Bayesian Poisson Regression for Crowd Counting

Antoni B. Chan
Department of Computer Science
City University of Hong Kong
abchan@ucsd.edu

Nuno Vasconcelos
Dept. of Electrical and Computer Engineering
University of California, San Diego
nuno@ece.ucsd.edu

Abstract

Poisson regression models the noisy output of a counting function as a Poisson random variable, with a log-mean parameter that is a linear function of the input vector. In this work, we analyze Poisson regression in a Bayesian setting, by introducing a prior distribution on the weights of the linear function. Since exact inference is analytically unobtainable, we derive a closed-form approximation to the predictive distribution of the model. We show that the predictive distribution can be kernelized, enabling the representation of non-linear log-mean functions. We also derive an approximate marginal likelihood that can be optimized to learn the hyperparameters of the kernel. We then relate the proposed approximate Bayesian Poisson regression to Gaussian processes. Finally, we present experimental results using Bayesian Poisson regression for crowd counting from low-level features.

1. Introduction

Recent work [1, 2] on crowd counting using low-level feature regression has shown promise in computer vision. One advantage with these methods is that they bypass intermediate processing stages, such as people detection or tracking, that may be susceptible to problems when the crowd is dense. In [1], the scene is segmented into crowds moving in different directions and various low-level features are extracted from each crowd segment (*e.g.* information on the shape, edges and texture of the segment). The crowd count in each segment is then estimated with a Gaussian process (GP) regression function that maps feature vectors to the crowd size. Experiments in [1] indicate that the counting algorithm is capable of producing accurate counts, for a wide range of crowd densities.

One problem with the system of [1] is that it uses GP regression, which models *continuous real-valued* functions, to predict *discrete counting numbers*. Because of this mismatch, regression may not be taking full advantage of Bayesian inference. For example, rounding of the real-valued predictions is not handled in a principled way, e.g. by reducing the confidence when the prediction is far from an integer. In addition, the confidence levels are currently measured in standard-deviations, which provides little intuition on the reliability of the estimates. A confidence measure based on posterior probabilities seems more intuitive for counting numbers. Finally, negative outputs of the GP must be truncated to zero, and it is unclear how this affects the optimality of the predictive distribution.

One common method of regression for counting numbers is Poisson regression [3], which models the noisy output of a counting function as a Poisson random variable, with a log-mean parameter that is a linear function of the input vector. This is analogous to standard linear regression, except that the mean is modeled as the exponential of a linear function to ensure non-negative values, and that the noise model is Poisson because the outputs are counting numbers. One way of extending Poisson regression to the Bayesian setting is to adopt a hierarchical model, where the log-mean function is modeled with a standard Gaussian process [4, 5, 6]. These solutions, however, have two disadvantages. First, because of the lack of conjugacy between the Poisson and the GP, [4, 5, 6] must approximate inference with Markov-chain Monte Carlo (MCMC), which limits these algorithms to small datasets. Second, the hierarchical model contains two redundant noise sources: 1) the Poisson-distributed observation noise, and 2) the Gaussian noise of the GP in the log-mean function. These two noise terms model essentially the same thing: the noise in observing the count. A more parsimonious representation would include only the observation noise, while modeling the mean as a deterministic function.

In this work, we analyze the standard Poisson regression model in a Bayesian setting, by adding a Gaussian prior on the weights of the linear log-mean function. Since exact inference is analytically unobtainable, approximate inference is still necessary. However, in contrast to previous work [4, 5, 6], we propose a closed-form approximation to Bayesian inference. The contributions of this paper, with respect to Bayesian Poisson regression (BPR), are five-fold: 1) we derive a closed-form approximation to the predictive distribution for BPR; 2) we kernelize the predictive distribution, enabling the representation of non-linear log-mean functions via kernel functions; 3) we derive an approximate marginal likelihood function for optimizing the hyperparameters of the kernel function with Type-II maximum likelihood; 4) we show that the proposed approximation to BPR is related to a Gaussian process with a special non-i.i.d. noise term; 5) finally, we present experimental results that show improvement in crowd counting accuracy when using the proposed model.

The remainder of this paper is organized as follows. In Sections 2 and 3, we briefly review Gaussian processes and Poisson regression. In Section 4, we present the Bayesian framework for Poisson regression, derive a closed-form approximation for the predictive distribution and marginal likelihood, and kernelize the regression model. Finally, in Section 5, we present experimental results on Bayesian Poisson regression for crowd counting.

2. Gaussian process regression

Gaussian process (GP) regression [7] is a Bayesian treatment for predicting a function value $f(\mathbf{x})$ from the input vector $\mathbf{x} \in \mathbb{R}^d$. Consider the case when $f(\mathbf{x})$ is linear, from which we observe a noisy target y, *i.e.*

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{w}, \qquad y = f(\mathbf{x}) + \epsilon, \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^d$ is the weight vector of the linear model, and the observation noise is Gaussian, $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$. The Bayesian model assumes a prior distribution on the weight vectors, $\mathbf{w} \sim \mathcal{N}(0, \Sigma_p)$, where Σ_p is the covariance matrix of the weight prior.

2.1. Bayesian prediction

Let $X = [\mathbf{x}_1, \cdots \mathbf{x}_N]$ be the matrix of observed input vectors $\mathbf{x}_i$, and $\mathbf{y} = [y_1 \ \cdots \ y_N]^T$ be the vector of observed outputs y_i. Bayesian inference on (1) is based on the posterior distribution of the weights $\mathbf{w}$, conditioned on the observed data $\{X, \mathbf{y}\}$, and is computed with Bayes' rule,

$$p(\mathbf{w}|X, \mathbf{y}) = \frac{p(\mathbf{y}|X, \mathbf{w})p(\mathbf{w})}{\int p(\mathbf{y}|X, \mathbf{w})p(\mathbf{w})d\mathbf{w}}. \tag{2}$$

Since the data-likelihood and weight prior are both Gaussian, (2) is also Gaussian [7],

$$p(\mathbf{w}|X, \mathbf{y}) = G(\mathbf{w}|\frac{1}{\sigma_n^2} A^{-1} X\mathbf{y}, A^{-1}), \tag{3}$$

where $G(x|\mu, \Sigma) = (2\pi)^{-\frac{d}{2}} |\Sigma|^{-\frac{1}{2}} \exp(-\frac{1}{2} \|x - \mu\|_\Sigma^2)$ is the equation of a multivariate Gaussian distribution, $\|x\|_\Sigma^2 = x^T \Sigma^{-1} x$, and $A = \frac{1}{\sigma_n^2} XX^T + \Sigma_p^{-1}$. Finally, given a novel input vector $\mathbf{x}_*$, the predictive distribution of $f_* = f(\mathbf{x}_*)$ is obtained by averaging over all possible model parameterizations, with respect to the posterior distribution of $\mathbf{w}$ [7],

$$\begin{aligned} p(f_*|\mathbf{x}_*, X, \mathbf{y}) &= \int p(f_*|\mathbf{x}_*, \mathbf{w})p(\mathbf{w}|X, \mathbf{y})d\mathbf{w} \qquad (4) \\ &= G(f_*|\frac{1}{\sigma_n^2} \mathbf{x}_*^T A^{-1} X\mathbf{y}, \mathbf{x}_*^T A^{-1} \mathbf{x}_*). \qquad (5) \end{aligned}$$

2.2. Kernelized regression

The predictive distribution in (5) can be rewritten to only depend on the inner products between the inputs $\mathbf{x}_i$. Hence, the "kernel trick" can be applied to obtain a kernel version of the Bayesian linear regression. Consider the model

$$f(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{w}, \tag{6}$$

where $\phi(\mathbf{x})$ is a high-dimensional feature transformation of $\mathbf{x}$ from dimension d to D, *i.e.* $\phi : \mathbb{R}^d \to \mathbb{R}^D$, and $\mathbf{w} \in \mathbb{R}^D$. Substituting into (5) and applying the matrix inversion lemma, the predictive distribution can be rewritten in terms of the kernel function $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \Sigma_p \phi(\mathbf{x}')$ [7]

$$p(f_*|\mathbf{x}_*, X, \mathbf{y}) = G(f_*|\mu_*, \Sigma_*), \tag{7}$$

where the predictive mean and covariance are

$$\begin{aligned} \mu_* &= \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{y}, \qquad (8) \\ \Sigma_* &= k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{k}_*, \qquad (9) \end{aligned}$$

and K is the kernel matrix with entries $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, and $k_* = [k(\mathbf{x}_*, \mathbf{x}_1) \cdots k(\mathbf{x}_*, \mathbf{x}_N)]^T$. Hence, non-linear regression is achieved by adopting different positive definite kernel functions. For example, using a linear kernel,

$$k_l(\mathbf{x}, \mathbf{x}') = \theta_1^2 (\mathbf{x}^T \mathbf{x}' + 1) + \theta_2^2, \tag{10}$$

results in standard Bayesian linear regression, while employing a squared-exponential (RBF) kernel,

$$k_r(\mathbf{x}, \mathbf{x}') = \theta_1^2 e^{-\frac{1}{\theta_2^2} \|\mathbf{x} - \mathbf{x}'\|^2} + \theta_3^2, \tag{11}$$

yields Bayesian regression for locally smooth, infinitely differentiable, functions. Finally, a compound kernel, such as the RBF-RBF kernel,

$$k_{rr}(\mathbf{x}, \mathbf{x}') = \theta_1^2 e^{-\frac{1}{\theta_2^2} \|\mathbf{x} - \mathbf{x}'\|^2} + \theta_3^2 e^{-\frac{1}{\theta_4^2} \|\mathbf{x} - \mathbf{x}'\|^2} + \theta_5^2, \tag{12}$$

which contains two RBF functions with different length scales, can simultaneously model both global non-linear trends and local deviations from the trend.

The hyperparameters θ of the kernel function $k(\mathbf{x}, \mathbf{x}')$ can be learned with Type-II maximum likelihood. The marginal likelihood of the training data $\{\mathbf{x}_i, y_i\}_{i=1}^N$ is maximized with respect to the hyperparameters ([7], Chapter 5)

$$p(\mathbf{y}|X,\theta) = \int p(\mathbf{y}|\mathbf{w},X,\theta)p(\mathbf{w}|\theta)d\mathbf{w} \quad (13)$$

$$= -\frac{1}{2}\mathbf{y}^T K_y^{-1}\mathbf{y} - \frac{1}{2}\log|K_y| - \frac{N}{2}\log 2\pi, \quad (14)$$

where $K_y = K + \sigma_n^2 I$. More details are available in [7].

3. Regression for counting numbers

While the GP provides a Bayesian framework for regressing to real-valued outputs, it is not clear how to use the GP when the outputs are counting numbers, *i.e.* non-negative integers, $y \in \mathbb{Z}_+ = \{0, 1, 2, \cdots\}$. A typical approach to regression for counting functions models the output as a Poisson random variable, where the mean parameter is a function of the input variable. In this section, we review two standard regression methods for counting numbers, Poisson regression and negative binomial regression.

3.1. Poisson regression

Poisson regression [3] models the noisy output y as a Poisson distribution, where the log-mean parameter is a linear function of the input vector $\mathbf{x} \in \mathbb{R}^d$, *i.e.*

$$\lambda(\mathbf{x}) = \mathbf{x}^T\beta, \quad \mu(\mathbf{x}) = e^{\lambda(\mathbf{x})}, \quad y \sim \text{Poisson}(\mu(\mathbf{x})), \quad (15)$$

where $\lambda(\mathbf{x})$ is the log-mean function, $\mu(\mathbf{x})$ is the mean function, $y \in \mathbb{Z}_+$, and $\beta \in \mathbb{R}^d$ is the weight vector. The likelihood of an output y given an input vector $\mathbf{x}$ is $p(y|\mathbf{x},\beta) = \frac{e^{-\mu(\mathbf{x})}\mu(\mathbf{x})^y}{y!}$. The mean and the variance of the predictive distribution are equal, *i.e.* $\mathbb{E}[y] = \text{var}(y) = \mu(\mathbf{x})$, and $\text{mode}(y) = \lfloor \mu(\mathbf{x}) \rfloor$.

Given a matrix of input vectors $X = [\mathbf{x}_1 \cdots \mathbf{x}_N]$ and a vector of outputs $\mathbf{y} = [y_1 \cdots y_N]^T$, the weight vector β can be learned by maximizing the data likelihood, $\log p(\mathbf{y}|X,\beta)$, which is a concave in β. Poisson regression is an example of a generalized linear model [8], which is a general regression framework when the underlying covariates are linear. Generalized kernel machines, and the resulting kernel Poisson regression, were proposed in [9].

3.2. Negative binomial regression

A Poisson random variable is *equidispersed*, *i.e.* the variance is equal to the mean. However, in many cases, the actual random variable is *overdispersed*, with variance greater than the mean, due to additional factors that are not accounted for by the input $\mathbf{x}$ or the model itself. Poisson regression is ill-suited to model overdispersion because it will bias the mean towards the variance, in order to keep the equidispersion property. One popular regression model for overdispersion replaces the Poisson noise with a *negative binomial* [3],

$$\mu(\mathbf{x}) = \exp(\mathbf{x}^T\beta), \qquad y \sim \text{NegBin}(\mu(\mathbf{x}), \alpha), \quad (16)$$

where α is the scale parameter of the negative binomial. The likelihood of y given an input vector $\mathbf{x}$ is

$$p(y|\mathbf{x},\beta,\alpha) = \frac{\Gamma(y+\alpha^{-1})}{\Gamma(y+1)\Gamma(\alpha^{-1})}(p)^{\alpha^{-1}}(1-p)^y, \quad (17)$$

where $p = \frac{\alpha^{-1}}{\alpha^{-1}+\mu(\mathbf{x})}$, and $\Gamma(\cdot)$ is the gamma function. Note that the negative binomial reduces to a Poisson distribution when $\alpha = 0$. The mean, variance, and mode of y are

$$\begin{aligned} \mathbb{E}[y] &= \mu(\mathbf{x}), & (18) \\ \text{var}(y) &= \mu(\mathbf{x})(1+\alpha\mu(\mathbf{x})), & (19) \\ \text{mode}(y) &= \begin{cases} \lfloor (1-\alpha)\mu(\mathbf{x}) \rfloor, & \alpha < 1 \\ 0, & \alpha \geq 1 \end{cases}. & (20) \end{aligned}$$

Hence, for $\alpha > 0$, the negative binomial has variance larger than that of an equivalent Poisson with mean $\mu(\mathbf{x})$. Similar to Poisson regression, the parameters $\{\alpha, \beta\}$ of the negative binomial model can be estimated by maximizing the data log-likelihood (see [3] for more details).

4. Bayesian Poisson regression

Although both Poisson and negative binomial regression provide methods for regressing a counting function, they do not do so in a Bayesian setting, *i.e.* by integrating over the posterior distribution of the weight vector β. In this section, we present a Bayesian regression model for counting functions. We adopt the standard Poisson regression model,

$$\lambda(\mathbf{x}) = \mathbf{x}^T\beta, \quad \mu(\mathbf{x}) = e^{\lambda(\mathbf{x})}, \quad y \sim \text{Poisson}(\mu(\mathbf{x})), \quad (21)$$

and introduce a Gaussian prior on the weight vector, $\beta \sim \mathcal{N}(0, \Sigma_p)$. The posterior distribution of β, given the training data $\{X, \mathbf{y}\}$, is computed with Bayes' rule

$$p(\beta|X,\mathbf{y}) = \frac{p(\mathbf{y}|X,\beta)p(\beta)}{\int p(\mathbf{y}|X,\beta)p(\beta)d\beta}. \quad (22)$$

However, a closed-form expression of (22) is analytically unobtainable because of the lack of conjugacy between the Poisson likelihood and the Gaussian prior. Instead, we will adopt the approximate posterior distribution of [10]. We will then derive a closed-form expression for the predictive distribution and marginal likelihood, using this approximate posterior distribution, and kernelize both quantities.

4.1. Log-gamma approximation

The approximate posterior distribution of [10] is based on approximating the log-gamma distribution with a

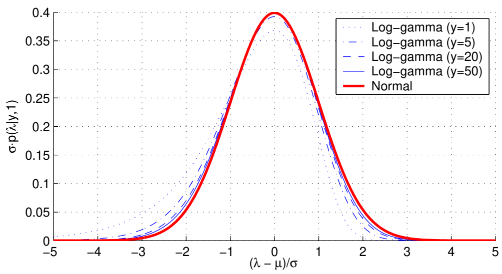

Figure 1. Gaussian approximation of the log-gamma distribution for different values of y. The plot is normalized so that the distributions are zero-mean and unit variance.

Gaussian. Consider a Gamma random variable $\mu \sim \mathrm{Gamma}(a, b)$, with distribution

$$p(\mu|a, b) = \frac{1}{\Gamma(a)b^a}\mu^{a-1}e^{-\frac{\mu}{b}}. \tag{23}$$

The transformed random variable $\lambda = \log \mu$ has a log-gamma distribution. It is well known that, for large a, the log-gamma distribution is approximately Gaussian [11, 12],

$$\lambda = \log \mu \sim \mathcal{N}(\log a + \log b, a^{-1}). \tag{24}$$

Setting $b = 1$ and $a = y \in \mathbb{Z}_+$, (23) becomes

$$p(\mu|y, 1) = \frac{1}{\Gamma(y)}\mu^{y-1}e^{-\mu}. \tag{25}$$

The distribution of λ is obtained with the change of variable formula, leading to the following approximation,

$$p(\lambda|y, 1) = p(\mu = e^{\lambda}|y, 1)\frac{\partial}{\partial\lambda}e^{\lambda} \tag{26}$$

$$= \frac{1}{(y-1)!}e^{\lambda y}e^{-e^{\lambda}} \approx G(\lambda|\log y, y^{-1}). \tag{27}$$

Figure 1 plots the Gaussian approximation of the log-gamma distribution for different values of y. As y increases, the log-gamma converges to the Gaussian approximation.

4.2. Approximate posterior distribution

We now present the approximation to the posterior distribution $p(\beta|X, \mathbf{y})$. The output y is Poisson, and hence the data-likelihood is

$$p(\mathbf{y}|X, \beta) = \prod_{i=1}^{N}\frac{1}{y_i!}\mu(\mathbf{x}_i)^{y_i}e^{-\mu(\mathbf{x}_i)} \tag{28}$$

$$= \prod_{i=1}^{N}\frac{1}{y_i(y_i-1)!}e^{\lambda(\mathbf{x}_i)y_i}e^{-e^{\lambda(\mathbf{x}_i)}}. \tag{29}$$

Using (27), this can be approximated as [10]

$$p(\mathbf{y}|X, \beta) \approx \prod_{i=1}^{N}\frac{1}{y_i}G(\lambda(\mathbf{x}_i)|\log y_i, y_i^{-1}) \tag{30}$$

$$= \frac{|\Sigma_y|^{\frac{1}{2}}}{(2\pi)^{\frac{N}{2}}}e^{-\frac{1}{2}\left\|X^T\beta-\mathbf{t}\right\|^2_{\Sigma_y}}, \tag{31}$$

where $\Sigma_y = \mathrm{diag}([\frac{1}{y_1} \cdots \frac{1}{y_N}])$, and $\mathbf{t} = \log(\mathbf{y})$ is the element-wise logarithm of $\mathbf{y}$. Substituting into (22),

$$\log p(\beta|X, \mathbf{y}) \propto \log p(\mathbf{y}|X, \beta) + \log p(\beta) \tag{32}$$

$$\approx -\frac{1}{2}\left\|X^T\beta - \mathbf{t}\right\|^2_{\Sigma_y} - \frac{1}{2}\left\|\beta\right\|^2_{\Sigma_p}, \tag{33}$$

where we have dropped terms that are not a function of β. Expanding the norm term and completing the square, the posterior distribution is approximately Gaussian,

$$p(\beta|X, \mathbf{y}) \approx G(\beta|\hat{\mu}_\beta, \hat{\Sigma}_\beta), \tag{34}$$

with mean and variance

$$\hat{\mu}_\beta = (X\Sigma_y^{-1}X^T + \Sigma_p^{-1})^{-1}X\Sigma_y^{-1}\mathbf{t}, \tag{35}$$

$$\hat{\Sigma}_\beta = (X\Sigma_y^{-1}X^T + \Sigma_p^{-1})^{-1}. \tag{36}$$

This approximate posterior distribution was originally derived in [10]. In the remainder of this section, we extend [10], by deriving an approximation to the predictive distribution and marginal likelihood for Bayesian Poisson regression, and apply the kernel trick to both quantities.

4.3. Bayesian prediction

Given a novel input $\mathbf{x}_*$, the predictive distribution of the output y_* is obtained by averaging over all possible parameters, with respect to the posterior distribution of β,

$$p(y_*|\mathbf{x}_*, X, \mathbf{y}) = \int p(y_*|\mathbf{x}_*, \beta)p(\beta|X, \mathbf{y})d\beta. \tag{37}$$

Let us define an intermediate random variable $\lambda_* = \mathbf{x}_*^T\beta$. Note that λ_* is a linear transformation of β, and that the posterior distribution of β is approximately Gaussian. Hence, the distribution of λ_* is also approximately Gaussian,

$$p(\lambda_*|\mathbf{x}_*, X, \mathbf{y}) = G(\lambda_*|\hat{\mu}_\lambda, \hat{\sigma}^2_\lambda) \tag{38}$$

where

$$\hat{\mu}_\lambda = \mathbf{x}_*^T(X\Sigma_y^{-1}X^T + \Sigma_p^{-1})^{-1}X\Sigma_y^{-1}\mathbf{t}, \tag{39}$$

$$\hat{\sigma}^2_\lambda = \mathbf{x}_*^T(X\Sigma_y^{-1}X^T + \Sigma_p^{-1})^{-1}\mathbf{x}_*. \tag{40}$$

Finally, we can obtain the predictive distribution by integrating over λ_*,

$$p(y_*|\mathbf{x}_*, X, \mathbf{y}) = \int p(y_*|\lambda_*)p(\lambda_*|\mathbf{x}_*, X, \mathbf{y})d\lambda_* \tag{41}$$

where $p(y_*|\lambda_*) = e^{-(e^{\lambda_*})}(e^{\lambda_*})^{y_*}/y_*!$ is a Poisson distribution. The integral in (41) does not have an analytic solution, and thus an approximation is necessary.

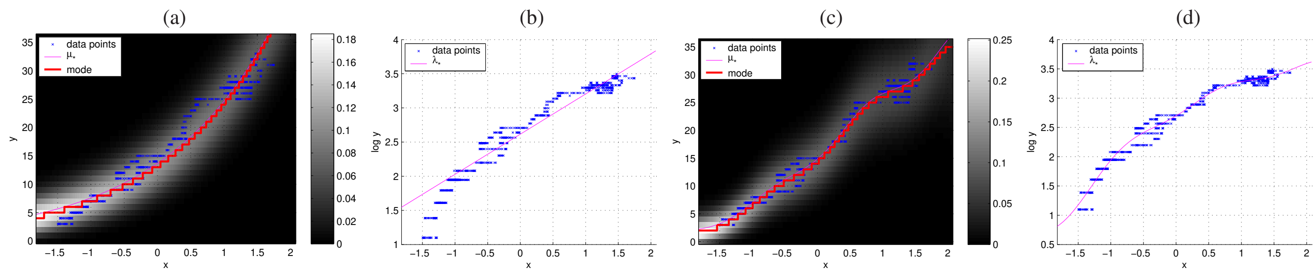

Figure 2. Examples of Bayesian Poisson regression using (a) the linear kernel, and (c) the RBF kernel. The mean parameter $e^{\hat{\mu}_\lambda}$ and the mode are plotted on top of the negative binomial predictive distribution. The corresponding log-mean functions are plotted in (b) and (d).

4.4. Closed-form approximate prediction

To obtain a closed-form approximation to the predictive distribution in (41), we note that we can define a random variable $\mu_* = \exp(\lambda_*)$, and hence $\lambda_* = \log \mu_*$. Since λ_* is approximately Gaussian, we can use (24) to approximate λ_* as a log-gamma random variable, or equivalently μ_* as a gamma random variable, $\mu_* \sim \text{Gamma}(\hat{a}, \hat{b})$, where

$$\hat{a} = \sigma_\lambda^{-2}, \qquad \hat{b} = \sigma_\lambda^2 e^{\hat{\mu}_\lambda}. \tag{42}$$

We can now rewrite the predictive distribution of (41) as the integral over μ_*,

$$p(y_*|\mathbf{x}_*, X, \mathbf{y}) = \int_0^\infty p(y_*|\mu_*) p(\mu_*|\mathbf{x}_*, X, \mathbf{y}) d\mu_*, \tag{43}$$

where $p(y_*|\mu_*) = e^{-\mu_*}\mu_*^{y_*}/y_*!$ is a Poisson distribution, and $p(\mu_*|\mathbf{x}_*, X, \mathbf{y})$ is a gamma distribution. The gamma is the conjugate prior of the Poisson, and thus the integral in (43) can be solved analytically, resulting in a negative binomial distribution [3]

$$p(y_*|\mathbf{x}_*, X, \mathbf{y}) = \frac{\Gamma(\hat{a} + y_*)}{\Gamma(y_* + 1)\Gamma(\hat{a})} (\hat{p})^{\hat{a}} (1 - \hat{p})^{y_*}, \tag{44}$$

where $\hat{p} = \frac{1}{1+\hat{b}} = \frac{\hat{\sigma}_\lambda^{-2}}{\hat{\sigma}_\lambda^{-2} + \exp(\hat{\mu}_\lambda)}$. Hence, the predictive distribution of y_* can be approximated as a negative binomial,

$$y_*|\mathbf{x}_*, X, \mathbf{y} \sim \text{NegBin}(e^{\hat{\mu}_\lambda}, \hat{\sigma}_\lambda^2) \tag{45}$$

with mean and scale parameter computed with (39, 40).

4.5. Kernelized regression

Similar to GP regression, we can extend BPR to represent non-linear log-mean functions using the kernel trick. Given a high-dimensional feature transformation $\phi(\mathbf{x})$, the log-mean function is

$$\lambda(\mathbf{x}) = \phi(\mathbf{x})^T \beta. \tag{46}$$

Rewriting (39, 40) in terms of $\phi(\mathbf{x})$ and applying the matrix inversion lemma, the parameters of the λ_* distribution can be computed using a kernel function,

$$\hat{\mu}_\lambda = \mathbf{k}_*^T (K + \Sigma_y)^{-1} \mathbf{t}, \tag{47}$$

$$\hat{\sigma}_\lambda^2 = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T (K + \Sigma_y)^{-1} \mathbf{k}_*, \tag{48}$$

where $k(\cdot, \cdot)$, K, and $\mathbf{k}_*$ are defined as in Section 2.2. After computing (47, 48), the predictive distribution is still (45).

The hyperparameters θ of the kernel $k(\mathbf{x}, \mathbf{x}')$ can be learned, in a manner similar to the GP, by maximizing the marginal likelihood $p(\mathbf{y}|X, \theta)$. Using the log-gamma approximation in (31), $p(\mathbf{y}|X, \theta)$ is approximated with

$$\log p(\mathbf{y}|X, \theta) \propto -\frac{1}{2}\log|K + \Sigma_y| - \frac{1}{2}\mathbf{t}^T (K + \Sigma_y)^{-1}\mathbf{t}. \tag{49}$$

Figure 2 presents two examples of learning a BPR function, by maximizing the marginal likelihood. Two different kernels were used, the linear kernel and RBF kernel, and the predictive distributions are shown in Figures 2a and 2c, respectively. The corresponding log-mean functions are plotted in Figures 2b and 2d. While the linear kernel can only model an exponential trend in the data, the RBF kernel is capable of adapting to the local deviations in the function.

4.6. Relationship with Gaussian processes

We now relate the proposed approximate Bayesian Poisson regression to Gaussian processes. The equations for the parameters of the approximate λ_* distribution, $\hat{\mu}_\lambda$ and $\hat{\sigma}_\lambda^2$ in (47, 48), are almost identical to those of the GP predictive distribution, μ_* and Σ_* in (8, 9). There are two main differences. First, while the GP noise term in (9) is i.i.d. ($\sigma_n^2 I$), the noise term of BPR in (48) is dependent on the output values ($\Sigma_y = \text{diag}[\frac{1}{y_1} \cdots \frac{1}{y_N}]$). This is a consequence of assuming a Poisson noise model. Second, the predictive mean $\hat{\mu}_\lambda$ in (47) is computed using the log-counts $\mathbf{t}$, rather than the counts $\mathbf{y}$, as with the GP in (8).

Hence, we have the following interpretation of approximate Bayesian prediction for Poisson regression: given observed data $\{X, \mathbf{y}\}$ and novel input $\mathbf{x}_*$, the approximation models the predictive distribution of the log-mean λ_* as a Gaussian process with non-i.i.d. observation noise with covariance $\Sigma_y = \text{diag}([\frac{1}{y_1} \cdots \frac{1}{y_N}])$, learned from the data $\{X, \log \mathbf{y}\}$. Given the distribution of λ_*, the predictive distribution for y_* is a negative binomial with mean $e^{\hat{\mu}_\lambda}$ and scale parameter $\hat{\sigma}_\lambda^2$. Note that the variance of λ_* plays a role as the scale parameter of the negative binomial. Hence, increased uncertainty in estimating λ_* with a GP leads to increased uncertainty in the y_* prediction.

The approximation to the BPR marginal likelihood in (49) differs from that of the GP in (14) in a similar manner as above, and hence we have a similar interpretation. In summary, we have shown that the proposed approximation to BPR is based on assuming a GP prior on the log-mean parameter of the Poisson output distribution. The GP prior uses a special noise term, which approximates the uncertainty that arises from the Poisson noise model. This is in contrast to other methods [4, 5, 6] that assume the standard i.i.d Gaussian noise in the GP prior.

5. Crowd Counting Experiments

In this section, we present experimental results on crowd counting using the proposed Bayesian Poisson regression. We use the crowd video database introduced in [1], which contains 4000 frames of video of a pedestrian walkway with a large number of moving people. The database is annotated with two crowd motion classes, "away" from or "towards" the camera, and the goal is to count the number of people in each motion class. The database was split into a training set of 1200 frames for learning the regression function, and a test set of 2800 frames.

5.1. Experimental Setup

We use the crowd counting system from [1] to compare different regression functions. The crowd was segmented into the two motion classes, using the mixture of dynamic textures [13]. A feature vector, composed of the 29 perspective-normalized features described in [1], was extracted from each crowd segment in each video frame. The feature vectors were normalized so that each dimension had zero mean and unit variance, based on the statistics from the training frames. A Bayesian Poisson regression function was learned, from the training frames, using the linear kernel in (10), and the RBF-RBF kernel in (12), which we denote "BPR-l" and "BPR-rr", respectively. For comparison, a GP regression function was also trained using the linear and RBF-RBF kernels (GPR-l and GPR-rr). A standard linear least-squares regression function and Poisson regression function were also learned.

For BPR, the count estimate is the mode of the predictive distribution. For the GP, the count estimate is obtained by rounding the predictive mean to the nearest non-negative integer. The quality of the count estimates are evaluated with the mean-squared error, $\mathrm{MSE} = \frac{1}{M}\sum_{i=1}^{M}(\hat{c}_i - c_i)^2$, and absolute error, $\mathrm{err} = \frac{1}{M}\sum_{i=1}^{M}|\hat{c}_i - c_i|$, between the count estimate $\hat{c}_i$ and the ground-truth counts c_i, averaged over the M test frames.

5.2. Experimental Results

Table 1 presents the counting error rates for the various regression functions. For Poisson regression, the MSE

Table 1. Comparison of regression functions for crowd counting.

	MSE		err	
Method	away	towards	away	towards
Poisson	3.1518	3.1179	1.3975	1.3750
BPR-l	3.0814	2.0936	1.3700	1.1686
BPR-rr	**2.4675**	**2.0246**	**1.2154**	1.1375
linear	3.3493	2.8718	1.4521	1.3304
GPR-l	3.2786	2.6929	1.4371	1.2800
GPR-rr	3.1725	2.0896	1.4561	**1.1011**

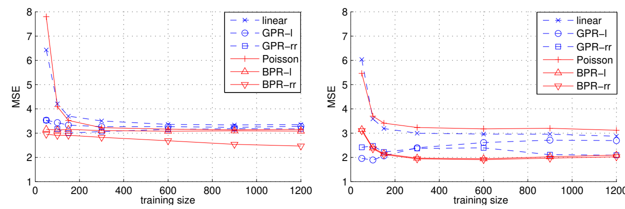

Figure 3. Error rate for training sets of different sizes for the (left) "away" crowd, and (right) "towards" crowd.

improves when using the Bayesian framework, decreasing from $3.152/3.118$ (away/towards) to $3.081/2.094$ for linear BPR. The MSE further decreases to $2.468/2.025$ when non-linear trends in the log-mean function are modeled with the RBF-RBF kernel (BPR-rr). Comparing the two Bayesian regression models with linear kernels, BPR-l outperforms GPR-l on both classes (MSE of $3.081/2.094$ vs. $3.279/2.693$). In the non-linear case, BPR-rr has a significantly lower MSE than GPR-rr on the "away" class (2.468 vs. 3.173), but shows only a slight improvement on the "towards" class (2.025 vs. 2.090). This indicates that BPR is improving the cases where GPR tends to have larger error.

We also measured the test error while varying the size of the training set, by picking a subset of the original training set. Figure 3 plots the MSE versus the training size. Overall, the Bayesian methods (BPR and GPR) are more robust when the training set is small, compared with standard linear or Poisson regression. This indicates that, in practice, a system could be trained with fewer examples, thus reducing the number of images that need to be annotated by hand.

Figure 4 plots the BPR-rr predictions and the true counts for the "away" and "towards" crowds. The predictions track the true counts in most of the test frames, with some errors occuring due to outliers in the video (*e.g.* bicycles and skateboarders). Finally, Figure 5 presents the original image, segmentation, and crowd estimates for several test frames.

6. Conclusions

In this paper, we have proposed an approximation to Bayesian Poisson regression for modeling counting functions. We derived a closed-form approximation to the predictive distribution of the model, and show that the model can be kernelized, enabling the representation of non-linear log-mean functions. We also propose an approximation to

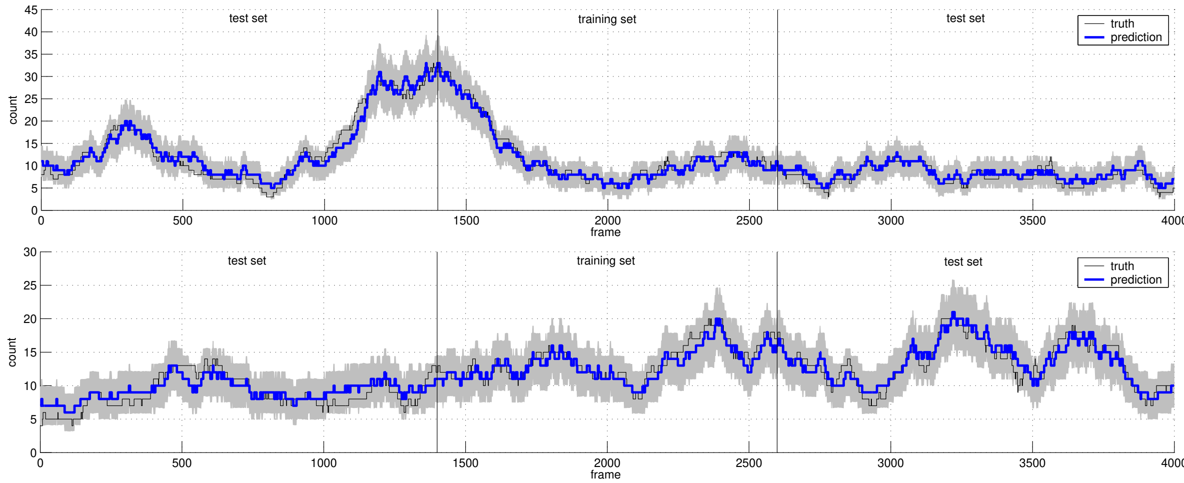

Figure 4. Crowd counting results over both the training and test sets for: (top) "away" crowd, and (bottom) "toward" crowd. The gray bars show the one standard-deviations error bars of the predictive distribution.

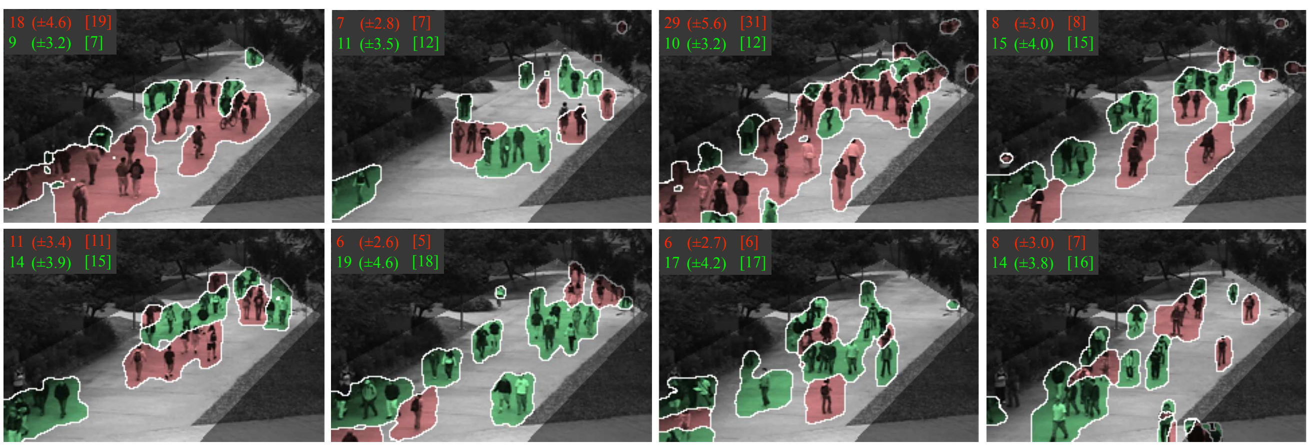

Figure 5. Crowd counting examples: The red and green segments are the "away" and "towards" crowds. The estimated crowd count for each segment is in the top-left, with the (standard-deviation of the Bayesian prediction) and the [ground-truth]. The ROI is also highlighted.

the marginal likelihood, for learning the kernel hyperparameters via type-II maximum likelihood. The proposed approximation is related to a Gaussian process with a special non-i.i.d. noise term that approximates the Poisson output noise. Finally, we apply BPR to feature-based crowd counting, and improve on the results obtained with GPR.

References

[1] A. B. Chan, Z. S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *CVPR*, 2008.

[2] D. Kong, D. Gray, and H. Tao, "Counting pedestrians in crowds using viewpoint invariant training," in *BMVC*, 2005.

[3] A. C. Cameron and P. K. Trivedi, *Regression analysis of count data*. Cambridge Univ. Press, 1998.

[4] P. J. Diggle, J. A. Tawn, and R. A. Moyeed, "Model-based geostatistics," *Applied Statistics*, vol. 47, no. 3, pp. 299–350, 1998.

[5] C. J. Paciorek and M. J. Schervish, "Nonstationary covariance functions for Gaussian process regression," in *NIPS*, 2004.

[6] J. Vanhatalo and A. Vehtari, "Sparse log gaussian processes via MCMC for spatial epidemiology," in *JMLR Workshop and Conference Proceedings*, 2007, pp. 73–89.

[7] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[8] J. A. Nedler and R. W. M. Wedderburn, "Generalized linear models," *J. of the Royal Stat. Society, Series A*, vol. 135, pp. 370–84, 1972.

[9] G. C. Cawley, G. J. Janacek, and N. L. C. Talbot, "Generalised kernel machines," in *Intl. Joint Conf. on Neural Networks*, 2007, pp. 1720–25.

[10] G. M. El-Sayyad, "Bayesian and classical analysis of poisson regression," *J. of the Royal Statistical Society. Series B (Methodological).*, vol. 35, no. 3, pp. 445–51, 1973.

[11] M. S. Bartlett and D. G. Kendall, "The statistical analysis of variance-heterogeneity and the logarithmic transformation," *Supplement to the J. of the Royal Statistical Society*, vol. 8, no. 1, pp. 128–38, 1946.

[12] R. L. Prentice, "A log gamma model and its maximum likelihood estimation," *Biometrika*, vol. 61, no. 3, pp. 539–44, 1974.

[13] A. B. Chan and N. Vasconcelos, "Modeling, clustering, and segmenting video with mixtures of dynamic textures," *IEEE Trans. on PAMI*, vol. 30, no. 5, pp. 909–926, May 2008.