mathbb{R}^{P \times D}$ with a much smaller number of data points than $\mathbf{X}$ (so that $P << N$). Then we assign a GP prior from $\mathbf{Z}$ to its so-called *inducing responses* $\mathbf{u}$. We assume that the responses $\mathbf{f}$ (i.e. the vector of latent values whose square gives the count densities of pixels) of our real data set $\mathbf{X}$ are generated as predictions from the GP on this small pseudo data set. The operator $diag(\cdot)$ returns the diagonal elements of the matrix in its argument as a diagonal matrix. Note that this matrix can easily be inverted by taking the reciprocal of its diagonal entries. The resultant model in Equation 1 converts the non-parametric GP into a parametric model that can express the training set only by $\mathbf{u}$ and $\mathbf{Z}$, regardless of its size. This approximation has close ties to the well-known Nyström approximation of kernel matrices [27].

5.3 Smoothing the Density Map

Mapping the pixel (neighborhood) features onto density counts is very prone to produce uninterpretable density maps, since edges, image acquisition arti-

facts, and tiny fluctuations in appearances may lead to larger changes in feature descriptions of pixels than intended. Furthermore, in some cases such as cell counting, the objects of interest may have easily-encodable characteristic shapes. Providing the model with prior knowledge about how the density counts of the neighboring pixels should affect each other would be very desirable. Such information can be plugged into our model very easily.

Given an image $\mathbf{I}$ and its latent function whose square is the count density map $\mathbf{f}$, we are interested in smoothing the GP response, rather than the input. Hence, we convolve $\mathbf{f}$ by a $J \times J$-pixel-sized linear filter $\mathbf{W} \in \mathbb{R}^{J \times J}$. For this we can simply generate a matrix $\mathbf{R}_w$ such that $\mathbf{R}_w\mathbf{f} = \mathbf{F} * \mathbf{W}$, where $\mathbf{F} \in \mathbb{R}^{N_x \times N_y}$ is $\mathbf{f}$ expressed on the input image coordinates and $*$ is the convolution operator that convolves the filter in its right argument on the matrix in its left argument. We define a new latent random vector $\mathbf{g}$, which encodes the smoothed version of $\mathbf{f}$. Remember that $\mathbf{f}$ follows a GP prior, hence we have

$$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K_{XX}}), \qquad p(\mathbf{g}|\mathbf{f}) = \mathcal{N}(\mathbf{g}|\mathbf{R}_w\mathbf{f}, \beta^{-1}\mathbb{I}),$$

where $\mathbb{I}$ is the identity matrix with size determined by the context. To see the effect of smoothing on the GP prior, let us integrate out $\mathbf{f}$:

$$\begin{aligned} p(\mathbf{g}|\mathbf{X}) &= \int p(\mathbf{g}|\mathbf{f})p(\mathbf{f}|\mathbf{X})d\mathbf{f} = \int \mathcal{N}(\mathbf{g}|\mathbf{R}_w\mathbf{f}, \beta^{-1}\mathbb{I})\mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K_{XX}})d\mathbf{f} \\ &= \mathcal{N}(\mathbf{g}|\mathbf{0}, \beta^{-1}\mathbb{I} + \mathbf{R}_w\mathbf{K_{XX}}\mathbf{R}_w^T). \end{aligned} \tag{2}$$

Since $\mathbf{K}$ is a positive semi-definite matrix governed by a proper kernel function, we have $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T\phi(\mathbf{x}')$. Hence, the kernel function projects our input observation $\mathbf{x}$ onto a higher-dimensional Hilbert space with $\phi(\cdot)$. Repeating the same on every data point by the capitalized argument $\mathbf{X}$, we can essentially express the entire $\mathbf{K_{XX}}$ on the Hilbert space as $\phi(\mathbf{X})^T\phi(\mathbf{X})$. Placing this into the covariance matrix in Equation 2 leads to

$$\mathbf{K'_{XX}} = \beta^{-1}\mathbb{I} + \mathbf{R}_w\phi(\mathbf{X})^T\phi(\mathbf{X})\mathbf{R}_w^T.$$

This eventually gives a GP on $\mathbf{g}$ with a new kernel function

$$k'(\mathbf{x}_i, \mathbf{x}_j) = \phi'(\mathbf{x}_i)^T\phi'(\mathbf{x}_j) + \delta_{ij}\beta^{-1}$$

such that $\phi'(\mathbf{X}) = \phi(\mathbf{X}) * \mathbf{W}$. Here, δ_{ij} denotes the Kronecker delta function. Hence, the response of the former $\phi(\cdot)$ is convolved *on the Hilbert space* by the filter $\mathbf{W}$. As we will show in Section 6, smoothing the kernel *response* on the image space leads to more interpretable density counts.

5.4 Final Model

Putting together everything above, our model reads

$$\begin{aligned} p(\mathbf{u}|\mathbf{Z}) &= \mathcal{N}(\mathbf{u}|\mathbf{0}, \mathbf{K_{ZZ}}), & p(\mathbf{f}|\mathbf{u}, \mathbf{X}, \mathbf{Z}) &= \mathcal{N}(\mathbf{f}|\mathbf{A}\mathbf{u}, \mathbf{B}), \\ p(\mathbf{g}|\mathbf{f}) &= \mathcal{N}(\mathbf{g}|\mathbf{R}_w\mathbf{f}, \beta^{-1}\mathbb{I}), & p(\mathbf{c}|\mathbf{g}) &= \prod_{b=1}^{B} \mathcal{N}\Big(c_b \Big| \mathbf{g}_b^T\mathbf{g}_b, \alpha^{-1}\Big), \end{aligned}$$

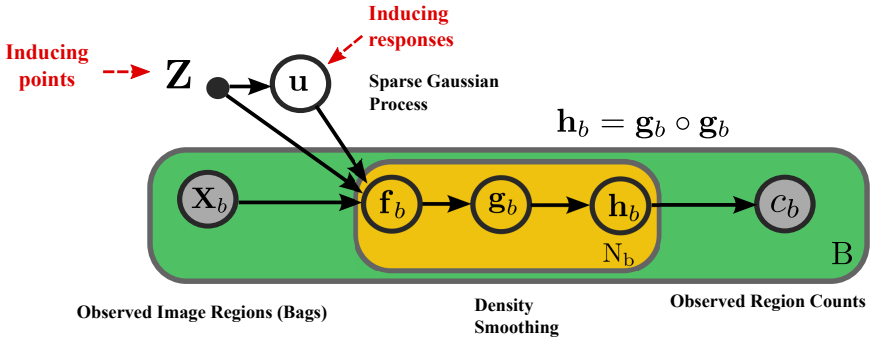

Fig. 2. Plate diagram of the proposed model. For simplicity, the diagram pretends that the annotated regions do not overlap, $\mathcal{B}_b \cap \mathcal{B}'_b = \emptyset$ for $b \neq b'$. Our actual model is not limited by this assumption.

where $\mathbf{g}_b$ denotes the subset of the smoothed pixel latent function whose square is the count density belonging to region b. Figure 2 depicts the plate diagram of the proposed model. Note here that the inner product $\mathbf{g}_b^T \mathbf{g}_b$ equals to the sum of the element-wise square of $\mathbf{g}_b$. With the former, the model performs density counting and with the latter non-negative counts are imposed. Principles of Bayesian statistics suggest integrating out all nuisance variables [30]. In this case, it is possible to integrate out $\mathbf{f}$, which leads to

$$p(\mathbf{g}|\mathbf{u}) = \int p(\mathbf{g}|\mathbf{f})p(\mathbf{f}|\mathbf{u})\mathrm{d}\mathbf{f} = \mathcal{N}(\mathbf{g}|\mathbf{R}_w\mathbf{A}\mathbf{u}, \beta^{-1}\mathbb{I} + \mathbf{R}_w\mathbf{B}\mathbf{R}_w^T).$$

Although integrating out $\mathbf{u}$ is possible [26], this would end up with a non-parametric model, preventing scalable inference. Following [31], we avoid this and keep $\mathbf{u}$ as a global parameter which paves our way towards stochastic variational inference [32].

5.5 Inference

In learning, our goal is to infer the posterior distribution

$$p(\mathbf{g}, \mathbf{u}|\mathbf{X}, \mathbf{c}) = \frac{p(\mathbf{c}, \mathbf{g}, \mathbf{u}|\mathbf{X})}{\int\int p(\mathbf{c}|\mathbf{g})p(\mathbf{g}|\mathbf{u}, \mathbf{X})p(\mathbf{u})d\mathbf{g}d\mathbf{u}} = \frac{p(\mathbf{c}|\mathbf{g})p(\mathbf{g}|\mathbf{u}, \mathbf{X})p(\mathbf{u})}{\int\int p(\mathbf{c}|\mathbf{g})p(\mathbf{g}|\mathbf{u}, \mathbf{X})p(\mathbf{u})d\mathbf{g}d\mathbf{u}}.$$

Since the integral in the denominator is not tractable, this posterior density needs to approximated. As shown in recent studies [31], FITC approximation leads to scalable variational Bayesian inference, which we also adopt in this work. As [33], we assume the variational distribution $Q = p(\mathbf{g}|\mathbf{u}, \mathbf{X}, \mathbf{Z})q(\mathbf{u})$ where $q(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\mathbf{m}, \mathbf{L}^T\mathbf{L})$ and $\mathbf{L} \in \mathbb{R}^{P \times P}$. The decomposition $\mathbf{L}^T\mathbf{L}$ is made to guarantee a positive-semi definite covariance matrix. It is possible to decompose

the marginal likelihood as:

$$\log\ p(\mathbf{c}|\mathbf{X},\mathbf{Z}) = \underbrace{\mathbb{E}_Q[\log p(\mathbf{c}|\mathbf{g})] - \mathbb{KL}(Q||p(\mathbf{u}|\mathbf{Z}))}_{ELBO(\mathcal{L})} + \mathbb{KL}(Q||p(\mathbf{g},\mathbf{u})),$$

where $\mathbb{KL}(\cdot||\cdot)$ denotes Kullback-Leibler (KL) divergence between the two densities in its arguments. The first two terms in this decomposition constitute the *Evidence Lower Bound (ELBO)*. During training, our goal is to maximize the ELBO, or in other words, minimize the KL divergence in the third term by updating the free parameters $\{\mathbf{m}, \mathbf{L}\}$ of our approximate distribution Q. Note that the third term vanishes only when we reach the real posterior. This only provides an asymptotical guarantee. In practice, this term is never exactly zero. After computing all expectations, our ELBO reads

$$\begin{aligned}\mathcal{L}(\mathbf{m},\mathbf{L},\boldsymbol{\theta},\mathbf{Z}) &= \frac{B}{2}\log\alpha + \sum_b \Big(-\frac{\alpha}{2}{c_b}^2 - \frac{\alpha}{2}\mathbb{E}_Q[({\mathbf{g_b}}^T\mathbf{g_b})^2] + \alpha c_b \mathbb{E}_Q[{\mathbf{g_b}}^T\mathbf{g_b}]\Big) \\ &+ \mathbb{E}_{q(\mathbf{u})}[\log p(\mathbf{u}|\mathbf{Z})] + \mathbb{H}[q(\mathbf{u})] - \frac{1}{2}\log\left(\frac{|\mathbf{K_{ZZ}^{-1}}|}{|\mathbf{L}^T\mathbf{L})|}\right) + \frac{dim(\mathbf{m})}{2} \\ =&\alpha\sum_b\Big[-\frac{1}{2}c_b^2 + c_b\left(\mathrm{Tr}\left[\mathbf{G}\right] + \mathbf{h}^T\mathbf{h}\right) - \Big(2\mathrm{Tr}\left[\mathbf{G}^2\right] + 4\mathbf{h}^T\mathbf{G}\mathbf{h} + \left(\mathrm{Tr}\left[\mathbf{G}\right] + \mathbf{h}^T\mathbf{h}\right)^2\Big)\Big] \\ &\frac{B}{2}\log\alpha - \frac{1}{2}\Big[tr\left[\mathbf{K_{ZZ}^{-1}}\mathbf{L}^T\mathbf{L}\right] + \mathbf{m}^T\mathbf{K_{ZZ}^{-1}}\mathbf{m}\Big],\end{aligned}$$

where $\mathbf{G} = \beta^{-1}\mathbb{I} + \mathbf{R}_{wb}\mathbf{B}_b\mathbf{R}_{wb}^T + \mathbf{R}_{wb}\mathbf{A}_b\mathbf{L}^T\mathbf{L}\mathbf{A}_b^T\mathbf{R}_{wb}^T$ and $\mathbf{h} = \mathbf{R}_{wb}\mathbf{A}_b\mathbf{m}$ and $\mathbf{G}^2$ is the element-wise square of $\mathbf{G}$. The subscript b of the variables $\mathbf{g}_b$, $\mathbf{A}_b$ $\mathbf{B}_b$ and $\mathbf{R}_{wb}$ indicates a subset of these vectors or matrices corresponding to the pixels of bag b. Using these region-level values speeds up inference significantly, as the dimensionality of the kernels is now reduced to the size of each individual region. Above, $\mathbb{E}_Q[\cdot]$ denotes the expected value of the argument with respect to the distribution Q, $dim(\cdot)$ returns the dimension of the vector in the argument, and $\mathrm{Tr}[\cdot]$ is the Trace operator. Here, $\mathcal{L}$ depends on the kernel hyperparameters $\boldsymbol{\theta}$, variational parameters $\mathbf{m}$, $\mathbf{L}$, and the inducing points $\mathbf{Z}$, all of which can be learned by gradient updates. To this end, we use stochastic gradient descent updates where we randomly chose one bag during each iteration. This approach is known as stochastic variational inference [32].

5.6 Prediction

Given a new region $\mathbf{X}^*$, the predictive distribution on the density maps is

$$p(\mathbf{g}^*|\mathbf{X}^*) = \int p(\mathbf{c}^*|\mathbf{g}^*)p(\mathbf{g}^*|\mathbf{u},\mathbf{X},\mathbf{Z})p(\mathbf{u},\mathbf{g}|\mathbf{X},\mathbf{c})d\mathbf{u}d\mathbf{g}.$$

We approximate the posterior $p(\mathbf{u},\mathbf{g}|\mathbf{X},\mathbf{c})$ by the distribution learned during training $Q = p(\mathbf{g}|\mathbf{u},\mathbf{X},\mathbf{Z}) \times q(\mathbf{u})$. Both of the integrals here are tractable in closed form, leading to

$$p(\mathbf{g}^*|\mathbf{X}^*) = \mathcal{N}(\mathbf{g}^*|\mathbf{R}_w\mathbf{A}\mathbf{m}, \beta^{-1}\mathbb{I} + \mathbf{R}_w\mathbf{B}\mathbf{R}_w^T).$$

6 Results

6.1 Baselines

Since no method exists that performs density counting from weak and sparse region annotations we adapted some existing methods to this new learning setup and treat as baselines. We also perform a lesion study on major components of our GPMIC. Consequently, we compare against the following models:

- **Linear Model** : This is the model introduced in Section 4, where the density of each pixel is a linear mapping of the feature vector of the pixel $\mathbf{x}_i$ with the parameter vector $\boldsymbol{\omega}$. The count c_b of region b is then given by the sum over the pixel densities. This is, in effect, a multiple instance regression model based on the instance relevance assumption: $c_b = \boldsymbol{\omega}^T \sum_b \mathbf{x}_b$.

- **MIR Cluster Bags** : This model is a variant of the above linear model and the one proposed in [21]. The key assumption of this model is that the bags have an internal structure. Hence, the individual instances in each bag belong to different abstract classes that correspond to clusters in the feature space. The model considers only the instances of one prime class to predict the count. This means the sum over the feature vectors $\mathbf{x}_{b_i}$ that belong to the prime cluster i of bag b is mapped to the count c_b of bag b using the parameter vector $\boldsymbol{\omega}$: $c_b = \boldsymbol{\omega}^T \sum_{b_i} \mathbf{x}_{b_i}$.

- **GPMIC No Square** : This is a simplified version of the proposed GPMIC with the difference that the GP prior is placed on the count density of each pixel instead of a latent value whose square is the count density. The probabilistic process if this baseline reads

$$p(\mathbf{u}|\mathbf{Z}) = \mathcal{N}(\mathbf{u}|\mathbf{0}, \mathbf{K_{ZZ}}), \qquad p(\mathbf{f}|\mathbf{u}, \mathbf{X}, \mathbf{Z}) = \mathcal{N}(\mathbf{f}|\mathbf{Au}, \mathbf{B}),$$
$$p(\mathbf{g}|\mathbf{f}) = \mathcal{N}(\mathbf{g}|\mathbf{R}_w\mathbf{f}, \beta^{-1}\mathbb{I}), \qquad p(\mathbf{c}|\mathbf{g}) = \prod_{b=1}^{B} \mathcal{N}\Big(c_b \Big| \mathbf{1}^T\mathbf{g}_b, \alpha^{-1}\Big).$$

- **GPMIC Unsmoothed**: This is another variant of GPMIC where the smoothing step introduced to enforce spatial smoothness of the density map is omitted. We introduce this baseline to demonstrate the benefit of smoothing. The resultant model is

$$p(\mathbf{u}|\mathbf{Z}) = \mathcal{N}(\mathbf{u}|\mathbf{0},\mathbf{K_{ZZ}}), \qquad p(\mathbf{g}|\mathbf{u}, \mathbf{X}, \mathbf{Z}) = \mathcal{N}(\mathbf{g}|\mathbf{Au}, \mathbf{B}),$$
$$p(\mathbf{c}|\mathbf{g}) = \prod_{b=1}^{B} \mathcal{N}\Big(c_b \Big| \mathbf{g}_b^T\mathbf{g}_b, \alpha^{-1}\Big).$$

- **Bag level histogram**: This method, introduced as a baseline in [2], does not perform *density* counting, as it does not predict pixel-level count densities but instead directly predicts the count of whole images or regions of an

image from a histogram that describes the image or the region. The count c_b of region b is given by a linear map of the histogram vector $\mathbf{h}_b$ with weight vector $\boldsymbol{\omega}$: $c_b = \boldsymbol{\omega}^T \mathbf{h}_b$.

– **Random Forest**: This baseline [3] is an application of random forests to density counting. The model in its vanilla form is trained with strong supervision at the pixel level. These labels are obtained by clicking on the center of an object and then placing a normal distribution with unit norm on this object center pixel to infer the count density for all the neighbouring pixels. To label background regions, users are asked to give a stroke to the background of an image and then all pixels that belong to that stroke are labeled with count density zero. We labeled the same parts of the training images with this method as we annotated for our weakly supervised methods.

– **Convolutional Neural Net (CNN)**: This baseline uses the CNN architecture proposed in [17] for cell counting. We trained the CNN on 50 regions of 40×40-pixels that are strongly annotated using the same Gaussian density prior as the random forest. We use elastic transformations to augment the annotated data. We train the model for 15 epochs with gradually decreasing learning rate. We evaluate this model only on the synthetic cell data set, as the CNN architecture has been specifically tailored for this kind of data.

6.2 Experiments

We evaluate our proposed GP-based weakly supervised density counting model and the baselines described in Section 6.1 on two benchmark tasks: i) synthetic cell counting, ii) pedestrian counting from [2], and one novel task: iii) erythrocyte counting in blood sample slides of malaria patients. The synthetic cell data set consists of a set of simulated fluorescence microscopy images containing round-shaped synthetic cells. The pedestrian counting application is based on a surveillance video of a street where pedestrians walk in two opposite directions. Finally, the Malaria data set consists of microscopy images of erythrocytes that are partly infected by Malaria and partly healthy. For all data sets, the position of the center pixel of each object is provided as ground truth. The general properties of the data sets are show in Table 1 and example images from the data sets are shown in the first column of Figure 3.

Table 1. General properties of the data sets.

Name	# Images	Average Count
Pedestrian	2000	29 ± 9
Synthetic cells	200	171 ± 64
Malaria	78	90 ± 84

For the results reported in Table 2, we use regions annotated by humans. For the pedestrian and the synthetic cells data sets, we use the same features and identical pre- and post-processing procedures as described in [3]. We characterized each pixel by the feature set consisting of Gaussian and Laplacian of Gaussian filters, Gaussian gradient magnitude, and the eigenvalues of structure tensors at scales 0.8, 1.6, and 3.2. Also for the Malaria data set we use the same features as for the synthetic cells data set described in [3] but this time for all three color channels of the RGB images. As the density smoothing kernel $\mathbf{W}$, we use a 11×11-pixels sized Gaussian density normalized to unity on that patch with a variance of four for all three experiments. We use stochastic variational inference to maximize the ELBO with respect to $\mathbf{m}$ and $\mathbf{L}$. To achieve faster convergence we initialize

$$\mathbf{m} \rightarrow \arg\min_{m} \sum_{b} \left(||\mathbf{A_b m}|| - c_b r_1 \right),$$
$$\mathbf{L} \leftarrow \mathbb{I} \cdot 1/r_2,$$

where r_1 and r_2 are scaling parameters that further improve the initialization. We have observed that $r_1 = \bar{P}_b/10$ and $r_2 = 1000$, where $\bar{P}_b$ is the average number of pixels per bag, is a suitable choice.

For the experiment on the synthetic cells data set we take the same approach as described by [2] and use the first 100 images as training set and the last 100 images as test set. We then annotate 5 images from the training set with 14 weak annotations each. On the Pedestrian data set we use five out of the 2000 frames, sparsely annotate ten regions on each frame, and leave the rest for evaluation. On the Malaria data set we use five images with ten weak annotations on each image as training set and the rest of the 78 images as test set. We train the baseline models on the same annotated regions. In Figure 3, we show the density maps obtained by our model and the baselines described in Section 6.1. Table 2 reports the mean average count prediction errors of the models in comparison, averaged over ten runs. GPMIC always gives the best performance in all three data sets, while it is tightly followed on the pedestrian data set by the random forest based density counting baseline. The results also reveal that the Malaria data set is harder than the other two benchmarks. This is understandable since the erythrocytes highly vary in shape when they are infected by malaria and they also overlap heavily no matter if they are healthy or diseased. Figure 3 clearly shows qualitative differences between the different methods. The smoothing step in GPMIC leads to a count density map that better reflects the real shape of the objects. A model without smoothing tends to learn a pronounced edge density in such cases as the Malaria data set. Note that GPMIC No Square allows negative count densities which also occur, as one can see in Figure 3, and lead to worse performance compared to the original GPMIC model as shown in Table 2.

Once strong annotations are provided, it is possible to achieve marginally better results on the synthetic cells and pedestrian data sets as reported in various earlier work [2–4, 7, 14, 17]. However, these models need considerably more annotation effort. Our model trains in 114, 32, and 341 seconds on pedestrian,

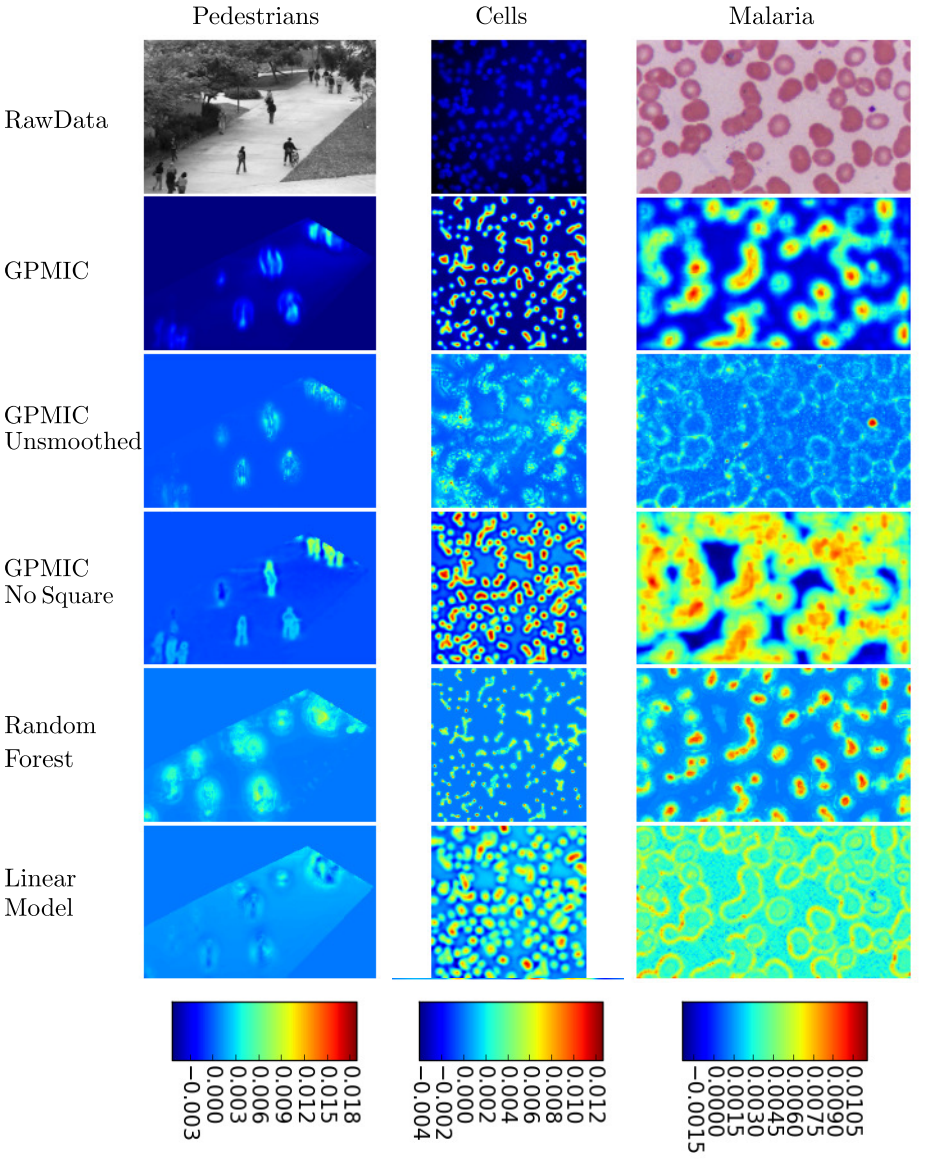

Fig. 3. Example densities for each dataset and all density counting methods. To illustrate the differences better, the Random Forest densities for the Pedestrian and Malaria data set were scaled by a factor of 0.35 and 0.47 respectively.

cells, and malaria data sets, respectively on a machine with 8 GB RAM and 2.3 GHz CPU. The training times for the closest competitor, Random Forest [3], for the same data sets are in the sub-second range. The CNN of [17] de-

signed specifically for the Cells data set trains in 5478 seconds. Consequently, our model achieves better prediction performance than the baselines within reasonable training time in all three applications.

Table 2. Mean Average Errors (MAE) of the models in comparison. The "Dense" column indicates whether a method provides a density map or not. GPMIC (Unsmoothed) and GPMIC (No Square) are more basic versions of the main GPMIC model. As the CNN of Xie et al. [17] is tailored specifically for the synthetic cell data set, we do not use it as a baseline in the remaining two applications.

Regression Model	**Pedestrians**	**Cells**	**Malaria**	**Dense**
Linear Model (Baseline Sec. 4)	21.3	22.1	23.7	Yes
MIR Cluster Bags [21]	4.8	15.5	19.6	No
Bag-level Histogram Linear [2]	10.7	17.4	23.7	No
Random Forest [3]	3.6	10.0	21.1	Yes
Convolutional Neural Net [17]	-	7.8	-	Yes
GPMIC (Unsmoothed)	15.8	21.2	26.2	Yes
GPMIC (No Square)	20.3	8.6	29.9	Yes
GPMIC (This work)	**3.5**	**6.7**	**18.0**	Yes

7 Conclusion

We propose a novel machine learning setup, weakly supervised density counting, and introduce a novel model that gives state-of-the-art performance on this setup. For the first time, we show the usability of GPs as effective prior functions on pixel count densities. This is made possible by building on the recent advances on scalable variational inference, which enables the GP prior to operate at the pixel level. Secondly, we propose an intermediary density smoothing scheme, which proves effective to regularize the count densities and achieve interpretable estimates. Lastly, we show that density counting can be successfully performed on a new medical application: counting blood cells of malaria patients. We believe this outcome to evoke new ideas for a number of clinical use cases.

An alternative way to enforce smooth density maps would be to introduce a normal distributed latent variable $\mathbf{g}$ with the regularized Laplacian precision matrix $\mathbf{Q}$: $p(\mathbf{g}|\mathbf{f}) = \mathcal{N}(\mathbf{g}|\mathbf{f}, \mathbf{Q}^{-1})$. The disadvantage of this formulation is that the inverse of the regularized Laplacian matrix is needed during inference, which is computationally very demanding.

The Bayesian nature of GPMIC allows closed-form calculation of the posterior predictive density, which provides a second-order uncertainty measure (variance) for predictions. This measure can easily be used to build effective interactive learning interfaces using information-theoretic active learning criteria, such as *Bayesian Active Learning by Disagreement (BALD)* [34]. We are encouraged to address such interesting implications of the proposed model in future work.

References

1. Vapnik, V.: Statistical learning theory. Springer (1998)
2. Lempitsky, V., Zisserman, A.: Learning to count objects in images. In: NIPS. (2010)
3. Fiaschi, L., Nair, R., Koethe, U., Hamprecht, F., et al.: Learning to count with regression forest and structured labels. In: ICPR. (2012)
4. Arteta, C., Lempitsky, V., Noble, J., Zisserman, A.: Interactive object counting. In: ECCV. (2014)
5. Ryan, D., Denman, S., Fookes, C., Sridharan, S.: Crowd counting using group tracking and local features. In: AVSS. (2010)
6. Cho, S.Y., Chow, T., Leung, C.T.: A neural-based crowd estimation by hybrid global learning algorithm. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **29**(4) (1999) 535–541
7. Kong, D., Gray, D., Tao, H.: A viewpoint invariant approach for crowd counting. In: ICPR. (2006)
8. Williams, C., Rasmussen, C.: Gaussian processes for machine learning. the MIT Press **2**(3) (2006) 4
9. Hensman, J., Rattray, M., Lawrence, N.: Fast variational inference in the conjugate exponential family. In: NIPS. (2012)
10. Bernardis, E., Stella, X.: Pop out many small structures from a very large microscopic image. Medical image analysis **15**(5) (2011) 690–707
11. Mualla, F., Schöll, S., Sommerfeldt, B., Maier, A., Steidl, S., Buchholz, R., Hornegger, J.: Unsupervised unstained cell detection by sift keypoint clustering and self-labeling algorithm. In: MICCAI. (2014)
12. Arteta, C., Lempitsky, V., Noble, J., Zisserman, A.: Learning to detect cells using non-overlapping extremal regions. In: MICCAI. (2012)
13. Kainz, P., Urschler, M., Schulter, S., Wohlhart, P., Lepetit, V.: You should use regression to detect cells. In: MICCAI. (2015)
14. Ryan, D., Denman, S., Fookes, C., Sridharan, S.: Crowd counting using multiple local features. In: DICTA. (2009)
15. Chan, A., Liang, Z.S., Vasconcelos, N.: Privacy preserving crowd monitoring: Counting people without people models or tracking. In: CVPR. (2008)
16. Chan, A., Vasconcelos, N.: Counting people with low-level features and Bayesian regression. Image Processing, IEEE Transactions on **21**(4) (2012) 2160–2177
17. Xie, W., Noble, J., Zisserman, A.: Microscopy cell counting with fully convolutional regression networks. In: MICCAI. (2015)
18. Wang, Z., Lan, L., Vucetic, S.: Mixture model for multiple instance regression and applications in remote sensing. Geoscience and Remote Sensing, IEEE Transactions on **50**(6) (2012) 2226–2237
19. Ray, S., Page, D.: Multiple instance regression. In: ICML. (2001)
20. Wagstaff, K., Lane, T.: Salience assignment for multiple-instance regression. Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space Administration (2007)
21. Wagstaff, K., Lane, T., Roper, A.: Multiple-instance regression with structured data. In: ICDMW. (2008)
22. Pappas, N., Marconi, R., Popescu-Belis, A.: Explaining the stars: Weighted multiple-instance learning for aspect-based sentiment analysis. In: EMNLP. (2014)
23. Rasmussen, C., Williams, C.: Gaussian processes for machine learning. MIT Press (2006)

24. Kim, M., de la Torre, F.: Gaussian processes multiple instance learning. In: ICML. (2010)
25. Kandemir, M., Zhang, C., Hamprecht, F.: Empowering multiple instance histopathology cancer diagnosis by cell graphs. In: MICCAI. (2014)
26. Snelson, E., Ghahramani, Z.: Local and global sparse Gaussian process approximations. In: AISTATS. (2007)
27. Williams, C., Seeger, M.: Using the Nyström method to speed up kernel machines. In: NIPS. (2001)
28. Lawrence, N.D. Gaussian process latent variable models for visualisation of high dimensional data. In: NIPS. (2004)
29. Lawrence, N.D. Deep Gaussian processes. In: AISTATS. (2013)
30. Gelman, A., Carlin, J., Stern, H., Rubin, D.: Bayesian data analysis. Volume 2. (2014)
31. Hensman, J., Fusi, N., Lawrence, N.: Gaussian processes for big data. arXiv preprint arXiv:1309.6835 (2013)
32. Hoffman, M., Blei, D., Wang, C., Paisley, J.: Stochastic variational inference. The Journal of Machine Learning Research **14**(1) (2013) 1303–1347
33. Titsias, M.K., Lawrence, N.D.: Bayesian Gaussian process latent variable model. In: AISTATS. (2010)
34. Houlsby, N., Huszar, F., Ghahramani, Z., Hernández-Lobato, J.: Collaborative Gaussian processes for preference learning. In: NIPS. (2012)