Online Learning with Bayesian Classification Trees
Supplementary Material

Samuel Rota Bulò
FBK-irst
Trento, Italy
rotabulo@fbk.eu

Peter Kontschieder
Microsoft Research
Cambridge, UK
Mapillary
Graz, Austria
pkontschieder@gmail.com

This document provides the following, additional contributions to our CVPR 2016 submission:

- in Section A we provide details about how to derive some formulæ that appear in the main paper;
- in Section B we provide a complexity analysis of our algorithm;
- in Section C we provide additional experimental analyses. Specifically, in Section C.1 we provide Matlab timings of our non-optimized implementation. In Section C.2, we demonstrate and discuss the positive effects of an increasing ensemble size for both, depth 7 and depth 8 BOF-S ensembles. Moreover, we provide a guide on how to perform model selection based on the online development of the ensemble training loss in Section C.3.

A. Detailed derivations

A.1. Derivation of (9) and (10)

We show how to obtain (9) and (10) from (6) when the unimodal surrogate posterior is adopted. The KL-divergence in (6) can be rewritten as follows:

$$\begin{aligned}
D_{\mathtt{KL}}(\hat{q}^{i+1}\|q) &= \mathbb{E}_{\hat{q}^{i+1}}\left[-\log q(t;h)\right] + \text{const} \\
&= \mathbb{E}_{\hat{q}^{i+1}}\left[-\log \delta_{\hat{S}}(S) - \sum_{n\in\mathcal{N}} \log q_n(\boldsymbol{\theta}_n;\boldsymbol{\mu}_n,\Sigma_n) - \sum_{\ell\in\mathcal{L}} \log q_\ell(\boldsymbol{\pi}_\ell;\boldsymbol{\alpha}_\ell)\right] + \text{const} \\
&= \mathbb{E}_{\hat{q}^{i+1}}\left[-\log \delta_{\hat{S}}(S)\right] + \sum_{n\in\mathcal{N}} \mathbb{E}_{\hat{q}^{i+1}}\left[-\log q_n(\boldsymbol{\theta}_n;\boldsymbol{\mu}_n,\Sigma_n)\right] + \sum_{\ell\in\mathcal{L}} \mathbb{E}_{\hat{q}^{i+1}}\left[-\log q_\ell(\boldsymbol{\pi}_\ell;\boldsymbol{\alpha}_\ell)\right] + \text{const}\,,
\end{aligned}$$

where we indicate with "const" a term that does not depend on $h = (\hat{S}, \boldsymbol{\mu}, \Sigma, \boldsymbol{\alpha})$. Hence,

$$\begin{aligned}
&\min_h D_{\mathtt{KL}}[p(\hat{q}^{i+1})\|q(t;h)] \\
&\quad= \min_{\hat{S}} \mathbb{E}_{\hat{q}^{i+1}}\left[-\log \delta_{\hat{S}}(S)\right] + \sum_{n\in\mathcal{N}} \min_{\boldsymbol{\mu}_n,\Sigma_n} \mathbb{E}_{\hat{q}^{i+1}}\left[-\log q_n(\boldsymbol{\theta}_n;\boldsymbol{\mu}_n,\Sigma_n)\right] + \sum_{\ell\in\mathcal{L}} \min_{\boldsymbol{\alpha}_\ell} \mathbb{E}_{\hat{q}^{i+1}}\left[-\log q_\ell(\boldsymbol{\pi}_\ell;\boldsymbol{\alpha}_\ell)\right] + \text{const}\,.
\end{aligned}$$

Therefrom, the independent minimizations in (9) and (10) follow. Moreover,

$$\hat{S}^{i+1} = \arg\min_{\hat{S}} \mathbb{E}_{\hat{q}^{i+1}}\left[-\log \delta_{\hat{S}}(S)\right] = \hat{S}^i\,,$$

for $\hat{q}^{i+1}$ is supported only at $\hat{S}^i$.

A.2. Update of the multi-modal surrogate posterior

The projection step in (6) takes the following form when the multi-modal surrogate posterior in (11) is adopted (we re-write it in terms of the distribution's parameter):

$$(h_{i+1}^1, \dots, h_{i+1}^{\mathsf{m}}) \in \arg\min_{h^1,\dots,h^{\mathsf{m}}} D_{\mathrm{KL}}(\hat{q}(t; h_i^1, \dots, h_i^{\mathsf{m}}) \| q(t; h^1, \dots, h^{\mathsf{m}}))\,. \tag{21}$$

where $\hat{q}(t; h_i^1, \dots, h_i^{\mathsf{m}})$ is the result of update rule (5), when the multi-modal surrogate posterior $q(t; h_i^1, \dots, h_i^{\mathsf{m}})$ is employed. We can then write

$$\hat{q}(t, h_i^1, .., h_i^{\mathsf{m}}) = \frac{1}{\mathsf{m}} \sum_{j=1}^{\mathsf{m}} \frac{p\Big(y_{i+1}; h_i^j, \boldsymbol{x}_{i+1}\Big)}{p(y_{i+1}; h_i^1, \dots, h_i^{\mathsf{m}}, \boldsymbol{x}_{i+1})} p\Big(t | y_{i+1}; h_i^j, \boldsymbol{x}_{i+1}\Big)$$

and apply the log-sum-inequality to obtain the following bound:

$$\begin{aligned} &D_{\mathrm{KL}}(\hat{q}(t|h_i^1, .., h_i^{\mathsf{m}}) \| q(t; h^1, \dots, h^{\mathsf{m}})) \\ &\qquad \leq \frac{1}{\mathsf{m}} \sum_{j=1}^{\mathsf{m}} \frac{p\Big(y_{i+1}; h_i^j, \boldsymbol{x}_{i+1}\Big)}{p(y_{i+1}; h_i^1, \dots, h_i^{\mathsf{m}}, \boldsymbol{x}_{i+1})} \left\{ D_{\mathrm{KL}}(\hat{q}(t; h_i^j) \| q(t; h^j) + \log \frac{p\Big(y_{i+1}; h_i^j, \boldsymbol{x}_{i+1}\Big)}{p(y_{i+1}; h_i^1, \dots, h_i^{\mathsf{m}}, \boldsymbol{x}_{i+1})} \right\}. \end{aligned}$$

Now, the minimization (21) can be replaced with the minimization of the upper bound that we derived above, to get approximate posterior updates. The minimization of the upper bound is equivalent to minimizing the KL-divergence terms with respect to each h^j independently, *i.e.* we perform the posterior update of single Bayesian trees, given the unimodal surrogate posteriors $q(t; h_i^j)$.

A.3. Posterior predictive distribution

From (7) we obtain

$$p(y; h_i, \boldsymbol{x}) = \mathbb{E}_{q^i}\left[p(y|t; \boldsymbol{x})\right] = \sum_{\ell \in \mathcal{L}} \mathbb{E}_{q^i}\left[r(\ell|t; \boldsymbol{x})\right] \mathbb{E}_{q_\ell^i}\left[\pi_{\ell y}\right] = \sum_{\ell \in \mathcal{L}} \mathbb{E}_{q^i}\left[r(\ell|t; \boldsymbol{x})\right] \frac{\alpha_{\ell y}^i}{|\boldsymbol{\alpha}_\ell^i|}\,.$$

As for the expectation of r we have

$$\mathbb{E}_{q^i}\left[r(\ell|t; \boldsymbol{x})\right] = \prod_{n \in \mathcal{N}} \mathbb{E}_{q_n^i}\left[\mathbb{1}_{z_n \geq 0}\right]^{\mathbb{1}_{\ell \swarrow n}} \mathbb{E}_{q_n^i}\left[\mathbb{1}_{z_n < 0}\right]^{\mathbb{1}_{n \searrow \ell}}\,, \tag{22}$$

where $z_n = \boldsymbol{\theta}_n^\top \xi_n(\boldsymbol{x})$. Since $\boldsymbol{\theta}_n$ is Gaussian distributed with mean $\boldsymbol{\mu}_n^i$ and covariance Σ_n^i, also z_n is Gaussian distributed with mean $\mathbb{E}[z_n] = \mathbb{E}[\boldsymbol{\theta}_n]^\top \xi_n(\boldsymbol{x}) = \boldsymbol{\mu}_n^{i\top} \xi_n(\boldsymbol{x})$ and variance $\mathbb{V}ar[z_n] = \xi_n(\boldsymbol{x})^\top \mathbb{V}ar[\boldsymbol{\theta}_n] \xi_n(\boldsymbol{x}) = \xi_n(\boldsymbol{x})^\top \Sigma_n^i \xi_n(\boldsymbol{x})$. This allows us to write

$$\mathbb{E}_{q_n^i}\left[\mathbb{1}_{z_n \geq 0}\right] = \mathbb{E}_{z_n}\left[\mathbb{1}_{z_n \geq 0}\right] = \Phi\left(\frac{\mathbb{E}[z_n]}{\sqrt{\mathbb{V}ar[z_n]}}\right) = \Phi(\boldsymbol{\mu}_n^{i\top} \tilde{\xi}_n^i(\boldsymbol{x})) = \beta_n^i(\boldsymbol{x})\,, \tag{23}$$

where $\tilde{\xi}_n^i(\boldsymbol{x})$ is defined as in (15). Moreover, we have that

$$\mathbb{E}_{q_n^i}\left[\mathbb{1}_{z_n < 0}\right] = 1 - \mathbb{E}_{q_n^i}\left[\mathbb{1}_{z_n \geq 0}\right] = 1 - \beta_n^i(\boldsymbol{x})\,. \tag{24}$$

We obtain (13), by substituting (23) and (24) into (22), and by writing $\rho(\ell; h_i, \boldsymbol{x})$ for the latter quantity.

A.4. Update rule for $\boldsymbol{\alpha}$

We show how to obtain (20) from (19). It is well-known that, if $\boldsymbol{\pi}_\ell$ is Dirichlet-distributed with parameter $\boldsymbol{\alpha}_\ell^{i+1}$, then the expectation of $\log(\pi_{\ell z})$ yields $\psi(\alpha_{\ell z}^{i+1}) - \psi(|\boldsymbol{\alpha}_\ell^{i+1}|_1)$, where ψ is the digamma function. Hence, the left-hand-side of (20) follows therefrom.

We focus now on the right-hand-side, *i.e.* the expectation $\mathbb{E}_{\hat{q}^{i+1}}\left[\log(\pi_{\ell z})\right]$, which involves the following marginal of $\hat{q}^{i+1}$:

$$\hat{q}^{i+1}(\boldsymbol{\pi}_\ell) = \frac{p(y_{i+1}|\boldsymbol{\pi}_\ell; h_i, \boldsymbol{x}_{i+1}) q_\ell(\boldsymbol{\pi}_\ell; \boldsymbol{\alpha}_\ell^i)}{p(y_{i+1}; h_i, \boldsymbol{x}_{i+1})}\,. \tag{25}$$

The likelihood term $p(y_{i+1}|\boldsymbol{\pi}_\ell; h_i, \boldsymbol{x}_{i+1})$ has the same form of $p(y_{i+1}; h_i, \boldsymbol{x}_{i+1})$ as per (13) with $\pi_{\ell y_{i+1}}$ replacing $\alpha^i_{\ell y_{i+1}}/|\boldsymbol{\alpha}^i_\ell|_1$. Accordingly, we can write

$$p(y_{i+1}|\boldsymbol{\pi}_\ell; h_i, \boldsymbol{x}_{i+1}) = p(y_{i+1}; h_i, \boldsymbol{x}_{i+1}) + \left(\pi_{\ell y_{i+1}} - \frac{\alpha^i_{\ell y_{i+1}}}{|\boldsymbol{\alpha}^i_\ell|_1}\right) \rho(\ell; h_i, \boldsymbol{x}_{i+1})\,.$$

By substituting back into (25) we obtain

$$\hat{q}^{i+1}(\boldsymbol{\pi}_\ell) = q_\ell(\boldsymbol{\pi}_\ell; \boldsymbol{\alpha}^i_\ell)\left[1 + \left(\pi_{\ell y_{i+1}} - \frac{\alpha^i_{\ell y_{i+1}}}{|\boldsymbol{\alpha}^i_\ell|_1}\right) a_\ell\right]\,,$$

where $a_\ell = \frac{\rho(\ell; h_i, \boldsymbol{x}_{i+1})}{p(y_{i+1}; h_i, \boldsymbol{x}_{i+1})}$.

We can now solve the expectation as follows:

$$\begin{aligned}\mathbb{E}_{\hat{q}^{i+1}}\left[\log(\pi_{\ell z})\right] &= \int \log(\pi_{\ell z})\hat{q}^{i+1}(\boldsymbol{\pi}_\ell)d\boldsymbol{\pi}_\ell \\ &= \left(1 - a_\ell \frac{\alpha^i_{\ell y_{i+1}}}{|\boldsymbol{\alpha}^i_\ell|_1}\right) \underbrace{\int \log(\pi_{\ell z}) q_\ell(\boldsymbol{\pi}_\ell; \boldsymbol{\alpha}^i_\ell)d\boldsymbol{\pi}_\ell}_{\psi(\alpha^i_{\ell z}) - \psi(|\boldsymbol{\alpha}^i_\ell|_1)} + a_\ell \int \log(\pi_{\ell z})\pi_{\ell y_{i+1}} q_\ell(\boldsymbol{\pi}_\ell; \boldsymbol{\alpha}^i_\ell)d\boldsymbol{\pi}_\ell\,,\end{aligned}$$

where the integral in the first term of the last equality is again the expectation of $\log(\pi_{\ell z})$ under the Dirichlet distribution. As for the last integral we can get rid of $\pi_{\ell y_{i+1}}$ by noting that

$$\pi_{\ell y_{i+1}} q_\ell(\boldsymbol{\pi}_\ell; \boldsymbol{\alpha}^i_\ell) = \frac{\alpha^i_{\ell y_{i+1}}}{|\boldsymbol{\alpha}_\ell|_1} q_\ell(\boldsymbol{\pi}_\ell; \boldsymbol{\beta}_\ell)$$

where $\beta_{\ell z} = \alpha^i_{\ell z}$ for all $z \neq y_{i+1}$ and $\beta_{\ell y_{i+1}} = \alpha^i_{\ell y_{i+1}} + 1$. By doing so, we find again a known integral:

$$a_\ell \int \log(\pi_{\ell z})\pi_{\ell y_{i+1}} q_\ell(\boldsymbol{\pi}_\ell; \boldsymbol{\alpha}^i_\ell)d\boldsymbol{\pi}_\ell = a_\ell \frac{\alpha^i_{\ell y_{i+1}}}{|\boldsymbol{\alpha}^i_\ell|_1} \int \log(\pi_{\ell z}) q_\ell(\boldsymbol{\pi}_\ell; \boldsymbol{\beta}_\ell)d\boldsymbol{\pi}_\ell = a_\ell \frac{\alpha^i_{\ell y_{i+1}}}{|\boldsymbol{\alpha}^i_\ell|_1}\left[\psi(\beta_{\ell z}) - \psi(|\boldsymbol{\beta}_\ell|_1)\right]\,.$$

By exploiting the digamma's recurrence relation $\psi(x+1) = \psi(x) + x^{-1}$, we can rewrite the last factor as

$$\psi(\beta_{\ell z}) - \psi(|\boldsymbol{\beta}_\ell|_1) = \psi(\alpha^i_{\ell z}) - \psi(|\boldsymbol{\alpha}^i_\ell|_1) + \frac{\mathbb{1}_{z=y_{i+1}}}{\alpha^i_{\ell z}} - \frac{1}{|\boldsymbol{\alpha}^i_\ell|_1}\,.$$

Finally, by exploiting the new relations we obtain

$$\begin{aligned}&\mathbb{E}_{\hat{q}^{i+1}}\left[\log(\pi_{\ell z})\right] \\ &= \left(1 - a_\ell \frac{\alpha^i_{\ell y_{i+1}}}{|\boldsymbol{\alpha}^i_\ell|_1}\right)\left[\psi(\alpha^i_{\ell z}) - \psi(|\boldsymbol{\alpha}^i_\ell|_1)\right] + a_\ell \frac{\alpha^i_{\ell y_{i+1}}}{|\boldsymbol{\alpha}^i_\ell|_1}\left[\psi(\alpha^i_{\ell z}) - \psi(|\boldsymbol{\alpha}^i_\ell|_1) + \frac{\mathbb{1}_{z=y_{i+1}}}{\alpha^i_{\ell z}} - \frac{1}{|\boldsymbol{\alpha}^i_\ell|_1}\right] \\ &= \psi(\alpha^i_{\ell z}) - \psi(|\boldsymbol{\alpha}^i_\ell|_1) + \frac{a_\ell(\mathbb{1}_{z=y_{i+1}} - u_\ell)}{|\boldsymbol{\alpha}^i_\ell|_1}\end{aligned}$$

where $u_\ell = \alpha^i_{\ell y_{i+1}}/|\boldsymbol{\alpha}^i_\ell|_1$.

A.5. Update rule for μ and Σ

The solution to (9) can be found by moment-matching. By [1], this yields the following explicit update formulas for the means and the covariances of each node $n \in \mathcal{N}$:

$$\boldsymbol{\mu}^{i+1}_n = \boldsymbol{\mu}^i_n + \Sigma^i_n \frac{\partial}{\partial \boldsymbol{\mu}^i_n} \log p(y_{i+1}; h_i, \boldsymbol{x}_{i+1})\,, \tag{26}$$

$$\Sigma^{i+1}_n = \Sigma^i_n + \Sigma^i_n \frac{\partial^2}{\partial \boldsymbol{\mu}^i_n \partial \boldsymbol{\mu}^{i\top}_n} \log p(y_{i+1}; h_i, \boldsymbol{x}_{i+1})\Sigma^{i\top}_n\,. \tag{27}$$

We will next simplify the derivatives to obtain (16) and (17). We start with the derivative in the update for the mean $\boldsymbol{\mu}_n$:

$$\frac{\partial}{\partial \boldsymbol{\mu}_n^i} \log p(y_{i+1}; h_i, \boldsymbol{x}_{i+1}) = \frac{\frac{\partial}{\partial \boldsymbol{\mu}_n^i} p(y_{i+1}; h_i, \boldsymbol{x}_{i+1})}{p(y_{i+1}; h_i, \boldsymbol{x}_{i+1})} \,.$$

Then

$$\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{\mu}_n^i} p(y_{i+1}; h_i, \boldsymbol{x}_{i+1}) = \rho(n; h_i, \boldsymbol{x}_{i+1}) \frac{\partial}{\partial \boldsymbol{\mu}_n^i} p(y_{i+1} | n; h_i, \boldsymbol{x}_{i+1}) \\
&= \rho(n; h_i, \boldsymbol{x}_{i+1}) \left[p(y_{i+1} | n_{\mathsf{L}}; h_i, \boldsymbol{x}_{i+1}) \frac{\partial}{\partial \boldsymbol{\mu}_n^i} \beta_n^i(\boldsymbol{x}_{i+1}) + p(y_{i+1} | n_{\mathsf{R}}; h_i, \boldsymbol{x}_{i+1}) \frac{\partial}{\partial \boldsymbol{\mu}_n^i} (1 - \beta_n^i(\boldsymbol{x}_{i+1})) \right] \\
&= \rho(n; h_i, \boldsymbol{x}_{i+1}) \phi(\boldsymbol{\mu}_n^{i\top} \tilde{\xi}_n^i) [p(y_{i+1} | n_{\mathsf{L}}; h_i, \boldsymbol{x}_{i+1}) - p(y_{s+1} | n_{\mathsf{R}}; h_i, \boldsymbol{x}_{i+1})] \tilde{\xi}_n^i \\
&= \rho(n; h_i, \boldsymbol{x}_{i+1}) \phi(\boldsymbol{\mu}_n^{i\top} \tilde{\xi}_n^i) (u_{n_{\mathsf{L}}} - u_{n_{\mathsf{R}}}) \tilde{\xi}_n^i \,,
\end{aligned}$$

where $\tilde{\xi}_n^i$ stands for $\tilde{\xi}_n^i(\boldsymbol{x}_{i+1})$, $u_n = p(y_{i+1} | n; h_i, \boldsymbol{x}_{i+1})$, and we explot the recursive formula $u_n = \beta_n^i(\boldsymbol{x}_{i+1})) u_{n_{\mathsf{L}}} + (1 - \beta_n^i(\boldsymbol{x}_{i+1})))u_{n_{\mathsf{R}}}$. By substituting back in the original derivative we obtain:

$$\frac{\partial}{\partial \boldsymbol{\mu}_n^i} \log p(y_{i+1}; h_i, \boldsymbol{x}_{i+1}) = \frac{\rho(n; h_i, \boldsymbol{x}_{i+1})}{p(y_{i+1}; h_i, \boldsymbol{x}_{i+1})} \phi(\boldsymbol{\mu}_n^{i\top} \tilde{\xi}_n^i) (u_{n_{\mathsf{L}}} - u_{n_{\mathsf{R}}}) \tilde{\xi}_n^i = \kappa_n \tilde{\xi}_n^i \,,$$

and by substitution in (26) we obtain (16).

As for the update rule for Σ_n, we have

$$\frac{\partial^2}{\partial \boldsymbol{\mu}_n^i \partial \boldsymbol{\mu}_n^{i\top}} \log p(y_{i+1}; h_i, \boldsymbol{x}_{i+1}) = \frac{\frac{\partial^2}{\partial \boldsymbol{\mu}_n^i \partial \boldsymbol{\mu}_n^{i\top}} p(y_{i+1}; h_i, \boldsymbol{x}_{i+1})}{p(y_{i+1}; h_i, \boldsymbol{x}_{i+1})} - \kappa_n^2 \tilde{\xi}_n^i \tilde{\xi}_n^{i\top} \,.$$

Then

$$\begin{aligned}
\frac{\partial^2}{\partial \boldsymbol{\mu}_n^i \partial \boldsymbol{\mu}_n^{i\top}} p(y_{i+1}; h_i, \boldsymbol{x}_{i+1}) &= \rho(n; h_i, \boldsymbol{x}_{i+1}) (u_{n_{\mathsf{L}}} - u_{n_{\mathsf{R}}}) \tilde{\xi}_n^i \frac{\partial}{\partial \boldsymbol{\mu}_n^{i\top}} \phi(\boldsymbol{\mu}_n^{i\top} \tilde{\xi}_n^i) \\
&= -\rho(n; h_i, \boldsymbol{x}_{i+1}) \phi(\boldsymbol{\mu}_n^{i\top} \tilde{\xi}_n^i) (u_{n_{\mathsf{L}}} - u_{n_{\mathsf{R}}}) (\boldsymbol{\mu}_n^{i\top} \tilde{\xi}_n^i) \tilde{\xi}_n^i \tilde{\xi}_n^{i\top} \,,
\end{aligned}$$

where we exploit the derivative $\phi'(x) = -x\phi(x)$. By substituting back in the previous derivative we get

$$\frac{\partial^2}{\partial \boldsymbol{\mu}_n^i \partial \boldsymbol{\mu}_n^{i\top}} \log p(y_{i+1}; h_i, \boldsymbol{x}_{i+1}) = -\kappa_n (\boldsymbol{\mu}_n^{i\top} \tilde{\xi}_n^i) \tilde{\xi}_n^i \tilde{\xi}_n^{i\top} - \kappa_n^2 \tilde{\xi}_n^i \tilde{\xi}_n^{i\top} = -(\kappa_n^2 + \kappa_n \boldsymbol{\mu}_n^{i\top} \tilde{\xi}_n^i) \tilde{\xi}_n^i \tilde{\xi}_n^{i\top} \,.$$

Finally, by substitution in (27) we obtain (17).

B. Complexity notes.

If we consider the online learning procedure with the multi-modal posterior, we are de facto training m tree models. Assume each tree to have n decision nodes. The per-sample training complexity is given by $O(\mathsf{mn}(\mathsf{d}^2 + |\mathcal{Y}|\nu))$. Indeed, the computation of the posterior updates require traversing the trees twice as shown Sec. 4 of the main paper. The most expensive operation per decision node is the covariance matrix update having complexity $O(\mathsf{d}^2)$, where d is the maximum node feature dimensionality. For the leaves, the update complexity is proportional to the number of classes $|\mathcal{Y}|$ and the average number ν of Newton-Raphson iterations. The computational complexity during inference per single tree is discussed in Sec. 4 of the main paper, *i.e.* for m trees, we have $O(\mathsf{md}^2\mathsf{n})$ in case the stochastic routing is applied. However, when applying the fast, single path inference trick as discussed in the main paper, complexity reduces to $O(\mathsf{md}\log_2(\mathsf{n}))$, which is identical to the complexity of offline, oblique random forests [2]. Since for our proposed method the number of trees m per ensemble as well as their average number of decision nodes n is much smaller than for oblique forests, we will typically experience a computational advantage over them.

	dna		satimages	USPS
#features	180	60	36	256
ORF	$2.0{\cdot}10^3$	-	$2.8{\cdot}10^3$	$4.0{\cdot}10^3$
MF	330	290	510	$1.6{\cdot}10^3$
EX-k	$5.0{\cdot}10^3$	-	$5.0{\cdot}10^3$	$2.9{\cdot}10^4$
BOF	32.3	4.3	63.7	$1.88{\cdot}10^3$
BOF-P	19.7	3.8	40.4	436.2
BOF-B	22.9	3.1	45.2	$1.29{\cdot}10^3$
BOF-BP	11.6	2.6	35.7	305.4

Table 2. Timing details for related works in top rows (Python implementation, numbers taken from [4]) and proposed Bayesian online forests in bottom rows (Matlab implementation) in $[s]$.

C. Experimental Results

C.1. Timings on Machine Learning and Kinect Datasets

Timing analysis In this section, we provide timing details in Tab. 2 for our parallel, though non-optimized Matlab implementation in the machine learning datasets evaluated in Sec. 5.1 of the main paper. We ran all experiments on a desktop machine with 16 cores. Please note that the timings for our method cannot directly be compared to those produced by the Python implementations from [3, 4], however, they show how fast our ensembles of Bayesian trees can be grown. Training times are provided for forest training, *i.e.* we have an advantage as we only have to train 8 trees of limited depth, while our competitors need to train 100 deep trees to obtain the reported accuracies. For future work, we plan a GPU-based implementation that is capable to pursue real-time updates of the model. Timings on test data were in the order of milliseconds. For the sake of implementation simplicity we only used balanced trees at the moment. Despite the quadratic term due to the covariance matrix update, our trees can be trained reasonably fast compared to the other approaches, because there is no need for exceptionally deep trees.

As for the experiment on the Kinect dataset in Sec. 5.2, the training time for 8 BOF-S trees on all training data is ≈20h for depth 8 (≈12h for depth 7), which can be reduced to ≈6h with the diagonal version for depth 8 BOF-SD and ≈3h for depth 7 BOF-SD (all Matlab timings).

C.2. Ensemble effect on Kinect dataset

In Fig. 5 we show the ensemble effect on test accuracy for Kinect data. The scores increase as we add more of our BOF-S trees to the ensemble, providing a stronger impact on the overall test scores as we found for the machine learning datasets. We explain this by the fact that the feature space in the Kinect experiment is much larger and more complex and also due to the inherent differences of BOF and BOF-S, *ie.* the enabled feature space subsampling in split nodes. Thus, for more complex classification scenarios we benefit from a larger ensemble size, akin to what is well-known and appreciated in conventional applications for random forests.

C.3. Model selection guide based on online development of ensemble training loss

Finally, we present a guide on how to perform tree topology selection, based on the analysis of the cumulative moving average of the training loss they exhibit (defined as $l_{s+1} = -\log\left((s+1)^{-1}\sum_{i=0}^{s} P(y_{i+1}|\boldsymbol{x}_{i+1}, h_i)\right)$. In Fig. 6, we plot the cumulative moving average loss for different tree depths 4-8 (averaged over 5 training repetitions) for the first 2000 training samples. Please note that depth 4 is in principle insufficient since the number of generated leaf nodes is smaller than the number of object classes. We observe that the loss development strongly correlates with tree depth (when considering sufficiently deep tree depths 5-8), *i.e.* depth 5 has lower loss than depth 6, *etc.* In such a way, one can select the model with lowest training loss if inference needs to be performed at this early stage of training (which we actually applied to obtain the scores for depth 8 BOF-S up to 1k samples in Fig. 4, center). As we approach closely to 2k samples, the differences in the loss become less pronounced, and different tree depths lead to similar loss developments. The depth 4 case (trees that are too shallow) also shows that the loss cannot be reasonably reduced after around 1300 samples, which is an expected trend and a clear indicator that the model choice was suboptimal.

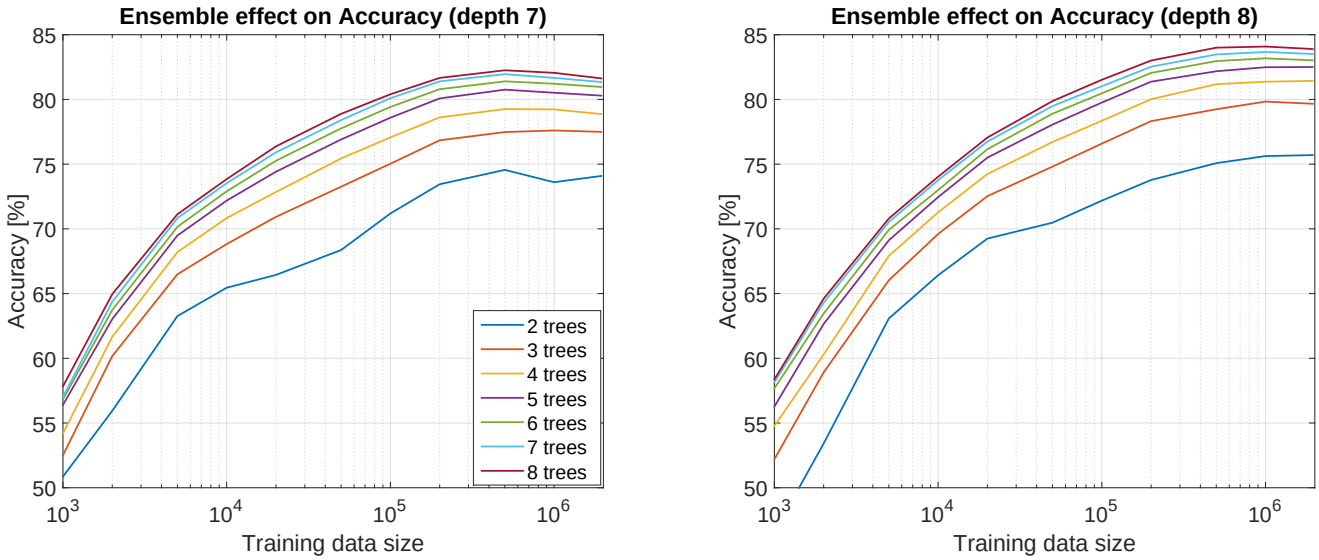

Figure 5. Effect of varying BOF-S ensemble sizes for depth 7 (left) and depth 8 (right) trees on Kinect test data accuracy as function of training data.

Figure 6. Demonstration of loss development on Kinect dataset using different depth selections. Losses are cumulative moving averages of the training loss (after averaging the losses over five repetitions per forest).

References

[1] Manfred Opper. A bayesian approach to on-line learning. In David Saad, editor, *On-line Learning in Neural Networks*, pages 363–378. Cambridge University Press, 1998. 3

[2] Bjoern H. Menze, B. Michael Kelm, Daniel N. Splitthoff, Ullrich Koethe, and Fred A. Hamprecht. On oblique random forests. In *Machine Learning and Knowledge Discovery in Databases*, volume 6912. Springer, 2011. 4

[3] M. Denil, D. Matheson, and N. de Freitas. Consistency of online random forests. In *(ICML)*, 2013. 5

[4] B. Lakshminarayanan, D. Roy, and Y. W. Teh. Mondrian forests: Efficient online random forests. In *Advances in Neural Inform. Process. Syst.*, 2014. 5