Bayesian Model Adaptation for Crowd Counts

Bo Liu   Nuno Vasconcelos
University of California, San Diego
La Jolla, CA 92093
boliu@eng.ucsd.edu, nuno@ece.ucsd.edu

Abstract

The problem of transfer learning is considered in the domain of crowd counting. A solution based on Bayesian model adaptation of Gaussian processes is proposed. This is shown to produce intuitive model updates, which are tractable, and lead to an adapted model (predictive distribution) that accounts for all information in both training and adaptation data. The new adaptation procedure achieves significant gains over previous approaches, based on multi-task learning, while requiring much less computation to deploy. This makes it particularly suited for the problem of expanding the capacity of crowd counting camera networks. A large video dataset for the evaluation of adaptation approaches to crowd counting is also introduced. This contains a number of adaptation tasks, involving information transfer across video collected by 1) a single camera under different scene conditions (different times of the day) and 2) video collected from different cameras. Evaluation of the proposed model adaptation procedure in this dataset shows good performance in realistic operating conditions.

1. Introduction

The problem of crowd counting [4] has recently received significant attention in computer vision. Given video of a crowded environment, the goal is to estimate the density of the crowd, by counting the number of people that it contains. Various methods have shown that the problem is solvable with fairly high accuracy [4, 5, 14, 21, 23]. In fact, state of the art results place the prediction error at around ± 1 person per video frame, for crowds with dozens of pedestrians. While this is sufficient for most applications of practical interest, the scalability problem remains open. Most works assume a large annotated training set per camera view. This is not practical for large camera networks, where crowd counting systems are most useful.

The scalability goal makes crowd counting a prime candidate for transfer learning [15]. This consists of sharing information (or models) across camera views, so as to minimize the amount of manual supervision *per view*. Like transfer learning in general, transfer learning methods for crowd counting can be of several types. They include *semi-supervised learning* (SSL) to account for unlabeled video in the training process, *multi-task learning* (MTL) to share a model across camera views, or *model adaptation,* to apply a model trained on a camera view to counting on another.

In principle, all of these are of interest to crowd counting. For example, [13] proposed a system that elegantly combines SSL and MTL with active learning and manifold learning to share information across camera views. The boundaries between different types of transfer learning are also loose. For example, 2-task MTL can transfer information between two views in a manner similar to model adaptation. There are, nevertheless, important differences. While adaptation subsumes the notion of a source and a target view, with asymmetric amounts of training data, this is usually not the case for MTL, where all sources usually have equal amounts of data, or for SSL, where the asymmetry is between labeled and unlabeled data. In result, model adaptation, which *adapts* an existing model to a *small* set of unseen data, tends to be computationally less intensive than MTL or SSL, which *learn* from *all* data.

These issues are of particular concern to this work, which addresses the problem of *expanding the capacity* of a crowd counting camera network, by adding cameras to an already installed system. This is usually done to expand the system *footprint,* i.e. increase the area of a scene that it covers, or its *resolution*, e.g. by adding views of areas where counting already takes place. Increased resolution is desirable when certain scene features, e.g. building entries or landmarks, justify detailed crowd counts. Capacity expansion has two constraints: it should require 1) little human effort, in the form of *small amounts of labeled data per added camera view,* and 2) little computational effort, by *requiring minimal model re-training* (ideally none). These make the problem more suited for model adaptation, due to its smaller labeling and computation requirements, than SSL or MTL.

In this work, we introduce a model adaptation procedure for the popular Gaussian process (GP) counting model of [19]. This procedure leverages the Bayesian nature of GPs, which supports the interpretation of the source model as a prior and the adaptation dataset as a set of observations. The two components can then be combined into a predictive distribution that captures the entire information in *both* the source and adaptation data. In this sense, the Bayesian formulation provides guarantees for the optimality of the adapted model that are not available for other approaches, e.g. gradient descent procedures that consider the original model a starting point for the optimization. The predictive distribution also provides a complete characterization of the uncertainty of the model predictions. For GP models, it is a Gaussian whose variance can be computed and acts as a confidence score for the predictions. Finally, the GP formulation is shown to enable *kernel adaptation without retraining*. We show that this is a major advantage over previous MTL formulations of the transfer problem, which are much more costly and have weaker transfer performance.

This theoretical contribution is complemented by the introduction of a large video dataset for the evaluation of count transfer. This dataset is unique in that it includes video from a network of several cameras, which cover different views of a sizeable outdoor environment. The video was collected to address both the footprint and resolution aspects of capacity expansion, including a mix of overlapping and non-overlapping camera views. In the model adaptation context, it tests the transfer of counts across 1) identical camera views under different crowd densities and imaging conditions (e.g. video collected at different times of the day), and 2) different camera views. The dataset includes a total of $27,000$ video frames and will be made publically available, from the author's website, upon publication of this work. A protocol is also introduced for the evaluation of count transfer and used to compare the proposed method to previous approaches. This comparison provides substantial evidence for the benefits of model adaptation.

2. Related work

Many methods have been proposed for people counting. Although object detection has been used for counting [12, 18], it tends to work only for low-density crowds. For dense crowds, with severe occlusions and few pixels per person, more attention has been devoted to feature-based methods, which directly map image features to crowd counts. Two main approaches have evolved. Region of interest (ROI) methods estimate the number of people in a region [4, 21, 23], line of interest (LOI) methods the number of people crossing a virtual gate [5, 14].

In this work, we consider ROI methods. These perform a preliminary segmentation of the scene into regions or blobs, extract features from each region, and use a feature regression to estimate the number of people per region [4, 23, 21, 13]. The regression function can be linear, e.g. least squares [21], or non-linear, e.g. based on GPs [4], manifold learning [13], or neural networks [23]. We adopt a GP, for its ability to account for crowd-counting non-linearities and its support for transfer learning by model adaptation. Since the complexity of transfer learning is a major factor in the cost of capacity expansion, model adaptation is a better solution to this problem than SSL or MTL.

Model adaptation has a long history in speech recognition, where recognizers learned from a large speech corpus are adapted to a new user or environment, e.g. a phone connection [20, 24]. In computer vision, adaptation has been proposed to bridge gap between camera views [8], data modalities [6, 16, 17], image conditions [22], or even object classes [10]. Recently, model adaptation has been used in the deep learning literature, to adapt a model learned from the Imagenet corpus [9] to other tasks [7]. Many of these approaches require specific classes of models or classification/regression architectures. We adopt a Bayesian formulation of the adaptation problem because this is well matched to GP-based regression and has tractable complexity.

The few works that have, so far, considered transfer learning for crowd counting have relied heavily on SSL. [23] presents an SSL procedure that uses sequential information in unlabeled frames to penalize sudden prediction changes. [13] proposes a system that combines SSL and MTL with active and manifold learning to share information between camera views. These approaches are feasible (albeit expensive) when there is unlimited time to jointly train a number of cameras. They are, however, less practical for capacity expansion, where they would require the solution of a complex optimization problem (neural net learning or manifold-regularized regression) whenever a camera is added to the network. More related to the solution now proposed is prior work on transfer learning for GPs [1, 11, 2]. While these are mostly MTL methods, they can be used as model adaptation procedures, albeit at some computational cost (retraining of kernel parameters). A detailed discussion of these approaches, and how they compare to the proposed method, is given in Section 4.5.

3. Crowd Counting

We start by briefly reviewing the GP formulation and then introduce the proposed adaptation procedure.

3.1. Counting as Gaussian process regression

GP-based crowd counting [4] formulates counting as a regression problem, where a count y is predicted by a real-valued function $f(\mathbf{x})$ of a vector $\mathbf{x} \in \mathbb{R}^d$ of d image features, according to

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

where $\phi(\mathbf{x}) \in \mathbb{R}^D$ is a high-dimensional embedding of $\mathbf{x}$, $\mathbf{w}$ a vector of regression parameters sampled from a prior Gaussian distribution $\mathbf{w} \sim \mathcal{N}(0, \Sigma_p)$, and $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ independent Gaussian noise.

Given a training sample $X = [\mathbf{x_1}, ..., \mathbf{x_N}]$, whose columns are feature observations, and the count vector $\mathbf{y} = [y_1, ..., y_N]^T$, $\mathbf{w}$ has Gaussian posterior distribution

$$\begin{aligned} p(\mathbf{w}|X, \mathbf{y}) &= \frac{p(\mathbf{y}|X, \mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|X)} && (2) \\ &= G(\mathbf{w}, \mu_{\mathbf{w}}, \Sigma_{\mathbf{w}}) && (3) \end{aligned}$$

where

$$\begin{aligned} \mu_{\mathbf{w}} &= \frac{1}{\sigma_n^2}\left(\frac{1}{\sigma_n^2}\Phi\Phi^T + \Sigma_p^{-1}\right)^{-1}\Phi\mathbf{y} && (4) \\ \Sigma_{\mathbf{w}} &= \left(\frac{1}{\sigma_n^2}\Phi\Phi^T + \Sigma_p^{-1}\right)^{-1} && (5) \end{aligned}$$

and $\Phi \in \mathbb{R}^{D \times N}$ is a matrix of columns $\phi(\mathbf{x}_i)$. The predictive distribution for the count y_* of a novel input $\mathbf{x}_*$ is

$$p(y_*|\mathbf{x}_*, X, \mathbf{y}) = G(y_*, \mu(\mathbf{x}_*), \sigma(\mathbf{x}_*)) \qquad (6)$$

with

$$\begin{aligned} \mu(\mathbf{x}) &= \phi(\mathbf{x})^T \mu_{\mathbf{w}} && (7) \\ \sigma(\mathbf{x}) &= \sigma_n^2 + \phi(\mathbf{x})^T \Sigma_{\mathbf{w}} \phi(\mathbf{x}). && (8) \end{aligned}$$

By introducing a kernel

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \Sigma_p \phi(\mathbf{x}'), \qquad (9)$$

(4)-(5) and (7)-(8) can be simplified into [4]

$$\begin{aligned} \mu(\mathbf{x}_*) &= \mathbf{k}_*{}^T(K + \sigma_n^2 I)^{-1}\mathbf{y} && (10) \\ \sigma(\mathbf{x}_*) &= k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*{}^T(K + \sigma_n^2 I)^{-1}\mathbf{k}_*, && (11) \end{aligned}$$

where $\mathbf{k}_* = \mathbf{k}(\mathbf{x}_*)$,

$$\mathbf{k}(\mathbf{x}) = [k(\mathbf{x}, \mathbf{x_1}), \ldots, k(\mathbf{x}, \mathbf{x_N})]^T, \qquad (12)$$

and K is the matrix of entries $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. Since K is available after training, the computation of the count prediction of (10) and confidence score of (11) basically reduces to computing the inner products of $\mathbf{k}_*$.

3.2. Learning

Rather than explicitly learning Σ_p and Φ, [4] used the standard trick of defining a parametric kernel $k(.,.)$ and learning its parameters. They used the RBF-RBF kernel

$$k(\mathbf{x}_i, \mathbf{x}_j) = \theta_1^2 e^{-\frac{1}{2\theta_2^2}||\mathbf{x}_i - \mathbf{x}_j||^2} + \theta_3^2 e^{-\frac{1}{2\theta_4^2}||\mathbf{x}_i - \mathbf{x}_j||^2}, \qquad (13)$$

where the first RBF term has a large scale parameter and models the overall trends in the data, while the second has a smaller scale and models local nonlinearities. We adopt this kernel and, following [4], estimate the kernel hyperparameters θ_i by maximizing the marginal likelihood

$$\begin{aligned} \log p(\mathbf{y}|X; \theta) &= -\frac{1}{2}\mathbf{y}^T(K + \sigma_n^2 I)^{-1}\mathbf{y} \\ &- \frac{1}{2}\log|K + \sigma_n^2 I| - \frac{N}{2}\log 2\pi \quad (14) \end{aligned}$$

using an optimization procedure proposed in [19].

4. Model Adaptation for crowd counting

In this section, we introduce the proposed Bayesian model adaptation procedure.

4.1. Bayesian formulation

The goal of adaptation is to transfer a model learned from a *source* view to a *target* view, using a small amount of additional training data, known as the *adaptation dataset.* Assume that, after learning the kernel parameters from *source data* $(X, \mathbf{y})$, we observe a small amount of *adaptation* data $X^+ = [\mathbf{x}_1^+, ..., \mathbf{x}_M^+], \mathbf{y}^+ = [y_1^+, ..., y_M^+]^T$, where $M \ll N$. GPs are amenable to a Bayesian treatment of this problem. If 1) the parameter $\mathbf{w}$ captures all information about the original training data $(X, \mathbf{y})$, i.e. $p(y^+|X^+, \mathbf{w}, y, X) = p(y^+|X^+, \mathbf{w})$, and 2) only the observation of counts y^+ alters its distribution, i.e. $p(\mathbf{w}|X^+, y, X) = p(\mathbf{w}|y, X)$, the updated posterior is

$$p(\mathbf{w}|X^+, \mathbf{y}^+, X, \mathbf{y}) \propto p(\mathbf{y}^+|X^+, \mathbf{w})p(\mathbf{w}|X, \mathbf{y}). \qquad (15)$$

Note that (15) is similar to (2), with the prior $p(\mathbf{w})$ replaced by the posterior $p(\mathbf{w}|X, y)$. Hence, the model learned from $(X, \mathbf{y})$ can be considered a prior for the adaptation process and the adaptation data can be seen as a new set of observations. This is a hallmark of Bayesian inference, which makes it a very natural solution for problems, such as model adaptation, where information collected from different observations of a stochastic process must be fused.

Given the updated posterior, it is possible to compute the updated predictive distribution

$$\begin{aligned} & p(y_*^+|\mathbf{x}_*^+, X^+, \mathbf{y}^+, X, \mathbf{y}) \qquad (16) \\ &= \int p(y_*^+|\mathbf{x}_*^+, \mathbf{w})p(\mathbf{w}|X^+, \mathbf{y}^+, X, \mathbf{y})d\mathbf{w}. \end{aligned}$$

This is the *updated model,* e.g. a GP that predicts crowd counts y_*^+ for new observations $\mathbf{x}_*^+$ under the new setting. Note that the updated model accounts for *all* data, from both training and adaptation datasets. The goal of Bayesian model adaptation is to compute this distribution.

4.2. Parameter posterior

We start by discussing the updated parameter posterior of (15). Given the equivalence between (15) and (2), it follows from (3) that

$$p(\mathbf{w}|X^+, \mathbf{y}^+, X, \mathbf{y}) \quad = \quad G(\mathbf{w}, \mu_{\mathbf{w}}^+, \Sigma_{\mathbf{w}}^+) \qquad (17)$$

where $\mu_{\mathbf{w}}^+$ and $\Sigma_{\mathbf{w}}^+$ are the *updated posterior mean and covariance,* respectively. An important difference between (15) and (2) is that, unlike $p(\mathbf{w})$, $p(\mathbf{w}|X, \mathbf{y})$ is not a zero mean distribution. This prevents the direct application of Section 3.1 to the adaptation scenario. In Appendix A of the supplementary materials, we show that the adapted posterior has hyper-parameters

$$\mu_{\mathbf{w}}^+ \quad = \quad \mu_{\mathbf{w}} + \frac{1}{\sigma_n^2}\Gamma_{\mathbf{w}}^{-1}\Phi^+\mathbf{z}^+ \qquad \Sigma_{\mathbf{w}}^+ = \Gamma_{\mathbf{w}}^{-1} \qquad (18)$$

where

$$\begin{aligned} \Gamma_{\mathbf{w}} &= \frac{1}{\sigma_n^2}\Phi^+\Phi^{+T} + \Sigma_{\mathbf{w}}^{-1} & (19)\\ \Phi^+ &= [\phi(\mathbf{x}_1^+), \ldots, \phi(\mathbf{x}_M^+)] & (20)\\ \mathbf{z}^+ &= \mathbf{y}^+ - \Phi^{+T}\mu_{\mathbf{w}} & (21) \end{aligned}$$

and $\mu_{\mathbf{w}}$ and $\Sigma_{\mathbf{w}}$ are as in (4) and (5), respectively.

This is the posterior distribution for $\mathbf{w}$, given *both* the training and adaptation data. Note that the mean prediction is identical to that of the original model ($\mu_{\mathbf{w}}$) plus a *correction* based on the adaptation data. From (7) and (21), the expected value of this correction, given by (18), is a function of the errors $y_i^+ - \mu(\mathbf{x}_i^+)$ of the original model on the adaptation observations $\mathbf{x}_i^+$. Hence, the model will stay approximately unchanged if it makes accurate predictions on the adaptation set. On the other hand, drastic prediction errors will produce a significantly different model.

4.3. Predictive distribution

In Appendix B of the supplement, we show that the predictive distribution of (16) is

$$p(y_*^+|\mathbf{x}_*^+, X^+, \mathbf{y}^+, X, \mathbf{y}) = G(y_*^+, \mu^+(\mathbf{x}_*^+), \sigma^+(\mathbf{x}_*^+))$$

with

$$\begin{aligned} \mu^+(\mathbf{x}_*^+) &= \mu(\mathbf{x}_*^+) + \mathbf{k}_*^{+T}[K^+ + \sigma_n^2 I]^{-1}\mathbf{e} & (22)\\ \sigma^+(\mathbf{x}_*^+) &= \sigma(\mathbf{x}_*^+) - \mathbf{k}_*^{+T}[K^+ + \sigma_n^2 I]^{-1}\mathbf{k}_*^+ & (23) \end{aligned}$$

where

$$\begin{aligned} \mathbf{e} &= \mathbf{y}^+ - \mu(\mathbf{X}^+) & (24)\\ \mu(\mathbf{X}^+) &= [\mu(\mathbf{x}_1^+), \ldots, \mu(\mathbf{x}_M^+)]^T, & (25)\\ \mathbf{k}_*^+ &= \Phi^{+T}\Sigma_{\mathbf{w}}\phi_*^+, & (26)\\ K^+ &= \Phi^{+T}\Sigma_{\mathbf{w}}\Phi^+. & (27) \end{aligned}$$

The parameters $\mu^+(\mathbf{x}_*^+)$ and $\sigma^+(\mathbf{x}_*^+)$ are the prediction and confidence score, under the adapted model, for the count y_*^+ of observation $\mathbf{x}_*^+$.

The prediction $\mu^+(\mathbf{x}_*^+)$ of the adapted model is equal to the prediction $\mu(\mathbf{x}_*^+)$ of the original model plus a correction term, $\mathbf{k}_*^{+T}[K^+ + \sigma_n^2 I]^{-1}\mathbf{e}$, determined by the prediction error $\mathbf{e}$ of the latter on the adaptation data $\mathbf{X}^+$. Hence, the impact of the adaptation stage increases with the mismatch between the predictions of the original model and the true counts, in the adaptation dataset. In fact, the correction term is a dot product of $\mathbf{e}$ with $\mathbf{k}_*^+ = \mathbf{k}^+(\mathbf{x}_*^+)$, where

$$\mathbf{k}^+(\mathbf{x}) = [k^+(\mathbf{x}, \mathbf{x}_{\mathbf{1}}^+), \ldots, k^+(\mathbf{x}, \mathbf{x}_{\mathbf{M}}^+)]^T. \qquad (28)$$

and

$$k^+(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T\Sigma_{\mathbf{w}}\phi(\mathbf{x}') \qquad (29)$$

is the kernel defined by the covariance $\Sigma_{\mathbf{w}}$. Since $\mathbf{k}_*^+$ is a vector of similarities, according to kernel $k^+(.,.)$, between $\mathbf{x}_*^+$ and the entries of $\mathbf{X}^+$, the correction is larger when $\mathbf{x}_*^+$ is most similar to the adaptation data $\mathbf{X}^+$. In summary, *the correction is most significant for the observations $\mathbf{x}_*^+$ that are most similar to the adaptation data $\mathbf{X}^+$ and have the poorest count predictions under the source model.*

4.4. Model adaptation

So far, we have seen how the predictions and confidence score of the GP can be adapted to new data, through (22)-(23). However, these equations depend on the kernel $k^+(.,.)$ of (29) rather than the original kernel of (9). Adaptation of the GP model requires the derivation of a relationship between $k^+(\mathbf{x}, \mathbf{x}')$ and $k(\mathbf{x}, \mathbf{x}')$. This relationship follows from the application of the matrix inversion lemma to the right-hand side of (5), which results in

$$\Sigma_{\mathbf{w}} = \Sigma_p - \Sigma_p\Phi(\Phi^T\Sigma_p\Phi + \sigma_n^2 I)^{-1}\Phi^T\Sigma_p. \qquad (30)$$

Using (9) and (29),

$$k^+(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - k^t(\mathbf{x}, \mathbf{x}') \qquad (31)$$

where

$$\begin{aligned} k^t(\mathbf{x}, \mathbf{x}') &= \phi^T(\mathbf{x})\Sigma_p\Phi(\Phi^T\Sigma_p\Phi + \sigma_n^2 I)^{-1}\Phi^T\Sigma_p\phi(\mathbf{x}')\\ &= \mathbf{k}^T(\mathbf{x})(K + \sigma_n^2 I)^{-1}\mathbf{k}(\mathbf{x}'), & (32) \end{aligned}$$

with $\mathbf{k}(\mathbf{x})$ given by (12), and $K = \Phi^T\Sigma_p\Phi$ as in (10)-(11). The kernel $k^t(.,.)$ is a *transfer kernel* that, when combined with the original kernel $k(.,.)$, results in the *adapted kernel* $k^+(.,.)$. It follows that the kernel matrix K^+ of (27) can be written as the matrix of entries

$$K_{ij}^+ = k(\mathbf{x}_i^+, \mathbf{x}_j^+) - \mathbf{k}^T(\mathbf{x}_i^+)(K + \sigma_n^2 I)^{-1}\mathbf{k}(\mathbf{x}_j^+). \quad (33)$$

Similarly, $\mathbf{k}_*^+$ can be written as the vector of entries

$$(\mathbf{k}_*^+)_i = k(\mathbf{x}_*^+, \mathbf{x}_i^+) - \mathbf{k}^T(\mathbf{x}_*^+)(K + \sigma_n^2 I)^{-1}\mathbf{k}(\mathbf{x}_i^+). \quad (34)$$

The updated prediction of (22) and score of (23) can thus be computed with the *original kernel* $k(\mathbf{x}, \mathbf{x}')$ and *no retraining*. Algorithm 1 summarizes the GP adaptation procedure.

Algorithm 1 Bayesian model adaptation

```
Input:
    Training data $X$ and $\mathbf{y}$; Adaptation data $X^+$, and $\mathbf{y}^+$;
    Feature vector $\mathbf{x}_*^+$ for count prediction
Output:
    Predictive distribution parameters: mean $\mu^+(\mathbf{x}_*^+)$ and variance $\sigma^+(\mathbf{x}_*^+)$;
    Training: Use $(X, \mathbf{y})$ and (14) to learn the parameters of the kernel $k(.,.)$ of (13)
    Original prediction: Given $\mathbf{x}_*^+$ compute the count estimate $\mu(\mathbf{x}_*^+)$ and confidence score $\sigma(\mathbf{x}_*^+)$ with (10) and (11), respectively
    Adaptation: Use $(X^+, \mathbf{y}^+)$ to compute $K^+$, using (33), $\mathbf{e}$, using (24)-(25), $(K^+ + \sigma_n^2\mathbf{I})^{-1}$ and $(K^+ + \sigma_n^2\mathbf{I})^{-1}\mathbf{e}$.
    Adapted prediction: compute $\mathbf{k}_*^+$ with (34), the adapted prediction $\mu^+(\mathbf{x}_*^+)$ with (22) and the adapted confidence score $\sigma^+(\mathbf{x}_*^+)$ with (23).
```

4.5. Relation to previous approaches

Model adaptation has received little attention in crowd counting. Instead, previous works [23, 13] rely on SSL. While SSL could be implemented with GP regression, e.g. by introducing an additional manifold regularization in the prior over functions $f(\mathbf{x})$, it is practical only when *all* counting models are learned *simultaneously,* e.g. when a camera counting network is first deployed. In general, its complexity is too high for the capacity expansion scenario, where it would require the solution of a learning problem involving *all* training video (both labeled and unlabeled) from *all* cameras, whenever a camera is added to the network.

The method of [13] also includes a simple model adaptation module. This assumes the existence of n observation pairs $\{\mathbf{x}^{source}, \mathbf{x}^{target}\}$ with identical counts ($y^{source} = y^{target}$) and consists of learning a linear *feature alignment* map $X^{target} = X^{source}\beta$, where β is a diagonal matrix learned by least squares. We refer to this method as feature alignment (FA). It should be noted that, in [13], it is mostly used to bring source and target data into alignment, so as to enable SSL across views. More related to the approach now proposed is prior machine learning research on transfer learning with GPs. The most popular among these methods is a procedure to jointly learn the parameters of a kernel shared by several GPs [1]. The shared kernel $k(\mathbf{x}, \mathbf{x}')$ is transferred to each task through a joint GP prior that induces correlations between tasks l and k according to

$$< f_l(\mathbf{x}), f_k(\mathbf{x}') >= K_{lk}^f k(\mathbf{x}, \mathbf{x}'), \tag{35}$$

where K^f is a positive semi-definite matrix that specifies inter task similarities and is learned from data of *all* tasks. While this learning is expensive, the methods can be applied with tractable complexity to the two-task (source-target) problem that we consider in this work.

In this case, the MT kernel reduces to

$$< f_l(\mathbf{x}), f_k(\mathbf{x}') >= \rho\, k(\mathbf{x}, \mathbf{x}'), \tag{36}$$

where ρ is a measure of similarity between source and target views. Several authors have noted limitations of this approach. [11] proposed an extension for asymmetric scenarios, where there is a primary and several secondary tasks, making $K_{lk}^f = \rho_l \rho_k$ with $\rho_p = 1$ if p is the primary task. However, in the two-task setting, this is identical to (36). [2] noted that [1] fails to fully exploit the statistical interpretation of the kernel a covariance for random functions, in the GP context. They treat the parameter ρ as a Gamma random variable and introduce a Bayesian procedure for its estimation. For the 2-class setting, this still results in the kernel of (36), albeit with ρ replaced by a constant derived from the Gamma (hyper-) parameters. However, because these are learned with the remaining GP parameters, this is not fundamentally different from (36).

For capacity expansion, all these approaches require relearning kernel parameters whenever a camera is added to the network. However, by only requiring data from the source and target views, this learning is more tractable than the combination of MTL and SSL [13]. Nevertheless, all these approaches suffer from the inconsistency between (36) and the statistical interpretation of a GP. Note that, while it is common to replace the covariance of (9) by a kernel function such as (13) (the "kernel trick"), this *continues to be* a covariance function. When data is observed, this function must be adapted by Bayesian inference, as shown in (31)-(34). While this results in the adapted kernel $k^+(\mathbf{x}, \mathbf{x}')$ of (29) with the covariance matrix of (30), the MTL kernel of (36) reduces this covariance to $\mathbf{\Sigma}'_{\mathbf{w}} = \rho\mathbf{\Sigma}_p$. This is a massive simplification that ignores the role of the source and adaptation data on kernel transfer. We thus refer to these approaches as *weak kernel transfer* (WKT) methods. Our experiments (see Section 6) show that the weak underlying covariance approximation renders these methods non-competitive with the *training free* model adaptation of (31)-(34), even when the WKT parameters are *relearned*.

A final approach is to resort to a hierarchical probability model, by introducing a (hyper-) prior distribution over the parameters of the GP prior $p(\mathbf{w})$ [25]. We refer to this method as *hierarchical GP* (HGP). Although statistically principled, HGP learning is based on an EM algorithm that requires a good initialization and can easily overfit when there is little data. The latter is a concern for the capacity expansion problem, where the goal is to rely on an adaptation dataset as small as possible. Perhaps due to this, HGP has not performed very well in our experiments.

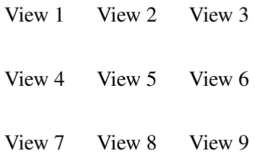

Figure 1. Typical image from the nine views in the proposed dataset for evaluation of transfer learning algorithms for crowd counting.

Table 1. Dataset properties. VI-J refers to View I, Sequence J.

Video	Range	avg	Video	Range	avg
V1-1	22-43	33.31	V1-2	13-58	41.38
V1-3	14-22	17.00	V2-1	14-24	18.68
V2-2	3-43	20.61	V2-3	11-21	16.24
V3-1	10-30	20.35	V3-2	17-40	28.33
V3-3	5-32	17.51	V4-1	6-15	10.64
V4-2	4-18	9.57	V4-3	5-25	12.33
V5-1	9-16	12.85	V5-2	2-17	9.59
V5-3	6-22	13.36	V6-1	7-15	11.33
V6-2	4-18	9.57	V6-3	6-26	12.82
V7-1	8-20	14.31	V7-2	6-18	10.54
V7-3	3-26	16.59	V8-1	12-24	17.52
V8-2	3-14	7.82	V8-3	16-30	24.30
V9-1	7-15	11.02	V9-2	4-13	8.53
V9-3	6-23	12.76			

5. Pedestrian count transfer dataset

In this section we introduce a new dataset, designed to test adaptation methods for crowd counting.

5.1. The dataset

The dataset was collected with a network of nine cameras, mounted on a building that overlooks a large pedestrian walkway. Figure 1 shows a typical image of each view. The views were selected to address both the resolution and footprint aspects of capacity expansion. Note how pairs of views have different overlap, ranging from none (e.g. views 3 and 7) to substantial (e.g. views 4 and 6). Some views, e.g. view 2, have no overlap with any other views, others, e.g view 5, have overlap with several others. The views are also at different resolutions, see e.g. views 1 and 9, and the dataset covers a salient scene feature, a courtyard in front of a building entrance, at higher resolution.

The dataset includes 3 video sequences per view, collected at different times of the day. These were chosen to induce significant variability of crowd-densities and lighting. Each sequence is 100s and 1,000 frames long (10 fps). Manual crowd count ground truth is provided for the $27,000$ frames in the dataset. Each video has between 2 and 58 people. Table 1 details the average and range of pedestrians per video. Since the videos were collected in a public area, without any staging or coaching, pedestrian configurations are highly variable. Pedestrians walk in different directions and at different speeds, sometimes stop, cast different shadows depending on time of day, and occlude each other in complex ways. Different views can have different resolutions (e.g. optical vs digital zoom). Occasionally, a bicycle, skateboarder, or other outlier enters the scene. All these factors pose challenges to count transfer.

5.2. Evaluation Protocol

To evaluate count transfer, we propose a protocol composed of 27 tasks and 702 adaptation experiments. Each task has one video as target and 26 experiments, using each of the remaining 26 videos as source. For each experiment, 400 frames are specified for model training, and $1,000$ frames for adaptation and testing. All experiments are repeated under two adaptation modes: “weak,” and “strong” adaptation. In the weak adaptation mode, the adaptation dataset has a small size of 20 frames. This leaves 980 frames for testing, per experiment. In the strong adaptation mode, the size of the adaptation set is 50 frames. Since the assembly of an adaptation dataset requires manual supervision, the size of this dataset determines the amount of manual labor involved. The two adaptation modes characterize counting performance in terms of manual labor (“weak” vs. “strong”). This is an important component of the cost of capacity expansion. For performance comparison, average error rates are reported per task, for each adaptation mode. The error rate is measured by the mean absolute error (MAE) $R = \frac{1}{N}\sum_{i=1}^{N} |\hat{y}_i - y_i|$, where N is the number of frames, $\hat{y}_i$ the count estimate and y_i the ground truth.

6. Experiments

In this section, we discuss experiments comparing various count transfer algorithms. Since a new representation for crowd counting was not our goal, we adopted the features of [4]. A mixture of dynamic textures [3] is fit to the video, in order to extract regions of crowd from the background, a region of interest (ROI) is defined, the video normalized for perspective, and several low-level features extracted per region. The information extracted from each frame is finally converted into a 30-dimensional vector, which is fed to a GP for crowd density estimation. For details on this representation see [4].

6.1. Comparison with adaptation methods

We start with a comparison to prior transfer learning methods using two datasets in the literature, ucsd [4] and mall [13]. The ucsd dataset has been used to evaluate several crowd counting algorithms. It includes 2000 frames at 10 fps. Mall includes 2000 frames at < 2 fps. These datasets are quite different, covering scenes of different

Table 2. Transfer counting accuracy (MAE) and time (seconds).

	ucsd to mall		mall to ucsd	
	MAE	Time (sec)	MAE	Time (sec)
FA [13]	7.47	796	4.44	832
HGP [25]	4.36	3.2	3.32	2.0
WKT [2]	4.18	263	3.76	134
GPA	4.18	0	2.79	0
GPTL	3.55	205	2.91	165

types (indoors for mall, outdoors for ucsd) and crowd densities, at different frame rates, from different camera angles, etc. Two adaptation tasks were defined, using one dataset as source and the other as target. For each task, 800 frames were used for training and 50 as adaptation set. The proposed approach was compared to the feature alignment (FA), weak kernel transfer (WKT), and hierarchical-GP (HGP) methods of Section 4.5. Two variants of the proposed approach were considered. The first is the pure model adaptation procedure of Section 4, denoted GP adaptation (GPA). The second, is a transfer learning extension, where the parameters of the kernel of (13) are relearned so as to maximize the marginal likelihood

$$\log p(\mathbf{y}^+|X^+, X, \mathbf{y}; \theta) = -\frac{1}{2}(\mathbf{y}^+)^T(K^+ + \sigma_n^2 I)^{-1}\mathbf{y}^+ \\ -\frac{1}{2}\log|K^+ + \sigma_n^2 I| - \frac{N}{2}\log 2\pi,$$

with K^+ as in (33). This is denoted as GP with transfer learning (GPTL).

Table 2 presents the results of all methods on the two transfer tasks. This includes MAE and total time used to relearn kernel parameters[1]. Unsurprisingly, the linear mapping of FA achieves the worst MAE. More interesting is the fact that the previous methods that relearn kernel parameters (HGP, WKT) have weaker MAE than the learning free GPA. This is strong evidence in support of the arguments of Section 4.5 namely 1) the weakness of the approximation of (31)-(34) by (36) by WTK, and 2) the overfitting potential of HGP. Finally, kernel transfer through (31)-(34) severely reduces this tendency to overfit, enabling GPTL to achieve the overall best results. In terms of training time, transfer learning methods require about 4 min for parameter relearning vs. no time for GPA. This increases as $O(n^3)$ on number of adaptation frames (15 min for 800 frames). For capacity expansion, it can be a non-trivial cost, specially when it does not guarantee better performance. Over all, only GPTL achieves lower MAE than GPA and only on one transfer task (ucsd to mall). Weaker performance on mall to ucsd suggests that parameter relearning can overfit. This is not surprising, given the small adaptation set.

6.2. Proposed dataset

Table 3 summarizes the results of count transfer on the dataset of Section 5. In addition to FA, WKT, HGP, GPA, and GPTL, results are reported for *no adaptation* (NA), where the model learned from the source is simply applied to the target data, and two *no transfer* modes: NTA, where a GP is learned from the adaptation set only and NTF, where a GP is learned from 400 frames of target data. This is the standard implementation of crowd counting (a model trained per view) and significantly more expensive in terms of video labeling. It is included as a lower bound for count transfer MAE. The table includes both the MAE and the retraining time per method, normalized by that of NTF.

Several observations ensue. First, all methods outperformed NA in both adaptation modes, showing that transfer learning is helpful for count transfer. Second, while no method underperformed NA for heavy adaptation, this was the case of both WKT and HGP for weak adaptation. These methods can thus have negative transfer, i.e. transfer learning results weaker than learning the counting model on the adaptation set. Only GPA, GPTL, and FA had no negative transfer on average. However, all methods had negative transfer in at least some experiments, as NTA had best results on 6 (5) experiments in heavy (weak) mode. This shows that all methods can be improved upon, highlighting the importance of testing transfer over a diversity of camera views, as required by the proposed dataset.

Third, among the methods that, on average, had no negative transfer, GPA achieved the best performance. In fact, it achieved the best results on an impressive 17 (13) of the 27 heavy (weak) adaptation experiments. It even outperformed GPTL - winner of only 3 (2) experiments. This is further evidence that parameter relearning can lead to overfitting. This hypothesis is also supported by the fact that the GPA gains over transfer learning were larger for weak adaptation. Fourth, GPTL clearly outperformed all other transfer learning methods. Again, kernel transfer through (31)-(34) seems to severely reduce the tendency for overfitting. Overall, GPA or GPTL achieved the top performance in 19 (17) of the 21 (22) heavy (weak) adaptation experiments where there was no negative transfer.

Finally, as expected, the error was larger in the weak than in the heavy adaptation regime. For heavy adaptation (annotation of 5s of video), the best methods had fairly high accuracy, with average error as low as 2.1 pedestrians (GPA). This is very close to the lower bound of NTF (1.85). On the other hand, errors were larger in weak adaptation (annotation of 2s of video), where even the average error of GPA was fairly high (3.18 people). This again suggests that there is room for future improvements.

7. Conclusion

The results above suggest that model adaptation is better suited for the capacity expansion problem than transfer learning. This is mostly due to a stronger kernel transfer by GPA and the fact that it is less prone to overfitting than

[1] All experiments were performed on a Intel Xeon E5504, 2.1 GHz. Note that GPA involves no parameter learning.

Table 3. MAE of various methods on proposed dataset. Confidence values are presented for average results.

Heavy	V1-1	V1-2	V1-3	V2-1	V2-2	V2-3	V3-1	V3-2	V3-3	V4-1	V4-2	V4-3	V5-1	V5-2
GPA	**2.78**	**3.09**	1.47	1.73	**2.39**	1.39	**2.14**	**2.30**	**2.26**	**1.86**	**2.02**	**2.01**	2.15	2.18
GPTL	3.36	3.31	1.47	**1.59**	2.81	**1.36**	2.27	3.51	2.53	1.93	2.13	2.38	2.25	2.53
HGP	5.00	5.60	2.33	2.92	3.26	2.29	3.00	3.51	3.03	2.41	2.81	3.20	2.59	2.93
WKT	3.30	6.58	2.95	2.44	9.19	2.68	3.87	6.15	8.59	2.78	2.55	3.06	1.94	2.67
FA	5.65	13.75	3.18	2.99	6.47	2.32	2.96	5.66	3.07	2.14	2.80	2.70	2.52	3.03
NTA	4.23	6.51	**1.34**	2.32	21.55	3.97	3.79	6.47	11.05	2.29	3.20	3.68	**1.37**	**2.05**
NA	18.33	28.08	7.76	8.46	10.19	7.14	9.78	14.51	8.41	8.08	7.62	7.40	6.88	7.47
NTF	1.89	1.64	0.89	1.07	1.67	0.85	1.07	1.23	1.20	1.27	2.02	2.16	1.38	2.42
V5-3	V6-1	V6-2	V6-3	V7-1	V7-2	V7-3	V8-1	V8-2	V8-3	V9-1	V9-2	V9-3	**Avg.**	Time
2.10	1.99	**2.14**	**2.30**	2.04	2.16	**2.00**	**2.24**	**2.05**	**2.10**	1.78	2.14	**1.99**	2.10 ± 0.82	0
3.96	1.61	2.32	2.84	2.15	**2.12**	2.23	2.73	2.07	2.76	1.77	2.46	2.14	2.39 ± 2.95	0.72
3.33	2.60	2.54	3.31	2.49	3.69	2.77	3.70	2.48	2.45	2.48	3.09	3.15	3.07 ± 1.29	0.01
1.80	1.68	2.77	3.06	2.52	3.20	2.66	3.42	4.12	3.32	**1.27**	2.71	2.41	3.47 ± 2.43	0.86
2.76	1.99	2.92	3.18	2.67	2.81	2.88	2.63	4.27	5.28	2.63	3.12	2.68	3.67 ± 3.10	3.32
3.17	**1.41**	3.20	3.10	**2.03**	2.14	2.57	3.86	3.78	3.30	1.45	**1.94**	2.31	4.00 ± 4.05	0.11
6.77	7.34	7.76	7.96	6.99	8.44	8.53	11.29	7.20	12.77	7.54	11.63	10.62	9.81 ± 8.77	0.05
2.64	1.34	2.37	3.39	1.64	1.76	1.63	2.86	3.43	2.50	1.56	2.28	1.72	1.85 ± 0.70	1.00

Weak	V1-1	V1-2	V1-3	V2-1	V2-2	V2-3	V3-1	V3-2	V3-3	V4-1	V4-2	V4-3	V5-1	V5-2
GPA	3.62	5.21	2.20	3.10	**5.30**	2.86	**2.62**	5.12	2.86	**2.38**	2.94	**2.71**	3.21	**2.88**
GPTL	3.91	6.02	3.27	**3.05**	7.43	2.84	4.62	**3.53**	4.59	3.92	**2.64**	3.62	4.27	3.18
HGP	7.14	19.33	3.47	5.17	8.53	3.64	5.14	5.71	4.33	2.70	4.11	3.40	3.61	5.41
WKT	4.69	11.62	3.76	3.26	7.37	2.91	3.81	4.94	7.18	2.78	7.43	3.55	**2.88**	4.66
FA	**3.23**	14.70	4.27	3.07	6.62	3.31	3.30	4.45	3.71	2.41	3.26	3.11	3.03	3.74
NTA	4.99	**3.83**	**2.12**	3.62	9.39	**2.70**	5.94	7.15	**2.64**	2.39	4.84	5.35	3.20	3.85
V5-3	V6-1	V6-2	V6-3	V7-1	V7-2	V7-3	V8-1	V8-2	V8-3	V9-1	V9-2	V9-3	**Avg.**	Time
2.65	2.46	**3.15**	**3.17**	2.89	**2.81**	**3.10**	**2.48**	**3.68**	2.95	3.07	3.13	**3.20**	3.18 ± 1.61	0
3.83	4.68	3.35	3.65	3.71	3.46	3.50	4.65	3.96	4.13	3.54	4.62	3.65	3.99 ± 4.06	0.64
3.15	2.91	5.27	6.02	3.13	3.56	4.67	4.90	3.88	7.59	5.01	4.51	5.14	5.24 ± 4.81	0.01
3.90	2.56	10.20	3.83	3.79	3.43	3.40	4.37	4.06	**2.78**	3.18	6.01	3.84	4.67 ± 0.94	0.53
3.19	**2.24**	3.58	3.64	**2.78**	2.96	3.20	3.61	4.09	5.01	**2.63**	3.64	3.24	3.93 ± 3.35	2.83
5.78	3.08	4.91	5.51	2.89	4.40	4.23	4.37	4.57	5.27	2.85	**2.75**	4.38	4.33 ± 1.61	0.12

methods that require parameter relearning. While GPA performance is quite strong, there is room for further progress in terms of 1) the complete elimination of negative transfer, and 2) more robust transfer in the weak adaptation scenario (or even with less than 2s of annotated video per new view). We believe that the dataset now introduced will enable further progress towards these goals.

References

[1] E. Bonilla, K. M. Chai, and C. Williams. Multi-task gaussian process prediction. 2008. 2, 5

[2] B. Cao, S. J. Pan, Y. Zhang, D.-Y. Yeung, and Q. Yang. Adaptive transfer learning. In *AAAI*, 2010. 2, 5, 7

[3] A. Chan and N. Vasconcelos. Modeling, clustering, and segmenting video with mixtures of dynamic textures. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(5):909–926, May 2008. 6

[4] A. Chan and N. Vasconcelos. Counting people with low-level features and bayesian regression. *Image Processing, IEEE Transactions on*, 21(4):2160–2177, April 2012. 1, 2, 3, 6

[5] Y. Cong, H. Gong, S. Zhu, and Y. Tang. Flow mosaicking: Real-time pedestrian counting without scene-specific learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 1, 2

[6] W. Dai, Y. Chen, G. Xue, Q. Yang, and Y. Yu. Translated learning: transfer learning across different feature spaces. In *Neural Information Processing Systems*, 2008. 2

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *EEE Conference on Computer Vision and Pattern Recognition, 2014*, 2014. 2

[8] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2066–2073, June 2012. 2

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. 2

[10] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 951–958, June 2009. 2

[11] G. Leen, J. Peltonen, and S. Kaski. Focused multi-task learning using gaussian processes. In *Machine Learning and Knowledge Discovery in Databases*, pages 310–325. Springer, 2011. 2, 5

[12] E. Liebe, B. Seeman, and B. Schiele. Pedestrian detection in crowded scenes. In *IEEE Conference in Computer Vision and Pattern Recognition*, 2005. 2

[13] C. C. Loy, S. Gong, and T. Xiang. From semi-supervised to transfer counting of crowds. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2256–2263. IEEE, 2013. 1, 2, 5, 6, 7

[14] Z. Ma and A. Chan. Crossing the line: Crowd counting by integer programming with local features. In *IEEE Conf. Computer Vision and Pattern Recognition*, 2013. 1, 2

[15] S. J. Pan and Q. Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010. 1

[16] J. Pereira and N. Vasconcelos. Cross-modal domain adaptation for text-based regularization of image semantics in image retrieval systems. *Computer Vision and Image Understanding*, 124:123–135, 2014. 2

[17] G. Qi, C. Aggarwal, and T. Huang. Towards semantic knowledge propagation from text corpus to web images. In *Proc. ACM International Conference on World Wide Web*, 2011. 2

[18] V. Rabaud and S. Belongie. Counting crowded moving objects. In *IEEE Conference in Computer Vision and Pattern Recognition*, 2006. 2

[19] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006. 2, 3

[20] D. Reynolds, T. Quatieri, and R. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10:19–41, 2000. 2

[21] D. Ryan, S. Denman, C. Fookes, and S. Sridharan. Crowd counting using multiple local features. In *Digital Image Computing: Techniques and Applications, 2009. DICTA '09.*, pages 81–88, Dec 2009. 1, 2

[22] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 213–226, Berlin, Heidelberg, 2010. Springer-Verlag. 2

[23] B. Tan, J. Zhang, and L. Wang. Semi-supervised elastic net for pedestrian counting. *Pattern Recognition*, 44:22972304, 2011. 1, 2, 5

[24] P. Woodland. Speaker adaptation for continuous density hmms: a review. In *ISCA Tutorial and Research Workshop on Adaptation Methods for Speech Recognition*, 2001. 2

[25] K. Yu, V. Tresp, and A. Schwaighofer. Learning gaussian processes from multiple tasks. In *Proceedings of the 22nd international conference on Machine learning*, pages 1012–1019. ACM, 2005. 5, 7