EBMaC: Empirical Bayes and Matrix Constraints for Label Shift

Anonymous Authors[1]

Abstract

We estimate the importance weights and their associated confidence set in label shift problems using hierarchical models via the Empirical Bayes and Matrix Constraints (EBMaC) method. Our approach accommodates dispersion beyond what is permitted by the classic multinomial model and produces exact confidence regions in finite samples for confusion matrix and predicted labels. In addition, we describe the dependence structure of the importance weights in matrix constraints. Through a linear programming technique, we are able to compute smaller confidence sets and shorter elementwise confidence intervals for importance weights compared to existing methods, while maintaining the probability guarantee. Applying the results to prediction in the target domain directly yields smaller conformal prediction set and PAC prediction set. Numerical experiments demonstrate the advantages of EBMaC in producing tighter confidence sets for the importance weights both marginally and jointly.

1. Introduction

When we simultaneously consider data sets from different sources, problems of distribution shift naturally arise. The most frequently studied distribution shifts are covariate shift and label shift. Here, we focus on label shift, which describes the scenario where the marginal distributions of the labels differ in the source and the target domains, but given the label, the conditional distributions of covariates remain unchanged. The key quantity of interest is importance weights, *i.e.* the ratios of the label proportions between the two domains.

Given a classifier, there are three types of approaches for estimating the importance weights. The first one mainly relies on the linear relationship of the confusion matrix and the predicted label distribution (Lipton et al., 2018; Azizzadenesheli et al., 2019), and is named the confusion matrix method. The classifier is used to produce the confusion matrix in the source domain and to generate the predicted label distribution in the target domain. In forming the confusion matrix, either hard assignments or soft assignments can be implemented (Garg et al., 2020). The difference between BBSE (Lipton et al., 2018) and RLLS (Azizzadenesheli et al., 2019) is that BBSE pioneered the method while RLLS refined it by adding a regularization term on the importance weights to address potential near-singularity issues in the confusion matrix. The second one estimates the importance weights by maximum likelihood estimator (MLE). To this end, Saerens et al. (2002) proposed MLLS which finds the MLE by EM algorithm. Alexandari et al. (2020) proposed BCTS and demonstrated that further calibrating a classifier on the source domain significantly improves the MLE. The improvement happens because a classifier trained on the source domain may not perfectly represent the true proportions of the labels, even if it achieves high prediction probabilities (Guo et al., 2017). Such miscalibration biases the label predicting probability in the source domain and thus the estimated importance weights. The last one solves an estimating equation, formed by the projected score function, and is named ELSA (Tian et al., 2023). ELSA has the feature of being robust to an uncalibrated classifier, and it outperforms BCTS in computational efficiency while maintaining competitive accuracy.

In terms of the confidence intervals of the importance weights, most results hold only in the asymptotic sense. In finite samples, BBSE and RLLS rely on expressing the estimators explicitly in terms of confusion matrix and predicted label distribution. On the other hand, Si et al. (2023) proposed the Gaussian elimination (GE) method, where they modified each step of the Gaussian elimination procedure when solving the linear system in the confusion matrix method. Nevertheless, these methods do not produce tight confidence sets.

We propose EBMaC (Empirical Bayes and Matrix Constraints) method in the confusion matrix method class. We first construct confidence regions for the confusion matrix and the predicted label distributions using empirical Bayes method in a hierarchical model. It incorporates the overdispersion phenomenon, which is often encountered in

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

practice. We further take into account by recognizing it as a linear programming problem. This allows us to bypass matrix inversion and to obtain the tightest confidence sets for the importance weights. Furthermore, we demonstrate that applying the resulting confidence set yields the smallest finite sample prediction sets in the target domain. The superiority of EBMaC is rigorously proven in theory and illustrated through extensive numerical experiments.

2. Problem Setup

Let $\boldsymbol{x} \in \mathcal{X} = \mathbb{R}^d$ and $y \in \mathcal{Y} = \{1, \ldots, K\} \equiv [K] \subset \mathbb{N}^+$ denote the feature and the label, respectively. Let P and Q represent the source and target distributions, respectively. Then, let $p(\cdot)$ denote the pmf/pdf for the source domain and $q(\cdot)$ for the target domain. The letters in the () reflect corresponding random variables. For example, $p(y)$ is the marginal pmf of the labels in the source domain. The label shift setting assumes that $p(\boldsymbol{x} \,|\, y) = q(\boldsymbol{x} \,|\, y)$ while $p(y) \neq q(y)$ in general. Suppose we have a classifier g, defined as $g : \mathcal{X} \to \mathcal{Y}$. Then the $K \times K$ confusion matrix $\mathbf{C}_g$ is defined by $(\mathbf{C}_g)_{ij} \equiv \mathbb{P}_{(\mathbf{X},Y)\sim P}\{g(\mathbf{X}) = i, Y = j\}$, for $i, j \in [K]$, where $\mathbb{P}_{(\mathbf{X},Y)\sim P}$ stands for probability in source domain. In the target domain, we define the predicted label proportions, $\mathbf{q}_g = (q_1, ..., q_K)^\top$. For each $k \in [K]$, $q_k = \mathbb{P}_{\mathbf{X}\sim Q_\mathbf{X}}\{g(\mathbf{X}) = k\}$, where $Q_\mathbf{X}$ is the distribution of $\mathbf{X}$ in the target domain. We also use $\mathbb{P}$ to denote the probability in the combined domain. Let $\boldsymbol{\omega} = (\omega_1, ..., \omega_K)^\top$ be the importance weights, where $\omega_y = q(y)/p(y)$ for $y \in [K]$. We aim at the estimation and inference for $\boldsymbol{\omega}$.

Remark 2.1. Under label shift, for any classifier g such that $\widehat{y} = g(\boldsymbol{x})$, we have $p(\widehat{y} \,|\, y) = q(\widehat{y} \,|\, y)$. It is clear that $\sum_{y\in\mathcal{Y}} p(\widehat{y}, y)\omega_y = q(\widehat{y})$. When the confusion matrix $\mathbf{C}_g$ is invertible, solving the linear system $\mathbf{C}_g\boldsymbol{\omega} = \mathbf{q}_g$ is a valid method for estimating the importance weights $\boldsymbol{\omega}$. In practice, since the true values of $\mathbf{C}_g$ and $\mathbf{q}_g$ are unknown, we estimate them by the sample versions using the source and target data, as described in (Lipton et al., 2018).

3. Main Results

EBMaC incorporates multiple classifiers instead of treating a single one (Lipton et al., 2018), and adopts the empirical Bayes (EB) approach to estimate model parameters of $\mathbf{C}_g$'s and $\mathbf{q}_g$'s. In addition, EBMaC employs a linear programming method to solve for confidence regions for the importance weights $\boldsymbol{\omega}$, rather than the classic Gaussian elimination method (Si et al., 2023). Through these innovations, EBMaC can directly produce the estimation and inference results simultaneously for $\mathbf{C}_{g^*}$ and $\mathbf{q}_{g^*}$ of a chosen classifier g^*, which facilitates inference for $\boldsymbol{\omega}$. Together, implementing linear programming in combination with multiple classifiers enables EBMaC to achieve tighter elementwise confidence intervals for $\boldsymbol{\omega}$. As an end result, given the confidence intervals for $\mathbf{C}_{g^*}$ and $\mathbf{q}_{g^*}$, EBMaC provides the smallest possible confidence region for $\boldsymbol{\omega}$.

3.1. Estimation and Inference by Empirical Bayes

3.1.1. Bayesian Modeling

Let $\{(\boldsymbol{x}_s, y_s)\}_{s=1}^m$ be the source data and $\{\boldsymbol{x}_{m+t}\}_{t=1}^n$ be the target data. Let $\mathcal{G} = \{g_1, ..., g_G\}$ be a collection of G classifiers. Given a classifier $g \in \mathcal{G}$, we apply it on the set $\{\boldsymbol{x}_s\}_{s=1}^m$, resulting in $\widehat{y}_s = g(\boldsymbol{x}_s)$ for all $s = 1, \ldots, m$. Let $M_{g,ij} = \sum_{s=1}^m \mathbb{1}\{\widehat{y}_s = i, y_s = j\}$, then $\sum_{i=1}^K \sum_{j=1}^K M_{g,ij} = m$. For simplicity, we denote the vectorization of $[M_{g,ij}]$ by $\mathbf{M}_g$, *i.e.* $\mathbf{M}_g = (M_{g,1}, \ldots, M_{g,K^2})^\top$. Similarly, we denote $\boldsymbol{c}_g = \mathrm{vec}(\mathbf{C}_g)$, *i.e.* $\boldsymbol{c}_g = (c_{g,1}, \ldots, c_{g,K^2})^\top \in \Delta^{K^2-1}$, which is a (K^2-1)-dimensional probability simplex. We assume a hierarchical model

$$\begin{aligned} \mathbf{M}_g \,|\, \boldsymbol{c}_g &\sim \mathrm{Multinomial}(m, \boldsymbol{c}_g), \\ \boldsymbol{c}_g &\sim \mathrm{Dir}(\boldsymbol{\alpha}_s), \end{aligned}$$

where $\mathrm{Dir}(\boldsymbol{\alpha}_s)$ denotes the Dirichlet distribution, and $\boldsymbol{\alpha}_s = (\alpha_{s,1}, ..., \alpha_{s,K^2})^\top$ is the concentration hyperparameter. Given $\boldsymbol{\alpha}_s$, we assume that $\mathbf{c}_{g_1}, ..., \mathbf{c}_{g_G}$ are independent.

Additionally, given a classifier g, we write $\widehat{y}_{m+t} = g(x_{m+t})$. Let $N_{g,k} = \sum_{t=1}^n \mathbb{1}\{\widehat{y}_{m+t} = k\}$. Note that $\sum_{k=1}^K N_{g,k} = n$. We assume the hierarchical model for the target domain to be

$$\begin{aligned} \mathbf{N}_g \,|\, \mathbf{q}_g &\sim \mathrm{Multinomial}(n, \mathbf{q}_g), \\ \mathbf{q}_g &\sim \mathrm{Dir}(\boldsymbol{\alpha}_t). \end{aligned}$$

Here, $\boldsymbol{\alpha}_t = (\alpha_{t,1}, ..., \alpha_{t,K})^\top$ is the hyperparameter for the target data. Similarly, $\mathbf{q}_{g_1}, ..., \mathbf{q}_{g_G}$ are assumed to be independent given $\boldsymbol{\alpha}_t$.

3.1.2. Estimation of Hyperparameters

Because $\mathbf{c}_g$ is latent, in Appendix A.2, we derive the the marginal distribution of $\mathbf{M}_g$ given $\boldsymbol{\alpha}_s$ to be

$$f(\boldsymbol{m}_g; \boldsymbol{\alpha}_s) = \frac{\Gamma(m_0)\Gamma(m+1)}{\Gamma(m_0+m)} \prod_{k=1}^{K^2} \frac{\Gamma(\alpha_{s,k} + m_{g,k})}{\Gamma(\alpha_{s,k})\Gamma(m_{g,k}+1)},$$

where $m_0 = \sum_{j=1}^{K^2} \alpha_{s,j}$ and $\Gamma(\cdot)$ is the Gamma function. When we observe $(\boldsymbol{m}_{g_1}, ..., \boldsymbol{m}_{g_G})$, the log-likelihood is

$$\begin{aligned} & \ell(\boldsymbol{\alpha}_s; \boldsymbol{m}_{g_1}, ..., \boldsymbol{m}_{g_G}) \qquad (1) \\ = \; & \sum_{i=1}^G \log \left\{ \frac{\Gamma(m_0)\Gamma(m+1)}{\Gamma(m_0+m)} \prod_{k=1}^{K^2} \frac{\Gamma(\alpha_{s,k} + m_{g_i,k})}{\Gamma(\alpha_{s,k})\Gamma(m_{g_i,k}+1)} \right\} \\ \propto \; & G\{\log\Gamma(m_0) - \log\Gamma(m_0+m)\} \\ & + \sum_{i=1}^G \sum_{k=1}^{K^2} \{\log(\alpha_{s,k} + m_{g_i,k}) - \log\Gamma(\alpha_{s,k})\}. \end{aligned}$$

The partial derivative of with respect to $\alpha_{s,k}$ is

$$\begin{aligned}&\frac{\partial \ell(\boldsymbol{\alpha}_s; \boldsymbol{m}_{g_1}, ..., \boldsymbol{m}_{g_G})}{\partial \alpha_{s,k}} \\ = \;& G\{\psi(m_0) - \psi(m_0 + m)\} \\ &+ \sum_{i=1}^{G} \{\psi(\alpha_{s,k} + m_{g_i,k}) - \psi(\alpha_{s,k})\},\end{aligned}$$

where $\psi(x)$ is the digamma function, defined as $\psi(x) = d \log \Gamma(x)/dx$. When K is small, we implement numerical optimization in `Scipy` to minimize the negative of the log-likelihood in (1). If K is large, following Minka (2000), we find the maximum by fixed-point iteration. In the $(t+1)$-th iteration, we set

$$\alpha_{s,k}^{(t+1)} = \alpha_{s,k}^{(t)} \frac{G^{-1} \sum_{i=1}^{G} \{\psi(\alpha_{s,k}^{(t)} + m_{g_i,k}) - \psi(\alpha_{s,k}^{(t)})\}}{\psi(m_0^{(t)} + m) - \psi(m_0^{(t)})}, \quad \text{for } k = 1, ..., K^2,$$

and $m_0^{(t+1)} = \sum_k \alpha_{s,k}^{(t+1)}$. We use the moment matching estimation to get the initial $\boldsymbol{\alpha}_s^{(0)}$, with details in Appendix A.1. Similar procedure is conducted to estimate $\boldsymbol{\alpha}_t$ based on the target model and data. The only differences are in changing K^2 to K, $\boldsymbol{\alpha}_s$ to $\boldsymbol{\alpha}_t$, and $\boldsymbol{m}_{g_i}$ to $\boldsymbol{n}_{g_i}$ for $i \in [G]$. Let the estimators be $\widehat{\boldsymbol{\alpha}}_s$ and $\widehat{\boldsymbol{\alpha}}_t$.

3.1.3. Inference Based on the Posterior Distribution

Given a new classifier g^*, we aim at estimating $\mathbf{C}_{g^*}$ and $\mathbf{q}_{g^*}$, which are simplified as $\mathbf{C}^*$ and $\mathbf{q}^*$. Because Dirichlet distribution is the conjugate prior of multinomial distribution, the posterior distributions of $\mathbf{C}^*$ and $\mathbf{q}^*$ are still the Dirichlet distributions with updated parameters $\widetilde{\boldsymbol{\alpha}}_s = \widehat{\boldsymbol{\alpha}}_s + \boldsymbol{m}_{g^*}$ and $\widetilde{\boldsymbol{\alpha}}_t = \widehat{\boldsymbol{\alpha}}_t + \boldsymbol{n}_{g^*}$. Using the mode of posterior distributions, we estimate $\mathbf{C}^*$ and $\mathbf{q}^*$ by

$$\begin{aligned}\widehat{\mathbf{C}} &= \max(\widetilde{\mathbf{A}}_s - 1, 0)/(m_0 + m - K^2), \\ \widehat{\mathbf{q}} &= \max(\widetilde{\boldsymbol{\alpha}}_t - 1, 0)/(n_0 + n - K),\end{aligned}$$

where $\widetilde{\mathbf{A}}_s$ is a $K \times K$ matrix reshaped from $\widetilde{\boldsymbol{\alpha}}_s$.

Because there is no closed form to build a confidence set for the Dirichlet distribution, we consider each component of $\mathbf{C}$ and $\mathbf{q}$ marginally. A nice feature is that the marginal distributions of Dirichlet distributions are Beta distributions. Specifically, the marginal posterior distributions are

$$C_{ij}^* \mid \boldsymbol{m}_{g^*} \sim \text{Beta}(\widetilde{A}_{s,ij}, m_0 + m - \widetilde{A}_{s,ij}), \quad (2)$$

$$q_k^* \mid \boldsymbol{n}_{g^*} \sim \text{Beta}(\widetilde{\alpha}_{t,k}, n_0 + n - \widetilde{\alpha}_{t,k}), \quad (3)$$

for $i, j, k \in [K]$. Note that $\text{Beta}(a, b)$ has dramatically different shapes depending on a and b, hence we set the confidence intervals differently. For $a > 1$ and $b > 1$, it is unimodal, and we set the confidence interval from $(\delta/2)$-th to $(1 - \delta/2)$-th quantile. For $a \le 1$ and $b > 1$, it is monotonically decreasing, and the confidence interval is chosen from 0 to $(1 - \delta)$-th quantile. For $a > 1$ and $b \le 1$, it is monotonically increasing, and the confidence interval is set from δ-th quantile to 1. We exclude the case of $a \le 1$ and $b \le 1$, which cannot occur. We use $[\underline{C}_{ij}, \overline{C}_{ij}]$ and $[\underline{q}_k, \overline{q}_k]$ to denote the confidence intervals of level $(1 - \delta)$ for C_{ij}^* and q_k^* for $i, j, k \in [K]$.

3.2. Estimation and Inference of Importance Weights

Following Lipton et al. (2018), we estimate the importance weights by

$$\widehat{\boldsymbol{\omega}} = \max(\widehat{\mathbf{C}}^{-1}\widehat{\mathbf{q}}, 0),$$

where $\widehat{\mathbf{C}}^{-1}$ is the inverse of $\widehat{\mathbf{C}}$. However, we construct confidence sets of $\boldsymbol{\omega}$ very differently from the literature Si et al. (2023).

Let $\underline{\mathbf{C}} = (\underline{C}_{ij})$, $\overline{\mathbf{C}} = (\overline{C}_{ij})$, $\underline{\mathbf{q}} = (\underline{q}_k)$, and $\overline{\mathbf{q}} = (\overline{q}_k)$ be the collection of endpoints of confidence intervals for C_{ij}^* and q_k^*. For any matrices $\mathbf{A}$ and $\mathbf{B}$ of the same size, define $\mathbf{A} \le \mathbf{B}$ if $A_{ij} \le B_{ij}$ for all i, j, and similarly define $\mathbf{A} < \mathbf{B}$, $\mathbf{A} > \mathbf{B}$, and $\mathbf{A} \ge \mathbf{B}$. Let $\mathbf{Z} \in [\mathbf{A}, \mathbf{B}]$ if and only if $\mathbf{A} \le \mathbf{Z} \le \mathbf{B}$. Let $\boldsymbol{\mathcal{C}} = [\underline{\mathbf{C}}, \overline{\mathbf{C}}]$ and $\boldsymbol{\mathcal{Q}} = [\underline{\mathbf{q}}, \overline{\mathbf{q}}]$.

Given the relation $\mathbf{C}^*\boldsymbol{\omega} = \mathbf{q}^*$, it is readily obtained that $\boldsymbol{\omega} = \mathbf{C}^{*-1}\mathbf{q}^*$. Based on this explicit expression and the availability of $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{Q}}$, Si et al. (2023) constructed confidence interval for $\boldsymbol{\omega}$ through computing $\mathbf{C}^{-1}\mathbf{q}$ using Gaussian elimination. However, during this process, many relaxations are implemented, which result in inflation of the confidence set. Rather than solving for the explicit solution of $\boldsymbol{\omega}$, we directly impose the linear constraints on $\boldsymbol{\omega}$, which leads to

$$\boldsymbol{\Omega} = \{\boldsymbol{\omega} : \exists\, \mathbf{C} \in \boldsymbol{\mathcal{C}}, \mathbf{C}\boldsymbol{\omega} \in \boldsymbol{\mathcal{Q}}, \boldsymbol{\omega} > \mathbf{0}\}.$$

Although the definition of $\boldsymbol{\Omega}$ is clear, it is hard to implement because to verify $\boldsymbol{\omega} \in \boldsymbol{\Omega}$, we have to find the particular $\mathbf{C}$ such that the requirement is satisfied. An important discovery is the equivalence of $\boldsymbol{\Omega}$ and $\{\boldsymbol{\omega} : \overline{\mathbf{C}}\boldsymbol{\omega} \ge \underline{\mathbf{q}}, \underline{\mathbf{C}}\boldsymbol{\omega} \le \overline{\mathbf{q}}, \boldsymbol{\omega} > \mathbf{0}\}$. We present the result in Theorem 3.1 and the proof in Appendix B.1.

Theorem 3.1. $\{\boldsymbol{\omega} : \exists\, \mathbf{C} \in \boldsymbol{\mathcal{C}}, \mathbf{C}\boldsymbol{\omega} \in \boldsymbol{\mathcal{Q}}, \boldsymbol{\omega} > \mathbf{0}\} = \{\boldsymbol{\omega} : \overline{\mathbf{C}}\boldsymbol{\omega} \ge \underline{\mathbf{q}}, \underline{\mathbf{C}}\boldsymbol{\omega} \le \overline{\mathbf{q}}, \boldsymbol{\omega} > \mathbf{0}\}$.

Note that if we have used level $1 - \delta_{ij}$ for confidence interval $[\underline{C}_{ij}, \overline{C}_{ij}]$ and $1 - \delta_k$ for $[\underline{q}_k, \overline{q}_k]$, then the overall confidence level is $1 - (\sum_{i,j \in [K]} \delta_{ij} + \sum_{k \in [K]} \delta_k)$. Thus, if we want to reach a reasonable overall confidence level, then the individual confidence levels should be much higher.

Although Theorem 3.1 gives the clear description of the confidence set, it may have irregular shape. In practice,

people are often interested in more regular shapes such as hyperrectangle. For this purpose, we try to find the bounding hyperrectangle for $\mathbf{\Omega}$ by solving $2K$ optimization problems

$$\min_{\boldsymbol{\omega}} \omega_k \text{ subject to } \boldsymbol{\omega} \in \mathbf{\Omega}, \tag{4}$$

$$\max_{\boldsymbol{\omega}} \omega_k \text{ subject to } \boldsymbol{\omega} \in \mathbf{\Omega}, \tag{5}$$

for $k \in [K]$. Thanks to Theorem 3.1, $\boldsymbol{\omega} \in \mathbf{\Omega}$ contains $3K$ linear constraints, and ω_k is also a linear function of $\boldsymbol{\omega}$, we can solve the optimization problems using linear programming, by simplex algorithm for example. We denote the resulting bounding hyperrectangle as $\mathbf{\Omega}_{\mathrm{BH}} = \prod_{k=1}^{K} [\underline{\omega}_k, \overline{\omega}_k]$, where $\underline{\omega}_k$ and $\overline{\omega}_k$ are the corresponding minimizers and maximizers. We denote the Gaussian elimination-based confidence set of Si et al. (2023) by $\mathbf{\Omega}_{\mathrm{GE}}$ with detailed explanation in Appendix C. Then Corollary 3.2 holds, where the proof is in Appendix B.2.

Corollary 3.2. *When $\mathbf{\Omega}_{\mathrm{GE}}$ exists, $\mathbf{\Omega} \subset \mathbf{\Omega}_{\mathrm{BH}} \subset \mathbf{\Omega}_{\mathrm{GE}}$.*

3.3. Finite Sample Prediction

Benefiting from the confidence set of $\boldsymbol{\omega}$, we propose two ways of constructing a prediction set, conformal prediction (Vovk et al., 2005) and probably approximately correct (PAC) prediction (Valiant, 1984). We provide finite sample guarantee of the prediction set, while in the literature only asymptotic properties can be achieved.

3.3.1. Conformal Prediction

We consider the split conformal prediction setting, where the source data is divided into calibration set $S_1 = \{z_i = (\boldsymbol{x}_i, y_i) : i = 1, ..., m_1\}$ and training set $S_2 = \{(\boldsymbol{x}_i, y_i) : i = m_1 + 1, ..., m\}$. We assume that the nonconformity score $r(\boldsymbol{x}, y) \in [0, 1]$ is trained in S_2 and that the prediction set is then derived from the calibration set S_1. Let $\boldsymbol{x}_0$ be a new covariate in the target domain with the potential label y_0. Under label shift assumption, the calibration data set S_1 and $z_0 = (\boldsymbol{x}_0, y_0)$ satisfy the weighted exchangeability condition in Tibshirani et al. (2019), that is,

$$q(z_0, z_1, ..., z_{m_1}) = p(z_0, z_1 ..., z_{m_1}) \prod_{i=0}^{m_1} \omega_{y_i},$$

where $p(z_0, ..., z_{m_1}) = p(z_{\sigma(0)}, ..., z_{\sigma(m_1)})$ for any permutation $\sigma : \{0, ..., m_1\} \to \{0, ..., m_1\}$. Let $r_i = r(\boldsymbol{x}_i, y_i)$ for $i = 1, ..., m_1$. We can then create the level $(1 - \alpha)$ conformal prediction set $F_{\mathrm{CP}}(\boldsymbol{x}_0; \boldsymbol{\omega})$, denoted by

$$F_{\mathrm{CP}}(\boldsymbol{x}_0; \boldsymbol{\omega}) = \{y_0 \in [K] : r(\boldsymbol{x}_0, y_0) \le \tau_{\mathrm{CP}}(y_0; \boldsymbol{\omega})\},$$

where $\tau_{\mathrm{CP}}(y_0; \boldsymbol{\omega})$ is defined as

$$\begin{aligned} & \tau_{\mathrm{CP}}(y_0; \boldsymbol{\omega}) \\ = \; & Q_{1-\alpha}\Big(\sum_{i=1}^{m_1} \delta_{r_i} \frac{\omega_{y_i}}{\sum_{j=1}^{m_1} \omega_{y_j} + \omega_{y_0}} + \delta_1 \frac{\omega_{y_0}}{\sum_{j=1}^{m_1} \omega_{y_j} + \omega_{y_0}}\Big), \end{aligned}$$

where δ_r denotes the Dirac measure on r, and $Q_{1-\alpha}$ denotes the $(1 - \alpha)$-th sample quantile. When the true importance weight $\boldsymbol{\omega}$ is known, $F_{\mathrm{CP}}(\boldsymbol{x}_0; \boldsymbol{\omega})$ has a $1 - \alpha$ coverage rate by Theorem 2 of Podkopaev & Ramdas (2021). However, $\boldsymbol{\omega}$ is unknown in practice. Given a potential confidence set $\mathbf{\Omega}_0$ of $\boldsymbol{\omega}$, we can construct a prediction set by computing

$$\tau_{\mathrm{CP}}(y_0; \mathbf{\Omega}_0) = \sup_{\boldsymbol{\omega} \in \mathbf{\Omega}_0} \tau_{\mathrm{CP}}(y_0; \boldsymbol{\omega}) \tag{6}$$

and

$$F_{\mathrm{CP}}(\boldsymbol{x}_0; \mathbf{\Omega}_0) = \{y_0 \in [K] : r(\boldsymbol{x}_0, y_0) \le \tau_{\mathrm{CP}}(y_0; \mathbf{\Omega}_0)\}. \tag{7}$$

Compared to the known $\boldsymbol{\omega}$ case, the construction in (7) increases the confidence interval. However, we can still control the prediction level at $1 - \delta - \alpha$, as established in Theorem 3.3. See Appendix B.3 for the proof.

Theorem 3.3. *If $\mathbb{P}(\boldsymbol{\omega} \in \mathbf{\Omega}_0) \ge 1 - \delta$, then*

$$\mathbb{P}_{(\mathbf{X}_0, Y_0) \sim Q}\{Y_0 \in F_{\mathrm{CP}}(\mathbf{X}_0; \mathbf{\Omega}_0)\} \ge 1 - \delta - \alpha.$$

Note that $\mathbf{\Omega}_0$ can be obtained using the entire source data, while to guarantee the conformal prediction probability, we have to perform data splitting. Additionally, the advantage of $\mathbf{\Omega}_{\mathrm{BH}}$ over $\mathbf{\Omega}_{\mathrm{GE}}$ in Corollary 3.2 is inherited in the conformal prediction set, in that at the same prediction level, the prediction set based on $\mathbf{\Omega}_{\mathrm{BH}}$ is always smaller than that based on $\mathbf{\Omega}_{\mathrm{GE}}$. This property and the more general result are summarized in Theorem 3.4. The proof is provided in Appendix B.4.

Theorem 3.4. *If $\mathbf{\Omega}_1 \subset \mathbf{\Omega}_2$, then $F_{\mathrm{CP}}(\boldsymbol{x}_0; \mathbf{\Omega}_1) \subset F_{\mathrm{CP}}(\boldsymbol{x}_0; \mathbf{\Omega}_2)$ for all $\boldsymbol{x}_0$. In particular, $F_{\mathrm{CP}}(\boldsymbol{x}_0; \mathbf{\Omega}_{\mathrm{BH}}) \subset F_{\mathrm{CP}}(\boldsymbol{x}_0; \mathbf{\Omega}_{\mathrm{GE}})$.*

3.3.2. PAC prediction

Si et al. (2023) constructed PAC prediction set which relies on confidence interval $\mathbf{\Omega}_{\mathrm{GE}}$. Similar to section 3.3.1, if we replace $\mathbf{\Omega}_{\mathrm{GE}}$ by $\mathbf{\Omega}_{\mathrm{BH}}$, we can obtain a smaller prediction set with the same PAC guarantee (Park et al., 2021).

Theorem 3.5. *If $\mathbf{\Omega}_1 \subset \mathbf{\Omega}_2$, then $F_{\mathrm{PAC}}(\boldsymbol{x}_0; \mathbf{\Omega}_1) \subset F_{\mathrm{PAC}}(\boldsymbol{x}_0; \mathbf{\Omega}_2)$ for all $\boldsymbol{x}_0$. In particular, $F_{\mathrm{PAC}}(\boldsymbol{x}_0; \mathbf{\Omega}_{\mathrm{BH}}) \subset F_{\mathrm{PAC}}(\boldsymbol{x}_0; \mathbf{\Omega}_{\mathrm{GE}})$.*

See Appendix D for the construction of $F_{\mathrm{PAC}}(\boldsymbol{x}_0; \mathbf{\Omega}_0)$ and Appendix B.5 for the proof of Theorem 3.5.

4. Experiments

In this section, we implemented the EBMaC to evaluate its performance on the MNIST (LeCun et al., 1998), CIFAR-10, and CIFAR-100 data sets (Krizhevsky et al., 2009).

4.1. Classifier Training

For all data sets, we randomly selected 40,000 observations from training data set, combined with the 10,000 testing data to train classifiers, and used the remaining data to perform the analysis. For the MNIST data set, we trained 11 classifiers with different random seeds using the same architectures as in Azizzadenesheli et al. (2019). Each model was trained 10 epochs, and the best performer was retained. The final accuracy ranges from 97.25% to 98.07%. For the CIFAR data sets, we trained five classifiers using different architectures as shown in Table 1. Each classifier was trained 200 epochs, and the best performer was retained. In implementing EBMaC, we used the classifier with the lowest testing accuracy as g^*, and the remaining classifiers as $g_1, ..., g_G$, where $G = 10$ for MNIST and $G = 4$ for CIFAR-10 and CIFAR-100.

Table 1. Trained classifiers for CIFAR data sets with different accuracy on corresponding testing data sets.

Model	CIFAR-10	CIFAR-100
VGG16[a]	92.38%	71.80%
ResNet18[b]	93.98%	75.47%
MobileNetV2[c]	93.77%	70.23%
PreActResNet18[d]	93.97%	60.15%
RegNetX[e]	93.93%	–
GoogLeNet[f]	–	75.78%

[a] Simonyan (2014)
[b] He et al. (2016a)
[c] Sandler et al. (2018)
[d] He et al. (2016b)
[e] Radosavovic et al. (2020)
[f] Szegedy et al. (2015)

4.2. Experimental Design

To generate the data sets that satisfy the label shift assumption, we performed the following construction using Dirichlet shift. In generating the source data, we first generated a random vector $\mathbf{v}$ from $\mathrm{Dir}(\alpha \mathbf{1}_K)$. For each $k \in [K]$, we randomly draw $m\mathbf{v}_k$ observations, from those with $y = k$. Here m is the source sample size, and $\alpha = 10,000$. We generated the target data in the same way, except that $\alpha = 10^p$ with $p = -3, -2, -1, 0, 1, 2, 3$, and we did not retain the labels. We chose eight different m values, ranging from 1000 to 8000, with a step size of 1000. This resulted in 56 different data sets. In each data set, the sizes of the source data and target data are both m. Note that a smaller α leads to less balanced label distribution. This design is applied to MNIST, CIFAR-10 and CIFAR-100.

4.3. Performance of EBMaC on Importance Weights

We compared EBMaC to BBSE (Lipton et al., 2018), RLLS (Azizzadenesheli et al., 2019), and MLLS (Azizzadenesheli et al., 2019), using MSE $\|\boldsymbol{\omega} - \widehat{\boldsymbol{\omega}}\|^2$ as a criterion. The implementation code for the existing methods was adapted from Ye et al. (2024) [1]. For CIFAR-100, as shown in the first plot of Figure 1, as the sample size and concentration parameter α increase, the MSE for EBMaC decreases, while the remaining three plots show that the MSE of EBMaC is generally smaller than other methods. The results for MNIST and CIFAR-10 are in Figures 5 and 6 in Appendix E.

In our analysis, we found that in some classes in the target data, the variance of predicted label counts sometimes exceeds its mean, which violates the property of the multinomial distribution. However, the hierarchical modeling allows overdispersion by introducing additional hyperparameters, which extends the model flexibility. As shown in Tables 2, 3, and 4 in Appendix E, we observe that the CIFAR-100 data set has larger average variance-mean ratios compared to that of MNIST and CIFAR-10.

4.4. Performance of EBMaC on Confidence Sets

In obtaining confidence sets for CIFAR-100, we fix the same confidence level for $\boldsymbol{\Omega}$, $\boldsymbol{\Omega}_{\mathrm{BH}}$, and $\boldsymbol{\Omega}_{\mathrm{GE}}$, and present results in Figure 2. In the left panel, the x-axis represents the average length ratios of the GE method to the LP method, computed as $K^{-1}\sum_{k=1}^{K}(l_{k,\mathrm{GE}}/l_{k,\mathrm{BH}})$, where $l_{k,\mathrm{BH}} = \overline{\omega}_k - \underline{\omega}_k$, and $l_{k,\mathrm{GE}}$ is defined similarly. Here, GE was implemented using the code from Si et al. (2023)[2]. Further, we perform a one-sided t-test to evaluate whether the log-ratio of the lengths $\log(l_{k,\mathrm{GE}}/l_{k,\mathrm{BH}})$ is greater than zero across the labels. The resulting $-\log_{10}(P\text{-value})$ is shown in the y-axis. The horizontal line is at $-\log_{10}(0.05)$, representing the statistical significance. In the right panel, we provide the bar plot of the ratio of the volume of $\boldsymbol{\Omega}$ to $\boldsymbol{\Omega}_{\mathrm{BH}}$ at each α value, presented in percentage. The results for CIFAR-10 and MNIST are similarly presented in Figures 3 and 4.

In Figure 2, in the left panel, the vast majority is above the horizontal line, indicating that the improvement of the length ratio is significant in most cases. When comparing the results across three data sets, we find that EBMaC exhibits the best performance on CIFAR-100 in terms of both length ratio and P-value, but shows less improvement on MNIST. Note that all classifiers for MNIST have the best accuracy, while those for CIFAR-100 have the worst. This reflects that worse performance of classifiers generates more improvement of BH over GE. This is because worse classifiers lead to a confusion matrix that is less diagonal

[1] https://github.com/ChangkunYe/MAPLS
[2] https://github.com/averysi224/pac-ps-label-shift

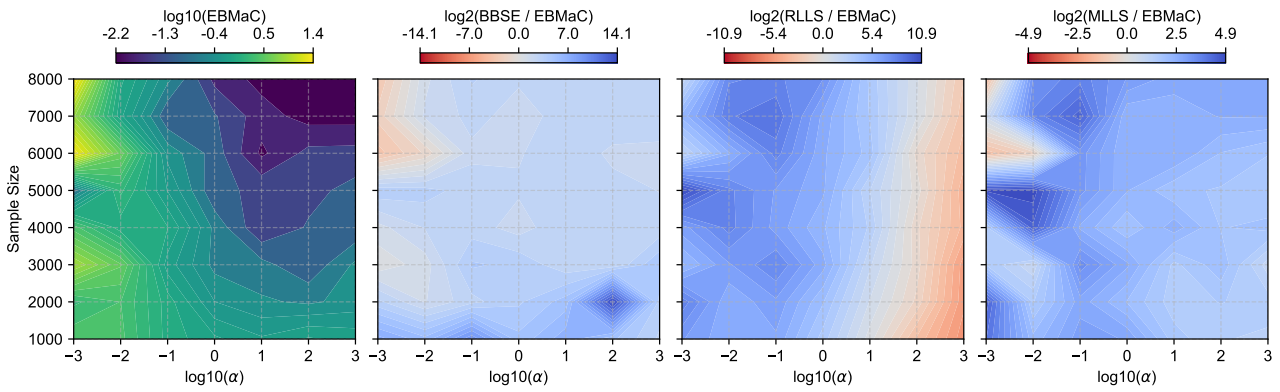

Figure 1. Comparison of label shift estimation methods on CIFAR-100. The first contour plot displays the average MSE of different classifiers in log10 scale for all data sets. The second contour plot shows the log2 ratio of MSE from BBSE to that from EBMaC. The third and fourth contour plots are similar to the second one, but they present the comparison results of RLLS and MLLS to that of EBMaC, respectively.

dominant, which can be handled by BH without any issue but results in much inflation of the confidence set by GE.

From the right panel of Figure 2, we can see that in the settings when α is large, the difference between $\mathbf{\Omega}$ and $\mathbf{\Omega}_{\mathrm{BH}}$ can be very large, reflected in very small volume ratio. This indicates that aiming for a hyperrectangular shape can be costly. In such case, it might be wiser to use $\mathbf{\Omega}$ to further perform conformal / PAC prediction. The performance in Figures 3 and 4 is slightly different in that the ratios of the volumes in the right panels are generally larger. This is because the shapes of $\mathbf{\Omega}$ resemble more a hyperrectangle for CIFAR-10 and MNIST, due to a more diagonal dominant confusion matrix.

5. Discussion

The main innovations of EBMaC are in proposing an empirical Bayesian approach in hierarchical modeling for label shift problems (EB) and in handling the matrix constraints via linear programming (MaC). EB is able to handle overdispersion, while MaC achieves the tightest confidence sets for importance weights. These two components can work separately. For example, we can combine Clopper-Pearson interval (ClP) with MaC to obtain ClPMaC. We can also combine EB with GE to create EBGE. One obvious advantage of EB is in handling overdispersion, while the advantage of MaC is established theoretically. When the collection of classifiers performs poorly, EBMaC showcases the significant improvement of MaC over GE. The nice property of EBMaC naturally leads to a better outcome of downstream analysis, such as the prediction performance described in 3.3.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Alexandari, A., Kundaje, A., and Shrikumar, A. Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation. In *International Conference on Machine Learning*, pp. 222–232. PMLR, 2020.

Azizzadenesheli, K., Liu, A., Yang, F., and Anandkumar, A. Regularized learning for domain adaptation under label shifts. *arXiv preprint arXiv:1903.09734*, 2019.

Garg, S., Wu, Y., Balakrishnan, S., and Lipton, Z. A unified view of label shift estimation. *Advances in Neural Information Processing Systems*, 33:3290–3300, 2020.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016a.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands,*

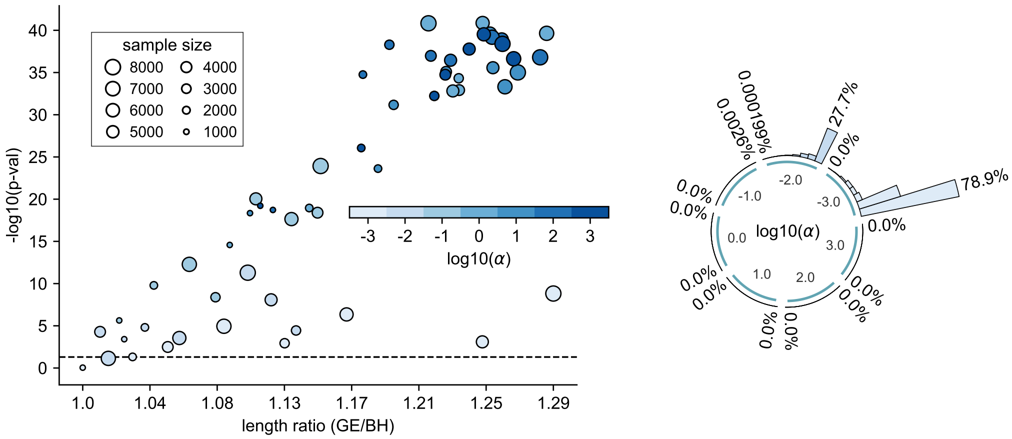

Figure 2. (CIFAR-100) Left panel: Comparison of $\mathbf{\Omega}_{\mathrm{GE}}$ and $\mathbf{\Omega}_{\mathrm{BH}}$. Right panel: Bar plot of ratios volume$(\mathbf{\Omega})/$ volume$(\mathbf{\Omega}_{\mathrm{BH}})$.

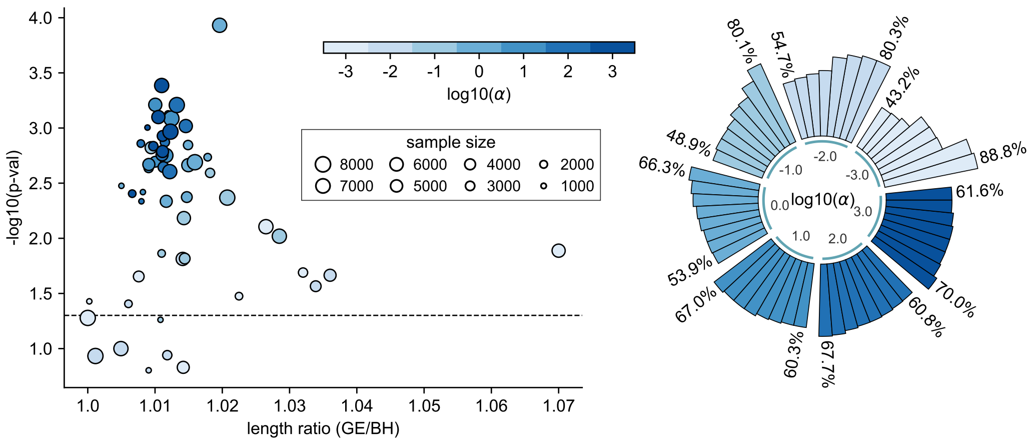

Figure 3. (CIFAR-10) Left panel: Comparison of $\mathbf{\Omega}_{\mathrm{GE}}$ and $\mathbf{\Omega}_{\mathrm{BH}}$. Right panel: Bar plot of ratios volume$(\mathbf{\Omega})/$ volume$(\mathbf{\Omega}_{\mathrm{BH}})$.

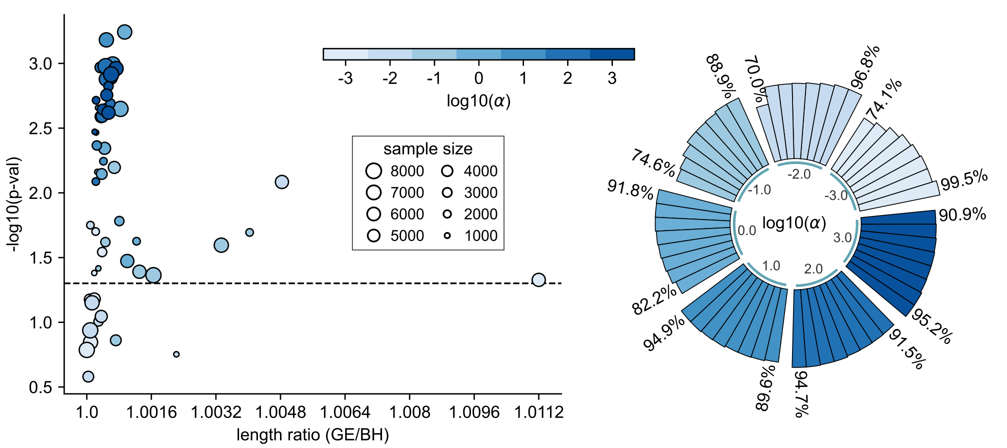

Figure 4. (MNIST) Left panel: Comparison of $\mathbf{\Omega}_{\mathrm{GE}}$ and $\mathbf{\Omega}_{\mathrm{BH}}$. Right panel: Bar plot of ratios volume$(\mathbf{\Omega})/$ volume$(\mathbf{\Omega}_{\mathrm{BH}})$.

October 11–14, 2016, Proceedings, Part IV 14, pp. 630–645. Springer, 2016b.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lipton, Z., Wang, Y.-X., and Smola, A. Detecting and correcting for label shift with black box predictors. In *International conference on machine learning*, pp. 3122–3130. PMLR, 2018.

Minka, T. Estimating a dirichlet distribution, 2000.

Park, S., Dobriban, E., Lee, I., and Bastani, O. PAC prediction sets under covariate shift. *arXiv preprint arXiv:2106.09848*, 2021.

Podkopaev, A. and Ramdas, A. Distribution-free uncertainty quantification for classification under label shift. In *Uncertainty in artificial intelligence*, pp. 844–853. PMLR, 2021.

Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K., and Dollár, P. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10428–10436, 2020.

Saerens, M., Latinne, P., and Decaestecker, C. Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural computation*, 14(1):21–41, 2002.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Si, W., Park, S., Lee, I., Dobriban, E., and Bastani, O. PAC prediction sets under label shift. *arXiv preprint arXiv:2310.12964*, 2023.

Simonyan, K. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Tian, Q., Zhang, X., and Zhao, J. ELSA: Efficient label shift adaptation through the lens of semiparametric models. In *International Conference on Machine Learning*, pp. 34120–34142. PMLR, 2023.

Tibshirani, R. J., Foygel Barber, R., Candes, E., and Ramdas, A. Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32, 2019.

Valiant, L. G. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Vovk, V. Conditional validity of inductive conformal predictors. In *Asian conference on machine learning*, pp. 475–490. PMLR, 2012.

Vovk, V., Gammerman, A., and Shafer, G. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

Ye, C., Tsuchida, R., Petersson, L., and Barnes, N. Label shift estimation for class-imbalance problem: A bayesian approach. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1073–1082, 2024.

A. Dirichlet-Multinomial model

A.1. Moment Matching Estimation for Dirichlet-Multinomial Model

We first give the moment matching estimation, which of course requires marginal statistics. Recall that we assume a common $\boldsymbol{\alpha}_s$ for all $g \in \mathcal{G}$ in Dirichlet prior. For any k in $\{1, ..., K^2\}$, the marginal expectation and variance of $m_{g,k}$ are

$$\begin{aligned} E(M_{g,k}) &= E\{E(M_{g,k} \mid \mathbf{c}_g)\} = E(mc_{g,k}) = mE(c_{g,k}) = m\mu_k \\ Var(M_{g,k}) &= E\{Var(M_{g,k} \mid \mathbf{c}_g)\} + Var\{E(M_{g,k} \mid \mathbf{c}_g)\} \\ &= E\{mc_{g,k}(1 - c_{g,k})\} + m^2 Var(c_{g,k}) \\ &= E(mc_{g,k}) - mE^2(c_{g,k}) - mVar(c_{g,k}) + m^2 Var(c_{g,k}) \\ &= m\mu_k(1 - \mu_k)\frac{m_0 + m}{m_0 + 1}, \end{aligned}$$

where $\mu_k = \alpha_{s,k}/m_0$ and $m_0 = \sum_{j=1}^{K^2} \alpha_{s,j}$. Next, matching them to the sample mean and the sample variance leads to following equations,

$$m\mu_k = G^{-1} \sum_{g=1}^{G} m_{g,k} \equiv \overline{m}_k \tag{8}$$

$$m\mu_k(1 - \mu_k)\frac{m_0 + m}{m_0 + 1} = (G - 1)^{-1} \sum_{g=1}^{G} (m_{g,k} - \overline{m}_k)^2 \equiv \widehat{V}_k. \tag{9}$$

From Equation 8, we know $\mu_k = \overline{m}_k/m$, Rearrange Equation 9 and replace μ_k with $\overline{m}_k/m$, we have

$$\begin{aligned} \widehat{M_0} &= \frac{m\overline{m}_k(m - \overline{m}_k) - m\widehat{V}_k}{m\widehat{V}_k - \overline{m}_k(m - \overline{m}_k)} \\ &= (m - 1)\left\{\frac{m\widehat{V}_k}{\overline{m}_k(m - \overline{m}_k)} - 1\right\}^{-1} - 1. \end{aligned}$$

Note that, we can have $\widehat{m_0}$ for each class, so we can average them to have a final $\widehat{M_0}$. Then we substitute it into $\alpha_{s,k} = m_0\overline{m}_k/m$ to obtain $\widehat{\alpha}_{s,k}$, for any $k \in \{1, ..., K^2\}$.

A.2. Marginal distribution for Dirichlet-Multinomial model

$$\begin{aligned} f(\boldsymbol{m}_g; \boldsymbol{\alpha}_s) &= \int_{\mathcal{C}} f(\boldsymbol{m}_{g,k} \mid \mathbf{c}_g) f(\mathbf{c}_g; \boldsymbol{\alpha}_s) d\mathbf{c}_g \\ &= \int_{\mathcal{C}} \frac{\Gamma(m_0)}{\prod_{k=1}^{K^2} \Gamma(\alpha_{s,k})} \prod_{k=1}^{K^2} c_{g,k}^{\alpha_{s,k}-1} \cdot \frac{\Gamma(m+1)}{\prod_{k=1}^{K^2} \Gamma(m_{g,k}+1)} \prod_{k=1}^{K^2} c_{g,k}^{m_{g,k}} \, d\mathbf{c}_g \\ &= \frac{\Gamma(m_0)}{\prod_{k=1}^{K^2} \Gamma(\alpha_k)} \frac{\Gamma(m+1)}{\prod_{k=1}^{K^2} \Gamma(m_{g,k}+1)} \int_{\mathcal{C}} \prod_{k=1}^{K^2} c_{g,k}^{\alpha_{s,k}-1} \prod_{k=1}^{K^2} c_{g,k}^{m_{g,k}} \, d\mathbf{c}_g \\ &= \frac{\Gamma(m_0)}{\prod_{k=1}^{K^2} \Gamma(\alpha_{s,k})} \frac{\Gamma(m+1)}{\prod_{k=1}^{K^2} \Gamma(m_{g,k}+1)} \int_{\mathcal{C}} \prod_{k=1}^{K^2} c_{g,k}^{\alpha_{s,k}+m_{g,k}-1} \, d\mathbf{c}_g \\ &= \frac{\Gamma(m_0)}{\prod_{k=1}^{K^2} \Gamma(\alpha_k)} \frac{\Gamma(m+1)}{\prod_{k=1}^{K^2} \Gamma(m_{g,k}+1)} B(\alpha_{s,1} + m_{g,1}, ..., \alpha_{s,K^2} + m_{g,K^2}) \\ &= \frac{\Gamma(m_0)\Gamma(m+1)}{\Gamma(m_0+m)} \prod_{k=1}^{K^2} \frac{\Gamma(\alpha_{s,k} + m_{g,k})}{\Gamma(\alpha_{s,k})\Gamma(m_{g,k}+1)} \end{aligned}$$

B. Proofs

B.1. Proof of Theorem 3.1

Proof. Suppose that $\mathbf{C} \in \boldsymbol{\mathcal{C}}, \mathbf{q} \in \boldsymbol{\mathcal{Q}}$, and that ω satisfies $\mathbf{C}\boldsymbol{\omega} = \mathbf{q}$ and $\omega > \mathbf{0}$. The linear equation $\mathbf{C}\boldsymbol{\omega} = \mathbf{q}$ is equivalent to $\sum_{j=1}^{K} c_{ij}\omega_j = q_i$. Since $\omega_i > 0$, we get that

$$\sum_{j=1}^{K} \overline{c}_{ij}\omega_j \geq \sum_{j=1}^{K} c_{ij}\omega_j = q_i \geq \underline{q}_i,$$
$$\sum_{j=1}^{K} \underline{c}_{ij}\omega_j \leq \sum_{j=1}^{K} c_{ij}\omega_j = q_i \leq \overline{q}_i,$$

which implies that $\boldsymbol{\omega} \in \boldsymbol{\Omega}$.

Now, we prove the other direction. Suppose that $\boldsymbol{\omega} \in \boldsymbol{\Omega}$. Then for each $i \in [K]$, we can apply the following procedure. If $\sum_{j=1}^{K} \overline{c}_{ij}\omega_j \leq \overline{q}_i, \forall i \in [K]$, take $\mathbf{c}_i^\top = (\overline{c}_{i1}, ..., \overline{c}_{iK})$ and $q_i = \sum_{j=1}^{K} \overline{c}_{ij}\omega_j$. Otherwise, for $l \in [K]$, define

$$q_i(l) = \sum_{j=1}^{l} \underline{c}_{ij}\omega_j + \sum_{j=l+1}^{K} \overline{c}_{ij}\omega_j.$$

Then $q_i(l)$ is a decreasing function of l, and $q_i(K) = \sum_{j=1}^{K} \underline{c}_{ij}\omega_j \leq \overline{q}_i$ by the condition. Thus, we can find l_0 such that $q_i(l_0 - 1) \geq \overline{q}_i \geq q_i(l_0)$. Let $\mathbf{c}_i^\top = (\underline{c}_{i1}, ..., \underline{c}_{i,l_0-1}, \widetilde{c}_{il_0}, \overline{c}_{i,l_0+1}, ..., \overline{c}_{iK})$ and $q_i = \overline{q}_i$, where

$$\widetilde{c}_{il_0} = \underline{c}_{il_0} + \frac{\overline{q}_i - q_i(l_0)}{\omega_{l_0}}.$$

Then $\widetilde{c}_{il_0} \in [\underline{c}_{il_0}, \overline{c}_{il_0}]$ and $\mathbf{c}_i^\top \omega = q_i$. Taking $\mathbf{c}_i^\top$ as the i-th row of $\mathbf{C}$ and q_i as the i-th element of $\mathbf{q}$, we get that $\mathbf{C}\omega = \mathbf{q}$, where $\mathbf{C} \in \boldsymbol{\mathcal{C}}$ and $\mathbf{q} \in \boldsymbol{\mathcal{Q}}$. □

B.2. Proof of Corollary 3.2

Proof. The first part, $\boldsymbol{\Omega} \subset \boldsymbol{\Omega}_{\mathrm{BH}}$, is trivial from its definition. For the second part, note that $\boldsymbol{\Omega} \subset \boldsymbol{\Omega}_{\mathrm{GE}}$ by Theorem 3.1. Since $\boldsymbol{\Omega}_{\mathrm{BH}}$ is the smallest hyperrectangle that contains $\boldsymbol{\Omega}$, we get $\boldsymbol{\Omega}_{\mathrm{BH}} \subset \boldsymbol{\Omega}_{\mathrm{GE}}$. □

B.3. Proof of Theorem 3.3

Proof. By Theorem 2 of Podkopaev & Ramdas (2021), we have $\mathbb{P}_{(\mathbf{X}_0, Y_0)\sim Q}\{Y_0 \in F_{\mathrm{CP}}(\mathbf{X}_0; \boldsymbol{\omega})\} \geq 1 - \alpha$. Also, if $\boldsymbol{\omega} \in \boldsymbol{\Omega}_0$, we get $F_{\mathrm{CP}}(\mathbf{X}_0; \boldsymbol{\omega}) \subset F_{\mathrm{CP}}(\mathbf{X}_0; \boldsymbol{\Omega}_0)$ by Theorem 3.4. Then

$$\begin{aligned}
& \mathbb{P}_{(\mathbf{X}_0, Y_0)\sim Q}\{Y_0 \notin F_{\mathrm{CP}}(\mathbf{X}_0; \boldsymbol{\Omega}_0)\} \\
= \; & \mathbb{P}_{(\mathbf{X}_0, Y_0)\sim Q}\{\boldsymbol{\omega} \in \boldsymbol{\Omega}_0 \text{ and } Y_0 \notin F_{\mathrm{CP}}(\mathbf{X}_0; \boldsymbol{\Omega}_0)\} + \mathbb{P}_{(\mathbf{X}_0, Y_0)\sim Q}\{\boldsymbol{\omega} \notin \boldsymbol{\Omega}_0 \text{ and } Y_0 \notin F_{\mathrm{CP}}(\mathbf{X}_0; \boldsymbol{\Omega}_0)\} \\
\leq \; & \mathbb{P}_{(\mathbf{X}_0, Y_0)\sim Q}\{Y \notin F_{\mathrm{CP}}(\mathbf{X}_0; \boldsymbol{\omega})\} + \mathbb{P}(\boldsymbol{\omega} \notin \boldsymbol{\Omega}_0) \\
\leq \; & \alpha + \delta.
\end{aligned}$$

□

B.4. Proof of Theorem 3.4

Proof. First, (6) implies $\tau_{\mathrm{CP}}(y; \boldsymbol{\Omega}_1) \leq \tau_{\mathrm{CP}}(y; \boldsymbol{\Omega}_2)$ for all y. Then (7) gives the result. □

B.5. Proof of Theorem 3.5

Proof. First, (11) implies $\tau_{\mathrm{PAC}}\{T(\boldsymbol{\Omega}_1, S_1, V, b)\} \leq \tau_{\mathrm{PAC}}\{T(\boldsymbol{\Omega}_2, S_1, V, b)\}$ for all y. Then (12) gives the result. □

C. Gaussian Elimination With Intervals

Given that $\mathbf{C}^* \in \mathcal{C} = [\underline{\mathbf{C}}, \overline{\mathbf{C}}]$ and $\mathbf{q}^* \in \mathcal{Q} = [\underline{\mathbf{q}}, \overline{\mathbf{q}}]$, Si et al. (2023) introduce an intuitive way, which they named Gaussian elimination with intervals, of finding $\mathbf{\Omega}$ that contains $\boldsymbol{\omega} = \mathbf{C}^{*-1}\mathbf{q}^*$. Suppose that $\underline{c}_{ij} \geq 0$, $\underline{q}_i > 0$, and $\omega_i > 0$ for $i, j \in [K]$. They follow two phases of Gaussian elimination when solving a system of linear equations $\mathbf{C}^*\boldsymbol{\omega} = \mathbf{q}^*$ and derive the elementwise interval for ω_i. First, set $\underline{c}_{ij}^0 = \underline{c}_{ij}$, $\overline{c}_{ij}^0 = \overline{c}_{ij}$, $\underline{q}_i^0 = \underline{q}_i$, and $\overline{q}_i^0 = \overline{q}_i$. In the first phase (forward elimination), the elementary row operations are applied sequentially for $k = 1, ..., K-1$ to delete the (i,k) element in the matrix for $i > k$ by adding the multiple of the k-th row. Then the lower bound $\underline{c}_{ij}^{k+1}$ and the upper bound $\overline{c}_{ij}^{k+1}$ are derived from the interval $[\underline{\mathbf{C}}^k, \overline{\mathbf{C}}^k]$ at the k-th step as

$$\underline{c}_{ij}^{k+1} = \begin{cases} 0, & \text{if } i > k, j \leq k, \\ \underline{c}_{ij}^k - \frac{\overline{c}_{ik}^k \overline{c}_{kj}^k}{\underline{c}_{kk}^k}, & \text{if } i, j > k, \\ \underline{c}_{ij}^k, & \text{otherwise.} \end{cases}$$

$$\overline{c}_{ij}^{k+1} = \begin{cases} 0, & \text{if } i > k, j \leq k, \\ \overline{c}_{ij}^k - \frac{\underline{c}_{ik}^k \underline{c}_{kj}^k}{\overline{c}_{kk}^k}, & \text{if } i, j > k, \\ \underline{c}_{ij}^k, & \text{otherwise.} \end{cases}$$

Simultaneously, $\underline{q}_i^{k+1}$ and $\overline{q}_i^{k+1}$ are obtained from the same row operations to be

$$\underline{q}_i^{k+1} = \begin{cases} \underline{q}_i^k - \frac{\overline{c}_{ik}^k \overline{q}_k^k}{\underline{c}_{kk}^k}, & \text{if } i > k, \\ \underline{q}_i^k, & \text{otherwise.} \end{cases}$$

$$\overline{q}_i^{k+1} = \begin{cases} \overline{q}_i^k - \frac{\underline{c}_{ik}^k \underline{q}_k^k}{\overline{c}_{kk}^k}, & \text{if } i > k, \\ \overline{q}_i^k, & \text{otherwise.} \end{cases}$$

Then $c_{ij}^{*,k+1}$ and $q_i^{*,k+1}$, which would have been obtained in the forward elimination step solving $\mathbf{C}^*\boldsymbol{\omega} = \mathbf{q}^*$, always lie in $[\underline{c}_{ij}^{k+1}, \overline{c}_{ij}^{k+1}]$ and $[\underline{q}_i^{k+1}, \overline{q}_i^{k+1}]$. In the second phase (back substitution), they compute $\underline{\omega}_i$ and $\overline{\omega}_i$, iteratively for $i = K, ..., 1$, replacing the truth with intervals as in the first phase.

$$\underline{s}_i = \sum_{j=i+1}^{K} \underline{c}_{ij}^K \underline{\omega}_j \quad \text{and} \quad \overline{s}_i = \sum_{j=i+1}^{K} \overline{c}_{ij}^K \overline{\omega}_j,$$
$$\underline{\omega}_i = \frac{\underline{q}_i - \overline{s}_i}{\overline{c}_{ii}^K} \quad \text{and} \quad \overline{\omega}_i = \frac{\overline{q}_i - \underline{s}_i}{\underline{c}_{ii}^K}.$$

Then $\mathbf{\Omega}_{\mathrm{GE}}$ is defined as the K-dimensional hyperrectangle $\prod_{i=1}^K [\underline{\omega}_i, \overline{\omega}_i]$. Si et al. (2023) provide a theoretical result that their method yields $\omega \in \mathbf{\Omega}_{\mathrm{GE}}$ if $\underline{c}_{ij}^k \geq 0$, $\underline{c}_{ii}^k > 0$, and $\underline{q}_i^k \geq 0$ for all $i, j, k \in [K]$. The basic assumption in order to satisfy the condition is that $\overline{c}_{ik} \ll \underline{c}_{kk}$. This is ensured when the classifier $g(X)$ is accurate, that is, when the diagonal terms c_{kk}^* in $\mathbf{C}^*$ dominate non-diagonal terms. If the assumption is violated, we may encounter a possibility that $c_{kk}^{*,k} \approx 0$, which may lead to $\underline{c}_{kk}^k \leq 0$. Then in the forward elimination phase, $\underline{c}_{ij}^{k+1}$ for all $i, j > k$ will be $-\infty$, which may make the algorithm impractical. Furthermore, if $\underline{q}_i \leq \overline{s}_i$ or $\underline{c}_{ii}^K \leq 0$ for some i, then the back substitution phase would lead to $\underline{\omega}_i \leq 0$ or $\overline{\omega}_i = \infty$, which does not provide any information about the interval of ω_i. In order to deal with the nonpositive bounds, they mention that choosing a wider margin, which would, however, make $\mathbf{\Omega}_{\mathrm{GE}}$ larger than its optimal size.

D. Details of PAC prediction

Let the calibration set be $S_1 = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m_1}$ and denote by $r(\boldsymbol{x}, y)$ the nonconformity score trained separately. The PAC prediction set $F_{\mathrm{PAC}}(\boldsymbol{x}; \boldsymbol{\omega}, S_1)$ under label shift (Vovk, 2012; Park et al., 2021; Si et al., 2023) is defined by

$$\mathbb{P}_{S_1 \sim P^{m_1}}[\mathbb{P}_{(\mathbf{X}_0, Y_0) \sim Q}\{Y_0 \in F_{\mathrm{PAC}}(\mathbf{X}_0; \boldsymbol{\omega}, S_1)\} \geq 1 - \epsilon] \geq 1 - \eta.$$

Si et al. (2023) constructed a set that satisfies a modification of PAC guarantee such that

$$\mathbb{P}_{S_1\sim P^{m_1},V}[\mathbb{P}_{(\mathbf{X}_0,Y_0)\sim Q}\{Y_0 \in F_{\text{PAC}}(\mathbf{X}_0;\boldsymbol{\omega},S_1,V,b)\} \geq 1-\epsilon] \geq 1-\eta, \tag{10}$$

where $V = (V_1, ..., V_{m_1})^\top \sim Unif([0,1])^{m_1}$ and $b = \max_{k\in[K]} \omega_k$. The set $F_{\text{PAC}}(\boldsymbol{x};\boldsymbol{\omega},S_1,V,b)$ is in the form of

$$F_{\text{PAC}}(x;\boldsymbol{\omega},S_1,V,b) = [y \in [K] : r(\boldsymbol{x},y) \leq \tau_{\text{PAC}}\{T(\boldsymbol{\omega},S_1,V,b)\}],$$

where $T(\boldsymbol{\omega},S_1,V,b) = \{(\boldsymbol{x}_i,y_i) \in S_1 : V_i \leq \omega_{y_i}/b\}$ is a target sample generated by rejection-sampling from S_1. Let $m_0 = |T(\boldsymbol{\omega},S_1,V,b)|$. Here, $\tau_{\text{PAC}}\{T(\boldsymbol{\omega},S_1,V,b)\}$ is chosen to satisfy

$$\begin{aligned}&\sum_{(\boldsymbol{x}_i,y_i)\in T(\boldsymbol{\omega},S_1,V,b)} \mathbb{1}\{y_i \notin F_{\text{PAC}}(\boldsymbol{x}_i;\boldsymbol{\omega},S_1,V,b)\}\\ &= \sum_{(\boldsymbol{x}_i,y_i)\in T(\boldsymbol{\omega},S_1,V,b)} \mathbb{1}[r(\boldsymbol{x}_i,y_i) > \tau_{\text{PAC}}\{T(\boldsymbol{\omega},S_1,V,b)\}] \leq k(m_0,\epsilon,\eta),\end{aligned}$$

that is, $\tau_{\text{PAC}}\{T(\boldsymbol{\omega},S_1,V,b)\}$ is the largest value that is less than the $k(m_0,\epsilon,\eta)$-th largest value of $\{r(\boldsymbol{x}_i,y_i) : (\boldsymbol{x}_i,y_i) \in T(\boldsymbol{\omega},S_1,V,b)\}$, where

$$k(m_0,\epsilon,\eta) = \max\{k : F_{\text{Binom}(m_0,\epsilon)}(k) \leq \eta\}.$$

Note that $F_{\text{Binom}(n,\epsilon)}(\cdot)$ is the CDF of $\text{Binom}(n,\epsilon)$. If the true importance weight $\boldsymbol{\omega}$ is used, then the modified PAC condition (10) is satisfied. When the confidence set $\boldsymbol{\Omega}_0$ with $\mathbb{P}(\boldsymbol{\omega} \in \boldsymbol{\Omega}) \geq 1-\delta$ is provided, we can define

$$\tau_{\text{PAC}}\{T(\boldsymbol{\Omega},S_1,V,b)\} = \sup_{\omega\in\boldsymbol{\Omega}} \tau_{\text{PAC}}\{T(\omega,S_1,V,b)\} \tag{11}$$

and

$$F_{\text{PAC}}(\boldsymbol{x};\boldsymbol{\Omega},S_1,V,b) = [y \in [K] : r(\boldsymbol{x},y) \leq \tau_{\text{PAC}}\{T(\boldsymbol{\Omega},S_1,V,b)\}]. \tag{12}$$

Then $F_{\text{PAC}}(\boldsymbol{x};\boldsymbol{\Omega},S_1,V,b)$ satisfies the modified PAC condition (10) with η being $\eta+\delta$.

Theorem D.1. *Suppose that* $\mathbb{P}(\boldsymbol{\omega} \in \boldsymbol{\Omega}_0) \geq 1-\delta$. *Then*

$$\mathbb{P}_{S_1\sim P^{m_1},V}[\mathbb{P}_{(\mathbf{X}_0,Y_0)\sim Q}\{Y_0 \in F_{\text{PAC}}(\mathbf{X}_0;\boldsymbol{\Omega},S_1,V,b)\} \geq 1-\epsilon] \geq 1-\eta-\delta.$$

Proof. The proof follows from Theorem 3 of Park et al. (2021). □

E. Data Dispersion

Table 2. (MNIST) Average variance-mean ratios for all classes under different sample size and Dirichlet shift combinations.

	$\log_{10}(\alpha)$						
sample size (m)	-3	-2	-1	0	1	2	3
8000	3.62	3.17	3.10	0.43	0.32	0.31	0.32
7000	6.05	8.06	5.78	0.58	0.30	0.25	0.29
6000	8.26	6.38	3.57	0.77	0.28	0.25	0.27
5000	3.77	4.73	1.44	0.38	0.22	0.23	0.26
4000	3.08	3.82	2.67	0.27	0.21	0.18	0.15
3000	4.97	3.07	1.07	0.33	0.15	0.13	0.15
2000	2.69	2.08	0.94	0.42	0.14	0.12	0.11
1000	1.04	1.15	0.79	0.09	0.08	0.09	0.07

Table 3. (CIFAR-10) Average variance-mean ratios for all classes under different sample size and Dirichlet shift combinations.

	$\log_{10}(\alpha)$						
sample size (m)	-3	-2	-1	0	1	2	3
8000	3.55	2.73	1.58	0.61	0.23	0.31	0.36
7000	5.70	5.72	2.33	0.63	0.23	0.18	0.17
6000	6.10	4.45	1.44	0.28	0.35	0.23	0.18
5000	3.84	3.58	0.77	0.22	0.23	0.14	0.20
4000	4.15	4.02	2.43	0.27	0.15	0.28	0.26
3000	2.74	3.92	0.79	0.27	0.23	0.13	0.14
2000	1.19	2.19	0.70	0.22	0.14	0.19	0.15
1000	1.06	0.85	0.83	0.16	0.16	0.21	0.13

Table 4. (CIFAR100) Average variance-mean ratios for all classes under different sample size and Dirichlet shift combinations.

	$\log_{10}(\alpha)$						
sample size (m)	-3	-2	-1	0	1	2	3
8000	25.95	12.42	3.70	1.36	0.95	0.86	0.90
7000	12.82	10.68	4.31	1.14	0.81	0.78	0.82
6000	11.02	12.91	3.68	1.18	0.75	0.77	0.74
5000	5.09	16.20	2.59	1.00	0.70	0.75	0.64
4000	10.08	8.03	3.07	0.93	0.77	0.57	0.65
3000	6.14	7.64	3.33	0.98	0.64	0.55	0.54
2000	2.93	2.96	2.02	0.71	0.53	0.53	0.55
1000	1.15	1.68	0.87	0.48	0.46	0.55	0.46

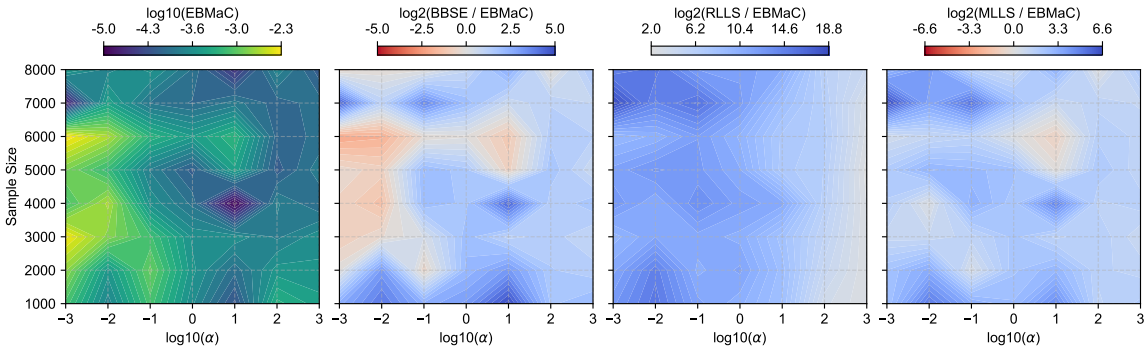

Figure 5. Comparison of label shift estimation methods on MNIST. The first contour plot displays the average MSE of different classifiers in log10 scale for all data sets. The second contour plot shows the log2 ratio of MSE from BBSE to that from EBMaC. The third and fourth contour plots are similar to the second one, but they present the comparison results of RLLS and MLLS to that of EBMaC, respectively.

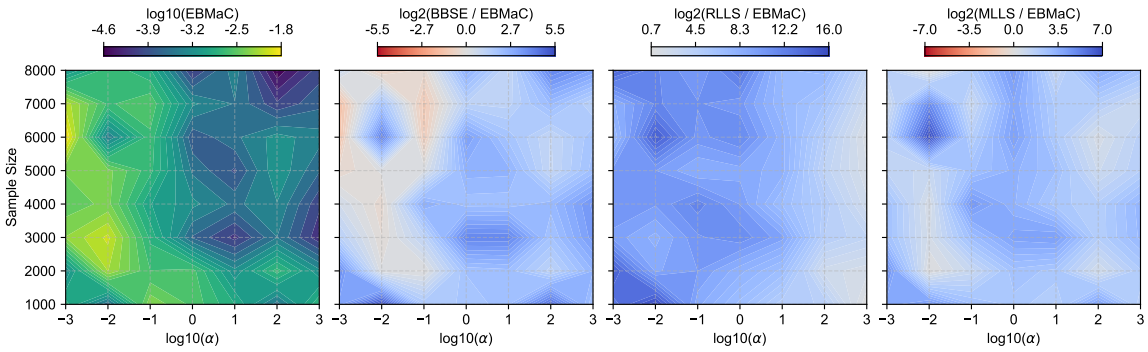

Figure 6. Comparison of label shift estimation methods on CIFAR-10. The first contour plot displays the average MSE of different classifiers in log10 scale for all data sets. The second contour plot shows the log2 ratio of MSE from BBSE to that from EBMaC. The third and fourth contour plots are similar to the second one, but they present the comparison results of RLLS and MLLS to that of EBMaC, respectively.