Maximum Likelihood with Bias-Corrected Calibration is Hard-To-Beat at Label Shift Adaptation

Amr M. Alexandari [*] [1] **Anshul Kundaje** [1] [2] **Avanti Shrikumar** [*] [1]

Abstract

Label shift refers to the phenomenon where the prior class probability $p(y)$ changes between the training and test distributions, while the conditional probability $p(\boldsymbol{x}|y)$ stays fixed. Label shift arises in settings like medical diagnosis, where a classifier trained to predict disease given symptoms must be adapted to scenarios where the baseline prevalence of the disease is different. Given estimates of $p(y|\boldsymbol{x})$ from a predictive model, Saerens et al. proposed an efficient maximum likelihood algorithm to correct for label shift that does not require model retraining, but a limiting assumption of this algorithm is that $p(y|\boldsymbol{x})$ is calibrated, which is not true of modern neural networks. Recently, Black Box Shift Learning (BBSL) and Regularized Learning under Label Shifts (RLLS) have emerged as state-of-the-art techniques to cope with label shift when a classifier does not output calibrated probabilities, but both methods require model retraining with importance weights and neither has been benchmarked against maximum likelihood. Here we (1) show that combining maximum likelihood with a type of calibration we call bias-corrected calibration outperforms both BBSL and RLLS across diverse datasets and distribution shifts, (2) prove that the maximum likelihood objective is concave, and (3) introduce a principled strategy for estimating source-domain priors that improves robustness to poor calibration. This work demonstrates that the maximum likelihood with appropriate calibration is a formidable and efficient baseline for label shift adaptation. See sec. 6 for links to code, video, and blogpost.

1. Introduction

Imagine we train a classifier to predict whether or not a person has a disease based on observed symptoms, and the classifier predicts reliably when deployed in the clinic. Suppose that there is a sudden surge in cases of the disease. During such an outbreak, the probability of persons having the disease given that they show symptoms rises, but the symptoms generated by the disease do not change. How can we adapt the classifier to cope with the difference in the baseline prevalence of the disease?

Formally, let y denote our labels (e.g. whether or not a person is diseased), and let $\boldsymbol{x}$ denote the observed symptoms. Let us denote the joint distribution $(\boldsymbol{x}, y)$ before the outbreak (our "source" domain) using the letter $\mathbb{P}$, and let us denote the distribution during the outbreak (our "target" domain, where we do not have labels) as $\mathbb{Q}$. How can we adapt a classifier trained to estimate $p(y|\boldsymbol{x})$ (the conditional probability in distribution $\mathbb{P}$) so that it can instead estimate $q(y|\boldsymbol{x})$ (the conditional probability in distribution $\mathbb{Q}$)? Absent assumptions about the nature of the shift between $\mathbb{P}$ and $\mathbb{Q}$, this problem is intractable. However, if the disease generates similar symptoms regardless of spread, we can assume that $p(\boldsymbol{x}|y) = q(\boldsymbol{x}|y)$, and that the shift in the joint distribution $q(\boldsymbol{x}, y)$ is due to a shift in the label proportion $q(y)$. Formally, we assume that $q(\boldsymbol{x}, y) = p(\boldsymbol{x}|y)q(y)$. This is known as *label shift* or *prior probability shift* (Storkey, 2008), and it corresponds to anti-causal learning (i.e. predicting the cause y from its effects $\boldsymbol{x}$) (Schoelkopf et al., 2012). Anti-causal learning is appropriate for diagnosing diseases given observations of symptoms because diseases cause symptoms.

Given estimates of $p(y)$ and $p(y|\boldsymbol{x})$, Saerens et al. (2002) proposed a simple Expectation Maximization (EM) procedure to estimate $q(y)$ without needing to estimate $p(\boldsymbol{x}|y)$. However, estimates of $p(y|\boldsymbol{x})$ derived from modern neural networks are often poorly calibrated (Guo et al., 2017), and the lack of calibration can decrease the effectiveness of EM. As an alternative, Lipton et al. (2018) developed a technique called Black Box Shift Learning (BBSL) that can work even when the predictions $p(y|\boldsymbol{x})$ are not calibrated. Azizzadenesheli et al. (2019) further improved upon BBSL in a technique known as Regularized Learning under Label

[*] Equal contribution [1] Department of Computer Science, Stanford University [2] Departments of Genetics, Stanford University. Correspondence to: Anshul Kundaje <anshul@kundaje.net>, Avanti Shrikumar <avanti.shrikumar@gmail.com>.

Proceedings of the 37th International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020. Copyright 2020 by the author(s).

Shifts (RLLS). Both BBSL and RLLS leverage information in a confusion matrix calculated on a held-out portion of the training set. To our knowledge, neither BBSL nor RLLS have been benchmarked against EM. Moreover, both BBSL and RLLS require model retraining using importance weighting, which does not work not as well as expected with deep neural networks (Byrd and Lipton, 2019), and RLLS also relies on a regularization hyperparameter. Conversely, EM requires neither retraining nor hyperparameter tuning.

Although the EM approach is limited by the assumption that the predictions $p(y|\boldsymbol{x})$ are calibrated, a number of recent techniques have been proposed to correct for miscalibration of $p(y|\boldsymbol{x})$ using a held-out portion of the training set (Guo et al., 2017). The held-out set can be thought of as analogous to the held-out set used in BBSL and RLLS to calculate a confusion matrix. This suggests a simple yet novel hybrid algorithm for adapting to label shift: first, calibrate predictions using the held-out training set, then perform domain adaptation on the calibrated predictions using EM. In this work, we studied the effectiveness of this hybrid algorithm. More generally, we studied the impact of calibration on domain adaptation to label shift.

1.1. Our Contributions

1. In experiments on MNIST, CIFAR10/CIFAR100, and Diabetic Retinopathy Detection, we found that EM achieves **state-of-the-art results** when used with an appropriate type of calibration. Although BBSL and RLLS both benefit from calibration, they did not tend to outperform EM when the probabilities were well-calibrated.

2. We observed that the popular calibration approach of Temperature Scaling (TS) (Guo et al., 2017) does not tend to achieve the best results for adaptation to label shift, possibly owing to large systematic biases in the calibrated probabilities (**Fig.** 1). Best results are obtained with variants of TS containing class-specific bias parameters that can correct for systematic bias.

3. We make two contributions to the algorithm for maximum likelihood label shift: first, we identify a principled strategy for computing the source-domain priors that improves robustness when the calibrated probabilities have systematic bias. Second, we prove that the likelihood function is concave and bounded; thus, the EM algorithm converges to a global maximum, and standard convex optimizers can be used as an alternative to EM.

2. Background

2.1. Temperature Scaling, Vector Scaling and Expected Calibration Error

Calibration has a long history in the machine learning literature (DeGroot and Fienberg, 1983; Platt, 1999; Zadrozny and Elkan, 2001; 2002; Niculescu-Mizil and Caruana, 2005; Kuleshov and Liang, 2015; Naeini et al., 2015; Kuleshov and Ermon, 2016). In the context of modern neural networks, Guo et al. (2017) showed that Temperature Scaling, a single-parameter variant of Platt Scaling (Platt, 1999), was effective at reducing miscalibration. Temperature scaling performs calibration by introducing a temperature parameter T to the logit vector of the softmax. Let $z(\boldsymbol{x}^k)$ represent a vector of the original softmax logits computed on input $\boldsymbol{x}^k$, and let y be a random variable representing the label. With temperature scaling, we have $p(y = i|\boldsymbol{x}^k) = \frac{e^{z(\boldsymbol{x}^k)_i/T}}{\sum_j e^{z(\boldsymbol{x}^k)_j/T}}$, where T is optimized with respect to the Negative Log Likelihood (NLL) on a held-out portion of the training set, such as the validation set. Guo et al. (2017) compared TS to an approach defined as Vector Scaling (VS), where a different scaling parameter was used for each class along with class-specfic bias parameters. Formally, in vector scaling, $p(y = i|\boldsymbol{x}^k) = \frac{e^{(z(\boldsymbol{x}^k)_i W_i)+b_i}}{\sum_j e^{(z(\boldsymbol{x}^k)_j W_j)+b_j}}$. Guo et al. (2017) found that vector scaling had a tendency to perform slightly worse than TS as measured by a metric known as the Expected Calibration Error (Naeini et al., 2015). To compute the ECE, the predicted probabilities for the output class are partitioned into M equally spaced bins, and the weighted average of the difference between the bin's accuracy and the bin's confidence is computed, where the weights are determined by the proportion of examples falling in the bin. Formally, ECE $= \sum_{m=1}^{M} \frac{|B_m|}{n} |\text{acc}(B_m) - \text{conf}(B_m)|$, where n is the number of samples.

2.2. Label Shift Adaptation via Maximum Likelihood

In an influential paper on label shift adaptation, Saerens et al. (2002) proposed an Expectation Maximization algorithm for estimating the shift in the class priors between the training and test distributions. Let $\hat{q}^{(s)}(y = i)$ denote the estimate (from EM iteration s) of the prior probability $q(y = i)$ of observing class i in the test set. The algorithm proceeds as follows: first, $\hat{q}^{(0)}(y = i)$ is initialized to be equal to the class priors $\hat{p}(y = i)$ estimated from the training set. Then, the conditional probabilities in the E-step are computed as $\hat{q}^{(s)}(y = i|\boldsymbol{x}_k) = \frac{\frac{\hat{q}^{(s)}(y=i)}{\hat{p}(y=i)} \hat{p}(y=i|\boldsymbol{x}_k)}{\sum_{j=1}^{m} \frac{\hat{q}^{(s)}(y=j)}{\hat{p}(y=j)} \hat{p}(y=j|\boldsymbol{x}_k)}$, where m is the number of output classes. Finally, the prior estimates in the M-step are updated as $\hat{q}^{(s+1)}(y = i) = \frac{1}{N} \sum_{k=1}^{N} \hat{q}^{(s)}(y = i|\boldsymbol{x}_k)$, where N is the number of examples in the testing set. The E and M steps are iterated until convergence. As there is

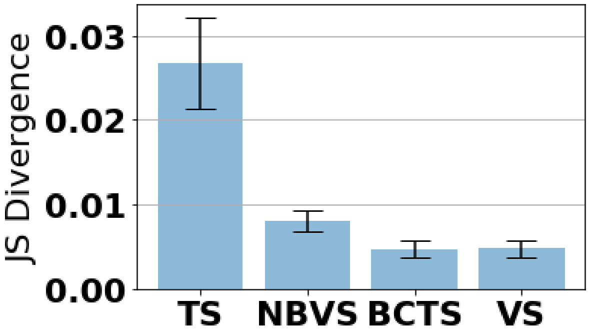

Figure 1. **Temperature Scaling exhibits systematic bias**. On CIFAR10 data, systematic bias was quantified by the JS divergence between the true class label proportions and the average class predictions on a held-out test set drawn from the same distribution as the dataset used for calibration. TS: Temperature Scaling, NBVS: No-Bias Vector Scaling, BCTS: Bias-Corrected Temperature Scaling, VS: Vector Scaling. BCTS and VS had significantly lower systematic bias compared to TS and NBVS. Results are averaged over multiple models and dataset samples (**Sec. 4.1**).

no need to estimate $p(\boldsymbol{x}|y)$ in any step of the EM procedure, the algorithm can scale to high-dimensional datasets, with a runtime of $O(mN)$ per iteration (independent of the input dimensions). The E-step is derived from Bayes' rule and is motivated by the assumption that $\hat{p}(y=i|\boldsymbol{x}_k)$ is calibrated.

2.3. Label Shift Adaptation via BBSL & RLLS

Following the EM approach of Saerens et al. (2002), several additional approaches for labels shift adaptation have emerged (Chan and Ng, 2005; Storkey, 2008; Schoelkopf et al., 2012; Zhang et al., 2013; Lipton et al., 2018; Azizzadenesheli et al., 2019). Many of these approaches build estimates $p(\boldsymbol{x}|y)$, which can scale poorly with dataset sizes and underperform on high-dimensional data (Lipton et al., 2018). Lipton et al. (2018) proposed Black-Box Shift Learning (BBSL), which strives to efficiently estimate the weights $[\boldsymbol{w}]_i = \frac{q(y=i)}{p(y=i)}$ even in cases where the prediction model $\hat{p}(y=i|\boldsymbol{x}_k)$ is poorly calibrated or biased. BBSL proceeds as follows: let f be a function that accepts an input and returns the model's predicted class, let $\boldsymbol{x}_k$ denote an example from a held-out portion of the training set, and let $\boldsymbol{x}'_k$ denote an example from the testing set. The empirical estimate of $\boldsymbol{w}$, denoted as $\hat{\boldsymbol{w}}$, is computed as $\hat{\boldsymbol{w}} = \hat{\boldsymbol{C}}_{\hat{y},y}^{-1}\hat{\boldsymbol{u}}_{\hat{y}}$, where $[\hat{\boldsymbol{u}}_{\hat{y}}]_i = \frac{\sum_k \mathbb{1}\{f(\boldsymbol{x}'_k)=i\}}{m}$ and $[\hat{\boldsymbol{C}}_{\hat{y},y}]_{ij} = \frac{1}{n}\sum_k \mathbb{1}\{f(\boldsymbol{x}_k) = i \text{ and } y_k = j\}$ (m and n denote the number of examples in the testing and held-out training set respectively). Because the approach above is not guaranteed to produce positive values for all elements of $\hat{\boldsymbol{w}}$, any negative elements of $\hat{\boldsymbol{w}}$ are set to 0 after they are estimated. Domain adaptation is then performed by retraining the model on the entire training set distribution with examples upweighted in accordance with $\hat{\boldsymbol{w}}$. Lipton et al. (2018) denote the version of BBSL described above as **BBSL-hard**. They also compare to a variant that they call **BBSL-soft**, which they describe as the case where where f outputs probabilities rather than hard classes. We interpreted this to mean $[\hat{\boldsymbol{u}}_{\hat{y}}]_i = \frac{\sum_k f(\boldsymbol{x}'_k)_i}{m}$ and $[\hat{\boldsymbol{C}}_{\hat{y},y}]_{ij} = \frac{1}{n}\sum_k f(\boldsymbol{x}_k)_i \mathbb{1}\{y_k = j\}$. Azizzadenesheli et al. (2019) further improved upon BBSL by including regularization terms in a technique known as Regularized Learning under Label Shift (RLLS). In our experiments, we compare to BBSL-hard, BBSL-soft, RLLS-hard and RLLS-soft. Regularization hyperparameters for RLLS were set in accordance with the hard-coded values given in the publicly available code provided by the authors at this url. Note that BBSL and RLLS both require a portion of the training set to be held out during the initial training phase in order to accurately estimate the confusion matrix $\hat{\boldsymbol{C}}_{\hat{y},y}$; in our experiments involving calibration, we use this same heldout set to calibrate the model.

3. Methods

3.1. No-Bias Vector Scaling and Bias-Corrected Temperature Scaling

As shown in **Fig. 1**, we often found that TS alone resulted in systematically biased estimates of $p(y=i|\boldsymbol{x}^k)$, while VS, a generalization of TS that contains both class-specific bias terms and class-specific scaling terms, did not exhibit as much systematic bias. Intrigued by this observation, we investigated the performance of two intermediaries between Temperature Scaling and Vector Scaling. The first, which we refer to as No Bias Vector Scaling (NBVS), is equivalent to vector scaling but with all the class-specific bias parameters fixed at zero: $p(y=i|\boldsymbol{x}^k) = \frac{e^{z(\boldsymbol{x}^k)_i W_i}}{\sum_j e^{z(\boldsymbol{x}^k)_j W_j}}$. The second, which we refer to as Bias-Corrected Temperature Scaling (BCTS), is equivalent TS Scaling but with the addition of the class-specific bias terms from VS: $p(y=i|\boldsymbol{x}^k) = \frac{e^{z(\boldsymbol{x}^k)_i/T+b_i}}{\sum_j e^{z(\boldsymbol{x}^k)_j/T+b_j}}$. As with TS and VS, the parameters are optimized to minimize the NLL on the validation set. Note that in the case of binary classification, the parameterization of BCTS reduces to Platt Scaling (Platt, 1999). Thus, BCTS can be viewed as a multi-class generalization of Platt scaling. Given fitted calibration parameters, performing calibration on test data takes $O(mN)$ time where m is the number of output classes and N is the number of datapoints.

3.2. Defining source-domain priors in the EM

The EM algorithm of Saerens et al. (2002) requires the user to provide estimates of the source-domain prior class probabilities $\hat{p}(y=i)$. Let us consider two possible approaches to estimating these probabilities. The first approach, considered in the original Saerens et al. (2002) paper, is to set $\hat{p}(y = i)$ to the expected value of the binary label $y = i$ over the source domain dataset. A second, less obvious, approach is to set it to the expected value of the predictions $\hat{p}(y = i|x)$ over the source domain dataset, formally denoted as $\mathbf{E}_{\boldsymbol{x} \sim p(\boldsymbol{x})}[\hat{p}(y = i|\boldsymbol{x})]$. If $\hat{p}(y = i|x)$ were unbiased, the two approaches would agree. However, depending on the calibration of $\hat{p}(y = i|\boldsymbol{x})$, this may not be the case, bringing us to:

Lemma A: In the absence of domain shift and in the limit of sufficient data, the EM algorithm will converge to the original priors $\hat{p}(y = i)$ if and only if $\hat{p}(y = i) := \mathbf{E}_{\boldsymbol{x} \sim p(\boldsymbol{x})}[\hat{p}(y = i|\boldsymbol{x})]$.

Proof: Note that the EM algorithm will converge when $\hat{q}^{(s+1)}(y = i) = \hat{q}^{(s)}(y = i)$. From the M-step, we know that $\hat{q}^{(s+1)}(y = i) = \frac{1}{N} \sum_{k=1}^{N} \hat{q}^{(s)}(y = i|\boldsymbol{x}_k)$, where the examples $\boldsymbol{x}_k$ are drawn from the target distribution. Substituting the formula for $\hat{q}^{(s)}(y = i|\boldsymbol{x}_k)$ from the E-step, we have $\hat{q}^{(s+1)}(y = i) = \frac{1}{N} \sum_{k=1}^{N} \frac{\frac{\hat{q}^{(s)}(y=i)}{\hat{p}(y=i)} \hat{p}(y=i|\boldsymbol{x}_k)}{\sum_{j=1}^{m} \frac{\hat{q}^{(s)}(y=j)}{\hat{p}(y=j)} \hat{p}(y=j|\boldsymbol{x}_k)}$. To prove **sufficiency**, we consider the scenario where $\hat{q}^{(s)}(y = i) = \hat{p}(y = i)$ and check whether convergence is attained. If the samples in the target distribution are drawn from the same distribution as the source, then in the limit of sufficient N, the value of $\hat{q}^{(s+1)}(y = 1)$ will approach $\mathbf{E}_{x \sim p(x)} \frac{\frac{1}{1}\hat{p}(y=i|\boldsymbol{x}_k)}{\sum_{j=1}^{m} \frac{1}{1}\hat{p}(y=j|\boldsymbol{x}_k)} = \mathbf{E}_{x \sim p(x)} \hat{p}(y = i|\boldsymbol{x}_k)$. As we defined $\hat{p}(y = i)$ to equal $\mathbf{E}_{x \sim p(x)} \hat{p}(y = i|\boldsymbol{x}_k)$, we get $\hat{q}^{(s+1)}(y = 1) = \hat{q}^{(s)}(y = 1)$, and convergence is attained. To prove **necessity**, we simply observe that if we had *not* defined $\hat{p}(y = i)$ to equal $\mathbf{E}_{x \sim p(x)} \hat{p}(y = i|\boldsymbol{x}_k)$, then we would not have $\hat{q}^{(s+1)}(y = 1) = \hat{q}^{(s)}(y = 1)$, and convergence would not be attained. ■

In the absence of domain shift, it is desirable that EM converge to the original priors $\hat{p}(y = i)$. In light of **Lemma A**, we set $\hat{p}(y = i)$ to the average value of $\hat{p}(y = i|\boldsymbol{x})$ over the source-domain validation set (we use the validation set to avoid the effects of overfitting on the training set; this is the same validation set used for calibration). If we instead compute $\hat{p}(y = i)$ by averaging binary labels in the validation set, we observe poor (even detrimental) performance with EM when the calibration lacks bias correction (**Tab.** A.1).

3.3. Convergence of Maximum Likelihood to the Global Optimum

We prove that the likelihood objective for label shift is concave. As a result, the EM algorithm of Saerens et al. (2002) converges to the global maximum likelihood estimate.

Lemma B: the maximum likelihood objective is concave.

Proof: Let ω_i denote membership in class i, and let $q(\omega_i)$ & $p(\omega_i)$ denote target & source domain priors. We seek target-domain priors $q(\boldsymbol{\omega})$ that maximize the log-likelihood $l(\boldsymbol{X}; q(\boldsymbol{\omega})) = \sum_k \log \sum_i q(\boldsymbol{x}_k|\omega_i) q(\omega_i)$. Because $q(\boldsymbol{x}_k|\omega_i)$ is not known explicitly, we rewrite the log likelihood in terms of $p(\omega_i|\boldsymbol{x}_k)$ and $p(\omega_i)$ as follows:

$$\begin{align*} l(\boldsymbol{X}; q(\boldsymbol{\omega})) &= \sum_k \log \sum_i q(\boldsymbol{x}_k|\omega_i) q(\omega_i) \\ &= \sum_k \log \sum_i p(\boldsymbol{x}_k|\omega_i) q(\omega_i) \tag{1} \\ &= \sum_k \log \sum_i \frac{p(\omega_i|\boldsymbol{x}_k) p(\boldsymbol{x}_k)}{p(\omega_i)} q(\omega_i) \tag{2} \\ &= \sum_k \log \left(p(\boldsymbol{x}_k) \sum_i \frac{p(\omega_i|\boldsymbol{x}_k)}{p(\omega_i)} q(\omega_i) \right) \\ &= \sum_k \log(p(\boldsymbol{x}_k)) + \log \sum_i \frac{p(\omega_i|\boldsymbol{x}_k)}{p(\omega_i)} q(\omega_i) \end{align*}$$

where (1) follows from the label shift assumptions and (2) follows from Bayes' rule. Now note that the maximization is independent of $p(\boldsymbol{x}_k)$. Using the constant C to denote $\sum_k \log(p(\boldsymbol{x}_k))$, the objective can be written as:

$$\begin{aligned} \max_{q(\boldsymbol{\omega})} \quad & C + \sum_k \log \sum_i \frac{p(\omega_i|\boldsymbol{x}_k)}{p(\omega_i)} q(\omega_i) \\ \text{s.t.} \quad & \mathbf{1}^T \cdot q(\boldsymbol{\omega}) = 1 \\ & q(\omega_i) \geq 0 \quad \forall i \end{aligned} \tag{3}$$

Both $p(\omega_i|\boldsymbol{x}_k)$ and $p(\omega_i)$ are constants with respect to $q(\boldsymbol{\omega})$. Hence, the objective is a constant plus the sum of logs of linear functions in our decision variable, and the constraints are affine. Therefore, the maximization problem is concave. ■

Lemma C: given that $p(\omega_i) \geq \epsilon \; \forall i$ (i.e. every class considered has a non-zero probability of occurrence in the source domain), the log likelihood is bounded above by $C + N \log(1/\epsilon)$.

Proof: We give a loose bound. Using the form of the likelihood derived in **Eqn.** 3, we have:

$$\begin{align*} l(\boldsymbol{X}; q(\boldsymbol{\omega})) &= C + \sum_k \log \left(\sum_i \frac{p(\omega_i|\boldsymbol{x}_k)}{p(\omega_i)} q(\omega_i) \right) \tag{4} \\ &\leq C + \sum_k \log \left(\sum_i p(\omega_i|\boldsymbol{x}_k) \cdot \frac{q(\omega_i)}{\epsilon} \right) \tag{5} \\ &\leq C + \sum_k \log(1/\epsilon \cdot \sum_i p(\omega_i|\boldsymbol{x}_k)) \tag{6} \\ &\leq C + N \cdot \log(1/\epsilon) \tag{7} \end{align*}$$

where (5) is using the given assumption on $p(\omega_i)$, (6) is using the fact that probabilities are ≤ 1, and (7) is due to the fact that probabilities sum up to one. ■

Note that a similar assumption to $p(\omega_i) \geq \epsilon \; \forall i$ is adopted by BBSL; if the source-domain probability of a class were 0, we would simply exclude that class from the optimization.

As a consequence of the likelihood being concave and bounded, standard convex optimization algorithms would converge to the global maximum likelihood estimate. We used the Expectation Maximization (EM) algorithm in this work for ease of implementation. For completeness, we include the convergence proof for EM in Appendix **I**.

3.4. Summary of Proposed Algorithm

1. Given a model f that outputs predicted probabilities, calibrated the predictions of f on a held-out validation using an appropriately strong calibration algorithm. In this work, we observed that Bias-Corrected Temperature Scaling and Vector Scaling work well.

2. Average the calibrated predictions $\hat{p}(y = i|\boldsymbol{x})$ over this held-out validation set to obtain the estimated source-domain class priors $\hat{p}(y = i)$.

3. Given samples from the target domain, use the estimated $\hat{p}(y = i)$ and the calibrated predictor $\hat{p}(y = i|\boldsymbol{x})$ to optimize the concave maximum likelihood objective in **Eqn. 3** w.r.t. the estimated target-domain class proportions $\hat{q}(y = i)$. The EM algorithm from **Sec 2.2** can be used for this.

4. After finding $\hat{q}(y = i)$, compute the adapted predictions for the target domain as follows (this is similar to the E-step of EM, and is derived by combining the label shift assumption with Bayes' rule):

$$\hat{q}(y = i|\boldsymbol{x}) = \frac{\frac{\hat{q}(y=i)}{\hat{p}(y=i)}\hat{p}(y = i|\boldsymbol{x})}{\sum_{j=1}^{m} \frac{\hat{q}(y=j)}{\hat{p}(y=j)}\hat{p}(y = j|\boldsymbol{x})} \tag{8}$$

3.5. Metrics for evaluating adaptation to label shift

The first metric we consider is the mean squared error in the true weights compared to the estimated weights (Azizzadenesheli et al., 2019; Lipton et al., 2018). Let us denote the true target-domain prior as $q(y = i)$ and the true source domain prior as $p(y = i)$. The true class weights are defined as $\boldsymbol{w}_i := q(y = i)/p(y = i)$. Both BBSL and RLLS directly output estimated weights $\hat{\boldsymbol{w}}_i$. For Maximum Likelihood, the weights can be obtained by dividing the estimated target-domain priors $\hat{q}(y = i)$ by the source-domain priors $\hat{p}(y = i)$ (where the source priors are computed as described in Sec. 3.2). The mean squared error of the weights is then simply $\frac{1}{N}\sum_i(\hat{\boldsymbol{w}}_i - \boldsymbol{w}_i)^2$, where N is the number of classes.

The second metric we consider is the improvement in accuracy of the domain-adapted model predictions relative to using the original model predictions. Given the ratio $\hat{q}(y = i)/\hat{p}(y = i)$, the adapted model predictions can be computed as in **Eqn. 8**. For Maximum Likelihood, we use these adapted predictions to assess accuracy. By contrast, in both the BBSL and RLLS papers, model retraining was performed to obtain adapted predictions. Due to computational constraints, as well as recent observations that retraining deep neural networks using importance weights does not work as well as expected (Byrd and Lipton, 2019), we did not perform model retraining. Thus, we use the MSE of importance weights to compare Maximum Likelihood to BBSL and RLLS, and use accuracy only for comparing the impact of different calibration algorithms on the Maximum Likelihood estimate[1].

4. Results

4.1. Experimental Setup

We evaluated the efficacy of BBSL, RLLS and Maximum Likelihood coupled to different calibration approaches on MNIST, CIFAR10, CIFAR100, and a diabetic retinopathy detection dataset.

For MNIST, CIFAR10, and CIFAR100, we trained ten different models (each with a different random seed), and reserved 10K examples of the training set as a held-out validation set. For MNIST, we used the architecture from Azizzadenesheli et al. (2019), and for CIFAR10 & CIFAR100, we used the architecture from Geifman and El-Yaniv (2017). Dirichlet label shift was simulated on the testing set by sampling (with replacement) according to class proportions generated by a dirichlet distribution with uniform α values of 0.1, 1.0 and 10.0 (smaller values of α result in more extreme label shift). Samples from the validation set were used for calibration, EM initialization and BBSL & RLLS confusion matrix estimation. Accuracy was reported on the label-shifted testing set, while the calibration metrics of NLL and ECE (with 15 bins) were reported on the unshifted testing set. In addition to exploring different degrees of dirichlet shift, we also investigated how the algorithms behaved when the number of samples used in the validation and testing set were varied. For example, in experiments with $n = 8000$, only 8000 samples from the validation set and 8000 samples from the shifted testing set were presented to the domain adaptation and calibration algorithms. For each model, for a given α and n, 10 trials were performed,

[1] Technically speaking, we could use the class ratios produced by BBSL and RLLS in conjunction with **Eqn. 8** to obtain adapted predictions without model retraining; however, when the predictions are not calibrated, this adaptation can sometimes *decrease* test-set accuracy. To avoid potentially misleading the reader into thinking BBSL and RLLS can harm test-set accuracy, we did not report the results of such adaptation. However, the overall performance trends were similar, in that EM with bias-corrected calibration performed the best.

Shift Estimator	**Calibration Method**	$\alpha = 0.1$			$\alpha = 1.0$			$\alpha = 10$		
		n=2000	n=4000	n=8000	n=2000	n=4000	n=8000	n=2000	n=4000	n=8000
EM	None	0.01049; 4.0	0.00763; 4.0	0.00795; 4.0	0.00283; 4.0	0.00202; 4.0	0.00147; 4.0	0.00191; 2.0	0.001; 2.0	0.0006; 2.0
BBSL-hard	None	0.00443; 3.0	0.00214; 2.0	0.00125; 2.0	0.00266; 2.0	0.00135; 3.0	0.00074; 2.0	0.00244; 3.0	0.00123; 3.0	0.00055; 2.5
BBSL-soft	None	**0.00309; 1.0**	**0.00133; 1.0**	**0.00092; 1.0**	**0.00188; 1.0**	**0.00108; 1.0**	**0.00057; 1.0**	0.00187; 1.0	**0.0009; 1.0**	**0.00047; 1.0**
RLLS-hard	None	0.00405; 2.0	0.00209; 2.0	0.00124; 2.0	0.00265; 2.5	0.00135; 2.0	0.00074; 3.0	0.00244; 3.0	0.00123; 3.0	0.00055; 3.0
RLLS-soft	None	**0.00316; 1.0**	**0.00127; 1.0**	**0.0009; 1.0**	**0.00183; 1.0**	0.00108; 1.0	0.00057; 1.0	**0.00187; 1.0**	**0.0009; 1.0**	**0.00047; 1.0**
EM	TS	0.00944; 2.0	0.00705; 2.0	0.00693; 2.0	0.00262; 2.0	0.00192; 2.0	0.00148; 2.0	0.00195; 2.0	0.00101; 2.0	0.00058; 2.0
BBSL-soft	TS	**0.0031; 1.0**	**0.00132; 1.0**	**0.00091; 1.0**	**0.00191; 1.0**	0.00102; 1.0	**0.0006; 0.0**	0.00179; 1.0	**0.00095; 1.0**	**0.00048; 1.0**
RLLS-soft	TS	**0.00312; 1.0**	**0.00125; 1.0**	**0.0009; 1.0**	**0.00186; 1.0**	0.00102; 1.0	0.0006; 1.0	**0.00179; 1.0**	**0.00095; 1.0**	**0.00048; 1.0**
EM	NBVS	**0.0014; 0.0**	**0.00106; 0.0**	**0.00075; 0.0**	**0.00133; 0.0**	**0.00071; 0.0**	**0.00043; 0.0**	0.00161; 0.0	**0.00079; 0.0**	0.00047; 0.0
BBSL-soft	NBVS	0.0028; 1.0	0.00137; 1.0	0.00098; 1.0	0.00175; 1.0	0.00096; 1.0	0.00061; 1.0	0.00179; 1.0	0.00081; 1.0	0.00048; 1.0
RLLS-soft	NBVS	0.00275; 1.0	0.00131; 1.0	**0.00094; 1.0**	0.00168; 1.0	0.00096; 1.0	0.00061; 1.0	0.00179; 1.0	0.00081; 1.0	0.00048; 1.0
EM	BCTS	**0.00037; 0.0**	**0.00034; 0.0**	**0.00022; 0.0**	**0.00126; 0.0**	**0.00071; 0.0**	**0.00043; 0.0**	**0.00161; 0.0**	**0.00077; 0.0**	**0.00047; 0.0**
BBSL-soft	BCTS	0.00288; 2.0	0.00126; 1.0	0.00104; 1.0	0.00171; 1.0	0.00102; 1.0	0.0006; 1.0	0.00176; 1.0	0.00083; 1.0	0.00048; 1.0
RLLS-soft	BCTS	0.00284; 1.0	0.00124; 1.0	0.00093; 1.0	0.00169; 1.0	0.00102; 1.0	0.0006; 1.0	0.00176; 1.0	0.00083; 1.0	0.00048; 1.0
EM	VS	**0.00061; 0.0**	**0.00031; 0.0**	**0.0002; 0.0**	**0.00118; 0.0**	**0.00067; 0.0**	**0.0004; 0.0**	**0.00167; 0.0**	**0.00076; 0.0**	0.00045; 0.0
BBSL-soft	VS	0.00306; 2.0	0.00134; 1.0	0.00103; 1.0	0.00173; 1.0	0.00102; 1.0	0.00061; 1.0	0.00172; 1.0	0.00082; 1.0	0.00047; 1.0
RLLS-soft	VS	0.00298; 1.0	0.00133; 1.0	0.00091; 1.0	0.00171; 1.0	0.00102; 1.5	0.00061; 1.0	**0.00172; 1.0**	0.00082; 1.0	0.00047; 1.0

Table 1. **CIFAR10: Comparison of EM, BBSL and RLLS (dirichlet shift).** Value before the semicolon is the median MSE in the estimated shift weights (as defined in **Sec. 3.5**). Value after the semicolon is the median rank of a method relative to the others in the group that use the same calibration. α represents the dirichlet shift parameter (larger α corresponds to less extreme shift), n represents the sample size for both the validation set and the label-shifted test set. A bold value in a group is not significantly different from the best-performing method in the group, as measured by a paired Wilcoxon test at $p < 0.01$. See **Table C.2** for an equivalent table but with statistical comparisons done across all calibration methods. EM tends to outperform BBSL and RLLS when calibration techniques involving class-specific bias parameters are used.

Shift Estimator	**Calibration Method**	$\rho = 0.01$			$\rho = 0.9$		
		n=2000	n=4000	n=8000	n=2000	n=4000	n=8000
EM	None	0.00197; 2.0	0.00107; 2.0	0.00067; 2.0	**0.00584; 1.0**	**0.00305; 0.0**	0.00316; 2.0
BBSL-hard	None	0.00197; 3.0	0.00122; 4.0	0.00079; 4.0	0.00897; 3.0	0.00689; 4.0	0.00608; 4.0
BBSL-soft	None	0.00161; 1.0	0.00099; 1.0	0.00057; 1.0	0.00662; 2.0	0.00379; 1.0	**0.00298; 1.0**
RLLS-hard	None	0.00197; 3.0	0.00122; 3.0	0.00079; 3.0	0.00868; 2.5	0.00676; 3.0	0.00593; 3.0
RLLS-soft	None	0.00161; 0.5	**0.00099; 1.0**	**0.00057; 0.0**	**0.00622; 1.0**	0.00379; 1.0	**0.00298; 1.0**
EM	TS	0.00162; 2.0	**0.00084; 0.0**	0.00054; 0.0	**0.00318; 0.0**	**0.00155; 0.0**	**0.00085; 0.0**
BBSL-soft	TS	0.00161; 1.0	0.00091; 1.0	0.00054; 2.0	0.00659; 2.0	0.00395; 1.0	0.00339; 1.0
RLLS-soft	TS	0.00161; 1.0	**0.00091; 1.0**	0.00054; 1.0	0.00627; 1.0	0.00393; 1.0	0.00339; 1.0
EM	NBVS	**0.00158; 0.0**	**0.00076; 0.0**	**0.00054; 0.0**	**0.00112; 0.0**	**0.00059; 0.0**	**0.00038; 0.0**
BBSL-soft	NBVS	0.00162; 1.0	0.00088; 2.0	0.00057; 2.0	0.00607; 2.0	0.00385; 1.0	0.00349; 1.0
RLLS-soft	NBVS	0.00162; 1.0	0.00088; 1.0	0.00057; 1.0	0.006; 1.0	0.0038; 1.0	0.00349; 2.0
EM	BCTS	**0.00153; 0.0**	**0.00076; 0.0**	**0.00052; 0.0**	**0.0006; 0.0**	**0.00037; 0.0**	**0.00028; 0.0**
BBSL-soft	BCTS	0.00158; 1.0	0.00087; 2.0	0.00055; 2.0	0.00623; 2.0	0.00383; 1.0	0.00338; 1.0
RLLS-soft	BCTS	**0.00158; 1.0**	0.00087; 1.0	0.00055; 1.0	0.00603; 1.0	0.00378; 2.0	0.00338; 2.0
EM	VS	**0.00155; 0.0**	**0.00076; 0.0**	**0.00052; 0.0**	**0.00067; 0.0**	**0.00042; 0.0**	**0.00033; 0.0**
BBSL-soft	VS	0.00171; 1.0	0.00086; 2.0	0.00056; 2.0	0.00641; 2.0	0.00407; 2.0	0.00393; 1.0
RLLS-soft	VS	0.00171; 1.0	0.00086; 1.0	0.00056; 1.0	0.00626; 1.0	0.00405; 1.0	0.00393; 2.0

Table 2. **MNIST: Comparison of EM, BBSL and RLLS (“tweak-one” shift)**. Value before the semicolon is the median MSE in the estimated shift weights. Value after semicolon is the median rank of a method relative to others in the group that use the same calibration. A bold value in a group is not significantly different from the best-performing method in the group, as measured by a paired Wilcoxon test at $p < 0.01$. See **Table D.2** for an equivalent table but with statistical comparisons done across all calibration methods. EM tends to outperform BBSL and RLLS when calibration techniques involving class-specific bias parameters are used.

where each trial consisted of a different sampling (without replacement) of the validation set as well as a different draw of test-set class proportions from the dirichlet distribution. This resulted in a total of 100 experiments per dataset (10 for each of the 10 different models). Statistical significance was calculated using a signed Wilcoxon test with a one-sided p-value threshold of 0.01. For MNIST and CIFAR10, we also explored “tweak-one” shift (Lipton et al., 2018), where the prior of the fourth class was set to a parameter ρ and the remaining class priors were set to $(1 - \rho)/9$. We explored

Shift Estimator	Calibration Method	$\alpha = 0.1$			$\alpha = 1.0$			$\alpha = 10.0$		
		n=7000	n=8500	n=10000	n=7000	n=8500	n=10000	n=7000	n=8500	n=10000
EM	None	1.80113; 4.0	1.67187; 4.0	1.7157; 4.0	0.55795; 4.0	0.54112; 4.0	0.55955; 4.0	0.3896; 4.0	0.37585; 4.0	0.36337; 4.0
BBSL-hard	None	0.83095; 3.0	0.68099; 3.0	0.58656; 3.0	0.33542; 3.0	0.27998; 3.0	0.24659; 3.0	0.25694; 3.0	0.21736; 3.0	0.20637; 3.0
BBSL-soft	None	0.5279; 2.0	0.47521; 2.0	0.44263; 2.0	0.23637; 2.0	0.21324; 2.0	0.18605; 2.0	0.18842; 2.0	0.16108; 2.0	0.14018; 2.0
RLLS-hard	None	0.4577; 1.0	0.38413; 1.0	0.37675; 1.0	0.20285; 1.0	0.16017; 1.0	0.14228; 1.0	0.16412; 1.0	0.13842; 1.0	0.12127; 1.0
RLLS-soft	None	**0.33882; 1.0**	**0.2728; 0.0**	**0.2878; 0.0**	**0.13859; 0.0**	**0.12697; 0.0**	**0.10567; 0.0**	**0.12528; 0.0**	**0.11011; 0.0**	**0.10056; 0.0**
EM	TS	0.41127; 1.0	0.34963; 1.0	0.30451; 1.0	0.23803; 2.0	0.21715; 2.0	0.19781; 2.0	0.16408; 2.0	0.14477; 2.0	0.13852; 2.0
BBSL-soft	TS	0.40279; 1.0	0.34713; 1.0	0.33012; 1.0	0.18774; 1.0	0.15665; 1.0	0.12639; 1.0	0.13409; 1.0	0.10951; 1.0	0.09703; 1.0
RLLS-soft	TS	0.27236; 1.0	0.22426; 1.0	0.22697; 1.0	**0.12255; 0.0**	**0.10639; 0.0**	**0.08221; 0.0**	**0.11395; 0.0**	**0.08803; 0.0**	**0.07868; 0.0**
EM	NBVS	**0.1637; 0.0**	**0.15904; 0.0**	**0.14348; 0.0**	**0.1042; 0.0**	**0.10523; 0.0**	**0.10777; 1.0**	**0.10122; 0.0**	**0.09874; 1.0**	0.09729; 2.0
BBSL-soft	NBVS	0.40856; 2.0	0.33122; 2.0	0.29594; 2.0	0.17545; 2.0	0.1409; 2.0	0.11336; 2.0	0.13601; 2.0	0.113; 1.0	0.09731; 1.0
RLLS-soft	NBVS	0.2797; 1.0	0.21465; 1.0	0.23037; 1.0	**0.11469; 1.0**	**0.09903; 1.0**	**0.08341; 1.0**	**0.11303; 1.0**	**0.08868; 1.0**	**0.07531; 0.0**
EM	BCTS	**0.1101; 0.0**	**0.10824; 0.0**	**0.11636; 0.0**	**0.08838; 0.0**	**0.09102; 0.0**	**0.0876; 1.0**	**0.09207; 0.0**	**0.08707; 1.0**	0.08781; 2.0
BBSL-soft	BCTS	0.40887; 2.0	0.32629; 2.0	0.30435; 2.0	0.17923; 2.0	0.14756; 2.0	0.11536; 2.0	0.13698; 2.0	0.11267; 2.0	0.09748; 1.0
RLLS-soft	BCTS	0.26594; 1.0	0.2305; 1.0	0.23604; 1.0	0.11498; 1.0	**0.09755; 1.0**	**0.08309; 1.0**	0.1085; 1.0	**0.08672; 1.0**	**0.07252; 0.0**
EM	VS	**0.09485; 0.0**	**0.08792; 0.0**	**0.0862; 0.0**	**0.07684; 0.0**	**0.07922; 0.0**	**0.07773; 0.0**	**0.09387; 0.0**	**0.08796; 1.0**	0.08981; 2.0
BBSL-soft	VS	0.43085; 2.0	0.3216; 2.0	0.30221; 2.0	0.16715; 2.0	0.13847; 2.0	0.11169; 2.0	0.13075; 2.0	0.10971; 1.5	0.09433; 1.0
RLLS-soft	VS	0.28941; 1.0	0.21363; 1.0	0.22405; 1.0	0.11627; 1.0	0.09894; 1.0	**0.08267; 1.0**	**0.11426; 1.0**	**0.08623; 1.0**	**0.07414; 0.5**

Table 3. **CIFAR100: Comparison of EM, BBSL and RLLS (dirichlet shift).** Value before the semicolon is the median MSE in the estimated shift weights. Value after the semicolon is the median rank of a method relative to the others in the group that use the same calibration. A bold value in a group is not significantly different from the best-performing method in the group, as measured by a paired Wilcoxon test at $p < 0.01$. See **Table E.1** for an equivalent table but with statistical comparisons done across all calibration methods. EM tends to outperform BBSL and RLLS when calibration techniques involving class-specific bias parameters are used.

Shift Estimator	Calibration Method	$\rho = 0.5$			$\rho = 0.9$		
		n=500	n=1000	n=1500	n=500	n=1000	n=1500
EM	None	**0.35073; 0.0**	**0.30498; 0.0**	**0.19709; 0.0**	0.0899; 2.0	0.0596; 2.0	0.05666; 3.0
BBSL-hard	None	1.83879; 4.0	1.4584; 4.0	0.81962; 4.0	0.31856; 4.0	0.10582; 4.0	0.0491; 3.0
BBSL-soft	None	1.4041; 2.0	0.71345; 2.0	0.43408; 2.0	0.10441; 2.5	0.04494; 2.0	0.02562; 2.0
RLLS-hard	None	1.07889; 2.0	1.08159; 3.0	0.74669; 3.0	0.05118; 1.0	**0.02865; 1.0**	0.02458; 1.0
RLLS-soft	None	0.91948; 2.0	0.63483; 1.0	0.41057; 1.0	**0.03602; 1.0**	**0.0225; 1.0**	**0.02351; 1.0**
EM	TS	**0.47048; 0.0**	**0.29683; 0.0**	**0.20944; 0.0**	0.08077; 1.0	0.05537; 2.0	0.05268; 2.0
BBSL-soft	TS	1.26587; 2.0	0.57077; 2.0	0.38372; 2.0	0.09552; 1.5	0.03859; 1.0	0.02165; 1.0
RLLS-soft	TS	0.80812; 1.0	0.56293; 1.0	0.37624; 1.0	**0.0352; 0.0**	**0.02457; 0.0**	**0.02054; 0.5**
EM	NBVS	**0.51515; 0.0**	**0.33256; 0.0**	**0.24066; 0.0**	0.09137; 1.5	0.08415; 2.0	0.08002; 2.0
BBSL-soft	NBVS	1.62832; 2.0	0.70875; 2.0	0.45962; 2.0	0.09014; 1.0	0.03608; 1.0	**0.01769; 1.0**
RLLS-soft	NBVS	0.77171; 1.0	0.56492; 1.0	0.40272; 1.0	**0.04301; 1.0**	**0.03258; 1.0**	**0.02541; 0.5**
EM	BCTS	**0.40245; 0.0**	**0.2475; 0.0**	**0.19461; 0.0**	**0.02267; 0.0**	**0.01354; 0.0**	**0.01043; 0.0**
BBSL-soft	BCTS	1.40016; 2.0	0.52628; 2.0	0.46372; 2.0	0.05524; 1.0	0.02529; 1.0	0.01419; 1.0
RLLS-soft	BCTS	0.87398; 1.0	0.50668; 1.0	0.42255; 1.0	**0.0384; 1.0**	0.02826; 1.0	0.02245; 1.0
EM	VS	**0.53081; 0.0**	**0.26442; 0.0**	**0.21641; 0.0**	**0.02388; 0.0**	**0.01307; 0.0**	**0.00953; 0.0**
BBSL-soft	VS	1.36527; 2.0	0.50393; 2.0	0.48291; 1.5	0.04678; 1.0	0.02485; 1.0	0.01543; 1.0
RLLS-soft	VS	0.82992; 1.0	0.47802; 1.0	0.42028; 1.0	**0.04096; 1.0**	0.02948; 1.0	0.022; 1.0

Table 4. **Kaggle Diabetic Retinopathy: Comparison of EM, BBSL and RLLS.** ρ represents proportion of healthy examples in shifted domain; source domain has $\rho = 0.73$. Value before semicolon is the median MSE in the estimated shift weights. Value after the semicolon is the median rank of a method relative to others in the group that use the same calibration. A bold value in a group is not significantly different from the best-performing method in the group (paired Wilcoxon test at $p < 0.01$). See **Table F.1** for an equivalent table but with statistical comparisons done across all calibration methods. EM tends to outperform BBSL and RLLS when calibration techniques involving class-specific bias parameters are used.

$\rho = 0.01$ and $\rho = 0.9$.

The Kaggle Diabetic Retinopathy dataset (Kaggle, 2015) is a collection of retinal fundus images and an associated "grade" from 0-4, where 0 indicates healthy and 1-4 indicate progressively more severe stages of retinopathy. For our experiments, we used the publicly-available pretrained model from De Fauw (2015), but modified it so as to make predictions on only one eye at a time (specifically, we supplied the mirror image of a given eye as the input for the second eye). Because test-set labels are unavailable, we separated the validation set used during the training of the model (consisting of 3514 examples) into "pseudo-validation" and

Shift Estimator	Calibration Method	$\alpha = 0.1$			$\alpha = 1.0$			$\alpha = 10$		
		n=2000	n=4000	n=8000	n=2000	n=4000	n=8000	n=2000	n=4000	n=8000
EM	None	5.275; 4.0	5.225; 4.0	5.319; 4.0	1.75; 4.0	1.65; 4.0	1.669; 4.0	0.275; 3.0	0.175; 3.0	0.25; 4.0
EM	TS	5.6; 2.0	5.513; 3.0	5.488; 3.0	1.725; 3.0	1.713; 3.0	1.775; 3.0	0.25; 4.0	0.188; 3.0	0.244; 3.0
EM	NBVS	**5.85; 2.0**	**5.725; 2.0**	5.806; 2.0	2.125; 2.0	2.013; 2.0	2.15; 2.0	0.7; 1.0	0.775; 1.0	0.788; 2.0
EM	BCTS	**5.825; 2.0**	**5.775; 1.0**	5.75; 1.0	**2.15; 1.0**	**2.075; 1.0**	2.081; 1.0	**0.725; 1.0**	**0.8; 1.0**	**0.838; 1.0**
EM	VS	5.625; 2.0	**5.675; 1.0**	5.775; 1.0	**2.15; 1.0**	**2.1; 1.0**	**2.187; 1.0**	**0.75; 1.0**	**0.812; 1.0**	**0.862; 1.0**

Table 5. **CIFAR10: Comparison of calibration methods when using EM adaptation to dirichlet shift, with $\Delta\%$accuracy as the metric**. Unlike BBSL and RLLS, the EM algorithm does not rely on retraining to produce domain adapted probabilities. Value before the semicolon is the median change in %accuracy relative to a baseline of no adaptation. Value after the semicolon is the median rank compared to other methods in the same column. Bold values in a column are not significantly different from the best performing method in the column, as measured by a paired Wilcoxon test at $p \leq 0.01$. Calibration techniques involving class-specific bias parameters (namely BCTS and VS) tend to achieve the best performance.

Shift Estimator	Calibration Method	$\alpha = 0.1$			$\alpha = 1.0$			$\alpha = 10.0$		
		n=7000	n=8500	n=10000	n=7000	n=8500	n=10000	n=7000	n=8500	n=10000
EM	None	14.136; 4.0	14.153; 4.0	14.22; 4.0	11.971; 4.0	12.112; 4.0	12.155; 4.0	11.779; 4.0	11.682; 4.0	11.89; 4.0
EM	TS	24.479; 2.0	24.371; 2.0	24.855; 2.0	21.157; 2.0	21.271; 2.0	20.99; 2.0	20.7; 2.5	20.894; 3.0	20.915; 2.5
EM	NBVS	24.079; 2.0	24.053; 2.0	24.575; 2.0	20.986; 2.0	21.382; 2.0	21.53; 2.0	21.036; 2.0	20.947; 2.0	21.025; 2.0
EM	BCTS	24.271; 2.0	24.112; 1.0	24.3; 1.0	**21.321; 1.0**	**21.506; 1.0**	21.71; 1.0	**21.143; 1.0**	**21.329; 1.0**	**21.125; 1.0**
EM	VS	**24.829; 1.0**	**24.518; 1.0**	24.51; 1.0	**21.3; 1.0**	**21.629; 1.0**	**21.93; 1.0**	**21.221; 1.0**	**21.282; 1.0**	**21.21; 1.0**

Table 6. **CIFAR100: Comparison of calibration methods when using EM adaptation to dirichlet shift, with $\Delta\%$accuracy as the metric**. Unlike BBSL and RLLS, the EM algorithm does not rely on retraining to produce domain adapted probabilities. Value before the semicolon is the median change in %accuracy relative to a baseline of no adaptation. Value after the semicolon is the median rank compared to other methods in the same column. Bold values in a column are not significantly different from the best performing method in the column, as measured by a paired Wilcoxon test at $p \leq 0.01$. Calibration techniques involving class-specific bias parameters (namely BCTS and VS) tend to achieve the best performance.

Shift Estimator	Calibration Method	$\rho = 0.5$			$\rho = 0.9$		
		n=500	n=1000	n=1500	n=500	n=1000	n=1500
EM	None	2.0; 3.0	2.0; 4.0	2.2; 4.0	1.4; 4.0	1.5; 4.0	1.567; 4.0
EM	TS	2.2; 3.0	2.3; 3.0	2.667; 3.0	1.6; 3.0	2.0; 3.0	2.133; 3.0
EM	NBVS	3.5; 2.0	4.1; 2.0	4.233; 2.0	2.2; 2.0	2.4; 2.0	2.467; 2.0
EM	BCTS	**3.8; 1.0**	**4.65; 1.0**	**4.633; 1.0**	**3.6; 0.0**	**3.6; 0.0**	**3.733; 0.0**
EM	VS	**3.8; 1.0**	**4.4; 1.0**	**4.633; 1.0**	**3.5; 1.0**	**3.6; 1.0**	**3.733; 1.0**

Table 7. **Kaggle Diabetic Retinopathy: Comparison of calibration methods when using EM adaptation to domain shift, with $\Delta\%$accuracy as the metric.** ρ represents proportion of healthy examples in shifted domain; source distribution has $\rho = 0.73$. Unlike BBSL and RLLS, the EM algorithm does not rely on retraining to produce domain adapted probabilities. Value before the semicolon is the median change in %accuracy relative to a baseline of no adaptation. Value after the semicolon is the median rank compared to other methods in the same column. Bold values in a column are not significantly different from the best performing method in the column, as measured by a paired Wilcoxon test at $p \leq 0.01$. Calibration techniques involving class-specific bias parameters (namely BCTS and VS) tend to achieve the best performance.

"pseudo-test" sets. Specifically, for each of 100 trials, we sampled n examples from the original validation set without replacement to form a pseudo-validation set, and kept the remaining examples as the pseudo-test set. Calibration was performed on the pseudo-validation set, and calibration metrics of NLL and ECE were reported on the pseudo-test set. Domain shift was then simulated by sampling from the pseudo-test set in such a way that the proportion of "healthy" labels was set to a fraction ρ, and the relative proportions of retinopathy grades was kept the same as in the source distribution. In the source distribution, $\rho = 0.73$; for the simulated domain shift, we explored $\rho = 0.5$ and $\rho = 0.9$.

4.2. Max Likelihood With Appropriate Calibration Performs Well At Estimating Shift Weights

We compared the performance of maximum likelihood (EM), BBSL and RLLS in the presence of different types of calibration, using MSE of the shift weights as the metric (**Sec. 3.5**). Results are in **Tables 1, 2, 3, 4, C.4 & D.3**. Across all datasets, we observed the following general trends: first, in the absence of calibration, BBSL and RLLS tend to outperform EM, with RLLS tending to perform the best (consistent with the results in Azizzadenesheli et al. (2019)). However, as calibration improves, so does the performance of EM. In particular, the best overall performance is achieved when using the variants of temperature scaling that contain class-specific bias parameters - namely BCTS and VS - in combination with EM.

We also computed the improvement in accuracy achieved by EM with different calibration methods compared to an unadapted baseline (**Tables 5, 6, 7, C.1**). Across datasets, we observed that either BCTS or VS tended to achieve the best accuracy. To reconcile this with the observation in Guo et al. (2017) that VS did not give the best ECE compared to TS, we calculated the Negative Log Likelihood (NLL) of different calibration methods on an unshifted test set and found that BCTS and VS tended to achieve the best NLL, even when they did not yield the best ECE (Sec. B), indicating that the ECE and NLL metrics do not always agree with each other. Empirically, we found that the NLL corresponds better with the improvement that a calibration method will give to domain adaptation (**Sec. G**). This is consistent with other reports stating that ECE computed using only information about the most confidently predicted class, as was done in Guo et al. (2017), is perhaps not the best metric (Vaicenavicius et al., 2019).

5. Discussion

In this work, we explored the effect of calibration on procedures designed to perform domain adaptation to label shift. In experiments on CIFAR10, MNIST, CIFAR100 and diabetic retinopathy detection, we found that maximum likelihood methods such as EM with specific types of calibration tends to outperform moment matching methods such as BBSL and RLLS. In particular, we found that the best results were achieved when the calibration method involves class-specific bias parameters that can reduce systematic bias in the class probabilities. This capacity for bias correction is absent from the popular Temperature Scaling approach recommended by Guo et al. (2017). We reconcile this by noting that Guo et al. evaluated calibration using ECE computed on only the most confidently predicted classes, which is known to be misleading (Vaicenavicius et al., 2019), and by observing that Vector Scaling (which does include class-specific bias parameters) performed almost as well as Temperature Scaling in their evaluation.

In addition, we observe that when the calibrated probabilities retain systematic bias, domain adaptation via maximum likelihood is sensitive to the strategy used to compute the source-domain priors. If the source-domain priors $\hat{p}(y = i)$ are not defined in a way that mirrors the systematic bias in the predicted probabilities $\hat{p}(y = i|x)$, then maximum likelihood will estimate a label shift even if the target domain is identical to the source domain (**Lemma A**) and can produce highly detrimental results (**Tables A.1**). By contrast, if the source domain priors for maximum likelihood are specified as recommend in **Sec. 3.2**, maximum likelihood becomes substantially more tolerant of systematic bias in the calibrated probabilities, although it does not tend to outperform BBSL or RLLS in the presence of poor calibration. We conjecture that maximum likelihood is sensitive to systematic bias because the estimate of $\hat{q}(y = i|\boldsymbol{x}_k)$ (e.g. as computed in the E-step of EM) relies heavily on the ratio $\frac{\hat{q}(y=i)}{\hat{p}(y=i)}$. Systematic bias is defined as error in $\hat{p}(y = i)$, which, as it appears in the denominator, could manifest as large errors in $\frac{\hat{q}(y=i)}{\hat{p}(y=i)}$ - particularly when $\hat{p}(y = i)$ is small.

One concern when using EM to find the maximum likelihood estimate is the possibility of getting trapped in local minima. To address this concern, we analyzed the likelihood function and determined that it is concave and bounded (**Sec. 3.3**). Thus, EM (and any standard convex optimization algorithm) converges to the global maximum of the likelihood.

Following the appearance of this work online, an independent study by Garg et al. (2020) verified that our proposed approach of coupling bias-corrected temperature scaling (BCTS) with maximum-likelihood estimation "uniformly dominates" compared to BBSL and RLLS. The same study also developed a theoretical analysis of the maximum likelihood approach that confirms the importance of calibration in achieving good performance.

In conclusion, we presented an algorithm that is simple, computationally efficient, and avoids both hyperparameter tuning and the pitfalls associated with retraining deep learning models with importance weighting (Byrd and Lipton, 2019). When tested empirically on a variety of datasets and data shifts, it produces better or comparable results compared to the current state-of-the-art. The algorithm can also be applied in a semi-supervised context (Appendix H). We posit that maximum likelihood with bias-corrected calibration will prove particularly useful in big data settings where deep learning models are more likely to be deployed.

6. Useful Links

code: `https://github.com/kundajelab/labelshiftexperiments`

video: `https://youtu.be/ZBXjE9QTruE`

blogpost: `https://bit.ly/3kTds7J`

7. Author Contributions

AS conceived of the method and implemented the calibration & label shift adaptation algorithms. AS and AA designed & conducted experiments, and wrote the manuscript, with guidance and feedback from AK.

References

Kamyar Azizzadenesheli, Anqi Liu, Fanny Yang, and Animashree Anandkumar. Regularized learning for domain adaptation under label shifts. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJl0r3R9KX.

Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 872–881, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/byrd19a.html.

Yee Seng Chan and Hwee Tou Ng. Word sense disambiguation with distribution estimation. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, pages 1010–1015, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc.

Jeffrey De Fauw. Jeffreydf/kaggle_diabetic_retinopathy: Fifth place solution of the kaggle diabetic retinopathy competition. https://github.com/JeffreyDF/kaggle_diabetic_retinopathy, Oct 2015. (Accessed on 01/22/2019).

Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2): 12–22, 1983.

Saurabh Garg, Yifan Wu, Sivaraman Balakrishnan, and Zachary C Lipton. A unified view of label shift estimation. *arXiv preprint arXiv:2003.07554*, 2020.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4878–4887. Curran Associates, Inc., 2017.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/guo17a.html.

Kaggle. Kaggle competition on diabetic retinopathy detection. https://www.kaggle.com/c/diabetic-retinopathy-detection#description, 2015.

Volodymyr Kuleshov and Stefano Ermon. Reliable confidence estimation via online learning. *arXiv preprint arXiv:1607.03594*, pages 2586–2594, 2016.

Volodymyr Kuleshov and Percy S Liang. Calibrated structured prediction. In *Advances in Neural Information Processing Systems*, pages 3474–3482, 2015.

Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3122–3130, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/lipton18a.html.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632. ACM, 2005.

John C Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, 1999.

Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural Comput.*, 14(1): 21–41, January 2002.

Bernhard Schoelkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On causal and anticausal learning. *Proceedings of the 29th International Conference on Machine Learning*, June 2012.

Amos Storkey. When training and test sets are different: Characterizing learning transfer. In *Dataset Shift in Machine Learning*. The MIT Press, 2008.

Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas Schön. Evaluating model calibration in classification. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 3459–3467. PMLR, 16–18 Apr 2019. URL http://proceedings.mlr.press/v89/vaicenavicius19a.html.

Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Proceedings of the Eighteenth Interacational Conference on Machine Learning*, volume 1, pages 609–616, 2001.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699. ACM, 2002.

Kun Zhang, Bernhard Schlkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 819–827, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL http://proceedings.mlr.press/v28/zhang13d.html.