Locking and Quacking: Stacking Bayesian models predictions by log-pooling and superposition

Anonymous Author(s)
Affiliation
Address
email

Abstract

Combining predictive distributions is a central problem in Bayesian inference and machine learning. Currently, predictives are almost exclusively combined using linear density-mixtures such as Bayesian model averaging, Bayesian stacking, and mixture of experts. Nonetheless, linear mixtures impose traits that might be undesirable for some applications, such as multi-modality. While there are alternative strategies (e.g., geometric bridge or superposition), optimizing their parameters usually implies computing intractable normalizing constant repeatedly. In this extended abstract, we present two novel Bayesian model combination tools. They are generalizations of *stacking*, but combine posterior densities by log-linear pooling (*locking*) and quantum superposition (*quacking*). To optimize model weights while avoiding the burden of normalizing constants, we maximize the Hyvärinen score of the combined posterior predictions. We demonstrate locking and quacking with an illustrative example.

1 Introduction

A general challenge in statistics is prediction in the presence of multiple candidate models or learning algorithms: we are interested in some outcome y on a measurable space $\mathcal{Y} \subset \mathbb{R}^d$; We fit different models to the data, or the same model on different parts of the dataset, and obtain a set of predictive distributions, $\{\pi_1(y), \dots, \pi_K(y)\}$, where each $\pi_k(y)$ is a (conditional[1]) probabilistic density such that $\int_{\mathcal{Y}} \pi_k(y) dy = 1$. When combining models, there are three subjective decisions to make: (1) individual models, (2) the "prior" assigned to each model, and (3) the form in which individual sampling models are combined in the predictive sampling distribution. Here, we focus on the latter.

The combination operation binds individual sampling distributions into a larger encompassing sampling model. A combination operator, parametrized by some parameter w, maps a sequence of probability densities into a single probability density:

$$h(\pi_1(\cdot), \dots, \pi_K(\cdot)|w) = \pi_*(\cdot), \text{ s.t. } \pi_*(\cdot) \geq 0, \mathbb{E}\,\pi_*(\cdot) = 1.$$

For example, a (linear) mixture can be represented by

$$h(\pi_1(\cdot), \dots, \pi_K(\cdot)|w) = \sum_k w_k \pi_k(\cdot), \quad \sum_k w_k = 1.$$

In Bayesian statistics, the *mixture* is the *de facto* combination operator to combine predictive distributions, and used in Bayesian model averaging [11], stacking [15], hierarchical stacking [16], and as its name has implied, mixture-of-experts [6, 7, 17], and hypothesis testing [8]. Despite its mathematical convenience, *mixture* has a few limitations:

[1] the dependence on covariate x is suppressed for brevity

Under review at the NeurIPS 2022 Workshop on Score-Based Methods. Do not distribute.

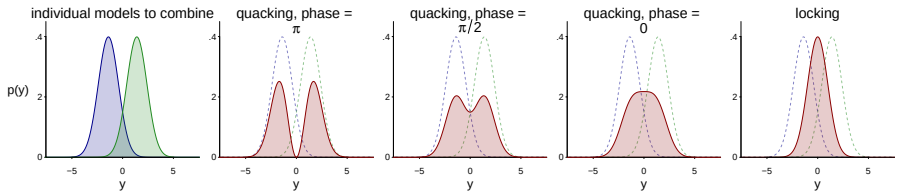

Figure 1: When combining two probabilistic predictions (panel 1), quacking combines them via superposition and locking combines them by geometric bridges.

Mixture is linear. If only relying on mixing to combine individual sampling models, the depth of the combination network is restricted to one. It only examines likelihoods through their evaluations at realized observations. Mixture of predictive densities typically results in a multimodal posterior prediction, which comes with unnatural interpretation and poor interval coverage.

In this extended abstract, we investigate two operators to combine Bayesian predictives (Figure 1):

1. Geometric bridge (log-linear pooling),

$$h(\pi_1(\cdot), \ldots, \pi_K(\cdot)|w) \coloneqq \frac{\prod_k \pi_k^{w_k}(\cdot)}{\int \prod_k \pi_k^{w_k}(y)dy}, \; w \in \mathcal{S}^k \,. \tag{1}$$

2. Superposition,

$$h(\pi_1(\cdot), \ldots, \pi_K(\cdot)|w, \alpha) \coloneqq \frac{\left| \sum_k \sqrt{w_k}\sqrt{\pi_k(\cdot)}e^{i\alpha_k} \right|^2}{\int_{\mathbb{R}} \left| \sum_k \sqrt{w_k}\sqrt{\pi_k(y')}e^{i\alpha_k} \right|^2 dy'}, \; \alpha \in [0, 2\pi)^K, \; w \in \mathcal{S}^k \,. \tag{2}$$

   where $\mathcal{S}^K$ is the k-dimensional simplex.

Compared with to mixture, these new operators have appealing features: When individual sampling models are log-concave, so is their geometric bridge, hence preserving the unimodality. Moreover, in superposition, when the phases α are uniformly distributed, we get back a mixture of densities. Even when there is only one single model, depending on the phase, the superposition and geometric bridge can make the combined distributions spikier, or flatter— approximately a power transformation, thereby automatically calibrating the prediction confidence. Finally, unlike the mixture, the superposition and geometric bridge can create a middle mode, leading to more flexible predictions.

The remaining question is then how to optimize the weights w such that the combined predictions can better fit the data. This is challenging because of the intractable normalizing constant, and existing log-linear pooling techniques rely on some non-testable normal approximation [1, 4, 10, 12]. In the next section, we provide a practical solution that incorporates the Hyvärinen score [5] and Bayesian posterior predictions.

2 Operator-oriented model averaging

2.1 Scoring rules and Hyvärinen score

In methods like stacking and mixture of experts, we need a scoring rule Gneiting and Raftery [2] to evaluate the combined prediction, in which the logarithmic scoring rule is the de facto choice for it is the only continuous proper local scoring rule. However, the log score does not apply to log-linear pooling and superposition: unless in trivial cases, the combined predictive densities contain an unknown normalization constant in the denominator.

To bypass the normalizing constant, we use the Hyvärinen score [5] to evaluate the unnormalized combined predictive density. In general, given an unnormalized density q, how good it fits the observed data y is quantified by

$$\mathcal{H}(y, p) = 2\Delta_y \log p(y) + ||\nabla_y \log p(y)||^2.$$

The Hyvärinen score can be interpreted as the L_2 norm of the difference between the score of the prediction and the true data generating process.

2.2 Importance weighted estimate of the score function

Another distinctive feature of the full Bayesian prediction is that the posterior prediction itself is a mixture of conditional sampling distributions. That is, in the k-th model, the posterior parameter inference given observed data $\mathbf{D}$ is $p_k(\theta|\,\mathbf{D})$, and the predictive density for future data $\tilde{y}$ is $\pi_k(\tilde{y}) = \int_{\mathbf{\Theta}} f(\tilde{y}|\theta) p_k(\theta|\,\mathbf{D})\, d\theta$. To compute the Hyvärinen score of this posterior prediction, we need the pointwise score function

$$\frac{\partial}{\partial y} \log \pi_k(y) = \frac{\pi_k'(y)}{\pi_k(y)} = \frac{\int_{\mathbf{\Theta}} \frac{\partial}{\partial y} f(y|\theta) p_k(\theta|\,\mathbf{D})\, d\theta}{\int_{\mathbf{\Theta}} f(y|\theta) p_k(\theta|\,\mathbf{D})\, d\theta}. \tag{3}$$

We will typically use Markov chain Monte Carlo (MCMC) methods for individual model inference, such that we have S simulation draws $\{\theta_{sk}, 1 \le s \le S\}$ from the model k posterior $p_k(\theta|\,\mathbf{D})$. The score function (3) is a ratio of integrals. We compute both the denominator and numerator by Monte Carlo sum, and a plug-in estimate of the score function is

$$\frac{\partial}{\partial y} \log \pi_k(y) \approx g_k(y) \coloneqq \frac{\sum_{s=1}^{S} \frac{\partial}{\partial y} f(y|\theta_{ks})}{\sum_{s=1}^{S} f(y|\theta_{ks})}. \tag{4}$$

There is no worry that the denominator and numerator are estimated using the same draws: We can view (4) as self-normalized importance sampling with a proposal density $p_k(\theta|\,\mathbf{D})$, target density $f(y|\theta)p_k(\theta|\,\mathbf{D})$, and a function $h(\theta) = \frac{\partial}{\partial y} f(y|\theta)/f(y|\theta)$ [2]. The usual convergence theory of self-normalized importance sampling [e.g., 9] guarantees the consistency and asymptotically normality of our score function estimate (4).

Similarly, for the second derivative, the Monte Carlo estimate given each model is also consistent,

$$\frac{\partial^2}{\partial y^2} \log \pi_k(y) = \frac{\pi_k(y)\pi_k''(y) - [\pi_k']^2}{[\pi_k(y)]^2} \approx h_{ik} \coloneqq \frac{\sum_{s=1}^{S} f''(y|\theta_{ks})}{\sum_{s=1}^{S} f(y|\theta_{ks})} - [g_{ik}]^2\,.$$

2.3 Proposed method: optimizing the Hyvärinen score of the combined posterior densities

Our general model combination method contains five steps as follows:

Step 1: fit each model to the data and obtain K predictive densities. In practice the posteriors $p_k(\theta|\,\mathbf{D})$ are represented by Monte Carlo draws, $\theta_{k1}, \ldots, \theta_{kS}$, leading to the estimate $\pi_k(\cdot) \coloneqq \frac{1}{S}\sum_{s=1}^{S} p_k(\cdot|\theta_{ks})$. **Step 2:** express the unnormalized predictive density via the combination operator. For example, in locking we have $q(\cdot|w) \coloneqq \prod_k \pi_k^{w_k}(\cdot)$. **Step 3:** evaluate $\nabla_y \log q(\cdot|w)$ and $\Delta_y \log q(\cdot|w)$ at every observed y_i points. They come in *closed form functions* of $\nabla_y \pi_k(y_i|\theta_{ks})$ and $\Delta_y \pi_k(y_i|\theta_{ks})$. In locking:

$$q_i'(w) \coloneqq \nabla_y \log q(y_i|w) = \sum_{k=1}^{K} w_k \nabla_y \log\left(\pi_k(y_i)\right) \approx \sum_{k=1}^{K} w_k \frac{\sum_{s=1}^{S} \nabla_y p_k(y_i|\theta_{ks})}{\sum_{s=1}^{S} p_k(y_i|\theta_{ks})}, \tag{5}$$

$$q_i''(w) \coloneqq \Delta_y \log q(y_i|w) \approx \sum_{k=1}^{K} \frac{w_k}{S} \sum_{s=1}^{S} \left(\Delta_y \log p_k(y_i|\theta_{ks})\right). \tag{6}$$

The quacking derivatives also come in closed form expression (functions of weight w and phase α), but we omit them here.

Step 4: optimize model weight vector w by the constrained optimization

$$\hat{w}_{\text{opt}} = \min_{w} \left(\sum_{i=1}^{n} \left(2q_i''(w) + |q_i'(w)|^2\right) - \log \text{prior}(w) \right), \;\; \text{s.t.} \sum_{i=1}^{K} w_k = 1,\; w_i \ge 0. \tag{7}$$

We use an non-informative prior Dirichlet (1.01) for weight regularization.

[2] Rewrite the score funciton into (3) into $\int_{\mathbf{\Theta}} h(\theta) \frac{f(y|\theta)p_k(\theta|\,\mathbf{D})}{\int_{\mathbf{\Theta}} f(y|\theta)p_k(\theta|\,\mathbf{D})d\theta} d\theta$ admits the self-normalized importance sampling estimate.

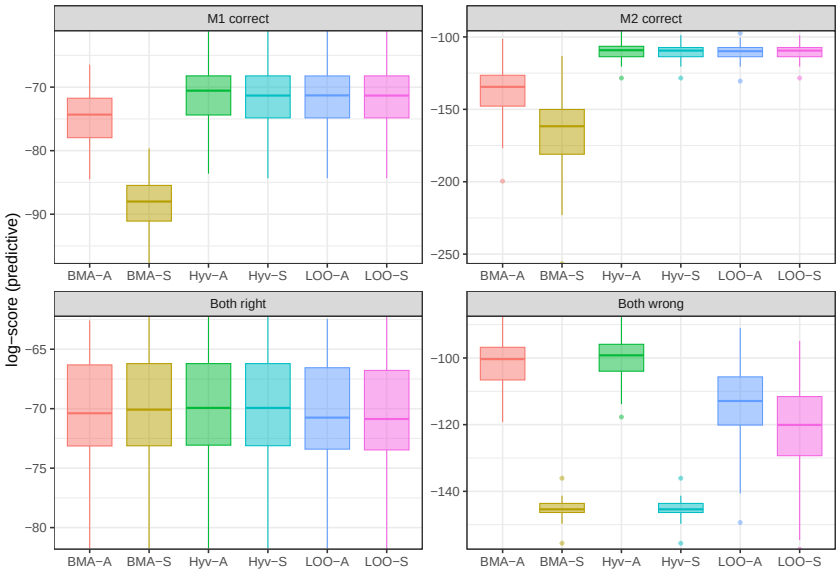

Figure 2: Log predictive scores. For each method, we show the overall predictive log score, $\sum_{j=1}^{N_{\text{pred}}} \log \pi_*(y_j)$. Our method (Hyvärinen model averaging, green) achieves higher log-scores in all four scenarios.

Complexity. The key blessing of applying scoring matching to Bayesian predictions is that the Mote Carlo integral is linear, and is exchangeable with gradient operators. Hence, all we need is to compute and store the gradient and hessian of the log likelihood (with respect to data) at the sampled parameters once, that is $\nabla_y \pi_k(y_i|\theta_{ks})$ and $\Delta_y \pi_k(y_i|\theta_{ks})$. In particular, the score functions have already been computed in gradient-based MCMC sampler, such as in dynamic Hamiltonian Monte Carlo [e.g., 3], hence nearly free. The summation in the objective function 7 contains nKS gradient evaluations in total, and can be computed in parallel.

3 Example

To manifest the flexibility of our new approach, in this section we run experiments and compare the locking method to other state of the art model averaging and selection tools. We adapt the experiment setting from Shao et al. [13]. Consider two normal belief models.

$$\begin{aligned}\mathcal{M}_1: &\quad Y_i \sim \text{Normal}(\theta_1, 1), \theta_1 \sim \text{Normal}(0, v_0),\\ \mathcal{M}_2: &\quad Y_i \sim \text{Normal}(0, \theta_2), \theta_2 \sim \text{Inverse-}\chi^2(\nu_0, \tau_0).\end{aligned}$$

Following Shao et al. [13], we picked $v_0 = 10$, $\nu_0 = 0.1$ and $\tau_0 = 1$. We simulate N_{train} data points from a true data generating process: a normal distribution with mean $\mu^\star$ and variance $v_\star$. We also generate a N_{test} independent test samples. We consider four scenarios: (1) $\mu^\star = 1$ and $v^\star = 1$ meaning that $\mathcal{M}_1$ is correctly specified but $\mathcal{M}_2$ is not; (2) $\mu^\star = 0$ and $v^\star = 5$ meaning that $\mathcal{M}_2$ is correctly specified but $\mathcal{M}_2$ is not; (3) $\mu^\star = 4$ and $v^\star = 3$, a situation in which neither model is correctly specified and (4) $\mu^\star = 0$ and $v^\star = 1$, in which both are correctly specified. We ran $M = 100$ replications of each scenario, with $N_{\text{train}} = 200$ and $N_{\text{test}} = 50$.

We compare six methods in total, (1) model selection using marginal likelihood, (2) Bayesian model averaging (3) model selection using leave-one-out log predictive densities [LOO-elpd, 14], (4) Bayesian stacking [15], (5) model selection using Hyvarinen score [13], and (6) *locking* (ours). We evaluate predictive performance of the learned combined model. To make the comparison fair, we pick a metric that we do not directly optimize over: the log predictive density on test data. As shown in Figure 2, our new *locking* method outperforms all its alternatives in terms of log-scores. We show that the Hyvärinen stacking (i.e. first optimising the weights using the Hyvärinen score on held-out data then forming a log-pool with the optmised weights) leads to higher log-predictive densities overall.

References

[1] Luiz M Carvalho, Daniel AM Villela, Flavio C Coelho, and Leonardo S Bastos. Bayesian inference for the weights in logarithmic pooling. *Bayesian Analysis*, 1(1):1–29, 2022.

[2] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

[3] Matthew D Hoffman, Andrew Gelman, et al. The No-U-Turn sampler: adaptively setting path lengths in hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

[4] Zaijing Huang and Andrew Gelman. Sampling for Bayesian computation with large datasets. *Technical Report, Columbia University*, 2005.

[5] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

[6] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[7] Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural computation*, 6(2):181–214, 1994.

[8] K Kamary, K Mengersen, CP Robert, and J Rousseau. Bayesian hypothesis testing as a mixture estimation model. *arXiv preprint arXiv:1412.2044*, 2018.

[9] Art B. Owen. *Monte Carlo theory, methods and examples*. 2013.

[10] David Poole and Adrian E Raftery. Inference for deterministic simulation models: the Bayesian melding approach. *Journal of the American Statistical Association*, 95(452):1244–1255, 2000.

[11] Adrian E Raftery, David Madigan, and Jennifer A Hoeting. Bayesian model averaging for linear regression models. *Journal of the American Statistical Association*, 92(437):179–191, 1997.

[12] MJ Rufo, J Martin, and CJ Pérez. Log-linear pool to combine prior distributions: A suggestion for a calibration-based approach. *Bayesian Analysis*, 7(2):411–438, 2012.

[13] Stephane Shao, Pierre E Jacob, Jie Ding, and Vahid Tarokh. Bayesian model comparison with the hyvärinen score: Computation and consistency. *Journal of the American Statistical Association*, 2019.

[14] Aki Vehtari, Andrew Gelman, and Jonah Gabry. Practical Bayesian model evaluation using leave-one-out cross-validation and waic. *Statistics and computing*, 27(5):1413–1432, 2017.

[15] Yuling Yao, Aki Vehtari, Daniel Simpson, and Andrew Gelman. Using stacking to average Bayesian predictive distributions. *Bayesian Analysis*, 13(3):917–1007, 2018.

[16] Yuling Yao, Gregor Pirš, Aki Vehtari, and Andrew Gelman. Bayesian hierarchical stacking: Some models are (somewhere) useful. *Bayesian Analysis*, 1(1):1–29, 2021.

[17] Seniha Esen Yuksel, Joseph N Wilson, and Paul D Gader. Twenty years of mixture of experts. *IEEE Transactions on Neural Networks and Learning Systems*, 23(8):1177–1193, 2012.