Agnostic Bayesian Learning of Ensembles

Alexandre Lacoste[*1] alexandre.lacoste.1@ulaval.ca
Hugo Larochelle[2] hugo.larochelle@usherbrooke.ca
Mario Marchand[1] Mario.Marchand@ift.ulaval.ca
François Laviolette[1] Francois.Laviolette@ift.ulaval.ca
[1]Département d'informatique et de génie logiciel, Université Laval, Québec, Canada, G1K-7P4
[2]Département d'informatique, Université de Sherbrooke, Québec, Canada, J1K-2R1

1. Description of the Learning Algorithms

To help describing the choice of hyperparameters, we use the function $\text{EXP}(x, y, c)$ to represent a set of c values ranging from 10^x to 10^y on a logarithmic scale.

SVM: We used the implementation of scikit-learn (Pedregosa et al., 2011) based on libsvm. The soft margin parameter is selected from $\text{EXP}(-1, 3, 20)$ and the RBF kernel width is selected from $\text{EXP}(-1, 3, 20)$.

ANN: We used a custom implementation with sigmoid activation function and quadratic weight regularization. The number of neurons on the second layer is fixed to the square root of the number on the first layer which range from 3 to 100. The regularization parameter is selected from $\text{EXP}(-2, 2, 20)$.

Random Forest: We used the implementation of scikit-learn with both the *gini* and *entropy* criterion for the split. We also selected various values for n_f, the maximum number of features explored at each split. The values are $\lceil \eta n_f \rceil$ where $\eta \in \text{EXP}(-2, 0, 10)$

Extra Randomized Trees: We used the implementation of scikit-learn with the same set of parameters as used for random forest.

Gradient Tree Boosting: We used the implementation of scikit-learn where the learning rate is selected from $\text{EXP}(-2, 0, 10)$ and the maximum depth parameter is a value in $\{1, 2, 3, 4, 5, 6\}$.

2. Marginal Likelihood of the t-Distribution

To be able to adjust the parameters of the prior, we report the marginal likelihood which, in this case, is simply the ratio of the marginalization constant of the prior and posterior

$$\begin{aligned}
&\mathcal{M}\left(\kappa_0, \nu_0, \mathbf{r}_0, T_0\right) \\
&\stackrel{\text{def}}{=} \Pr(L) = \mathop{\mathbf{E}}_{\mathbf{r},\Lambda\sim\Pr(\cdot,\cdot)} \Pr\left(L|\mathbf{r}, \Lambda\right) \\
&= \frac{1}{\pi^{md/2}} \frac{\Gamma_d\left(\nu_m/2\right)}{\Gamma_d\left(\nu_0/2\right)} \frac{|T_0|^{\nu_0/2}}{|T_m|^{\nu_m/2}} \left(\frac{\kappa_0}{\kappa_m}\right)^{d/2}, \qquad (1)
\end{aligned}$$

where $\Gamma_d(\cdot)$ is the multivariate gamma function $\Gamma_d\left(\nu/2\right) \stackrel{\text{def}}{=} \pi^{d(d-1)/4} \prod_{i=1}^{d} \Gamma\left(\frac{\nu+1-i}{2}\right)$.

To simplify the derivatives, we work with the logarithm of the marginal likelihood. Let $g\left(\kappa_0, \nu_0, \mu_0, t_0\right) \stackrel{\text{def}}{=} 2\log\left(\mathcal{M}\left(\kappa_0, \nu_0, \mu_0\mathbf{1}, t_0 I\right)\right)$. Then,

$$\begin{aligned}
\frac{\partial g}{\partial \kappa_0} &= \frac{dm}{\kappa_0\kappa_m} - \frac{\nu_m m^2}{\kappa_m^2} \operatorname{tr}\left(T_m^{-1}\left(\mathbf{r}_0 - \bar{\boldsymbol{l}}\right)\left(\mathbf{r}_0 - \bar{\boldsymbol{l}}\right)^T\right) \\
\frac{\partial g}{\partial \nu_0} &= \log\left(\frac{|T_0|}{|T_m|}\right) + \sum_{i=1}^{d} \psi\left(\tfrac{\nu_m+1-i}{2}\right) - \sum_{i=1}^{d} \psi\left(\tfrac{\nu_0+1-i}{2}\right) \\
\frac{\partial g}{\partial \mu_0} &= -\nu_m \operatorname{tr}\left(T_m^{-1} \tfrac{\kappa_0 m}{\kappa_m} B\right) \\
\frac{\partial g}{\partial t_0} &= \frac{\nu_0 d}{t_0} - \nu_m \operatorname{tr}\left(T_m^{-1}\right),
\end{aligned}$$

where $\psi(\cdot)$ is the digamma function and $B_{ij} \stackrel{\text{def}}{=} 2\mu_0 - \bar{\boldsymbol{l}}_i - \bar{\boldsymbol{l}}_j$.

References

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.