Revisiting Logistic-softmax Likelihood in Bayesian Meta-learning for Few-shot Classification

Tianjun Ke[†*] Haoqun Cao[†*] Zenan Ling[‡] Feng Zhou[†§]

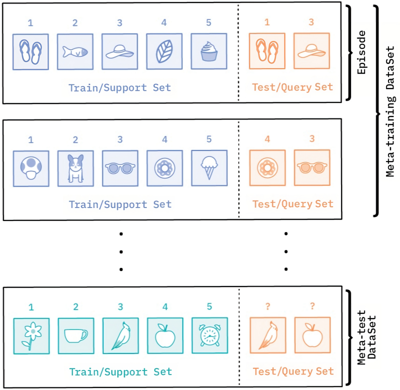
[†]Center for Applied Statistics and School of Statistics, Renmin University of China
[‡]School of EIC, Huazhong University of Science and Technology

November 11, 2023

[*]Equal contributions.
[§]Corresponding author.

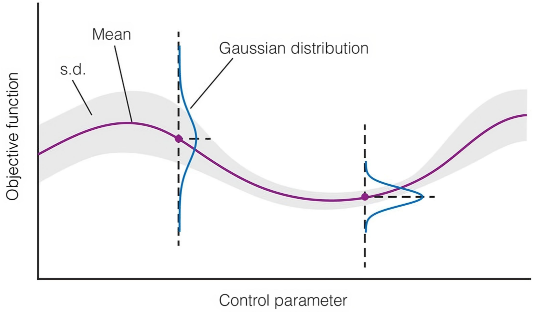
Meta-learning

Meta-learning involves learning from a set of tasks in order to acquire knowledge and generalize to new tasks.



Gaussian Process (GP) Classification

A GP is a probability distribution over functions, where $f(x)$ evaluated at a set of inputs have a joint Gaussian distribution. In the context of a C-class classification problem, separate GP latent functions $\{f^c\}_{c=1}^{C}$ are employed to model the logits for each class.



Deep Kernel

Deep Kernel combines kernel methods and neural networks, extending traditional covariance functions by integrating a deep architecture into the base kernel formulation. The deep kernel is defined as

$$k(\mathbf{x}, \mathbf{x}' \mid \boldsymbol{\theta}, \mathbf{w}) = k'(g_{\mathbf{w}}(\mathbf{x}), g_{\mathbf{w}}(\mathbf{x}') \mid \boldsymbol{\theta}),$$

where k' represents the base kernel with parameters $\boldsymbol{\theta}$ and g is a deep neural network parametrized by $\mathbf{w}$.

Motivation

1. The widely used softmax likelihood does not lead to conjugacy for GPs, making posterior inference intractable in classification.
2. While being conditional conjugate, the logistic-softmax function tends to exhibit an inherent lack of confidence.
3. Most GP-based meta-learning models employ Gibbs sampling for posterior inference, which can be computationally demanding for convergence.

Definition of Logistic-softmax with Temperature

We define the logistic-softmax function with temperature as:

$$p(y = k \mid \mathbf{f}_n) = \frac{\sigma(f_n^k/\tau)}{\sum_{c=1}^{C} \sigma(f_n^c/\tau)},$$

where we assume C classes, $f_n^c = f^c(\mathbf{x}_n)$, $\mathbf{f}_n = [f_n^1, \ldots, f_n^C]^\top$, $k \in [C] := \{1, \ldots, C\}$, τ is the temperature parameter and $\sigma(\cdot)$ is the logistic function.

Limiting Behavior

Although reminiscent of the softmax likelihood with temperature, the logistic-softmax likelihood displays distinct limiting behavior.

Limiting Behavior

Denote the logistic-softmax function with temperature as $\mathrm{LS}(\mathbf{f}_n, \tau)$. Define $I := \{i : f_n^i > 0\} \subset [C]$, we have

$$\lim_{\tau \to 0^+} \mathrm{LS}(\mathbf{f}_n, \tau) = \begin{cases} \boldsymbol{e}_{c^*}, & \text{if } \max_{c \in [C]} f_n^c < 0 \text{ and } c^* = \operatorname{argmax}_{c \in [C]} f_n^c \\ \frac{1}{|I|} \sum_{c \in I} \boldsymbol{e}_c, & \text{if } \max_{c \in [C]} f_n^c > 0 \end{cases}$$

where $\boldsymbol{e}_c \in \mathbb{R}^C$ is the one-hot vector with a 1 in its c-th coordinate.

Comparison of Logistic-softmax and Softmax with Temperature

We present several results demonstrating that logistic-softmax surpasses softmax as a versatile categorical likelihood function theoretically.

Pointwise Convergence

For all $\mathbf{f}_n \in \mathbb{R}^C$, $\tau \in \mathbb{R} \setminus \{0\}$ and $C_0 \in \mathbb{R}$, we have

$$\lim_{C' \to +\infty} \mathrm{LS}(\mathbf{f}_n - C', \tau) = \mathrm{S}(\mathbf{f}_n, \tau) = \mathrm{S}(\mathbf{f}_n - C_0, \tau),$$

where $\mathrm{S}(\mathbf{f}_n, \tau)$ denotes the softmax function with temperature.

Comparison of Logistic-softmax and Softmax with Temperature

Larger Size of Data Modelling Distribution Family

Assume the logits $f^c \sim \mathcal{GP}(a, k^c)$, where a is the mean function and k^c is the kernel function for each class $c \in [C]$. Denote $\mathbf{y} = [y_1, \ldots, y_N]^\top$ as the random label vector of N given points. Suppose $a \in \mathscr{A}$ and $k^c \in \mathscr{K}$, where $\mathscr{A}$ and $\mathscr{K}$ are two function classes. Define $\mathscr{F}(\ell \mid \mathscr{A}, \mathscr{K})$ as the family of the marginal distribution $p(\mathbf{y}|\mathbf{X}, a, k^c)$ induced by $a \in \mathscr{A}$ and $k^c \in \mathscr{K}$ on given points $\mathbf{X} \in \mathbb{R}^{N \times p}$ with a likelihood function ℓ. Under mild condition on $\mathscr{A}$, we have

$$\mathscr{F}(\mathrm{S} \mid \mathscr{A}, \mathscr{K}) = \mathscr{F}(\mathrm{S} \mid \mathscr{K}).$$

Furthermore, we have

$$\mathscr{F}(\mathrm{S} \mid \mathscr{A}, \mathscr{K}) \subset \mathscr{F}(\mathrm{LS} \mid \mathscr{K}).$$

Comparison of Logistic-softmax and Softmax with Temperature

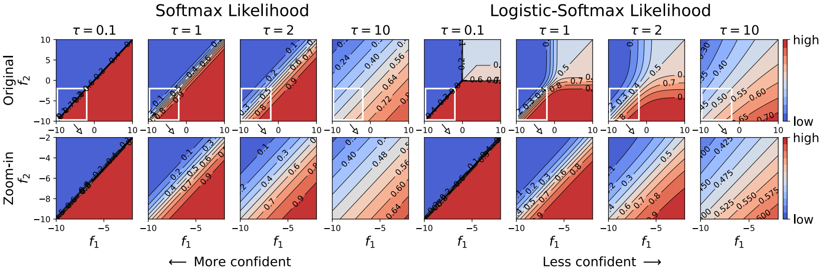
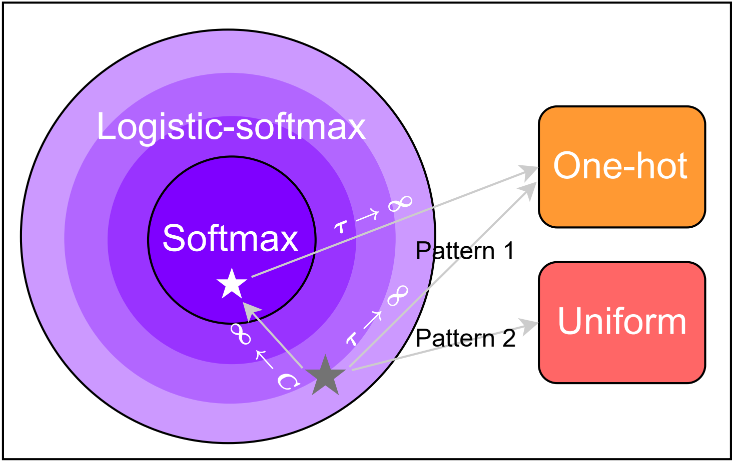


Figure: Plot of $p(y = 1|\mathbf{f})$ where f_3 clamped to -100. We provide separate zoom-in plots of softmax and logistic-softmax in the 2nd row. In the upper-right area (where all f_1 and f_2 are greater than 0), the logistic-softmax function exhibits unique probability patterns that softmax cannot model. In the bottom-left area (where all f_1 and f_2 are smaller than 0), logistic-softmax accurately approximates softmax.

Comparison of Logistic-softmax and Softmax with Temperature



Framework of Bayesian Meta-learning

Denote the input support and query data of task t as D_t^x, the target data as D_t^y. D^x and D^y are the collections of these datasets over all tasks. The marginal likelihood takes the form

task-specific parameters

$$p(D^y \mid D^x, \boldsymbol{\Theta}) = \prod_t \int p(D_t^y \mid D_t^x, \boldsymbol{\phi}_t)p(\boldsymbol{\phi}_t \mid \boldsymbol{\Theta})d\boldsymbol{\phi}_t.$$

task-common hyperparameters of deep kernel

The goal is to learn a generalizable $\boldsymbol{\Theta}$ via iterative bi-level optimization.

Task-level Bayesian Inference

Three sets of auxiliary latent variables are augmented to expand the logistic-softmax likelihood to obtain a conditional conjugate model for each task, including Gamma variables $\boldsymbol{\lambda}$, Poisson variables $\mathbf{M}$, and Pólya-Gamma variables $\boldsymbol{\Omega}$.

$$
\begin{aligned}
& p(\mathbf{Y}, \boldsymbol{\lambda}, \mathbf{M}, \boldsymbol{\Omega}, \mathbf{F}) \\
= & \prod_{n=1}^{N} \prod_{c=1}^{C} 2^{-(y_n^c + m_n^c)} \exp\left(\frac{y_n^c - m_n^c}{2} \frac{f_n^c}{\tau} - \frac{\omega_n^c}{2} \left(\frac{f_n^c}{\tau} \right)^2 \right) \\
& \cdot \mathrm{PG}(\omega_n^c \mid m_n^c + y_n^c, 0) \frac{\lambda_n^{m_n^c}}{m_n^c!} \exp(-\lambda_n) \cdot \prod_{c=1}^{C} \mathcal{N}(\mathbf{f}^c \mid \mathbf{a}^c, \mathbf{K}^c).
\end{aligned}
$$

Mean-field Approximation

In the mean-field algorithm, we need to approximate the true posterior $p(\boldsymbol{\lambda}, \mathbf{M}, \boldsymbol{\Omega}, \mathbf{F} \mid \mathbf{Y})$ by a variational distribution. Here, we assume $q(\boldsymbol{\lambda}, \mathbf{M}, \boldsymbol{\Omega}, \mathbf{F}) = q_1(\mathbf{M}, \boldsymbol{\Omega}) q_2(\boldsymbol{\lambda}, \mathbf{F})$ and obtain the optimal density for each factor:

$$q_1(\boldsymbol{\Omega}|\mathbf{M}) = \prod_{n,c=1}^{N,C} \mathsf{PG}(\omega_n^c \mid m_n^c + y_n^c, \widetilde{f}_n^c), \quad q_2(\boldsymbol{\lambda}) = \prod_{n=1}^{N} \mathsf{Ga}(\lambda_n \mid \alpha_n, C),$$

$$q_1(\mathbf{M}) = \prod_{n,c=1}^{N,C} \mathsf{Po}(m_n^c \mid \gamma_n^c), \quad q_2(\mathbf{F}) = \prod_{c=1}^{C} \mathcal{N}(\mathbf{f}^c \mid \widetilde{\boldsymbol{\mu}}^c, \widetilde{\boldsymbol{\Sigma}}^c),$$

Meta-level Optimization

The marginal likelihood is not tractable. Therefore we maximize the evidence lower bound (ELBO) to optimize $\boldsymbol{\Theta}$, which has an analytical expression because of the data augmentation. Moreover, we also consider predictive likelihood (PL), whose approximate gradient estimator is given by

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{PL}} \approx \frac{1}{M} \sum_{m=1}^{M} \nabla_{\boldsymbol{\theta}} \log p(y_* = k \mid \mathbf{x}_*, \mathbf{X}, \mathbf{Y}, \widehat{\boldsymbol{\Theta}}),$$

where M denotes the number of samples.

Prediction

The predictive probability of test label $y_* = k$ is:

$$p(y_* = k \mid \mathbf{x}_*, \mathbf{X}, \mathbf{Y}, \widehat{\mathbf{\Theta}}) = \int p(y_* = k \mid \mathbf{f}_*) \prod_{c=1}^{C} q(f_*^c \mid \mathbf{X}, \mathbf{Y}, \widehat{\mathbf{\Theta}}) d\mathbf{f}_*,$$

$$q(f_*^c \mid \mathbf{X}, \mathbf{Y}, \widehat{\mathbf{\Theta}}) = \int p(f_*^c \mid \mathbf{f}^c) q(\mathbf{f}^c \mid \mathbf{X}, \mathbf{Y}, \widehat{\mathbf{\Theta}}) d\mathbf{f}^c = \mathcal{N}(f_*^c \mid \mu_*^c, \sigma_*^{2^c}),$$

where $\sigma_*^{2^c} = k_{**}^c - \mathbf{k}_{*l}^c {\mathbf{K}_{ll}^c}^{-1} \mathbf{k}_{l*}^c + \mathbf{k}_{*l}^c {\mathbf{K}_{ll}^c}^{-1} \widetilde{\mathbf{\Sigma}}^c {\mathbf{K}_{ll}^c}^{-1} \mathbf{k}_{l*}^c$ and $\mu_*^c = \mathbf{k}_{*l}^c {\mathbf{K}_{ll}^c}^{-1} \widetilde{\boldsymbol{\mu}}^c$.

Few-shot Classification and Domain Transfer

Table: Average 1-shot and 5-shot accuracy and standard deviation on 5-way few-shot classification. Results are evaluated over 5 batches of 600 episodes with different random seeds. We highlight the best results in bold.

Method	CUB 1-shot	CUB 5-shot	mini-ImageNet 1-shot	mini-ImageNet 5-shot	mini-ImageNet → CUB 1-shot	mini-ImageNet → CUB 5-shot
Feature Transfer	46.19 ± 0.64	68.40 ± 0.79	39.51 ± 0.23	60.51 ± 0.55	32.77 ± 0.35	50.34 ± 0.27
Baseline++	61.75 ± 0.95	78.51 ± 0.59	47.15 ± 0.49	66.18 ± 0.18	39.19 ± 0.12	**57.31 ± 0.11**
MatchingNet	60.19 ± 1.02	75.11 ± 0.35	48.25 ± 0.65	62.71 ± 0.44	36.98 ± 0.06	50.72 ± 0.36
ProtoNet	52.52 ± 1.90	75.93 ± 0.46	44.19 ± 1.30	64.07 ± 0.65	33.27 ± 1.09	52.16 ± 0.17
RelationNet	62.52 ± 0.34	78.22 ± 0.07	48.76 ± 0.17	64.20 ± 0.28	37.13 ± 0.20	51.76 ± 1.48
MAML	56.11 ± 0.69	74.84 ± 0.62	45.39 ± 0.49	61.58 ± 0.53	34.01 ± 1.25	48.83 ± 0.62
DKT + Cosine	63.37 ± 0.19	77.73 ± 0.26	48.64 ± 0.45	62.85 ± 0.37	40.22 ± 0.54	55.65 ± 0.05
Bayesian MAML	55.93 ± 0.71	72.87 ± 0.26	44.46 ± 0.30	62.60 ± 0.25	33.52 ± 0.36	51.35 ± 0.16
Bayesian MAML (Chaser)	53.93 ± 0.72	71.16 ± 0.32	43.74 ± 0.46	59.23 ± 0.34	36.22 ± 0.50	51.53 ± 0.43
ABML	49.57 ± 0.42	68.94 ± 0.16	37.65 ± 0.22	56.08 ± 0.29	29.35 ± 0.26	45.74 ± 0.33
LS (Gibbs) + Cosine (ML)	60.23 ± 0.54	74.58 ± 0.25	46.75 ± 0.20	59.93 ± 0.31	36.41 ± 0.18	50.33 ± 0.13
LS (Gibbs) + Cosine (PL)	60.07 ± 0.29	78.14 ± 0.07	47.05 ± 0.20	66.01 ± 0.25	36.73 ± 0.26	56.70 ± 0.31
OVE PG GP + Cosine (ML)	63.98 ± 0.43	77.44 ± 0.18	**50.02 ± 0.35**	64.58 ± 0.31	39.66 ± 0.18	55.71 ± 0.31
OVE PG GP + Cosine (PL)	60.11 ± 0.26	79.07 ± 0.05	48.00 ± 0.24	**67.14 ± 0.23**	37.49 ± 0.11	57.23 ± 0.31
CDKT + Cosine (ML) ($\tau < 1$)	**65.21 ± 0.45**	**79.10 ± 0.33**	47.54 ± 0.21	63.79 ± 0.15	**40.43 ± 0.43**	55.72 ± 0.45
CDKT + Cosine (ML) ($\tau = 1$)	60.85 ± 0.38	75.98 ± 0.33	43.50 ± 0.17	59.69 ± 0.20	35.57 ± 0.30	52.42 ± 0.50
CDKT + Cosine (PL) ($\tau < 1$)	59.49 ± 0.35	76.95 ± 0.28	44.97 ± 0.25	60.87 ± 0.24	39.18 ± 0.34	56.18 ± 0.28
CDKT + Cosine (PL) ($\tau = 1$)	52.91 ± 0.29	73.34 ± 0.40	40.29 ± 0.14	60.23 ± 0.16	37.62 ± 0.32	54.32 ± 0.19

Uncertainty Quantification

Table: Expected calibration error (ECE) and maximum calibration error (MCE) for 5-shot 5-way tasks on CUB, mini-ImageNet, and domain-transfer. All metrics are computed on 3,000 random tasks from the test set.

Method	CUB ECE	CUB MCE	mini-ImageNet ECE	mini-ImageNet MCE	mini-ImageNet→CUB ECE	mini-ImageNet→CUB MCE
Feature Transfer	0.187	0.250	0.368	0.641	0.275	0.646
Baseline++	0.421	0.502	0.395	0.598	0.315	0.537
MatchingNet	0.023	0.031	0.019	0.043	0.030	0.079
ProtoNet	0.034	0.059	0.035	0.050	0.009	0.025
RelationNet	0.438	0.593	0.330	0.596	0.234	0.554
DKT + Cosine	0.187	0.250	0.287	0.446	0.236	0.426
Bayesian MAML	0.018	0.047	0.027	0.049	0.048	0.077
Bayesian MAML (Chaser)	0.047	0.104	0.010	0.071	0.066	0.260
LS (Gibbs) + Cosine (ML)	0.371	0.478	0.277	0.490	0.220	0.513
LS (Gibbs) + Cosine (PL)	0.024	0.038	0.026	0.041	0.022	0.042
OVE PG GP + Cosine (ML)	0.026	0.043	0.026	0.039	0.049	0.066
OVE PG GP + Cosine (PL)	**0.005**	**0.023**	**0.008**	0.016	0.020	0.032
CDKT + Cosine (ML)	**0.005**	0.036	0.009	**0.015**	**0.007**	**0.020**
CDKT + Cosine (PL)	0.018	0.223	0.025	0.140	0.010	0.029

1. Introduction
2. Method
3. Experiments
4. Conclusion

Conclusion

- Introduced the logistic-softmax function with temperature
- Delved into the theoretical property of the redesigned logistic-softmax function and its comparison with softmax
- Applied mean-field approximation for deep kernel based GP meta-learning for the first time
- Verified the results via extensive real-data experiments
- Shed some light on the coordination problem between the inner loop and the outer loop that appeared in bi-level optimization

Thanks!