

Supplementary Material – Ambiguity Helps: Classification with Disagreements in Crowdsourced Annotations

Viktoriia Sharmanska^{*,1}, Daniel Hernández-Lobato^{*,2}, José Miguel Hernández-Lobato³, Novi Quadrianto¹

¹SMiLe CLiNiC, University of Sussex, Brighton, UK

²Universidad Autónoma de Madrid, Madrid, Spain

³Harvard University, Cambridge, Massachusetts, US

1. Expectation Propagation (EP)

In this section we give all the necessary details to implement the EP algorithm [1] for the GPC^{conf} method described in the main manuscript. We show how to compute the EP posterior approximation from the product of all approximate factors and how to implement the EP updates to refine each approximate factor. We also show how to compute the EP approximation of the marginal likelihood and its gradients.

Recall from the main manuscript that in EP the approximate factors replace the corresponding exact factors of the likelihood in the joint distribution $p(\mathbf{y}|\mathbf{X}, \mathbf{X}^{\text{conf}}, \mathbf{f}, \mathbf{g})p(\mathbf{f})p(\mathbf{g})$. The resulting approximate joint distribution is then normalized to get the EP posterior approximation, and the normalization constant is the approximation to the marginal likelihood. Furthermore, we would like to recall that in the case of GPC^{conf}, the n -th likelihood factor to be approximated by EP is:

$$\begin{aligned} h_n(f(\mathbf{x}_n), g(\mathbf{x}_n^{\text{conf}})) \\ = \Theta(y_n f(\mathbf{x}_n))^{1-\Theta(g(\mathbf{x}_n^{\text{conf}}))} (1/2)^{\Theta(g(\mathbf{x}_n^{\text{conf}}))}. \end{aligned} \quad (1)$$

The corresponding approximate factor whose parameters are found by EP is:

$$\begin{aligned} \tilde{h}_n(f(\mathbf{x}_n), g(\mathbf{x}_n^{\text{conf}})) \\ = \tilde{s}_n \mathcal{N}(f(\mathbf{x}_n) | \tilde{m}_n, \tilde{v}_n) \mathcal{N}(g(\mathbf{x}_n^{\text{conf}}) | \tilde{\mu}_n, \tilde{\nu}_n). \end{aligned} \quad (2)$$

1.1. Reconstruction of the Posterior Approximation

In this section we show how to compute the posterior approximation given the approximate factors. This step is carried out in EP after updating all approximate factors. To implement it, we use the joint approximation \tilde{q} , which is defined as the product of all approximate factors and the factors that need not be approximated. Assume N training instances. Then,

$$\tilde{q}(\mathbf{f}, \mathbf{g}) = p(\mathbf{f})p(\mathbf{g}) \prod_{n=1}^N \tilde{h}_n(f(\mathbf{x}_n), g(\mathbf{x}_n^{\text{conf}})), \quad (3)$$

where

$$\begin{aligned} \tilde{h}_n(f(\mathbf{x}_n), g(\mathbf{x}_n^{\text{conf}})) \\ = \tilde{s}_n \mathcal{N}(f(\mathbf{x}_n) | \tilde{m}_n, \tilde{v}_n) \mathcal{N}(g(\mathbf{x}_n^{\text{conf}}) | \tilde{\mu}_n, \tilde{\nu}_n). \end{aligned} \quad (4)$$

Recall also from the main manuscript that the priors for \mathbf{f} and \mathbf{g} are Gaussian. That is, $p(\mathbf{f}) = \mathcal{N}(\mathbf{f} | \mathbf{0}, \mathbf{C}_f)$ and $p(\mathbf{g}) = \mathcal{N}(\mathbf{g} | m_g, \mathbf{C}_g)$, where m_g is the prior mean of g . Thus, $\tilde{q}(\mathbf{f}, \mathbf{g})$ is an unnormalized distribution inside a family \mathcal{F} of exponential distributions. This family is a product of a multi-variate Gaussian on \mathbf{f} and another multi-variate Gaussian on \mathbf{g} . The consequence is that after the normalization of \tilde{q} , we get the EP posterior approximation which is:

$$q(\mathbf{f}, \mathbf{g}) = \frac{\tilde{q}(\mathbf{f}, \mathbf{g})}{Z_q} = \mathcal{N}(\mathbf{f} | \boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f) \mathcal{N}(\mathbf{g} | \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g), \quad (5)$$

where the parameters of this distribution are computed simply by summing the natural parameters of all the Gaussian distributions that appear in \tilde{q} . For this, the formulas relative to the product of Gaussians that appear in the appendix of [2] can be used. Using these formulas, we have that:

$$\begin{aligned} \boldsymbol{\Sigma}_f &= (\boldsymbol{\Pi}_f + \mathbf{C}_f^{-1})^{-1}, & \boldsymbol{\Sigma}_g &= (\boldsymbol{\Pi}_g + \mathbf{C}_g^{-1})^{-1}, \\ \boldsymbol{\mu}_f &= \boldsymbol{\Sigma}_f \mathbf{v}_f, & \boldsymbol{\mu}_g &= \boldsymbol{\Sigma}_g (\mathbf{v}_g + \mathbf{C}_g^{-1} \mathbf{m}_g), \end{aligned} \quad (6)$$

where $\boldsymbol{\Pi}_f$ and $\boldsymbol{\Pi}_g$ are diagonal matrices with all entries equal to \tilde{v}_n^{-1} and $\tilde{\nu}_n^{-1}$, respectively; \mathbf{v}_f is a vector whose i -th component is equal to \tilde{m}_n/\tilde{v}_n ; \mathbf{v}_g is a vector whose i -th component is equal to $\tilde{\mu}_n/\tilde{\nu}_n$; and \mathbf{m}_g is a vector with all components equal to m_g , the prior mean of g .

The computations just described scale like $\mathcal{O}(N^3)$, where N is the number of training instances. A better cost is obtained if the FITC approximation is used. In that case, as explained in the main manuscript, we have that

$$\mathbf{C}_f = \mathbf{D}_f + \mathbf{P}_f \mathbf{R}_f^T \mathbf{R}_f \mathbf{P}_f^T, \quad \mathbf{C}_g = \mathbf{D}_g + \mathbf{P}_g \mathbf{R}_g^T \mathbf{R}_g \mathbf{P}_g^T, \quad (7)$$

where $\mathbf{D}_f = \text{diag}(\mathbf{C}_f - \mathbf{K}_f)$, with $\mathbf{K}_f = \mathbf{C}_{\bar{x}\bar{x}}\mathbf{C}_{\bar{x}\bar{x}}^{-1}\mathbf{C}_{\bar{x}\bar{x}}^T$; and $\mathbf{D}_g = \text{diag}(\mathbf{C}_g - \mathbf{K}_g)$, with $\mathbf{K}_g = \mathbf{C}_{\bar{x}^{\text{conf}}\bar{x}^{\text{conf}}}\mathbf{C}_{\bar{x}^{\text{conf}}\bar{x}^{\text{conf}}}^{-1}\mathbf{C}_{\bar{x}^{\text{conf}}\bar{x}^{\text{conf}}}^T$. Furthermore, $\mathbf{P}_f = \mathbf{C}_{\bar{x}\bar{x}}$ and $\mathbf{P}_g = \mathbf{C}_{\bar{x}^{\text{conf}}\bar{x}^{\text{conf}}}$. Finally, \mathbf{R}_f and \mathbf{R}_g are the Cholesky decompositions of $\mathbf{C}_{\bar{x}\bar{x}}^{-1}$ and $\mathbf{C}_{\bar{x}^{\text{conf}}\bar{x}^{\text{conf}}}^{-1}$, respectively. See [3] for further details. The Woodbury formula provides an efficient method to compute \mathbf{C}_g^{-1} , which is required to evaluate μ_g .

If the FITC approximation is used, then Σ_f and Σ_g have a special structure that can be computed with cost $\mathcal{O}(M^2N)$, where M is the number of pseudo-inputs (recall that in general $M \ll N$). In particular,

$$\begin{aligned}\Sigma_f &= \mathbf{D}_f^{\text{new}} + \mathbf{P}_f^{\text{new}}(\mathbf{R}_f^{\text{new}})^T\mathbf{R}_f^{\text{new}}(\mathbf{P}_f^{\text{new}})^T, \\ \Sigma_g &= \mathbf{D}_g^{\text{new}} + \mathbf{P}_g^{\text{new}}(\mathbf{R}_g^{\text{new}})^T\mathbf{R}_g^{\text{new}}(\mathbf{P}_g^{\text{new}})^T,\end{aligned}\quad (8)$$

where $\mathbf{R}_f^{\text{new}}$ and $\mathbf{R}_g^{\text{new}}$ are matrices of size $M \times M$, and $\mathbf{P}_f^{\text{new}}$ and $\mathbf{P}_g^{\text{new}}$ are matrices of size $N \times M$. Finally, the matrices $\mathbf{D}_f^{\text{new}}$ and $\mathbf{D}_g^{\text{new}}$ are diagonal. The practical computation of all these matrices is described in detail in [3] and also in the appendix of [4].

1.2. Computation of an Old Distribution

To update the parameters of each of the approximate factors we have to compute first a (cavity) distribution q^{-n} that is obtained by removing the corresponding approximate factor \tilde{h} from the posterior q . Namely, $q^{-n} = q^{\text{old}}/\tilde{h}_n$. The natural parameters of q^{-n} are exactly the same natural parameters of q , but where we have subtracted from the natural parameters of q the natural parameters of the approximate factor \tilde{h}_n . Importantly, because \tilde{h}_n only depends on $f(\mathbf{x}_n)$ and $g(\mathbf{x}_n^{\text{conf}})$, we can simply marginalize all other latent variables of the model and consider q^{-n} to be the product of two uni-variate Gaussians. This is known as the EP *locality* property [5]. Thus, we have that:

$$\begin{aligned}q^{-n}(f(\mathbf{x}_n), g(\mathbf{x}_n^{\text{conf}})) \\ = \mathcal{N}(f(\mathbf{x}_n)|(\mu_f)_n^{\text{old}}, (\Sigma_f)_{n,n}^{\text{old}})\mathcal{N}(g(\mathbf{x}_n^{\text{conf}})|(\mu_g)_n^{\text{old}}, (\Sigma_g)_{n,n}^{\text{old}}),\end{aligned}\quad (9)$$

where the parameters of q^{-n} can be easily obtained using again the formulas for the ratio of Gaussians that appear in the appendix of [2]. In particular,

$$\begin{aligned}(\Sigma_f)_{n,n}^{\text{old}} &= ((\Sigma_f)_{n,n}^{-1} - \tilde{v}_n^{-1})^{-1}, \\ (\mu_f)_n^{\text{old}} &= (\Sigma_f)_{n,n}^{\text{old}}((\Sigma_f)_{n,n}^{-1}(\mu_f)_n - \tilde{m}_n/\tilde{v}_n), \\ (\Sigma_g)_{n,n}^{\text{old}} &= ((\Sigma_g)_{n,n}^{-1} - \tilde{v}_n^{-1})^{-1}, \\ (\mu_g)_n^{\text{old}} &= (\Sigma_g)_{n,n}^{\text{old}}((\Sigma_g)_{n,n}^{-1}(\mu_g)_n - \tilde{\mu}_n/\tilde{v}_n),\end{aligned}\quad (10)$$

where all variables with the superscript $\tilde{\cdot}$ are the corresponding parameters of \tilde{h}_n . The other variables involved in the computations are simply the parameters of the Gaussian

marginals of q . Thus, the only things required to compute q^{-n} are the marginals of the posterior approximation q and the parameters of the approximate factor to be updated \tilde{h}_n .

1.3. Update of the Approximate Factors

In this section we describe how to update each approximate factor \tilde{h}_n . In EP we refine in parallel each \tilde{h}_n , as in [6]. That is, we will update all approximate factors at the same time. This means that we will compute q^{-n} for all approximate factors. This is a relatively easy step that can be carried out with cost linear in N , *i.e.*, the total number of training points. As indicated in the main manuscript, the first step towards updating each \tilde{h}_n is to compute the normalization constant of $h_n q^{-n}$, denoted by Z_{h_n} . Recall that h_n is the exact likelihood factor described in the main manuscript. In this step, we exploit the fact that all computations have a closed form expression which is given below. In particular,

$$\begin{aligned}Z_{h_n} &= \int q^{-n}(f(\mathbf{x}_n), g(\mathbf{x}_n^{\text{conf}}))h_n(f(\mathbf{x}_n), g(\mathbf{x}_n^{\text{conf}}))df(\mathbf{x}_n)dg(\mathbf{x}_n^{\text{conf}}) \\ &= \int q^{-n}(f(\mathbf{x}_n), g(\mathbf{x}_n^{\text{conf}}))\Theta(y_n f(\mathbf{x}_n))^{1-\Theta(g(\mathbf{x}_n^{\text{conf}}))} \times \\ &\quad \times (1/2)^{\Theta(g(\mathbf{x}_n^{\text{conf}}))}df(\mathbf{x}_n)dg(\mathbf{x}_n^{\text{conf}}) \\ &= \Phi\left(\frac{(\mu_f)_n^{\text{old}}}{\sqrt{(\Sigma_f)_{n,n}^{\text{old}}}}\right)\Phi\left(\frac{-(\mu_g)_n^{\text{old}}}{\sqrt{(\Sigma_g)_{n,n}^{\text{old}}}}\right) + \\ &\quad + (1/2)\Phi\left(\frac{(\mu_g)_n^{\text{old}}}{\sqrt{(\Sigma_g)_{n,n}^{\text{old}}}}\right),\end{aligned}\quad (11)$$

where $\Phi(\cdot)$ is the c.d.f. of a standard Gaussian distribution.

Given Z_{h_n} , we can find the parameters of the updated distribution q^{new} , that minimizes $\text{KL}(h_n q^{-n}/Z_{h_n}||q^{\text{new}})$ very easily, for each factor h_n . In particular, because q^{new} has the same form as q^{-n} , and q^{-n} belongs to the exponential family (it is the product of two uni-variate Gaussians), we simply have to match the first and second moments between $h_n q^{-n}/Z_{h_n}$ and q^{new} [7]. This will give the optimal parameters of q^{new} , which are needed to compute the parameters of the updated factor \tilde{h}_n . Importantly, it is possible to find the required moments of $h_n q^{-n}/Z_{h_n}$ from the derivatives of $\log Z_{h_n}$, as indicated by [5]. In particular, we have

that for $n = 1, \dots, N$:

$$\begin{aligned}
\mathbb{E}_{h_n q^{-n}/Z_{h_n}}[f(\mathbf{x}_n)] &= (\mu_f)_n^{\text{old}} + (\Sigma_f)_{n,n}^{\text{old}} \frac{d \log Z_{h_n}}{d(\mu_f)_n^{\text{old}}}, \\
\mathbb{E}_{h_n q^{-n}/Z_{h_n}}[g(\mathbf{x}_n^{\text{conf}})] &= (\mu_g)_n^{\text{old}} + (\Sigma_g)_{n,n}^{\text{old}} \frac{d \log Z_{h_n}}{d(\mu_g)_n^{\text{old}}}, \\
\mathbb{E}_{h_n q^{-n}/Z_{h_n}}[f(\mathbf{x}_n)^2] - \mathbb{E}_{h_n q^{-n}/Z_{h_n}}[f(\mathbf{x}_n)]^2 \\
&= ((\Sigma_f)_{n,n}^{\text{old}})^2 \frac{d^2 \log Z_{h_n}}{d((\mu_f)_n^{\text{old}})^2} + (\Sigma_f)_{n,n}^{\text{old}}, \\
\mathbb{E}_{h_n q^{-n}/Z_{h_n}}[g(\mathbf{x}_n^{\text{conf}})^2] - \mathbb{E}_{h_n q^{-n}/Z_{h_n}}[g(\mathbf{x}_n^{\text{conf}})]^2 \\
&= ((\Sigma_g)_{n,n}^{\text{old}})^2 \frac{d^2 \log Z_{h_n}}{d((\mu_g)_n^{\text{old}})^2} + (\Sigma_g)_{n,n}^{\text{old}}. \tag{12}
\end{aligned}$$

See the appendix of [2] for further details on these expressions. We do not show here the computation of the required derivatives of $\log Z_{h_n}$ because, although tedious, they can be easily obtained using standard derivation rules.

Given the moments described above we can compute q^{new} for each factor h_n . In particular,

$$\begin{aligned}
q^{\text{new}}(f(\mathbf{x}_n), g(\mathbf{x}_n^{\text{conf}})) \\
= \mathcal{N}(f(\mathbf{x}_n) | (\mu_f)_n^{\text{new}}, (\Sigma_f)_{n,n}^{\text{new}}) \mathcal{N}(g(\mathbf{x}_n^{\text{conf}}) | (\mu_g)_n^{\text{new}}, (\Sigma_g)_{n,n}^{\text{new}}), \tag{13}
\end{aligned}$$

where

$$\begin{aligned}
(\mu_f)_n^{\text{new}} &= \mathbb{E}_{h_n q^{-n}/Z_{h_n}}[f(\mathbf{x}_n)], \\
(\mu_g)_n^{\text{new}} &= \mathbb{E}_{h_n q^{-n}/Z_{h_n}}[g(\mathbf{x}_n^{\text{conf}})], \\
(\Sigma_f)_{n,n}^{\text{new}} &= \mathbb{E}_{h_n q^{-n}/Z_{h_n}}[f(\mathbf{x}_n)^2] - \mathbb{E}_{h_n q^{-n}/Z_{h_n}}[f(\mathbf{x}_n)]^2, \\
(\Sigma_g)_{n,n}^{\text{new}} &= \mathbb{E}_{h_n q^{-n}/Z_{h_n}}[g(\mathbf{x}_n^{\text{conf}})^2] - \mathbb{E}_{h_n q^{-n}/Z_{h_n}}[g(\mathbf{x}_n^{\text{conf}})]^2. \tag{14}
\end{aligned}$$

Once that we have computed q^{new} we can simply obtain the corresponding approximate factor as $\tilde{h}_n = Z_{h_n} q^{\text{new}}/q^{-n}$. This means that we can compute the parameters of \tilde{h}_n using the formulas relative to the ratio of Gaussians that appear in the appendix of [2]. In particular, for each \tilde{h}_n we have that:

$$\begin{aligned}
\tilde{v}_n &= \left(((\Sigma_f)_{n,n}^{\text{new}})^{-1} - ((\Sigma_f)_{n,n}^{\text{old}})^{-1} \right)^{-1}, \\
\tilde{\nu}_n &= \left(((\Sigma_g)_{n,n}^{\text{new}})^{-1} - ((\Sigma_g)_{n,n}^{\text{old}})^{-1} \right)^{-1}, \\
\tilde{m}_n &= \tilde{v}_n \left(((\Sigma_f)_{n,n}^{\text{new}})^{-1} (\mu_f)_n^{\text{new}} - ((\Sigma_f)_{n,n}^{\text{old}})^{-1} (\mu_f)_n^{\text{old}} \right), \\
\tilde{\mu}_n &= \tilde{\nu}_n \left(((\Sigma_g)_{n,n}^{\text{new}})^{-1} (\mu_g)_n^{\text{new}} - ((\Sigma_g)_{n,n}^{\text{old}})^{-1} (\mu_g)_n^{\text{old}} \right). \tag{15}
\end{aligned}$$

The parameter \tilde{s}_n of \tilde{h}_n is set so that $\tilde{h}_n q^{-n}$ and $h_n \tilde{q}^{\text{old}}$

integrate the same. This gives:

$$\begin{aligned}
\tilde{s}_n &= Z_{h_n} \sqrt{\frac{2\pi \tilde{v}_n (\Sigma_f)_{n,n}^{\text{old}}}{(\Sigma_f)_{n,n}^{\text{new}}}} \sqrt{\frac{2\pi \tilde{\nu}_n (\Sigma_g)_{n,n}^{\text{old}}}{(\Sigma_g)_{n,n}^{\text{new}}}} \\
&\times \exp \left\{ \frac{1}{2} \left(\frac{\tilde{m}_n^2}{\tilde{v}_n} + \frac{((\mu_f)_n^{\text{old}})^2}{(\Sigma_f)_{n,n}^{\text{old}}} - \frac{((\mu_f)_n^{\text{new}})^2}{(\Sigma_f)_{n,n}^{\text{new}}} \right) \right\} \\
&\times \exp \left\{ \frac{1}{2} \left(\frac{\tilde{\mu}_n^2}{\tilde{\nu}_n} + \frac{((\mu_g)_n^{\text{old}})^2}{(\Sigma_g)_{n,n}^{\text{old}}} - \frac{((\mu_g)_n^{\text{new}})^2}{(\Sigma_g)_{n,n}^{\text{new}}} \right) \right\}. \tag{16}
\end{aligned}$$

In these updates we have ignored the damping effect. Damping is very easy to implement. It simply consists in enforcing the natural parameters of the new \tilde{h}_n to be a convex combination of the natural parameters of the old \tilde{h}_n and the new \tilde{h}_n , as described by the equations above¹. That is, we set $\tilde{h}_n = (\tilde{h}_n^{\text{new}})^\alpha (\tilde{h}_n^{\text{old}})^{1-\alpha}$, where \tilde{h}_n^{new} and \tilde{h}_n^{old} denote the new and the old approximate factor and $\alpha \in [0, 1]$ is a damping parameter. By using a damping scheme we avoid strong changes in each \tilde{h}_n and improve convergence [8], in general. For further details, we refer the reader to the R code provided.

1.4. Marginal Likelihood Approximation

In this section we describe how to compute, once EP has converged, the normalization constant Z_q of \tilde{q} . We recall that this constant can be used to approximate the marginal likelihood of the model. To compute its value, we use again the formulas for the product of Gaussian distributions that are given in the appendix of [2]. In particular,

$$\begin{aligned}
Z_q &= \left[\prod_{n=1}^N \tilde{s}_n \right] \sqrt{\frac{|\Sigma_f|}{|\mathbf{C}_f|}} \left[\prod_{n=1}^N \frac{1}{\sqrt{2\pi \tilde{v}_n}} \right] \sqrt{\frac{|\Sigma_g|}{|\mathbf{C}_g|}} \left[\prod_{n=1}^N \frac{1}{\sqrt{2\pi \tilde{\nu}_n}} \right] \times \\
&\times \exp \left\{ -\frac{1}{2} \left(\sum_{n=1}^N \frac{\tilde{m}_n^2}{\tilde{v}_n} - \boldsymbol{\mu}_f^T \boldsymbol{\Sigma}_f^{-1} \boldsymbol{\mu}_f \right) \right\} \times \\
&\times \exp \left\{ -\frac{1}{2} \left(\sum_{n=1}^N \frac{\tilde{\mu}_n^2}{\tilde{\nu}_n} + \mathbf{m}_g^T \mathbf{C}_g^{-1} \mathbf{m}_g - \boldsymbol{\mu}_g^T \boldsymbol{\Sigma}_g^{-1} \boldsymbol{\mu}_g \right) \right\}, \tag{17}
\end{aligned}$$

where all computations depending on the matrices \mathbf{C}_f , \mathbf{C}_g , $\boldsymbol{\Sigma}_f$ and $\boldsymbol{\Sigma}_g$ can be carried out efficiently with cost $\mathcal{O}(M^2 N)$, using the special form for these matrices described in Section 1.1. The details are found in [3] and the appendix of [4].

1.5. Optimization of the Marginal Likelihood Approximation

To find all hyper-parameters of the model (e.g., covariance hyper-parameters and pseudo-inputs in the FITC ap-

¹The first natural parameters of \tilde{h}_n are \tilde{v}_n^{-1} and $\tilde{\nu}_n^{-1}$. The second natural parameters are \tilde{m}_n/\tilde{v}_n and $\tilde{\mu}_n/\tilde{\nu}_n$.

proximation) we employ a type-II maximum likelihood approach that maximizes $\log Z_q$ using gradient ascent. For this, we require the gradients of $\log Z_q$ with respect to all the hyper-parameters. As described in [5] it is possible to easily compute these gradients if EP has converged because the gradients of $\log Z_q$ with respect to the parameters of each \tilde{h}_n are equal to zero. This means that for gradient computation purposes, we can consider the approximate factors to be fixed. Consider θ to be the natural parameters of the prior (either the prior for f or g). [5] shows that

$$\nabla_{\theta} \log Z_q = \boldsymbol{\eta}^{\text{post}} - \boldsymbol{\eta}^{\text{prior}}, \quad (18)$$

where $\boldsymbol{\eta}^{\text{prior}}$ are the moments (the first and second moments) of the prior distribution and $\boldsymbol{\eta}^{\text{post}}$ are the moments of the EP posterior approximation. For example, if θ contains the first natural parameter of the prior for f , *i.e.*, the inverse of the covariance matrix, \mathbf{C}_f^{-1} , $\boldsymbol{\eta}^{\text{prior}}$ is $-0.5\mathbf{C}_f$ and $\boldsymbol{\eta}^{\text{post}}$ is $-0.5(\boldsymbol{\Sigma}_f + \boldsymbol{\mu}_f\boldsymbol{\mu}_f^{\text{T}})$, because the prior mean of f is zero. By contrast, if θ contains the first natural parameter of the prior for g , \mathbf{C}_g^{-1} , $\boldsymbol{\eta}^{\text{prior}}$ is $-0.5(\mathbf{C}_g + \mathbf{m}_g\mathbf{m}_g^{\text{T}})$ and $\boldsymbol{\eta}^{\text{post}}$ is $-0.5(\boldsymbol{\Sigma}_g + \boldsymbol{\mu}_g\boldsymbol{\mu}_g^{\text{T}})$. Finally, if θ contains the second natural parameter of the prior for g , *i.e.*, the inverse of the covariance matrix times the mean vector, $\mathbf{C}_g^{-1}\mathbf{m}_g$, $\boldsymbol{\eta}^{\text{prior}}$ is \mathbf{m}_g and $\boldsymbol{\eta}^{\text{post}}$ is $\boldsymbol{\mu}_g$.

We show now how to compute the gradients with respect to the covariance matrix of each prior distribution. That is, \mathbf{C}_f and \mathbf{C}_g , since we are in particular interested in computing the gradients with respect to the covariance function parameters and the pseudo-inputs. For this, we recall that the first natural parameter is the inverse of the covariance function. Thus, we have that

$$\nabla_{\mathbf{C}_f} \log Z_q = -\frac{1}{2}\mathbf{C}_f^{-1} (\mathbf{C}_f - \boldsymbol{\Sigma}_f - \boldsymbol{\mu}_f\boldsymbol{\mu}_f^{\text{T}}) \mathbf{C}_f^{-1}. \quad (19)$$

where we have used the chain rule for matrix derivation. See [9] for further details. Consider that θ is a parameter of the covariance function of f (*e.g.*, the length-scale or a pseudo-input). If we use again the chain rule for matrix derivation we have that

$$\nabla_{\theta} \log Z_q = -\frac{1}{2}\text{trace} \left(\mathbf{C}_f^{-1} (\mathbf{C}_f - \boldsymbol{\Sigma}_f - \boldsymbol{\mu}_f\boldsymbol{\mu}_f^{\text{T}}) \mathbf{C}_f^{-1} \frac{d\mathbf{C}_f}{d\theta} \right). \quad (20)$$

See [9] for further details. Thus, all derivatives can be obtained by computing the matrix $\mathbf{C}_f^{-1} (\mathbf{C}_f - \boldsymbol{\Sigma}_f - \boldsymbol{\mu}_f\boldsymbol{\mu}_f^{\text{T}}) \mathbf{C}_f^{-1}$. In practice, after some algebra, it is possible to employ the special structure of all the covariance matrices given in Section 1.1 to evaluate these quantities efficiently with cost $\mathcal{O}(NM^2)$. For example, \mathbf{C}_f^{-1} can be easily evaluated using the Woodbury formula. See the appendices of [10] and [4] for further details.

A similar derivation can be carried out in the case of the parameters corresponding to the covariance function of the g matrix. However, in this case we have to take into account that the prior mean of g is the vector \mathbf{m}_g , which is different from zero. In particular,

$$\nabla_{\mathbf{C}_g} \log Z_q = -\frac{1}{2}\mathbf{C}_g^{-1} (\mathbf{C}_g + \mathbf{m}_g\mathbf{m}_g^{\text{T}} - \boldsymbol{\Sigma}_g - \boldsymbol{\mu}_g\boldsymbol{\mu}_g^{\text{T}}) \mathbf{C}_g^{-1} + \mathbf{C}_g^{-1} (\mathbf{m}_g\mathbf{m}_g^{\text{T}} - \boldsymbol{\mu}_g\mathbf{m}_g^{\text{T}}) \mathbf{C}_g^{-1}. \quad (21)$$

Consider that θ is a parameter of the covariance function of g . If we use again the chain rule for matrix derivation we have that

$$\begin{aligned} \nabla_{\theta} \log Z_q &= -\frac{1}{2}\text{trace} \left(\mathbf{C}_g^{-1} (\mathbf{C}_g + \mathbf{m}_g\mathbf{m}_g^{\text{T}} - \boldsymbol{\Sigma}_g - \boldsymbol{\mu}_g\boldsymbol{\mu}_g^{\text{T}}) \mathbf{C}_g^{-1} \frac{d\mathbf{C}_g}{d\theta} \right) \\ &+ \text{trace} \left(\mathbf{C}_g^{-1} (\mathbf{m}_g\mathbf{m}_g^{\text{T}} - \boldsymbol{\mu}_g\mathbf{m}_g^{\text{T}}) \mathbf{C}_g^{-1} \frac{d\mathbf{C}_g}{d\theta} \right). \quad (22) \end{aligned}$$

Thus, all derivatives can be obtained simply by pre-computing two matrices. These matrices are $\mathbf{C}_g^{-1} (\mathbf{C}_g + \mathbf{m}_g\mathbf{m}_g^{\text{T}} - \boldsymbol{\Sigma}_g - \boldsymbol{\mu}_g\boldsymbol{\mu}_g^{\text{T}}) \mathbf{C}_g^{-1}$ and $\mathbf{C}_g^{-1} (\mathbf{m}_g\mathbf{m}_g^{\text{T}} - \boldsymbol{\mu}_g\mathbf{m}_g^{\text{T}}) \mathbf{C}_g^{-1}$. Furthermore, as in the previous case, in practice, and after some algebra, it is possible to employ the special structure of all the covariance matrices given in Section 1.1 to evaluate these quantities efficiently with cost $\mathcal{O}(NM^2)$. Again, \mathbf{C}_g^{-1} can be easily evaluated using the Woodbury formula.

Finally, the gradient with respect to the prior mean of g , \mathbf{m}_g , is:

$$\nabla_{\mathbf{m}_g} \log Z_q = \mathbf{C}_g^{-1}\mathbf{m}_g - \mathbf{C}_g^{-1}\boldsymbol{\mu}_g. \quad (23)$$

2. Complete Experimental Results

2.1. Recognizing 57 scene attributes

	Test scenario A					Test scenario B				
	GPC image	GPC ^{conf} image+conf	GPC+ [11] image+conf	SVM+ [12] image+conf	SVM image	GPC image	GPC ^{conf} image+conf	GPC+ [11] image+conf	SVM+ [12] image+conf	SVM image
sailing	78.10 ± 1.42	79.60 ± 1.17	79.40 ± 1.34	77.40 ± 1.47	78.30 ± 1.48	91.40 ± 1.25	93.61 ± 1.10	93.80 ± 1.20	90.40 ± 1.21	93.80 ± 0.81
driving	76.30 ± 0.72	78.20 ± 0.87	78.30 ± 1.05	77.10 ± 1.12	78.30 ± 1.20	90.81 ± 1.20	93.42 ± 1.52	93.02 ± 1.40	90.59 ± 0.92	92.61 ± 0.99
biking	76.20 ± 1.71	78.70 ± 1.57	78.90 ± 1.52	75.50 ± 2.09	76.20 ± 1.68	88.40 ± 1.48	91.98 ± 1.56	91.38 ± 1.66	87.39 ± 2.30	88.00 ± 1.59
transport	78.70 ± 0.91	78.90 ± 0.90	79.20 ± 0.91	79.80 ± 1.07	78.40 ± 1.04	90.98 ± 0.86	90.79 ± 1.06	90.59 ± 1.17	90.57 ± 1.30	90.39 ± 1.38
vacation	74.70 ± 1.69	74.60 ± 1.96	75.20 ± 1.99	75.40 ± 1.31	74.60 ± 1.44	83.75 ± 1.31	84.55 ± 1.52	84.35 ± 1.57	83.96 ± 1.21	84.55 ± 1.05
hiking	76.70 ± 1.20	76.80 ± 1.22	77.10 ± 1.09	75.60 ± 1.31	78.30 ± 1.39	92.18 ± 1.00	92.38 ± 0.79	91.79 ± 0.91	91.57 ± 0.98	93.00 ± 1.10
climbing	81.50 ± 0.86	82.20 ± 0.81	82.20 ± 0.81	80.90 ± 0.95	81.40 ± 0.53	93.99 ± 0.94	94.20 ± 1.08	94.20 ± 1.08	92.98 ± 1.14	93.58 ± 0.88
camping	78.80 ± 1.16	79.40 ± 1.21	79.60 ± 1.24	77.70 ± 0.86	78.40 ± 0.89	93.20 ± 1.02	94.21 ± 0.99	94.21 ± 1.14	91.80 ± 1.39	94.40 ± 1.01
reading	75.10 ± 1.38	76.10 ± 1.40	76.10 ± 1.49	76.80 ± 1.24	76.90 ± 1.49	88.17 ± 1.51	90.17 ± 1.37	89.77 ± 1.73	88.98 ± 1.39	90.78 ± 1.80
swimming	76.30 ± 1.24	77.00 ± 0.92	77.00 ± 0.80	76.50 ± 0.85	76.40 ± 1.04	91.60 ± 1.26	94.19 ± 1.08	94.60 ± 0.94	91.41 ± 1.37	91.19 ± 1.26
eating	80.10 ± 0.75	80.40 ± 0.68	80.40 ± 0.80	77.60 ± 3.01	81.40 ± 0.75	92.39 ± 1.12	92.59 ± 0.80	92.39 ± 0.79	88.10 ± 4.05	93.20 ± 1.15
socializing	79.60 ± 0.92	79.50 ± 0.86	79.20 ± 0.83	78.80 ± 1.01	79.00 ± 0.84	92.20 ± 1.20	92.00 ± 1.28	92.01 ± 1.25	89.98 ± 1.20	92.38 ± 1.26
congregating	79.80 ± 1.06	80.20 ± 1.08	79.80 ± 1.02	78.20 ± 1.24	80.20 ± 1.22	92.96 ± 1.39	94.18 ± 1.38	93.58 ± 1.33	91.76 ± 1.39	93.98 ± 0.91
competing	79.00 ± 0.62	80.30 ± 0.85	79.80 ± 0.88	78.90 ± 0.75	79.50 ± 1.12	90.98 ± 0.81	93.97 ± 0.90	93.77 ± 0.96	89.98 ± 0.94	91.56 ± 1.27
sports	79.90 ± 1.24	80.30 ± 1.01	80.80 ± 1.08	79.90 ± 1.43	79.70 ± 1.01	93.18 ± 1.07	94.19 ± 0.99	94.39 ± 1.05	91.17 ± 1.48	93.98 ± 0.90
exercise	78.00 ± 1.12	79.00 ± 1.46	79.10 ± 1.42	77.80 ± 1.33	78.60 ± 1.52	89.39 ± 1.26	90.19 ± 1.30	90.39 ± 1.12	88.58 ± 1.33	89.79 ± 1.79
spectating	82.90 ± 1.26	83.00 ± 1.14	83.30 ± 1.23	81.40 ± 1.21	81.70 ± 0.97	93.60 ± 1.54	94.99 ± 1.07	95.19 ± 1.10	93.18 ± 1.24	94.59 ± 1.10
shopping	81.90 ± 0.77	82.00 ± 0.69	82.10 ± 0.75	80.90 ± 0.77	81.20 ± 0.60	94.58 ± 1.13	95.18 ± 0.81	95.18 ± 0.81	94.00 ± 0.89	94.99 ± 1.17
working	79.50 ± 1.21	79.40 ± 1.34	79.30 ± 1.39	79.90 ± 1.55	79.90 ± 1.31	88.75 ± 1.70	90.37 ± 1.25	89.36 ± 1.44	88.17 ± 1.99	88.37 ± 1.54
railing	71.80 ± 0.97	73.20 ± 1.02	73.40 ± 0.95	71.60 ± 1.23	71.10 ± 1.39	81.54 ± 1.84	82.75 ± 1.70	82.95 ± 1.61	79.78 ± 2.24	80.99 ± 2.24
trees	74.80 ± 0.97	75.90 ± 0.94	75.20 ± 0.82	75.80 ± 0.60	76.30 ± 1.05	89.17 ± 1.50	91.38 ± 1.02	90.98 ± 1.03	89.78 ± 1.11	89.77 ± 1.92
grass	76.60 ± 1.76	77.40 ± 1.34	77.50 ± 1.42	77.70 ± 1.58	77.40 ± 1.72	93.61 ± 1.22	94.01 ± 1.15	94.01 ± 1.09	93.60 ± 1.05	93.01 ± 1.09
vegetation	78.10 ± 1.65	78.60 ± 1.39	78.80 ± 1.35	78.00 ± 1.59	77.20 ± 1.52	91.36 ± 1.35	92.57 ± 1.53	92.78 ± 1.32	90.35 ± 1.58	90.15 ± 1.75
shrubby	76.50 ± 0.91	77.00 ± 0.84	76.70 ± 0.66	77.00 ± 1.07	77.10 ± 1.42	89.18 ± 0.89	89.98 ± 0.89	89.78 ± 0.72	89.37 ± 1.22	90.16 ± 1.55
foliage	79.00 ± 1.07	78.70 ± 1.08	78.60 ± 0.81	77.00 ± 0.98	77.10 ± 1.16	89.20 ± 1.62	91.60 ± 1.14	91.80 ± 1.20	87.99 ± 1.53	90.80 ± 1.28
leaves	75.00 ± 0.89	76.30 ± 0.85	76.10 ± 0.83	74.10 ± 1.58	75.50 ± 1.36	89.60 ± 1.14	91.00 ± 1.04	90.80 ± 1.04	88.99 ± 1.23	89.38 ± 0.98
asphalt	80.80 ± 0.80	81.00 ± 0.93	80.90 ± 0.83	81.50 ± 0.51	81.70 ± 0.62	91.78 ± 0.66	91.78 ± 1.08	91.78 ± 1.08	91.59 ± 1.12	91.19 ± 1.02
shingles	80.90 ± 1.07	80.80 ± 1.04	81.20 ± 1.07	81.10 ± 1.39	81.50 ± 1.19	93.21 ± 0.98	93.22 ± 1.44	93.61 ± 1.31	93.22 ± 1.04	93.20 ± 1.24
brick	76.60 ± 0.97	77.70 ± 0.95	77.60 ± 0.94	76.60 ± 0.55	77.70 ± 1.11	84.58 ± 0.88	85.59 ± 1.22	84.98 ± 1.10	84.77 ± 0.86	85.39 ± 0.95
metal	73.30 ± 1.07	73.60 ± 1.29	73.40 ± 1.23	72.00 ± 1.64	73.10 ± 0.92	85.57 ± 1.41	87.19 ± 0.93	87.58 ± 1.01	82.38 ± 2.35	87.17 ± 1.18
wood	74.10 ± 1.11	75.10 ± 1.01	75.20 ± 1.02	74.10 ± 1.42	72.60 ± 1.54	86.37 ± 1.44	88.17 ± 1.57	88.77 ± 0.95	85.97 ± 1.16	85.15 ± 1.19
rubber	75.80 ± 1.17	77.10 ± 1.10	76.40 ± 1.16	78.40 ± 1.33	78.10 ± 0.91	87.97 ± 1.49	89.97 ± 1.64	89.57 ± 1.66	90.17 ± 1.72	91.78 ± 1.29
cloth	77.60 ± 1.46	76.80 ± 1.06	76.80 ± 1.16	73.70 ± 1.53	78.00 ± 0.80	89.16 ± 1.31	89.17 ± 1.11	89.37 ± 1.03	86.18 ± 1.48	88.37 ± 0.79
rock	77.50 ± 1.41	76.90 ± 1.28	77.30 ± 1.41	74.80 ± 1.26	76.00 ± 1.73	88.38 ± 0.61	88.58 ± 0.80	88.97 ± 0.82	85.58 ± 0.82	87.99 ± 1.48
dir/soil	76.80 ± 1.00	76.60 ± 1.07	76.80 ± 0.98	75.90 ± 1.17	75.90 ± 1.30	89.14 ± 1.35	90.57 ± 0.86	90.57 ± 0.77	89.75 ± 1.11	89.36 ± 1.20
glass	65.40 ± 1.08	67.30 ± 1.45	67.00 ± 1.73	67.40 ± 1.42	67.10 ± 0.94	76.73 ± 1.88	77.54 ± 2.10	77.14 ± 2.19	78.32 ± 1.73	77.33 ± 1.80
ocean	81.40 ± 1.34	81.50 ± 1.21	82.40 ± 1.19	81.80 ± 0.87	80.90 ± 0.62	95.79 ± 0.83	96.40 ± 0.84	96.80 ± 0.51	96.38 ± 0.84	95.20 ± 0.64
stillwater	74.80 ± 1.23	76.00 ± 0.96	75.30 ± 1.01	75.00 ± 1.26	75.80 ± 1.73	87.59 ± 1.58	92.00 ± 1.02	91.00 ± 0.98	87.37 ± 1.17	90.96 ± 1.57
clouds	70.10 ± 1.47	69.30 ± 1.66	69.90 ± 1.68	70.50 ± 1.46	70.40 ± 2.01	83.96 ± 1.90	83.95 ± 2.06	83.56 ± 1.98	83.17 ± 1.79	84.37 ± 2.18
nat.light	77.70 ± 0.98	80.00 ± 1.36	80.20 ± 1.33	77.30 ± 1.29	73.30 ± 1.29	92.97 ± 1.44	94.39 ± 1.12	94.79 ± 1.03	90.75 ± 1.67	93.38 ± 1.02
sunny	70.10 ± 1.46	70.60 ± 1.50	70.30 ± 1.59	68.80 ± 1.56	70.50 ± 1.69	83.36 ± 1.94	85.38 ± 2.11	84.77 ± 2.17	82.73 ± 1.93	84.57 ± 2.31
el.lighting	75.40 ± 1.63	74.90 ± 1.33	74.90 ± 1.23	74.70 ± 1.27	75.10 ± 1.49	86.57 ± 2.08	87.58 ± 1.70	87.38 ± 1.95	86.78 ± 1.96	86.98 ± 2.18
glossy	71.70 ± 1.64	71.90 ± 1.69	72.40 ± 1.63	71.30 ± 1.38	73.00 ± 1.54	80.18 ± 1.12	80.79 ± 1.27	80.99 ± 1.21	78.38 ± 1.25	80.97 ± 0.92
dry	74.50 ± 1.22	75.20 ± 1.20	74.90 ± 1.31	74.60 ± 1.24	74.20 ± 1.38	86.20 ± 1.58	88.59 ± 1.31	87.59 ± 1.47	87.19 ± 0.97	86.79 ± 1.32
warm	72.60 ± 1.34	73.20 ± 1.55	73.40 ± 1.43	72.50 ± 1.37	73.00 ± 1.30	83.78 ± 1.20	82.58 ± 1.48	83.39 ± 1.59	82.58 ± 1.71	83.19 ± 1.55
cold	85.40 ± 0.96	85.50 ± 0.92	85.60 ± 0.90	85.30 ± 0.78	85.20 ± 0.64	94.39 ± 0.84	94.19 ± 0.91	94.19 ± 0.91	94.59 ± 0.95	94.40 ± 0.74
natural	80.10 ± 0.91	80.10 ± 0.74	80.00 ± 0.72	79.50 ± 0.83	80.30 ± 0.66	95.01 ± 0.75	95.41 ± 0.79	95.21 ± 0.80	95.21 ± 0.80	96.59 ± 0.94
man-made	71.90 ± 1.03	73.60 ± 1.27	73.60 ± 1.34	73.60 ± 1.54	73.30 ± 1.61	86.19 ± 1.63	88.39 ± 1.57	87.58 ± 1.62	87.58 ± 1.35	86.79 ± 1.57
open	75.50 ± 1.29	77.50 ± 1.19	77.20 ± 1.11	77.10 ± 1.05	76.80 ± 0.99	93.77 ± 0.84	95.98 ± 0.76	95.98 ± 0.76	93.38 ± 1.07	94.59 ± 0.81
enclosed	81.90 ± 0.68	82.60 ± 0.87	82.40 ± 0.70	80.90 ± 0.92	81.80 ± 0.97	94.40 ± 0.73	96.00 ± 0.80	95.40 ± 0.69	94.42 ± 1.34	96.40 ± 0.97
far-horizon	80.50 ± 0.98	81.20 ± 1.01	80.80 ± 0.96	81.40 ± 1.25	80.40 ± 1.00	93.80 ± 0.99	93.60 ± 1.05	93.60 ± 1.08	93.20 ± 0.94	93.60 ± 0.96
nohorizon	79.00 ± 1.38	79.10 ± 1.24	78.50 ± 1.27	77.30 ± 2.05	78.50 ± 1.37	89.58 ± 1.26	90.37 ± 1.13	89.77 ± 1.16	88.41 ± 2.13	90.19 ± 1.13
rugged	81.70 ± 1.12	82.20 ± 1.10	82.50 ± 1.12	81.00 ± 1.19	80.90 ± 1.33	93.78 ± 0.72	94.99 ± 0.91	94.59 ± 0.80	93.17 ± 0.87	93.38 ± 1.14
vertical	73.30 ± 1.38	74.30 ± 1.36	74.80 ± 1.37	74.60 ± 0.97	73.50 ± 1.25	84.98 ± 1.47	87.18 ± 1.55	87.98 ± 1.62	88.19 ± 1.27	86.39 ± 1.43
horizontal	68.00 ± 1.13	69.30 ± 0.91	69.50 ± 0.82	68.10 ± 1.45	68.40 ± 1.39	79.56 ± 1.67	79.97 ± 1.62	80.37 ± 1.63	79.57 ± 1.93	77.95 ± 1.98
cluttered	78.50 ± 0.98	77.80 ± 0.89	77.40 ± 0.93	77.30 ± 1.11	77.10 ± 0.92	89.39 ± 1.25	89.19 ± 1.28	89.19 ± 1.41	89.59 ± 1.27	88.58 ± 1.38
soothing	71.90 ± 0.70	71.90 ± 1.19	72.50 ± 1.10	72.10 ± 1.20	71.40 ± 1.32	82.77 ± 1.53	83.36 ± 1.71	83.77 ± 1.81	83.37 ± 1.76	82.37 ± 1.79

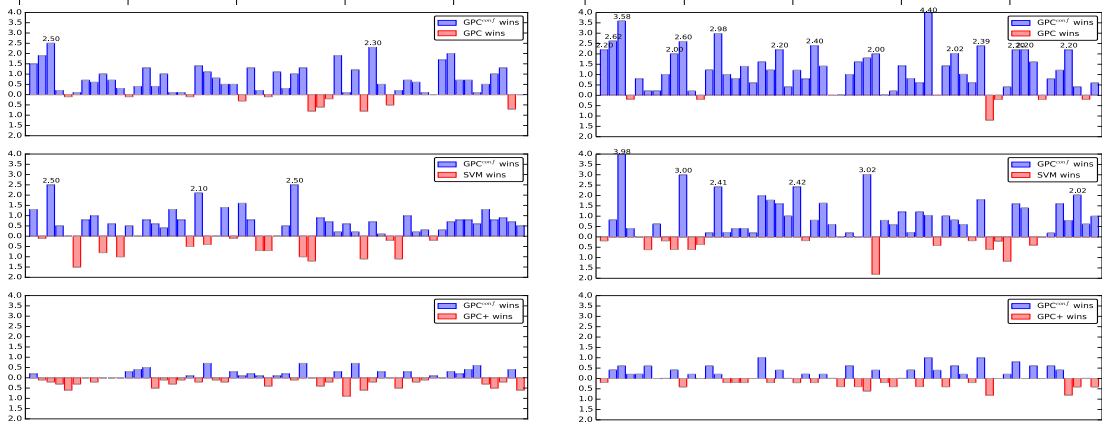


Table 1. Recognizing 57 scene attributes. **Test scenario A**: test images with both 1.0 and 0.66 agreement scores. **Test scenario B**: images with 1.0 score only (high confidence annotations). In both scenarios, training data is the same. **Top**: The numbers are mean and standard error of the accuracy over 10 runs of the 57 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** and an extra **blue** for GPC^{conf} and GPC+. **Bottom**: We also provide a pairwise comparison of the proposed GPC^{conf} and three other methods in terms of difference between the accuracies. The length of the bar corresponds to relative improvement of the accuracy for each of the 57 attributes.

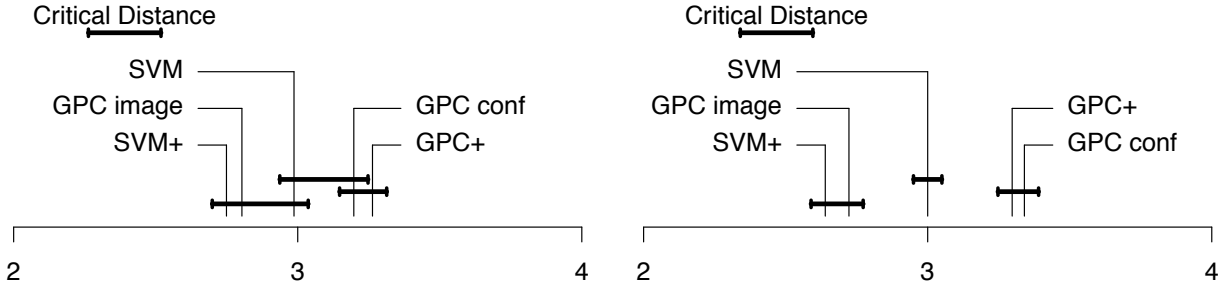


Figure 1. Statistical summary of the results based on Demšar [13] analysis for **Test scenario A** (left), where test images are those with both 1.0 and 0.66 agreement scores, and for **Test scenario B** (right), where test images are those with 1.0 agreement score only. Average rank of the methods (x axis) is computed based on accuracy (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 $> 5\%$).

	GPC image	GPC ^{conf} image+conf	GPC+ [11] image+conf	SVM+ [12] image+conf	SVM image
Chimpanzee	74.86 ± 0.81	74.93 ± 0.76	75.14 ± 0.86	75.07 ± 0.78	73.71 ± 0.90
Giant panda	80.64 ± 0.55	81.17 ± 0.62	80.69 ± 0.57	81.33 ± 0.58	80.53 ± 0.67
Leopard	81.67 ± 0.73	82.00 ± 0.71	81.83 ± 0.63	80.58 ± 0.66	80.42 ± 0.80
Persian cat	79.72 ± 0.44	80.14 ± 0.45	80.21 ± 0.42	79.15 ± 0.77	78.17 ± 1.01
Hippopotamus	72.85 ± 1.05	72.78 ± 1.12	72.85 ± 1.12	73.33 ± 1.41	73.06 ± 1.10
Raccoon	78.57 ± 1.09	78.81 ± 0.88	78.49 ± 0.89	76.98 ± 0.88	76.51 ± 0.59
Rat	84.33 ± 1.58	84.00 ± 1.53	83.83 ± 1.56	83.50 ± 1.80	81.50 ± 1.86
Seal	48.00 ± 1.45	48.10 ± 1.23	48.20 ± 1.33	48.50 ± 0.80	49.20 ± 0.80

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}, GPC+ and SVM+ methods utilize easy-hard score annotation available per image.

2.2. Average Precision as performance measure

	GPC image	GPC ^{conf} image+conf	GPC+ [11] image+conf	SVM+ [12] image+conf	SVM image		GPC image	GPC ^{conf} image+conf	GPC+ [11] image+conf	SVM+ [12] image+conf	SVM image
sailing	84.77 ± 1.16	85.49 ± 1.16	85.41 ± 1.14	84.21 ± 1.29	84.07 ± 0.96	metal	77.50 ± 1.58	78.48 ± 1.67	78.17 ± 1.62	77.12 ± 1.64	79.14 ± 1.41
driving	86.13 ± 1.24	87.47 ± 1.27	87.38 ± 1.24	86.35 ± 1.53	87.77 ± 1.26	paper	77.76 ± 1.71	78.77 ± 1.71	78.32 ± 1.55	77.31 ± 1.56	77.68 ± 1.87
biking	85.12 ± 1.40	85.25 ± 1.34	85.27 ± 1.36	84.90 ± 1.06	84.93 ± 1.09	wood	78.37 ± 1.70	78.60 ± 1.83	78.44 ± 1.70	77.10 ± 2.09	76.54 ± 2.13
transport	87.90 ± 0.79	88.04 ± 1.07	88.12 ± 1.08	87.41 ± 1.04	87.93 ± 1.12	vinyl	71.50 ± 2.15	71.13 ± 2.09	71.77 ± 2.17	71.40 ± 1.96	72.15 ± 2.07
vacation	83.74 ± 1.24	84.29 ± 1.38	84.35 ± 1.36	82.33 ± 1.51	83.61 ± 1.25	rubber	86.23 ± 1.40	86.41 ± 1.59	86.57 ± 1.55	86.07 ± 1.43	86.36 ± 1.63
hiking	85.08 ± 1.80	85.92 ± 2.00	85.75 ± 2.06	85.55 ± 1.93	85.57 ± 2.13	cloth	81.74 ± 1.60	84.04 ± 1.68	83.89 ± 1.65	82.48 ± 1.86	82.00 ± 1.85
climbing	86.97 ± 1.49	87.96 ± 1.52	88.00 ± 1.52	86.89 ± 1.30	87.09 ± 1.47	sand	84.97 ± 1.40	85.63 ± 1.43	85.16 ± 1.42	84.76 ± 1.58	85.42 ± 1.19
camping	81.35 ± 1.59	82.89 ± 1.48	82.84 ± 1.48	81.21 ± 1.89	81.17 ± 1.79	rock	79.22 ± 1.47	79.52 ± 1.55	79.63 ± 1.57	79.50 ± 1.51	80.39 ± 1.26
reading	82.34 ± 2.05	82.81 ± 2.08	83.13 ± 1.98	82.49 ± 1.95	83.20 ± 1.87	dir/soil	84.67 ± 1.34	85.12 ± 1.34	85.12 ± 1.36	82.85 ± 1.76	83.91 ± 1.40
teaching	82.29 ± 1.20	82.74 ± 1.35	82.59 ± 1.28	82.81 ± 0.91	83.35 ± 1.09	glass	74.53 ± 1.22	74.88 ± 1.14	75.21 ± 1.03	74.37 ± 1.48	75.89 ± 1.03
diving	80.86 ± 1.76	79.49 ± 1.40	80.32 ± 1.49	81.63 ± 1.39	79.88 ± 1.65	ocean	88.80 ± 1.12	88.78 ± 1.23	88.74 ± 1.21	88.40 ± 1.13	88.05 ± 1.50
swimming	80.60 ± 0.96	80.29 ± 1.03	80.41 ± 1.00	80.46 ± 0.92	79.74 ± 0.67	run.water	91.05 ± 0.58	91.26 ± 0.51	91.07 ± 0.60	90.60 ± 0.69	90.28 ± 0.89
eating	84.19 ± 1.51	86.48 ± 1.10	86.59 ± 1.12	83.32 ± 1.45	85.46 ± 1.66	stillwater	81.36 ± 1.61	80.66 ± 2.04	80.95 ± 1.96	81.52 ± 1.90	80.53 ± 2.07
socializing	84.14 ± 1.56	84.69 ± 1.60	84.80 ± 1.57	85.60 ± 1.74	85.81 ± 1.44	snow	91.62 ± 1.00	92.79 ± 0.99	92.58 ± 0.98	91.99 ± 0.85	92.17 ± 0.81
congregating	87.16 ± 1.42	87.48 ± 1.28	87.49 ± 1.27	86.24 ± 1.38	86.79 ± 1.32	clouds	78.21 ± 1.62	78.64 ± 1.68	78.47 ± 1.55	78.42 ± 1.40	77.15 ± 1.70
queuing	76.12 ± 0.81	77.08 ± 1.13	77.11 ± 1.11	75.26 ± 0.79	75.37 ± 0.99	nat.light	81.07 ± 2.01	83.37 ± 1.96	83.15 ± 1.98	80.01 ± 2.38	81.92 ± 2.28
competing	86.99 ± 1.61	87.87 ± 1.54	87.90 ± 1.52	87.31 ± 1.79	88.17 ± 1.56	sunny	76.93 ± 1.17	77.78 ± 1.25	77.68 ± 1.31	75.34 ± 1.73	78.13 ± 1.51
sports	87.54 ± 1.54	88.73 ± 1.32	88.79 ± 1.33	85.67 ± 1.92	88.02 ± 1.36	el.lighting	81.42 ± 0.88	80.86 ± 0.96	81.04 ± 0.89	81.26 ± 1.05	80.06 ± 0.89
exercise	84.76 ± 1.60	84.75 ± 1.51	84.89 ± 1.53	84.48 ± 1.49	85.67 ± 1.22	aged/worn	76.42 ± 1.16	76.86 ± 1.15	76.86 ± 1.17	77.47 ± 1.35	77.35 ± 1.16
playing	84.74 ± 0.79	85.38 ± 0.85	85.42 ± 0.77	83.74 ± 1.14	84.92 ± 1.04	glossy	77.87 ± 2.28	79.02 ± 2.17	78.86 ± 2.16	77.04 ± 2.24	77.65 ± 2.38
spectating	87.42 ± 1.30	88.41 ± 1.34	88.39 ± 1.34	88.43 ± 1.18	88.63 ± 1.31	matte	71.86 ± 1.10	72.23 ± 1.38	72.31 ± 1.27	74.63 ± 1.63	74.41 ± 1.65
farming	87.12 ± 1.00	87.17 ± 1.09	87.28 ± 1.07	87.63 ± 1.03	87.11 ± 1.01	moist/damp	79.26 ± 2.18	79.58 ± 2.09	79.59 ± 2.05	79.03 ± 1.72	78.73 ± 2.03
shopping	88.47 ± 1.72	89.62 ± 1.41	89.73 ± 1.42	87.33 ± 1.59	89.42 ± 1.69	dry	85.08 ± 1.08	85.67 ± 1.05	85.56 ± 1.00	86.00 ± 0.72	86.20 ± 0.98
working	84.37 ± 1.68	85.12 ± 1.76	85.15 ± 1.77	84.21 ± 1.45	83.88 ± 1.52	dirty	84.18 ± 1.92	84.85 ± 1.82	84.61 ± 1.88	84.17 ± 1.69	83.97 ± 1.79
us.tools	80.64 ± 0.89	80.10 ± 1.17	80.68 ± 1.11	80.17 ± 1.16	80.85 ± 0.90	rusty	74.79 ± 2.20	75.15 ± 2.00	75.22 ± 2.02	73.81 ± 1.73	75.49 ± 2.07
business	72.77 ± 1.50	72.71 ± 1.48	72.66 ± 1.24	70.21 ± 1.13	71.76 ± 1.87	warm	77.49 ± 1.87	77.59 ± 1.82	77.72 ± 1.85	75.66 ± 1.51	75.68 ± 1.92
praying	83.19 ± 1.03	83.51 ± 1.53	83.54 ± 1.51	83.07 ± 0.89	83.06 ± 1.29	cold	94.18 ± 0.66	94.40 ± 0.61	94.36 ± 0.62	94.63 ± 0.69	94.60 ± 0.59
fencing	74.83 ± 1.20	77.26 ± 1.36	76.95 ± 1.40	74.09 ± 1.64	75.42 ± 1.62	natural	90.10 ± 0.90	90.13 ± 0.99	90.33 ± 0.84	89.38 ± 1.00	89.89 ± 0.91
railing	76.01 ± 1.73	77.60 ± 1.62	77.65 ± 1.59	76.64 ± 1.89	77.70 ± 1.56	man-made	83.06 ± 1.40	83.81 ± 1.16	83.77 ± 1.21	81.89 ± 1.22	82.80 ± 1.66
wire	78.91 ± 1.86	78.85 ± 1.85	79.24 ± 1.70	80.04 ± 1.36	79.21 ± 1.70	open	85.92 ± 0.86	87.57 ± 0.82	87.30 ± 0.79	86.26 ± 0.78	86.64 ± 0.95
trees	82.88 ± 1.16	83.24 ± 1.27	83.38 ± 1.21	81.69 ± 1.26	81.69 ± 1.25	semi-enclosed	84.96 ± 1.76	85.47 ± 1.35	85.39 ± 1.36	85.05 ± 1.94	85.62 ± 1.48
grass	86.51 ± 1.67	87.41 ± 1.49	87.25 ± 1.58	86.53 ± 1.70	86.67 ± 1.54	enclosed	84.56 ± 1.75	85.03 ± 1.63	85.30 ± 1.62	85.87 ± 1.37	85.45 ± 1.72
vegetation	85.29 ± 1.25	85.29 ± 1.20	85.30 ± 1.13	84.27 ± 1.63	85.64 ± 1.39	far-horizon	87.89 ± 1.18	87.91 ± 1.15	87.76 ± 1.17	86.92 ± 0.96	86.76 ± 1.35
shrubbery	87.14 ± 1.34	87.30 ± 1.40	87.30 ± 1.40	85.70 ± 1.54	85.41 ± 1.95	nohorizon	80.76 ± 2.34	81.61 ± 2.31	81.74 ± 2.26	80.87 ± 1.64	80.94 ± 1.80
foliage	83.99 ± 1.32	84.31 ± 1.42	84.73 ± 1.46	82.65 ± 1.31	83.49 ± 1.50	rugged	88.23 ± 0.97	88.32 ± 1.01	88.21 ± 0.98	87.97 ± 1.08	87.65 ± 1.00
leaves	83.49 ± 1.55	83.53 ± 1.58	83.68 ± 1.62	82.71 ± 1.40	82.82 ± 1.46	vertical	81.25 ± 1.25	81.34 ± 1.77	81.24 ± 1.72	81.02 ± 1.41	81.09 ± 1.10
asphalt	88.13 ± 0.83	88.56 ± 0.78	88.71 ± 0.79	87.47 ± 1.07	88.26 ± 0.69	horizontal	77.94 ± 1.30	79.56 ± 1.40	79.55 ± 1.22	77.50 ± 1.15	77.66 ± 1.26
pavement	80.61 ± 1.30	79.05 ± 1.30	79.44 ± 1.16	81.22 ± 1.02	80.55 ± 0.98	symmetrical	79.13 ± 1.47	79.57 ± 1.50	79.42 ± 1.46	75.88 ± 1.66	79.99 ± 1.30
shingles	89.04 ± 1.29	89.03 ± 1.11	89.36 ± 1.13	88.95 ± 1.28	88.84 ± 1.40	cluttered	84.53 ± 1.60	85.21 ± 1.63	85.50 ± 1.61	83.69 ± 1.65	85.08 ± 1.80
carpet	79.17 ± 1.14	79.38 ± 1.22	79.54 ± 1.21	76.99 ± 1.59	77.96 ± 1.12	soothing	79.25 ± 0.88	79.97 ± 0.90	79.75 ± 0.95	79.16 ± 1.66	79.91 ± 0.94
brick	85.76 ± 1.18	85.41 ± 1.40	85.71 ± 1.37	86.17 ± 1.14	85.24 ± 1.63	stressful	79.96 ± 1.77	81.01 ± 1.87	80.77 ± 1.86	81.32 ± 1.33	80.51 ± 1.43
concrete	74.80 ± 1.43	75.11 ± 1.34	75.22 ± 1.35	73.90 ± 1.74	75.19 ± 1.36						

Table 3. Recognizing 83 attributes. The numbers are mean and std error of the **average precision** over 10 runs. GPC^{conf}, GPC+ and SVM+ methods utilize confidence information, whereas GPC and SVM do not. The best result is highlighted in **boldface** and an extra **blue** for GPC^{conf} and GPC+.

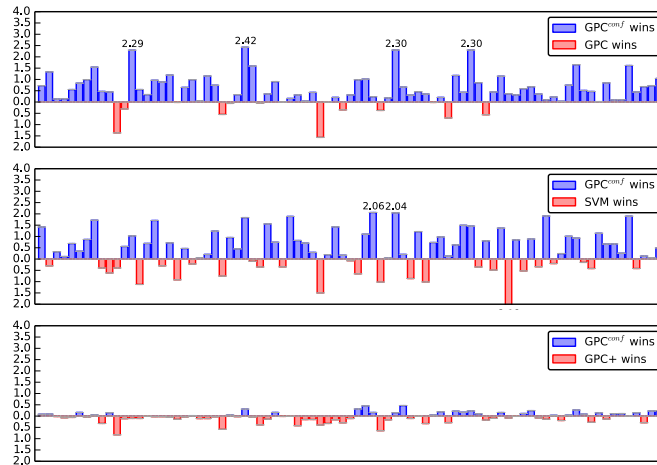


Figure 2. Pairwise comparison of the proposed GPC^{conf} and three other methods in terms of difference between the average precisions. The length of the bar corresponds to **relative improvement of the average precision** for each of the 83 attributes. The attributes and their classifier performance is reported in Table 3. Best viewed in color.

	GPC image	GPC ^{conf} image+conf	GPC+ [11] image+conf	SVM+ [12] image+conf	SVM image
sailing	84.41 ± 1.14	84.78 ± 1.33	84.84 ± 1.19	84.16 ± 1.02	83.18 ± 1.41
driving	84.91 ± 0.83	85.73 ± 0.99	85.62 ± 0.96	84.62 ± 1.41	85.36 ± 1.47
biking	85.77 ± 1.08	86.95 ± 1.14	87.08 ± 1.09	84.25 ± 1.45	85.62 ± 1.15
transport	85.82 ± 0.76	85.96 ± 0.96	85.96 ± 0.89	87.12 ± 1.09	85.72 ± 1.04
vacation	82.75 ± 1.37	83.18 ± 1.57	83.28 ± 1.47	84.05 ± 1.31	83.89 ± 1.29
hiking	83.37 ± 1.45	83.45 ± 1.45	84.01 ± 1.55	81.43 ± 1.81	82.06 ± 1.80
climbing	88.81 ± 0.59	88.80 ± 0.50	88.67 ± 0.52	88.53 ± 0.63	88.08 ± 0.59
camping	82.46 ± 1.53	83.87 ± 1.50	83.78 ± 1.43	82.25 ± 1.39	82.63 ± 1.69
reading	80.83 ± 1.64	81.30 ± 1.71	81.40 ± 1.69	81.05 ± 1.59	81.13 ± 1.67
swimming	82.95 ± 1.75	83.34 ± 1.36	83.34 ± 1.32	81.88 ± 1.52	83.43 ± 1.62
eating	84.44 ± 1.37	86.07 ± 1.28	86.06 ± 1.28	85.13 ± 1.21	85.24 ± 1.23
socializing	85.61 ± 0.86	86.61 ± 0.94	86.62 ± 0.90	85.93 ± 0.91	86.06 ± 0.90
congregating	85.35 ± 0.90	85.65 ± 0.96	85.90 ± 0.92	84.09 ± 1.21	84.56 ± 1.00
competing	87.43 ± 0.86	87.97 ± 0.90	88.03 ± 0.90	87.34 ± 1.03	88.06 ± 0.89
sports	88.20 ± 1.21	89.16 ± 1.27	88.90 ± 1.32	86.54 ± 2.14	89.17 ± 1.07
exercise	84.14 ± 1.36	84.05 ± 1.46	84.02 ± 1.43	85.20 ± 1.32	84.12 ± 1.45
spectating	88.09 ± 0.98	88.36 ± 1.09	88.39 ± 1.11	88.08 ± 1.19	88.56 ± 1.14
shopping	87.59 ± 1.17	87.75 ± 1.10	87.93 ± 1.10	86.81 ± 0.96	87.25 ± 0.95
working	87.64 ± 0.89	87.99 ± 0.79	88.04 ± 0.85	88.32 ± 0.94	87.83 ± 1.01
railing	78.27 ± 1.32	79.66 ± 0.98	79.67 ± 1.04	78.86 ± 1.47	78.45 ± 1.25
trees	86.03 ± 0.87	86.18 ± 0.80	86.06 ± 0.74	84.97 ± 0.46	84.90 ± 0.75
grass	85.56 ± 1.22	85.85 ± 1.17	86.08 ± 1.17	85.97 ± 1.44	85.74 ± 1.33
vegetation	85.45 ± 1.04	85.06 ± 1.26	85.27 ± 1.19	84.81 ± 1.00	84.67 ± 0.94
shrubby	84.34 ± 0.60	84.34 ± 0.65	84.55 ± 0.68	85.00 ± 0.79	84.16 ± 1.00
foliage	86.38 ± 1.23	86.29 ± 1.31	86.55 ± 1.26	83.97 ± 1.27	85.11 ± 1.34
leaves	78.40 ± 1.02	80.13 ± 1.18	79.89 ± 1.11	77.61 ± 1.52	79.25 ± 1.06
asphalt	88.33 ± 1.09	89.13 ± 1.07	89.12 ± 1.10	88.45 ± 0.92	89.27 ± 0.95
shingles	86.66 ± 0.73	86.46 ± 0.62	86.77 ± 0.65	86.71 ± 0.99	87.03 ± 1.01
brick	83.04 ± 1.62	83.76 ± 1.25	83.26 ± 1.43	83.04 ± 1.52	82.62 ± 1.45
metal	80.05 ± 0.62	80.86 ± 1.01	80.76 ± 0.92	77.79 ± 1.64	79.80 ± 0.84
wood	79.19 ± 1.23	79.59 ± 1.40	79.22 ± 1.53	77.31 ± 2.06	77.14 ± 2.07
rubber	85.61 ± 1.33	86.01 ± 1.38	85.89 ± 1.33	85.45 ± 1.53	86.54 ± 1.17
cloth	83.01 ± 1.42	83.86 ± 1.41	83.71 ± 1.44	81.55 ± 1.52	83.35 ± 1.25
rock	81.65 ± 1.84	81.83 ± 1.77	81.64 ± 1.85	80.35 ± 1.37	81.32 ± 2.01
dirt/soil	84.80 ± 1.23	84.53 ± 1.29	84.64 ± 1.30	83.10 ± 1.39	84.23 ± 1.22
glass	72.95 ± 1.15	74.40 ± 1.18	73.81 ± 1.43	72.69 ± 1.54	74.31 ± 0.96
ocean	87.99 ± 1.12	88.40 ± 1.02	88.60 ± 1.06	88.05 ± 1.12	88.85 ± 0.67
stillwater	81.22 ± 1.54	80.29 ± 1.70	80.30 ± 1.65	79.21 ± 1.55	80.38 ± 1.96
clouds	76.05 ± 2.04	76.10 ± 2.00	75.92 ± 2.10	77.33 ± 1.98	75.39 ± 2.05
nat.light	79.11 ± 1.64	79.90 ± 2.25	80.41 ± 2.06	78.43 ± 1.55	79.22 ± 1.80
sunny	73.90 ± 2.42	74.93 ± 2.26	74.85 ± 2.28	72.24 ± 1.66	75.08 ± 1.81
el.lighting	78.53 ± 1.28	78.81 ± 1.79	79.11 ± 1.68	79.40 ± 1.64	81.05 ± 1.48
glossy	76.14 ± 2.40	76.52 ± 2.48	76.44 ± 2.43	76.07 ± 2.39	76.15 ± 2.15
dry	84.51 ± 1.02	85.45 ± 1.10	85.44 ± 1.14	84.88 ± 1.09	84.55 ± 1.21
warm	77.90 ± 0.66	78.72 ± 0.87	78.54 ± 0.86	77.77 ± 1.23	78.15 ± 1.17
cold	93.28 ± 0.80	93.33 ± 0.79	93.29 ± 0.81	92.88 ± 0.97	93.56 ± 0.53
natural	85.79 ± 0.95	85.74 ± 0.86	86.16 ± 0.79	85.44 ± 1.09	85.79 ± 1.18
man-made	83.97 ± 1.28	84.10 ± 1.04	84.01 ± 1.09	83.33 ± 1.31	84.14 ± 1.29
open	81.87 ± 1.52	83.95 ± 1.27	83.54 ± 1.36	82.19 ± 1.26	82.39 ± 1.18
enclosed	85.39 ± 1.13	86.53 ± 1.25	86.78 ± 1.14	86.02 ± 0.98	86.20 ± 1.20
far-horizon	87.89 ± 1.16	88.12 ± 1.14	88.16 ± 1.12	87.68 ± 1.30	88.12 ± 0.89
nohorizon	78.94 ± 2.09	80.05 ± 1.96	79.93 ± 1.92	77.68 ± 2.17	78.94 ± 2.22
rugged	90.06 ± 0.83	90.53 ± 0.91	90.58 ± 0.85	90.14 ± 0.95	90.14 ± 1.06
vertical	80.14 ± 1.53	82.38 ± 1.72	81.84 ± 1.76	81.63 ± 1.80	82.02 ± 1.91
horizontal	77.13 ± 0.58	78.14 ± 0.78	78.13 ± 0.75	75.80 ± 1.23	77.60 ± 1.14
cluttered	84.63 ± 0.93	85.01 ± 1.05	84.96 ± 1.06	84.23 ± 1.47	84.38 ± 1.17
soothing	78.20 ± 1.27	78.24 ± 1.32	78.18 ± 1.31	77.50 ± 1.46	77.59 ± 1.42

Table 4. Recognizing 57 attributes (test scenario A). The numbers are mean and std error of the average precision over 10 runs. GPC^{conf}, GPC+ and SVM+ methods utilize confidence information, whereas GPC and SVM do not. The best result is highlighted in boldface and an extra blue for GPC^{conf} and GPC+.

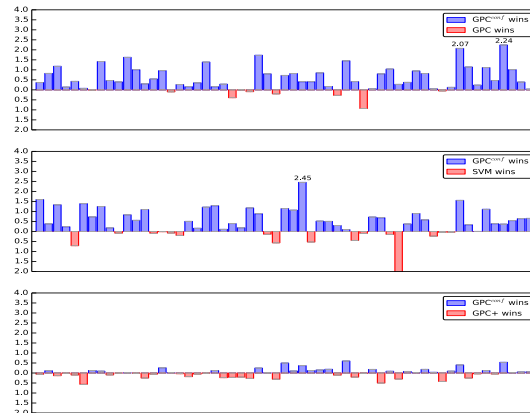


Figure 3. Pairwise comparison of the proposed GPC^{conf} and three other methods in terms of difference between the average precisions. The length of the bar corresponds to relative improvement of the average precision for each of the 57 attributes. Best viewed in color.

References

- [1] T. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Massachusetts Institute of Technology, 2001. [1](#)
- [2] D. Hernández-Lobato. *Prediction Based on Averages over Automatically Induced Learners: Ensemble Methods and Bayesian Techniques*. PhD thesis, Universidad Autónoma de Madrid, 2009. [1](#), [2](#), [3](#)
- [3] A. Naish-Guzman and S. Holden. The generalized FITC approximation. In *NIPS*, 2008. [2](#), [3](#)
- [4] M. Lázaro-Gredilla. *Sparse Gaussian Processes for Large-Scale Machine Learning*. PhD thesis, Universidad Carlos III de Madrid, 2010. [2](#), [3](#), [4](#)
- [5] M. Seeger. Expectation propagation for exponential families. Technical report, University of California, Berkeley, 2006. [2](#), [4](#)
- [6] D. Hernández-Lobato, J. M. Hernández-Lobato, and P. Dupont. Robust multi-class Gaussian process classification. In *NIPS*, 2011. [2](#)
- [7] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. [2](#)
- [8] T. Minka. Power EP. Technical report, Carnegie Mellon University, 2004. [3](#)
- [9] K. B. Petersen and M. S. Pedersen. The matrix cookbook, 2012. [4](#)
- [10] E. Snelson. *Flexible and efficient Gaussian process models for machine learning*. PhD thesis, University College London, 2007. [4](#)
- [11] 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. [5](#), [6](#), [7](#), [8](#)
- [12] V. Vapnik and A. Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 2009. [5](#), [6](#), [7](#), [8](#)
- [13] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research (JMLR)*, 2006. [6](#)