Ambiguity Helps: Classification with Disagreements in Crowdsourced Annotations

Viktoriia Sharmanska[⋆,1], Daniel Hernández-Lobato[⋆,2], José Miguel Hernández-Lobato[3], Novi Quadrianto[1]
[1]SMiLe CLiNiC, University of Sussex, Brighton, UK
[2]Universidad Autónoma de Madrid, Madrid, Spain
[3]Harvard University, Cambridge, Massachusetts, US

Abstract

Imagine we show an image to a person and ask her/him to decide whether the scene in the image is warm or not warm, and whether it is easy or not to spot a squirrel in the image. For exactly the same image, the answers to those questions are likely to differ from person to person. This is because the task is inherently ambiguous. Such an ambiguous, therefore challenging, task is pushing the boundary of computer vision in showing what can and can not be learned from visual data. Crowdsourcing has been invaluable for collecting annotations. This is particularly so for a task that goes beyond a clear-cut dichotomy as multiple human judgments per image are needed to reach a consensus. This paper makes conceptual and technical contributions. On the conceptual side, we define disagreements among annotators as privileged information about the data instance. On the technical side, we propose a framework to incorporate annotation disagreements into the classifiers. The proposed framework is simple, relatively fast, and outperforms classifiers that do not take into account the disagreements, especially if tested on high confidence annotations.

1. Introduction

There exists an anecdote in the computer vision community that every graduate student has to annotate at least one dataset during the graduate school life. With a rise of crowdsourcing platforms, such as Amazon Mechanical Turk (MTurk), Microworkers, and CrowdFlower, it becomes possible instead to crowdsource annotations from people everywhere in the world. One might expect that the graduate life becomes more of an easy and manageable ride. However, the ambitions and visions in the community evolve together with the possibility afforded by the crowdsourcing platforms. With MTurk, now it becomes possible to collect annotations for large datasets such as ImageNet [26], TinyImages [31], COCO [14], and Places [38]. Moreover, it becomes prevalent to collect task-specific datasets, for example for studying the attributes and their strength [20] and for determining the easiness or hardness of a particular classification task [22]. Those task-specific datasets often require annotations that are more *ambiguous* than typical object annotations 'present' or 'not present'.

Working with a crowdsourcing platform has its own positive and negative aspects. On the positive side, we could mass collect annotations within a short period of time. The annotations come from people with a diverse background, therefore to some degree it reflects on an unbiased process of data collection. On the other hand, this process requires a tight control over the quality of annotations. It is common that we get noisy or even random annotations by the inexperienced and distrusted MTurk annotators. Escaping from a task to annotate a dataset ourselves, we are confronted with a task to master a whole collection of tricks on the quality control in a crowdsourcing scenario [30, 10, 1].

This paper focuses on: a) the case that the quality control has been successfully applied and the annotations are collected from those *highly reputed* annotators, and b) the annotation task goes beyond a clear-cut dichotomy such as, for example, a squirrel is 'present' or 'not present' in the given image. In the ambiguous task, such as determining 'how easy' it is to spot a squirrel in an image, or deciding whether the scene is 'warm' or 'not warm' from the image, the necessity of obtaining multiple human judgments for each image data grows. A confidence in the label annotations, ranging from a simple percentage agreement to the kappa statistic, is then computed from the multiple annotations at each data point. For sufficiently high confidence annotations, a majority voting scheme is then widely adopted as the ground-truth label [31, 14, 38, 20, 22, 26].

It is now worth highlighting the following two observations regarding *disagreements* in multiple annotations. The first is that the removal of disputed cases results in a reduced size of datasets; those datasets are typically already moderate in size. The second is that the notion of a true ground-truth label might not be available, at least for a subset of the

data, given that the task is ambiguous and the annotations are all human judgments. We propose to incorporate annotation disagreements into the learning process of a classifier. Our method *will utilize* all data points with varying degree of confidence and *will follow* the commonly used rules to define a ground-truth label, for example, using maximum voting. We propose a classifier which uses the confidence in the label annotations to determine the level of influence of each data point in the training process. Data points with high level of disagreements will have less influence during training hopefully improving the performance. We instantiate this classifier in the form of Support Vector Classifier as well as Gaussian Process Classifier.

2. Related Work

The problem of learning a classifier from data points annotated with multiple noisy labels per instance dates back to at least the work of [29] in 1990s. The work adopts the latent variable modeling of [3]. Fast forward to 2008 and beyond, the needs for learning with multiple noisy annotations are further exemplified by the advent of crowdsourcing platforms [30, 24, 37, 1, 2, 15]. Prior work ranges from a simple majority voting where all annotators are weighted equally [20, 26] to a weighted voting by quantifying the expertise of the annotators [8]. Work that actively selects both the informative instances and the high-quality annotators also exists [25, 15]. However, the expertise of the annotators is not readily quantifiable and the notion of true label might not exist. In this paper, we focus on data annotated by highly reputed annotators, therefore they should have an equal weight, and we *will not* attempt to infer a true ground-truth label. Moreover, we are not in the active setting where we can control *what* and *how much* labeled data to be generated by annotators. Related to this, we are not trying to redefine a new task that aims to disambiguate the ambiguous task (e.g. [19, 11]). Instead, we are in a *passive* mode where we are given data sources consisting of data instances, their associated labels, and an *additional* information per data instance capturing the level of label disagreement, for example, in the forms of average annotator response for that label. *This additional information could be readily available in most vision datasets*.

There are two general perspectives in addressing the learning model of classification: probabilistic and non-probabilistic. We will focus on models that use the max-margin principle (SVM-based) as a representative of non-probabilistic approach and models that exploit Gaussian Process framework (GP-based) as a prototypical example of probabilistic method. Probabilistic methods output a probability distribution over labels, in contrast to non-probabilistic methods that only output the most likely label for a particular data point. Consequently, we would expect that the GP method will be better in capturing the *model uncertainty* over labels and can inherently steer this model uncertainty based on the confidence in label annotations. In the next sections, we will first continue our description of related work discussing how to incorporate disagreements in label annotations into the SVM-based models and then we will describe our proposed GP-based approach.

3. Instillation of Disagreements into Classifiers

We will now restrict our attention to the typical computer vision dataset: images and their annotations including the class labels based on, for example, the majority voting scheme over MTurk responses and the agreement scores among annotators defined as confidence.

3.1. SVM-based methods: state-of-the-art overview

Perhaps, the most frequently used classification setup is an SVM model trained on crowdsourced data with labels defined by majority voting among annotators. This means all data points are equally important without considering the size of the majority. We can instead use this majority size, i.e. the disagreement level in the label annotation, to distinguish between easy and difficult data points. One approach will be to instill this additional information as *an upper bound* to the hinge loss function in the SVM, resulting in a state-of-the-art method called SVM+ [34, 32]. First, let us define the learning setup. We assume that there are some image data in the form of a matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^{\mathrm{T}}$ of observed features and a vector of associated class labels $\mathbf{y} = (y_1, \ldots, y_n)^{\mathrm{T}}$, $y_n \in \{-1, 1\}$. However, besides $\mathbf{X}$ and $\mathbf{y}$, there is also another set of attributes associated with each training data point. Namely, the crowdsourcing confidence in label annotations $\mathbf{X}^{\text{conf}} = (\mathbf{x}_1^{\text{conf}}, \ldots, \mathbf{x}_n^{\text{conf}})^{\mathrm{T}}$. The goal of learning is to infer a classifier f that will output a label y_{new} for an un-seen data point $\mathbf{x}_{\text{new}}$, given the labeled training data *plus* the crowdsourcing confidence level for each data point. It is important to note that f cannot use $\mathbf{X}^{\text{conf}}$ as the *input argument* because it will not be available at test time. This label confidence is privileged information [34, 33] (available during training but not at test time).

The method SVM+ tries to predict the slack variables ξ_n of the data using $\mathbf{X}^{\text{conf}}$. Intuitively, we try to predict the difficulty of each data point based on the label disagreement among annotators for that particular instance, thereby creating a data dependent upper bound on the hinge loss. SVM+ sets $\xi_n = \langle \mathbf{w}^{\text{conf}}, \mathbf{x}_n^{\text{conf}} \rangle + b^{\text{conf}}$ in the SVM, where $\mathbf{w}^{\text{conf}}$ and b^{conf} are some parameters. The SVM+ objective function is:

$$\underset{\substack{\mathbf{w},\mathbf{w}^{\text{conf}} \\ b,b^{\text{conf}}}}{\text{minimize}} \|\mathbf{w}\|^2 + \alpha \left\|\mathbf{w}^{\text{conf}}\right\|^2 + \beta \sum_{n=1}^{N} \left[\langle \mathbf{w}^{\text{conf}}, \mathbf{x}_n^{\text{conf}} \rangle + b^{\text{conf}}\right]$$

$$\text{subject to, for all } n = 1, \ldots, N;\ \langle \mathbf{w}^{\text{conf}}, \mathbf{x}_n^{\text{conf}} \rangle + b^{\text{conf}} \geq 0$$

$$1 - y_n[\langle \mathbf{w}, \mathbf{x}_n \rangle + b] \leq \langle \mathbf{w}^{\text{conf}}, \mathbf{x}_n^{\text{conf}} \rangle + b^{\text{conf}}. \quad (1)$$

The scalar parameters α and β are the trade-off parameters. The model in (1) assumes a linear functional form for f and ξs. This can be relaxed via the kernel trick [34]. Gaussian kernels are often used in the $\mathbf{X}$ and $\mathbf{X}^{\text{conf}}$ domains, but with different bandwidths. Theoretically, SVM+ exploits that if one had access to the optimal slack variables, the convergence to the Bayes' error would be faster than the standard SVM, *i.e.*, $\mathcal{O}(1/n)$ convergence versus $\mathcal{O}(1/\sqrt{n})$ [34].

3.2. GP-based methods: Our proposed GPC with annotation disagreements

In a probabilistic approach to binary classification, the goal is to model probabilities of a data point $\mathbf{x}_n$ belonging to one of two classes. These probabilities must lie in the interval $[0, 1]$, however, a Gaussian process defines a probability over real-valued functions. GPC model turns the output of a Gaussian process into a class probability using a non-linear activation function.

GPC: Consider that the class label $y_n \in \{-1, 1\}$ of $\mathbf{x}_n$ is generated as $y_n = \text{sign}(f(\mathbf{x}_n))$, where $f(\cdot)$ is a latent function and $\text{sign}(0) = 1$. For GPC, a zero-mean Gaussian process prior is assumed for f [23]. To model a *noisy* class label generation process, the class membership probability can be written as $p(y_n = 1|\mathbf{x}_n, \epsilon_n, f) = \Theta([f(\mathbf{x}_n) + \epsilon_n])$ with a zero-mean and σ^2-variance Gaussian noise ϵ_n. We have used $\Theta(\cdot)$ to denote the Heaviside step function. In this paper, we will use this explicit noise representation[1]. Given N training data points, the latent function for those training points can be written as $\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N))^{\text{T}}$. Invoking the Gaussian marginalization property, the prior on $\mathbf{f}$ will then simply be an N-variate Gaussian distribution *i.e.*, $p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{C})$, where $\mathcal{N}(\cdot|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multivariate Gaussian with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Each entry in $\mathbf{C}$, C_{nm}, is $k(\mathbf{x}_n, \mathbf{x}_m)$, with $k(\cdot, \cdot)$ being a covariance function. A typical covariance function is the squared exponential $k(\mathbf{x}_n, \mathbf{x}_m) = \theta \exp(-0.5||\mathbf{x}_n - \mathbf{x}_m||^2/\ell)$, where θ controls the amplitude and ℓ controls the smoothness of f.

Assuming *i.i.d.* noise, the likelihood of the latent function $\mathbf{f}$ given N data points is[2]: $p(\mathbf{y}|\mathbf{X}, \mathbf{f}, \epsilon_{1,\ldots,N}) =$

$$= \prod_{n=1}^{N} p(y_n|\mathbf{x}_n, \epsilon_n, \mathbf{f}) = \prod_{n=1}^{N} \Theta(y_n f(\mathbf{x}_n)) \,. \qquad (2)$$

We have absorbed the noise term ϵ_n into the covariance function of f and re-defined $C_{nm} = k(\mathbf{x}_n, \mathbf{x}_m) + \sigma^2 \delta_{nm}$ where $\delta_{nm} = 1$ if $n = m$ and 0 otherwise and σ^2 is the variance of the additive Gaussian noise around f. Bayes' rule is then used to compute $p(\mathbf{f}|\mathbf{X}, \mathbf{y}) = p(\mathbf{y}|\mathbf{X}, \mathbf{f})p(\mathbf{f})/p(\mathbf{y}|\mathbf{X})$, which can be used for prediction. Moreover, $p(\mathbf{y}|\mathbf{X})$ may be maximized to estimate the parameters of $k(\cdot, \cdot)$ and the noise variance σ^2 [23]. Computing $p(\mathbf{f}|\mathbf{X}, \mathbf{y})$ is intractable, and several methods can be used for approximate inference [12, 18] with expectation propagation (EP) as the preferred method.

GPC$^{\text{conf}}$: To instill the confidence in the label annotation into the GPC framework, there exists GPC+ model [7]. The GPC+ is a heteroscedastic Gaussian process classification model that considers additive Gaussian noise around f with data-dependent variance given by $\exp(g(\mathbf{x}_n^\star))$, where g is another Gaussian process evaluated on the privileged information $\mathbf{x}_n^\star$. Similarly to the SVM+, the GPC+ can also be used to incorporate user agreement scores. The main drawback of the GPC+ model comes with an expensive inference procedure. This motivates us to propose a fast and scalable version of the GPC+ method. In the EP algorithm suggested in [7], the updates have no closed form expression and they must be approximated using numerical quadratures. Our proposed model admits an inference technique via EP algorithm which is quadrature-free (Section 4.2), therefore much faster than [7] (see Table 3) with no sacrifice in the predictive performance.

We follow the same intuition as in Section 3.1, but instead of using the additional information to upper bound the hinge loss, we will use it to modify the likelihood function. For this, we consider, besides f, another function g evaluated in the $\mathbf{X}^{\text{conf}}$ domain. The following properties of g are assumed. Whenever g is *negative*, the data point is easy-to-classify and the likelihood in (2) is considered. Whenever g is *positive*, the data point is difficult-to-classify and the influence of this data point towards the likelihood function on f should be reduced. For the difficult instance, we will consider a constant likelihood of $1/2$ in f, i.e. the probability that minimizes an impurity measure for a binary label. During the EP inference, the difficult instance will have either the constant likelihood (*being ignored*) or a mixture of constant and step likelihood (*reduced* influence); refer to Section 5.1-Analysis of the confidence in annotations.

Let $\mathbf{g} = (g(\mathbf{x}_1^{\text{conf}}), \ldots, g(\mathbf{x}_n^{\text{conf}}))^{\text{T}}$. Given $\mathbf{y}$, $\mathbf{X}$ and $\mathbf{X}^{\text{conf}}$, the likelihood for $\mathbf{f}$, $\mathbf{g}$ is now: $p(\mathbf{y}|\mathbf{X}, \mathbf{X}^{\text{conf}}, \mathbf{f}, \mathbf{g}) =$ $= \prod_{n=1}^{N} p(y_n|\mathbf{x}_n, \mathbf{x}_n^{\text{conf}}, \mathbf{f}, \mathbf{g}) =$

$$= \prod_{n=1}^{N} \Theta\big(y_n f(\mathbf{x}_n)\big)^{1-\Theta\left(g(\mathbf{x}_n^{\text{conf}})\right)} \big(1/2\big)^{\Theta\left(g(\mathbf{x}_n^{\text{conf}})\right)} \,. \qquad (3)$$

A similar likelihood is used for multi-class classification in [6]. However, the likelihood in [6] does not consider correlations through the function g about the classification difficulty of each data point.

We assume independence between f and g, $p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{C}_f)$ and $p(\mathbf{g}) = \mathcal{N}(\mathbf{g}|\mathbf{1}m_g, \mathbf{C}_g)$, where each entry in $\mathbf{C}_f$ and $\mathbf{C}_g$ is equal to $k(\mathbf{x}_n, \mathbf{x}_m) + \sigma_f^2 \delta_{nm}$ and $k(\mathbf{x}_n^{\text{conf}}, \mathbf{x}_m^{\text{conf}}) + \sigma_g^2 \delta_{nm}$ respectively. We denote the set of

[1] By integrating out the noise ϵ_n, we end up at the familiar form of probit classifiers where the class membership probability is simply $p(y_n = 1|\mathbf{x}_n, f) = \Phi_{(0,\sigma^2)}(f(\mathbf{x}_n))$ [23] with $\Phi(\cdot)$ as the probit function.

[2] We will drop the conditional notation on ϵ to avoid clutter.

different parameters in $k(\cdot,\cdot)$ for f and g along with noise variances as $\mathbf{\Omega} = \{\theta_f, \sigma_f^2, \ell_f, \theta_g, \sigma_g^2, \ell_g\}$. The posterior for $\mathbf{f}$ and $\mathbf{g}$ is:

$$p(\mathbf{f},\mathbf{g}|\mathbf{y},\mathbf{X},\mathbf{X}^{\text{conf}}) = \frac{p(\mathbf{y}|\mathbf{X},\mathbf{X}^{\text{conf}},\mathbf{f},\mathbf{g})p(\mathbf{f})p(\mathbf{g})}{p(\mathbf{y}|\mathbf{X},\mathbf{X}^{\text{conf}})}, \quad (4)$$

where $p(\mathbf{y}|\mathbf{X},\mathbf{X}^{\text{conf}})$ can be maximized with respect to $\mathbf{\Omega}$ to find good hyper-parameter values [23]. The posterior is used to compute a predictive distribution for the label y_{new} of a new data point $\mathbf{x}_{\text{new}}$: $p(y_{\text{new}}|\mathbf{y},\mathbf{X},\mathbf{X}^{\text{conf}},\mathbf{x}^{\text{new}}) =$

$$= \int \Theta(y_{\text{new}} f_{\text{new}}(\mathbf{x}_{\text{new}})) p(f_{\text{new}}|\mathbf{f}) p(\mathbf{f},\mathbf{g}|\mathbf{y},\mathbf{X},\mathbf{X}^{\text{conf}})\, d\mathbf{f} d\mathbf{g} df_{\text{new}}, \quad (5)$$

where $p(f_{\text{new}}|\mathbf{f})$ is a Gaussian conditional distribution. Note that (5) is approximate because we do not consider the label confidence $\mathbf{x}_{\text{new}}^{\text{conf}}$ associated to $\mathbf{x}_{\text{new}}$. However, if we are only interested in the label being predicted, *i.e.*, the one with the largest probability, (5) may be treated as exact.

We refer to the model specification described above as GPC$^{\text{conf}}$. Nevertheless, the computation of (4) and (5) is not feasible. These quantities must be approximated in the next section using expectation propagation (EP) [16].

4. Optimization

In this section, we will review an optimization method for the SVM+ problem in (1). Moreover, we will propose a quadrature-free EP algorithm for computing the posterior (4) and the predictive (5) distributions of GPC$^{\text{conf}}$.

4.1. SVM+ optimization

The optimization problem in (1) can be written as a quadratic programming problem and in [21] a sequential minimal optimization (SMO) algorithm is derived to find its solution. The SVM+ is more difficult to train than the SVM since it has more hyper-parameters to adjust, *i.e.*, the bandwidths of the two kernels and trade-off parameters α and β. These parameters are typically tuned using a grid search guided by cross-validation, which is *very expensive*.

4.2. GPC$^{\text{conf}}$ approximate inference

We briefly describe here a quadrature-free EP algorithm for GPC$^{\text{conf}}$. Full details can be found in the supplementary material. In EP each likelihood factor in $p(\mathbf{y}|\mathbf{X},\mathbf{X}^{\text{conf}},\mathbf{f},\mathbf{g})p(\mathbf{f})p(\mathbf{g})$, *i.e.*, the numerator in the r.h.s. of (4), is approximated by an un-normalized Gaussian [16]. We note that $p(\mathbf{f})$ and $p(\mathbf{g})$ are a Gaussian distribution, therefore no approximation is needed for those two terms. Let $f_n = f(\mathbf{x}_n)$ and $g_n = g(\mathbf{x}_n^{\text{conf}})$. Then, we can approximate the n-th likelihood term as follows: $p(y_n|\mathbf{x}_n,\mathbf{x}_n^{\text{conf}},\mathbf{f},\mathbf{g}) \equiv h_n(f_n,g_n) \approx$

$$\approx \tilde{s}_n \cdot \mathcal{N}(f_n|\tilde{m}_n,\tilde{v}_n) \cdot \mathcal{N}(g_n|\tilde{\mu}_n,\tilde{\nu}_n) \equiv \tilde{h}_n(f_n,g_n),$$

where $\tilde{s}_n$, $\tilde{m}_n$, $\tilde{v}_n$, $\tilde{\mu}_n$ and $\tilde{\nu}_n$ are to be estimated by EP, and we have assumed independence between f and g. The EP approximation of the numerator in the r.h.s. of (4) is $\tilde{q}(\mathbf{f},\mathbf{g}) = p(\mathbf{f})p(\mathbf{g})\prod_{n=1}^N \tilde{h}_n(f(\mathbf{x}_n), g(\mathbf{x}_n^{\text{conf}}))$, where each h_n has been replaced by the corresponding $\tilde{h}_n$. After normalization, $\tilde{q}$ becomes the EP posterior approximation. Namely, $q(\mathbf{f},\mathbf{g}) = \tilde{q}(\mathbf{f},\mathbf{g})Z_q^{-1} = \mathcal{N}(\mathbf{f}|\boldsymbol{\mu}_f,\mathbf{\Sigma}_f)\mathcal{N}(\mathbf{g}|\boldsymbol{\mu}_g,\mathbf{\Sigma}_g)$, which is a product of two multivariate Gaussians since the Gaussian distribution is closed under the product operation [27]. Furthermore, the normalization constant Z_q can be readily computed. Moreover, Z_q is used to approximate the denominator in the r.h.s. of (4).

EP updates until converge each factor $\tilde{h}_n$. We will denote $\tilde{h}_n^{\text{old}}$ as the value of this approximate factor at current iteration. The updated value at next iteration will be denoted as $\tilde{h}_n^{\text{new}}$. First, we *remove* the n-th approximate factor from the q approximate posterior, that is $q^{-n} \propto {}^{q^{\text{old}}}/_{\tilde{h}_n^{\text{old}}}$. The term q^{-n} is a Gaussian because both q and $\tilde{h}_n$ are. Next, an *updated posterior* distribution q^{new} is obtained by minimizing the Kullback-Leibler divergence between $h_n q^{-n}$ and q^{new}, i.e. $\text{KL}({}^{h_n q^{-n}}/_{Z_{h_n}} || q^{\text{new}})$ with the normalization constant Z_{h_n}. Minimizing the KL term involves moment matching between $h_n q^{-n}$ and q^{new}. The moments of $h_n q^{-n}$ can be obtained from the derivatives of $\log Z_{h_n}$ with respect to the (natural) parameters of q^{-n} [27]. Let m^{-n}, v^{-n}, μ^{-n} and ν^{-n} be the mean and variance under q^{-n} for f_n and g_n, respectively. Then, Z_{h_n} has a *closed-form*[3], *i.e.*, $Z_{h_n} = = \Phi({}^{m^{-n}}/_{\sqrt{v^{-n}}})\Phi({}^{-\mu^{-n}}/_{\sqrt{\nu^{-n}}}) + \Phi({}^{\mu^{-n}}/_{\sqrt{\nu^{-n}}})/2$, where $\Phi(\cdot)$ is the c.d.f. of a standard Gaussian distribution. The updated n-th approximate factor is now $\tilde{h}_n^{\text{new}} = {}^{Z_{h_n} q^{\text{new}}}/_{q^{-n}}$.

4.3. FITC approximation for scalable GPC$^{\text{conf}}$

EP requires the inverses of $\mathbf{C}_f$ and $\mathbf{C}_g$, *i.e.*, the covariance matrices of $p(\mathbf{f})$ and $p(\mathbf{g})$. This scales like $\mathcal{O}(N^3)$, which can be expensive if number of data points N is large. To reduce this cost, we employ the full independent training conditional (FITC) approximation [5, 17]. Under the FITC, the covariance matrix $\mathbf{C}$ is replaced by an approximation $\mathbf{Q}$ given by the sum of a low-rank matrix and a diagonal matrix. $\mathbf{Q}$ is parameterized by $M \leq N$ pseudo-inputs $\overline{\mathbf{X}} = (\overline{\mathbf{x}}_1,\ldots,\overline{\mathbf{x}}_M)^{\text{T}}$. That is, $\mathbf{Q} = \text{diag}(\mathbf{C}-\mathbf{K})+\mathbf{K}$, with $\mathbf{K} = \mathbf{C}_{\mathbf{x}\overline{\mathbf{x}}}\mathbf{C}_{\overline{\mathbf{x}}\overline{\mathbf{x}}}^{-1}\mathbf{C}_{\mathbf{x}\overline{\mathbf{x}}}^{\text{T}}$, where $\text{diag}(\mathbf{C}-\mathbf{K})$ is a diagonal matrix with the diagonal entries of $\mathbf{C}-\mathbf{K}$ and $\mathbf{C}_{\mathbf{x}\overline{\mathbf{x}}}$ is a $N \times M$ matrix whose entries are given by $k(\mathbf{x}_i,\overline{\mathbf{x}}_j)$, with $k(\cdot,\cdot)$ the covariance function. Similarly, for $\mathbf{C}_{\overline{\mathbf{x}}\overline{\mathbf{x}}}$. The cost of computing $\mathbf{Q}^{-1}$ is $\mathcal{O}(NM^2)$, if $M \ll N$. We approximate both $\mathbf{C}_f$ and $\mathbf{C}_g$ using the FITC approximation. For this, we use M pseudo-inputs, $\overline{\mathbf{X}}$ and $\overline{\mathbf{X}}^{\text{conf}}$, with the locations optimized by maximizing Z_q.

[3] Please, refer to the Eq. (11) in the supplementary material.

5. Experiments

In the first experiment, we address the task of recognizing various scene attributes whether they are 'present' or 'not present' in the images. We use a publicly available SUN Attribute dataset (*SUNAttribute*)[4] [20] that comes with the averaged score over MTurk annotations of attribute being present in the image. In the second experiment, we focus on differentiating between 'easy' and 'hard' images of animal classes. We use a subset of Animals with Attributes dataset (*AwA*)[5] [13] for which the annotation of easy-hard scores is available[6] [22]. For each class the annotation specifies ranking scores of its images from easiest to hardest.

Methods. We compare the performance of the proposed method GPC$^{\text{conf}}$ with baselines including the standard GPC (with the likelihood in (2)), the heteroscedastic model GPC+ [7], and the SVM-based methods, SVM and SVM+. GPC$^{\text{conf}}$, GPC+, and SVM+ methods utilize user agreement scores for each image label, whereas GPC and SVM do *not*. In GPC, for f, and in GPC$^{\text{conf}}$ and GPC+, for both f and g, we use a squared exponential covariance function. All hyper-parameters, *i.e.*, $\mathbf{\Omega} = \{\theta, \sigma^2, \ell\}$ are found by optimizing the marginal likelihood $p(\mathbf{y}|\mathbf{X}, \mathbf{X}^{\text{conf}})$; this is called type-II maximum likelihood. All GPC-based methods use the FITC approximation and the number of pseudo-inputs M is set to 100, which is smaller than the total number of training instances, 400 and 520 on average. The pseudo-inputs are initially chosen randomly from $\mathbf{X}$, in the case of the f function, and from $\mathbf{X}^{\text{conf}}$, in the case of the g function. Then, they are also *optimized* via type-II maximum likelihood. Finally, in SVM and SVM+ we use Gaussian RBF kernels, and a grid search coupled with 5 fold cross-validation to find all hyper-parameters. SVM has 2 hyper-parameters (kernel bandwidth and regularization) and SVM+ has 4 hyper-parameters (2 kernel bandwidths and 2 regularization parameters) to be tuned. The R code of all the methods is available at the authors' homepage.

Evaluation metric. We use the classification accuracy as the performance measure. We repeat each experiment 10 times using random train/test splits of the data and report mean and standard error across repeats. At each split, we take 80% of the data to train and 20% to test the models. Additionally we also provide the results using average precision as the performance metric (in the supplementary).

5.1. Ambiguity in recognizing semantic attributes

We focus on the Amazon MTurk study carried out on the SUNAttribute dataset. This dataset consists of 14,340 scene images taken from the SUN dataset [36] annotated with 102 scene attributes such as sunny, natural, man-made among others. The main task of this MTurk study is to annotate whether an attribute is 'present' or 'not present' in the image. The difficulty of this annotation task is multifaceted: (i) some attributes are rare e.g. smoke, scary, and others are rather common, e.g. natural, man-made; (ii) some attributes are visually obvious, e.g. ice, fire, and others are not, e.g. natural light, warm, cold; and (iii) images typically have only few attributes that are present.

In this dataset, presence of the attribute is measured as an average score of three binary user responses. The score is 1.00, 0.66, 0.33, or 0.0 when three, two, one, or zero workers accordingly decide that the attribute is 'present' in the image. In the description of the dataset [20], an attribute is considered 'present' if it receives at least two votes and 'not present' if it receives zero votes of three trusted workers. The authors mentioned that a single positive vote (average score equals 0.33) happens in a transition state between the attribute being present or not, so those images were *excluded* from the experiments. Our framework can automatically take this transition state into account and therefore does *not* discard any valuable annotations. The attribute label 'present'(+1) or 'not present'(-1) is defined via the maximum voting scheme (or in this case thresholding at 0.5) as it is widely adopted in crowdsourcing studies. Finally, the additional privileged data in terms of annotation confidence is 1.0 if all three MTurk users agree, and 0.66 if two out of three users agree on the label.

Setup. In [20], the authors consider two scenarios to study the attribute learning performance. We follow closely the first scenario, where the train and test sets are half positive and half negative so that recognizing each attribute is not influenced by the attribute popularity. We randomly select 500 samples with different confidence values to train/test the attribute classifiers. In total we train 83 binary attribute classifiers[7]. Additionally we study a balanced case of images with confidence 1.00 and 0.66 for training/testing the attribute classifiers. The results of this setting with 57 attributes in total[8] can be found in the supplementary material (entitled 57 attributes case). As our feature representation, we use 4096 dimensional deep CNNs features extracted from the fc7 activation layer in CaffeNet [9] pretrained on the large scene recognition database Places [38]. We normalize the features to have zero mean and unit variance for each dimension.

Results. We report the results of this experiment in Table 1. To ease the presentation of our results, we also visualize the pairwise differences between the performance of our proposed GPC$^{\text{conf}}$ method and four other methods, GPC, GPC+, SVM and SVM+, in the form of bar plots located

[4] https://cs.brown.edu/~gen/sunattributes.html

[5] http://attributes.kyb.tuebingen.mpg.de/

[6] http://smileclinic.alwaysdata.net/lupi/AwA_easyhardscore.zip

[7] In [20], 87 attributes are learned with 350 samples for training/testing. Since we use more samples, we account for 83 out of 87 of attributes.

[8] The number of attributes is restricted by the amount of positive samples available with confidence 1.00 and 0.66.

	GPC image	GPC^{conf} image+conf	GPC+ [7] image+conf	SVM+ [34] image+conf	SVM image		GPC image	GPC^{conf} image+conf	GPC+ [7] image+conf	SVM+ [34] image+conf	SVM image
sailing	79.70 ± 0.79	**80.70 ± 0.71**	79.90 ± 0.85	80.00 ± 0.84	79.30 ± 0.88	metal	71.50 ± 1.57	**72.00 ± 1.33**	71.70 ± 1.51	71.90 ± 1.45	70.40 ± 0.70
driving	77.20 ± 1.78	**78.70 ± 1.67**	78.50 ± 1.63	77.90 ± 1.47	77.70 ± 1.61	paper	72.90 ± 1.33	**74.00 ± 1.00**	73.70 ± 0.85	72.80 ± 1.29	72.80 ± 1.00
biking	75.00 ± 1.39	**76.80 ± 1.24**	76.70 ± 1.29	75.60 ± 1.71	**76.80 ± 1.19**	wood	72.00 ± 1.15	72.30 ± 1.21	**72.40 ± 1.25**	71.90 ± 2.09	72.20 ± 1.55
transport	**81.20 ± 0.58**	80.60 ± 1.03	80.90 ± 1.09	81.10 ± 1.02	80.00 ± 1.13	vinyl	69.80 ± 1.70	69.10 ± 1.68	69.20 ± 1.64	67.30 ± 1.48	**70.60 ± 1.34**
vacation	**76.50 ± 1.63**	76.10 ± 1.67	75.70 ± 1.55	74.20 ± 1.41	75.10 ± 1.12	rubber	77.40 ± 1.08	**78.20 ± 1.16**	77.80 ± 1.21	77.20 ± 1.42	77.80 ± 1.21
hiking	78.20 ± 2.11	**80.10 ± 1.68**	79.60 ± 1.74	78.70 ± 1.93	78.30 ± 1.56	cloth	74.10 ± 1.53	75.70 ± 1.09	**76.00 ± 1.10**	74.30 ± 1.18	75.10 ± 1.11
climbing	82.20 ± 1.29	**83.30 ± 1.18**	82.90 ± 1.14	81.80 ± 1.21	81.70 ± 1.10	sand	**78.50 ± 1.39**	77.80 ± 1.59	77.40 ± 1.46	77.70 ± 1.52	78.20 ± 1.36
camping	75.10 ± 1.25	**76.40 ± 1.33**	76.10 ± 1.37	75.10 ± 1.42	76.10 ± 1.31	rock	74.70 ± 1.31	**75.50 ± 1.37**	**75.50 ± 1.53**	74.20 ± 1.27	75.30 ± 1.69
reading	78.40 ± 0.94	**78.70 ± 1.13**	78.30 ± 1.15	77.70 ± 0.99	78.10 ± 1.22	dirt/soil	76.10 ± 1.34	**77.70 ± 1.30**	77.30 ± 1.38	75.50 ± 1.74	75.60 ± 1.31
teaching	75.80 ± 0.95	76.10 ± 1.35	**76.80 ± 1.26**	76.20 ± 1.30	76.10 ± 1.58	glass	67.70 ± 1.10	**68.20 ± 0.91**	68.00 ± 1.17	65.80 ± 1.39	67.90 ± 1.03
diving	75.70 ± 1.32	75.60 ± 1.04	76.10 ± 0.98	**76.60 ± 0.80**	74.40 ± 0.74	ocean	**81.00 ± 0.51**	80.60 ± 0.71	80.90 ± 0.77	80.60 ± 0.78	79.60 ± 0.97
swimming	76.20 ± 0.82	78.00 ± 1.00	**78.40 ± 1.01**	78.00 ± 0.65	78.20 ± 0.86	run.water	79.80 ± 1.33	80.20 ± 0.87	80.10 ± 0.99	**80.80 ± 1.03**	80.30 ± 1.41
eating	79.40 ± 1.00	80.40 ± 0.78	80.10 ± 0.94	77.30 ± 1.42	**80.60 ± 0.80**	stillwater	76.30 ± 1.51	76.00 ± 1.26	**76.40 ± 1.31**	75.40 ± 0.98	75.60 ± 1.11
socializing	76.20 ± 1.00	76.70 ± 1.43	77.30 ± 1.23	**78.10 ± 1.47**	77.30 ± 1.50	snow	84.40 ± 0.89	**85.50 ± 0.85**	85.40 ± 0.78	85.00 ± 0.96	85.40 ± 0.99
congregat.	80.00 ± 1.17	**81.20 ± 1.11**	80.60 ± 1.13	77.80 ± 1.04	79.90 ± 1.19	clouds	71.80 ± 1.68	70.50 ± 1.50	**71.90 ± 1.20**	70.80 ± 1.41	69.40 ± 1.10
queuing	71.30 ± 1.15	71.10 ± 1.20	71.20 ± 1.19	70.10 ± 1.07	**71.60 ± 1.37**	nat.light	78.80 ± 1.26	81.50 ± 1.00	**81.60 ± 1.07**	79.70 ± 1.18	81.30 ± 1.23
competing	78.30 ± 1.61	80.30 ± 1.40	80.00 ± 1.48	79.60 ± 1.63	**80.90 ± 1.46**	sunny	71.90 ± 1.20	72.20 ± 0.81	**72.70 ± 1.04**	69.30 ± 1.35	71.90 ± 1.36
sports	80.90 ± 1.10	**81.60 ± 0.84**	81.50 ± 0.86	79.90 ± 1.27	81.40 ± 1.06	el.lighting	73.20 ± 0.99	73.00 ± 1.17	73.20 ± 1.09	**73.80 ± 1.12**	72.10 ± 0.95
exercise	77.70 ± 1.48	**80.40 ± 1.14**	79.90 ± 1.43	79.00 ± 0.80	78.90 ± 1.10	aged/worn	72.70 ± 1.18	73.50 ± 0.95	73.10 ± 1.10	72.70 ± 1.08	**73.70 ± 0.93**
playing	76.70 ± 0.88	77.00 ± 1.32	**77.10 ± 1.29**	76.60 ± 1.66	76.90 ± 1.55	glossy	74.40 ± 1.56	**75.00 ± 1.81**	74.10 ± 1.69	72.60 ± 1.52	72.40 ± 1.71
spectating	81.90 ± 1.45	**83.20 ± 0.90**	**83.20 ± 0.86**	81.10 ± 1.20	82.20 ± 1.00	matte	69.40 ± 1.02	67.70 ± 0.99	67.90 ± 1.00	68.80 ± 1.04	**69.80 ± 0.96**
farming	79.00 ± 0.76	79.20 ± 1.08	79.20 ± 1.11	**79.30 ± 0.94**	77.80 ± 0.96	moist	73.80 ± 1.65	75.30 ± 1.62	74.20 ± 1.56	72.70 ± 1.64	**75.70 ± 1.42**
shopping	**81.50 ± 0.74**	81.20 ± 0.94	81.10 ± 0.98	80.20 ± 1.58	81.00 ± 1.07	dry	75.80 ± 1.60	75.70 ± 1.50	**76.20 ± 1.59**	**76.20 ± 0.90**	**76.20 ± 1.26**
working	76.90 ± 1.62	76.80 ± 1.59	**77.00 ± 1.60**	76.30 ± 1.50	75.70 ± 1.66	dirty	**77.40 ± 1.09**	76.70 ± 1.47	77.10 ± 1.40	75.70 ± 1.31	75.70 ± 1.20
us.tools	75.20 ± 1.17	74.80 ± 1.26	**75.30 ± 1.35**	73.20 ± 1.16	75.10 ± 1.18	rusty	69.60 ± 1.24	70.10 ± 1.14	**70.30 ± 1.16**	69.40 ± 1.10	69.50 ± 1.31
business	71.20 ± 1.48	**71.50 ± 1.51**	70.90 ± 1.63	68.60 ± 1.54	**71.50 ± 1.58**	warm	71.50 ± 1.63	**72.20 ± 2.08**	71.90 ± 2.02	70.60 ± 1.46	70.20 ± 1.55
praying	78.60 ± 0.60	78.70 ± 1.02	**79.20 ± 1.08**	78.40 ± 0.55	77.20 ± 1.21	cold	86.90 ± 1.27	87.20 ± 1.19	87.10 ± 1.17	**87.30 ± 0.83**	87.00 ± 0.99
fencing	69.40 ± 0.87	69.40 ± 0.99	**69.80 ± 1.00**	68.30 ± 1.71	69.30 ± 1.09	natural	**82.80 ± 1.16**	82.20 ± 1.10	82.40 ± 0.93	81.90 ± 1.40	81.70 ± 1.25
railing	65.70 ± 1.71	69.60 ± 1.12	**69.70 ± 1.30**	68.00 ± 1.26	68.20 ± 1.77	man-made	71.40 ± 1.14	71.80 ± 0.99	71.70 ± 1.15	71.70 ± 0.81	**73.90 ± 1.14**
wire	70.80 ± 1.55	70.20 ± 1.85	70.50 ± 1.80	**72.10 ± 1.24**	71.00 ± 1.26	open	78.00 ± 0.72	**79.60 ± 1.10**	79.50 ± 1.12	78.60 ± 0.74	79.30 ± 1.19
trees	**76.20 ± 1.35**	76.00 ± 1.22	76.00 ± 1.29	74.40 ± 1.18	75.20 ± 0.94	semi-encl.	76.90 ± 1.14	77.30 ± 1.01	77.60 ± 0.98	77.10 ± 1.86	**77.80 ± 1.37**
grass	77.90 ± 1.51	79.10 ± 1.14	79.00 ± 1.22	**79.30 ± 1.67**	77.70 ± 1.05	enclosed	80.60 ± 1.14	81.90 ± 1.34	**82.10 ± 1.25**	80.80 ± 1.08	**82.10 ± 1.05**
vegetation	78.30 ± 1.26	**78.90 ± 1.47**	78.50 ± 1.54	77.40 ± 1.23	76.70 ± 1.10	far-horizon	80.40 ± 1.52	**81.00 ± 1.31**	80.20 ± 1.35	79.90 ± 1.39	79.70 ± 1.24
shrubbery	78.50 ± 1.68	79.50 ± 1.31	**79.70 ± 1.43**	77.10 ± 1.63	76.20 ± 1.69	nohorizon	79.10 ± 1.28	**79.60 ± 1.43**	**79.60 ± 1.43**	78.60 ± 1.43	78.60 ± 1.44
foliage	76.20 ± 0.91	**77.30 ± 0.96**	77.10 ± 0.97	76.20 ± 0.69	76.80 ± 1.27	rugged	81.20 ± 0.87	81.30 ± 0.60	81.30 ± 0.57	80.80 ± 1.26	**81.90 ± 0.83**
leaves	75.50 ± 1.55	76.60 ± 1.59	**77.40 ± 1.46**	76.80 ± 1.16	77.20 ± 1.08	vertical	72.50 ± 0.93	**75.10 ± 1.53**	74.40 ± 1.39	71.80 ± 1.38	73.60 ± 1.23
asphalt	80.90 ± 1.25	**81.30 ± 1.32**	81.20 ± 1.60	80.30 ± 0.85	80.90 ± 1.16	horizontal	72.40 ± 1.27	**73.60 ± 0.94**	**73.60 ± 0.91**	71.70 ± 1.47	70.90 ± 1.25
pavement	73.70 ± 0.88	73.70 ± 1.10	73.90 ± 0.88	73.90 ± 1.10	**74.30 ± 0.68**	symmetr.	73.30 ± 1.24	73.10 ± 1.27	**73.40 ± 1.23**	70.60 ± 1.58	72.30 ± 1.66
shingles	83.60 ± 0.89	83.50 ± 0.82	**83.90 ± 0.98**	82.90 ± 1.26	82.20 ± 1.33	cluttered	**79.30 ± 1.18**	78.90 ± 1.20	78.70 ± 1.18	78.20 ± 1.35	78.40 ± 0.98
carpet	74.60 ± 1.27	74.40 ± 1.10	**74.90 ± 1.08**	74.40 ± 0.98	73.20 ± 1.36	soothing	71.90 ± 1.16	**73.70 ± 1.14**	73.20 ± 1.17	73.00 ± 1.32	73.10 ± 1.35
brick	79.10 ± 1.57	78.50 ± 1.46	78.30 ± 1.36	**80.10 ± 1.27**	78.70 ± 1.56	stressful	72.90 ± 1.58	73.70 ± 1.70	**73.80 ± 1.72**	72.80 ± 1.23	72.70 ± 1.64
concrete	66.40 ± 1.45	65.70 ± 1.10	**67.40 ± 1.34**	65.80 ± 1.33	67.30 ± 1.15						

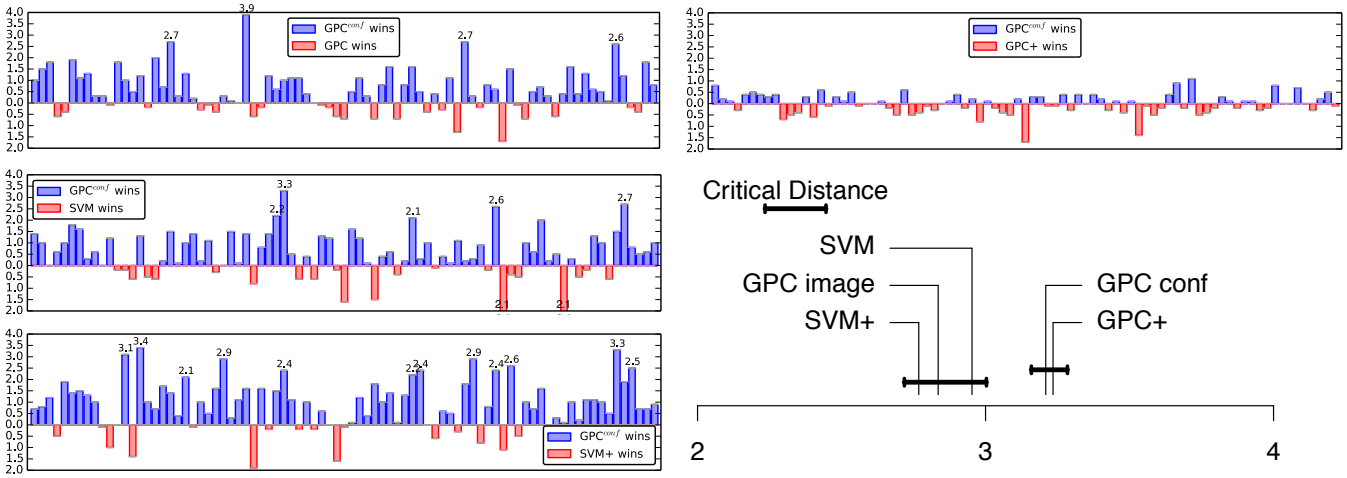

Table 1. Recognizing 83 scene attributes. **Top:** The numbers are mean and standard error of the accuracy over 10 runs of the 83 attribute classifiers. GPC^{conf}, GPC+ and SVM+ methods utilize user agreement scores for each image label, whereas GPC and SVM do *not*. The best result is highlighted in **boldface** with an extra blue for GPC^{conf} and GPC+. We consider GPC^{conf} as a faster variant of GPC+ (please refer to Table 3 for running time comparison). **Bottom:** To summarize the full results in the above table, we also provide a pairwise comparison of the proposed GPC^{conf} and four other methods in terms of difference in accuracies. The length of the bar corresponds to relative improvement of the accuracy for each of the 83 attributes. Finally, we also include statistical summary of the results based on Demšar [4] analysis (bottom right). Average rank of the methods (x axis) is computed based on accuracy over all repeats (the higher the better). A critical distance measures significant differences between methods based on their ranks. We link two methods with a solid line if they are not statistically different (p-value $> 5\%$). Best viewed in color.

at the bottom of the table. Additionally we also analyzed our experimental results using the multiple dataset statistical comparison method of [4] (bottom right). A bird's eye view of Table 1 and its figures tells that it is certainly beneficial to incorporate label confidence information into the learning process. Overall, the GPC-based LUPI methods, GPC$^{\text{conf}}$ and GPC+, outperform other baselines in the majority of cases. We counted 30 wins for GPC$^{\text{conf}}$, 28 wins for GPC+, 9 for GPC, 10 for SVM+, and 15 for SVM including 9 draws. GPC$^{\text{conf}}$ is superior to the standard GPC in 56 out of 83 cases, whereas SVM+ performs on par with the standard SVM baseline. Furthermore, GPC$^{\text{conf}}$ is able to utilize the label confidence better than SVM+ *in this experiment*. We suspect the SVM+ does not perform well due to the difficulty in finding suitable values for the four hyperparameters. The statistical analysis based on Demšar test [4] further supports these claims, as well as the fact that there is no significant difference between the GPC$^{\text{conf}}$ and GPC+ methods.

Analysis of the predictive performance. Additionally we perform the analysis using a balanced case of images with confidence 1.00 and 0.66 for training/testing the attribute classifiers (the results are in the supplementary material). We report the performance on test images with both 1.0 and 0.66 agreement scores (test scenario **A**), and when using test images with 1.0 agreement scores only (test scenario **B**). GPC$^{\text{conf}}$ shows an advantage over other baselines, particularly in **B**, verifying its robustness against label noise. As a practical note, we can see that there is a clear difference when testing the methods in **A** and **B**, i.e. the overall performance of the methods in **B** is higher than those in **A**. Hence, it is important to take into account the information about MTurk users agreement when collecting new datasets and designing the evaluation setup.

Analysis of the confidence in annotations. The main principled advantage of the GPC-based over SVM-based methods is that the label confidence is inherently encoded in the probabilistic output of the GPC-based methods. Given that we have confidence of the MTurk annotations during training ($\mathbf{X}^{\text{conf}}$), we analyze confidence in labels produced by the GPC$^{\text{conf}}$ method in two ways. First, we visualize few representative examples of the learned function $\mathbf{g}$ that acts on MTurk confidence $\mathbf{X}^{\text{conf}}$ in Figure 1–**right** (thick blue line, left y-axis shows function values). This function is designed to reflect the value of the MTurk confidence: instances with low confidence in annotations receive less emphasis in the training process ($\mathbf{g}$ is positive) and instances with high confidence receive more emphasis ($\mathbf{g}$ is negative). On this plot one can also see the probability $p(g(\mathbf{x}_n^{\text{conf}}) > 0)$ as a function of MTurk confidence (thick red line, right y-axis shows probability scores). For samples where MTurk confidence was equal to 1, $p(g(\mathbf{x}_n^{\text{conf}}) > 0)$ is almost 0 (our design enforces instances with high confidence to have

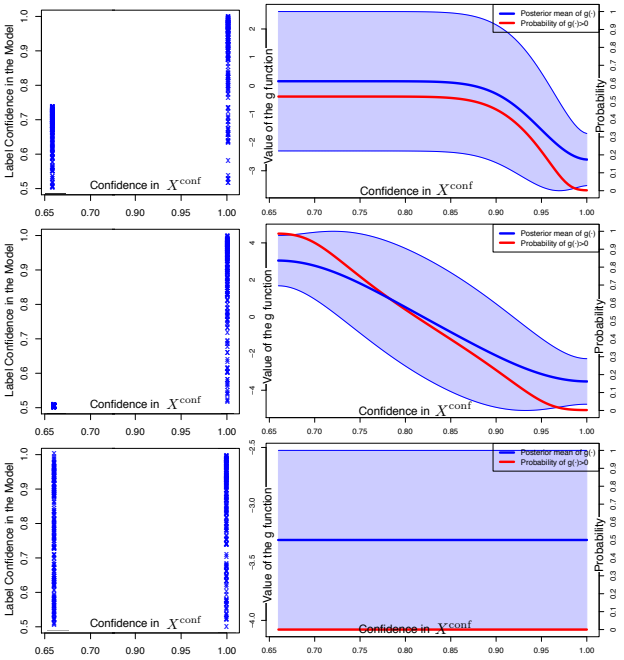

Figure 1. Label confidence analysis. **Left**: Scatter plot of MTurk confidence versus marginal likelihood $p(\mathbf{y}|\mathbf{X}, \mathbf{X}^{\text{conf}})$ at training time. **Right**: Representative posterior mean of the g function and 1-standard-deviation confidence interval (solid blue curve) alongside with the probability of $g > 0$ (solid red curve) for three different cases. Best viewed in color.

negative $\mathbf{g}$). However, for samples with MTurk confidence 0.66, we have 3 scenarios[9]: **a)** $p(g(\mathbf{x}_n^{\text{conf}}) > 0) = 1.0$ corresponds to low confidence data points being *ignored* with likelihood contributions of $1/2$ (Figure 1–**middle**), **b)** $p(g(\mathbf{x}_n^{\text{conf}}) > 0) = 0.0$ corresponds to low confidence data points being *as informative as* any data point with the step likelihood contributions (Figure 1–**bottom**), and **c)** $0.0 < p(g(\mathbf{x}_n^{\text{conf}}) > 0) < 1.0$ corresponds to influence of low confidence data points being *reduced* with a mixture of $1/2$ and the step likelihood contributions (Figure 1–**top**).

Second, we also show the scatter plots how label confidence scores $p(\mathbf{y}|\mathbf{X}, \mathbf{X}^{\text{conf}})$ produced by the classifier (in the training split) reflect the MTurk agreement scores in Figure 1–**left**. The scatter plot shows similar trends between the MTurk confidence and the classifier confidence of GPC$^{\text{conf}}$. When MTurk confidence is 1, the GPC$^{\text{conf}}$ classifier confidence tends to be high (in the top right corner), and if MTurk score is 0.66, the classifier confidence stays low (in the bottom left corner).

[9]For a particular instance $\mathbf{x}_n$, $\mathbf{x}_n^{\text{conf}}$, y_n the associated term in the likelihood function of f and g in Eq. 3 is: $p(y_n|\mathbf{x}_n, \mathbf{x}_n^{\text{conf}}, f, g) = 1/2^{\Theta(g(\mathbf{x}_n^{\text{conf}}))} \times \Theta(y_n f(\mathbf{x}_n))^{1-\Theta(g(\mathbf{x}_n^{\text{conf}}))}$. By marginalizing g, the likelihood term of f given the instance is: $p(g(\mathbf{x}_n^{\text{conf}}) > 0) \times \frac{1}{2} + (1 - p(g(\mathbf{x}_n^{\text{conf}}) > 0)) \times \Theta(y_n f(\mathbf{x}_n))$.

	GPC image	GPC^{conf} (ours) image+conf	SVM+ [34] image+conf	SVM image
Chimp.	74.86 ± 0.8	74.93 ± 0.7	$\mathbf{75.07 \pm 0.7}$	73.71 ± 0.9
G.panda	80.64 ± 0.5	81.17 ± 0.6	$\mathbf{81.33 \pm 0.5}$	80.53 ± 0.6
Leo	81.67 ± 0.7	$\mathbf{82.00 \pm 0.7}$	80.58 ± 0.6	80.42 ± 0.8
Pers.cat	79.72 ± 0.4	$\mathbf{80.14 \pm 0.4}$	79.15 ± 0.7	78.17 ± 1.0
Hippo	72.85 ± 1.0	72.78 ± 1.1	$\mathbf{73.33 \pm 1.4}$	73.06 ± 1.1
Raccoon	78.57 ± 1.0	$\mathbf{78.81 \pm 0.8}$	76.98 ± 0.8	76.51 ± 0.6
Rat	$\mathbf{84.33 \pm 1.5}$	84.00 ± 1.5	83.50 ± 1.8	81.50 ± 1.8
Seal	48.00 ± 1.4	48.10 ± 1.2	48.50 ± 0.8	49.20 ± 0.8

Table 2. Distinguishing easy from hard images on AwA dataset. The numbers are mean and standard error of the accuracy over 10 runs. To define confidence of image label, GPC^{conf} and SVM+ methods utilize easy-hard score annotation available per image.

	GPC	GPC^{conf} (ours)	GPC+[7]	SVM	SVM+[34]
SUNAttribute	27m.	32m.	51m.	6m.	106m.
AwA	32m.	42m.	73m.	10m.	252m.

Table 3. Average training time in minutes for SUNAttribute with $\approx$ 400 tr. samples and AwA experiments with $\approx$ 520 tr. samples.

5.2. Ambiguity to distinguish easy from hard images

We focus on the Amazon MTurk study carried out on 8 classes of the AwA dataset: *chimpanzee, giant panda, leopard, persian cat, hippopotamus, raccoon, rat* and *seal*. In this study, a worker is shown a set of images of one class and is asked to rank the images from the easiest to the hardest to spot the animal. Finally, each image gets an easy-hard score in the range from 1 (hardest) to 16 (easiest) as the average score over all worker responses across multiple sets of images.

Setup. This task mimics human learning to classify easy and hard samples. For each class, we label half of the images as 'easy'(+1) and half of the images as 'hard'(-1) with respect to the easy-hard scores, and solve a binary classification problem. The important information about the easy-hard score can be encoded in the label confidence. We iteratively assign the highest confidence, *i.e.*, 1.0 to the top 10% of the easiest images and bottom 10% of the hardest images. Next, we assign a less confident score equal to 0.9 to the 10% of the remaining data from the top and from the bottom, and repeat. Images with an average score slightly above/below the threshold get a confidence score of 0.6. We use all available data per class to form the train/test splits, ranging from 300 (*rat*) to 900 images (*giant panda*). As our feature representation, we use 4096 dimensional deep CNNs features extracted from the fc7 activation layer in CaffeNet [9] pretrained on ImageNet (ILSVRC12) [26]. We normalize the features to have zero mean and unit variance for each dimension.

Results. We present the results of this experiment in Table 2. First we would like to point out that in all cases but *seal* the overall performance of the methods is better than chance. This means that there is a difference between easy and hard instances and we can learn a classifier to capture this difference. In case of *seal*, majority of the images are indifferent, i.e. easy-hard score is homogeneously distributed in the middle range, so the image is as easy as it is hard. Since there is no signal for classifier to learn we exclude this class from further analysis. From the table with the results we can conclude that it is certainly beneficial to incorporate label confidence information into the learning process. Overall, the methods that utilize confidence information, GPC^{conf} and SVM+, outperform their counterparts, GPC and SVM, in all cases but *rat*, which has the least amount of train/test data available. In this experiment, the proposed GPC^{conf} performs on par with the SVM+ method, which shows the benefits of both GPC-based and SVM-based frameworks for LUPI.

6. Discussion and Conclusion

All you need in this life is ignorance and confidence, and then success is sure.

Mark Twain

We studied learning using privileged information (LUPI) as a framework to incorporate annotation disagreements into the classifiers. In this framework, there is extra information (in our case *confidence* in annotations) for each data sample that is only accessible at training time. We exploit this extra information as the way to discriminate between easy and difficult examples. We proposed a model based on Gaussian Process framework in which the influence of difficult instances in the training process is reduced, retained, or even ignored. The proposed method uses an efficient quadrature-free expectation propagation algorithm for approximate inference therefore it is faster to train than existing LUPI methods: 42m for ours v. 1h13m for GPC+ and 4h12m for SVM+, the two baseline LUPI methods. We showed that classifiers could benefit from incorporating annotation ambiguities into the learning process.

There are emerging research trends in coupling kernel methods/Gaussian processes and deep CNNs models (see for example: [28, 35]). The main idea is to learn the convolutional layers as in deep CNNs and replace fully connected layers with the nonparametric kernel functions. Coupling our GPC^{conf} model and the deep-er kernels concept is an attractive future direction. To achieve this in practice, we will need a large dataset with confidence annotations.

Acknowledgment

VS and NQ thank AWS Cloud Credits for Research. DHL acknowledges support from Plan Nacional I+D+i, grants TIN2013-42351-P and TIN2015-70308-REDT, and from Comunidad de Madrid, grant S2013/ICE-2845 CASI-CAM-CM. JMHL acknowledges support from the Rafael del Pino foundation. VS and DHL contributed equally.

References

[1] M. Buhrmester, T. Kwang, and S. D. Gosling. Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, pages 3–5, 2011. 1, 2

[2] T. Cohn and L. Specia. Modelling annotator bias with multi-task gaussian processes: An application to machine translation quality estimation. In *ACL*, 2013. 2

[3] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, pages 20–28, 1979. 2

[4] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research (JMLR)*, 2006. 6, 7

[5] E.Snelson and Z.Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *NIPS*, 2006. 4

[6] D. Hernández-Lobato, J. M. Hernández-Lobato, and P. Dupont. Robust multi-class Gaussian process classification. In *NIPS*, 2011. 3

[7] D. Hernández-Lobato, V. Sharmanska, K. Kersting, C. H. Lampert, and N. Quadrianto. Mind the nuisance: Gaussian process classification using privileged noise. In *NIPS*, 2014. 3, 5, 6, 8

[8] P. G. Ipeirotis, F. Provost, V. Sheng, and J. Wang. Repeated labeling using multiple noisy labelers. *Data Mining and Knowledge Discovery*, pages 402–441, 2014. 2

[9] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 5, 8

[10] G. Kazai, J. Kamps, and N. Milic-Frayling. Worker types and personality traits in crowdsourcing relevance labels. In *CIKM*, 2011. 1

[11] A. Kovashka and K. Grauman. Attribute adaptation for personalized image search. In *ICCV*, 2013. 2

[12] M. Kuss and C. E. Rasmussen. Assessing approximate inference for binary Gaussian process classification. *Journal of Machine Learning Research (JMLR)*, 2005. 3

[13] C. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Trans. on Pattern Analysis and Machine Intelligence (T-PAMI)*, pages 453–465, 2014. 5

[14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1

[15] C. Long and G. Hua. Multi-class multi-annotator active learning with robust Gaussian Process for visual recognition. In *ICCV*, 2015. 2

[16] T. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Massachusetts Institute of Technology, 2001. 4

[17] A. Naish-Guzman and S. Holden. The generalized FITC approximation. In *NIPS*, 2008. 4

[18] H. Nickisch and C. E. Rasmussen. Approximations for binary Gaussian process classification. *Journal of Machine Learning Research (JMLR)*, 2008. 3

[19] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011. 2

[20] G. Patterson, C. Xu, H. Su, and J. Hays. The SUN Attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision (IJCV)*, pages 59–81, 2014. 1, 2, 5

[21] D. Pechyony and V. Vapnik. Fast optimization algorithms for solving SVM+. In *Statistical Learning and Data Science*, 2011. 4

[22] A. Pentina, V. Sharmanska, and C. H. Lampert. Curriculum learning of multiple tasks. In *CVPR*, 2015. 1, 5

[23] C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006. 3, 4

[24] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, et al. Learning from crowds. *Journal of Machine Learning Research (JMLR)*, pages 1297–1322, 2010. 2

[25] F. Rodrigues, F. Pereira, and B. Ribeiro. GP classification and active learning with multiple annotators. In *ICML*, 2014. 2

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, pages 1–42, 2015. 1, 2, 8

[27] M. Seeger. Expectation propagation for exponential families. Technical report, University of California, Berkeley, 2006. 4

[28] J. Shawe-Taylor. Deep-er kernels. In *ICPRAM*, 2014. 8

[29] P. Smyth, U. M. Fayyad, M. C. Burl, P. Perona, and P. Baldi. Inferring ground truth from subjective labelling of venus images. In *NIPS*, 1995. 2

[30] A. Sorokin and D. Forsyth. Utility data annotation with amazon mechanical turk. In *CVPR Workshop*, 2008. 1, 2

[31] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: a large database for non-parametric object and scene recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence (T-PAMI)*, pages 1958–1970, 2008. 1

[32] V. Vapnik. *Estimation of Dependences Based on Empirical Data: Springer Series in Statistics*. Springer, 1982. 2

[33] V. Vapnik and R. Izmailov. Learning using privileged information: Similarity control and knowledge transfer. *Journal of Machine Learning Research (JMLR)*, pages 2023–2049, 2015. 2

[34] V. Vapnik and A. Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 2009. 2, 3, 6, 8

[35] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep kernel learning. In *AISTATS*, 2016. 8

[36] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 5

[37] Y. Yan, R. Rosales, G. Fung, M. W. Schmidt, G. H. Valadez, L. Bogoni, L. Moy, and J. G. Dy. Modeling annotator expertise: Learning when everybody knows a bit of something. In *AISTATS*, 2010. 2

[38] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014. 1, 5