Shift Happens: Adjusting Classifiers

Theodore James Thibault Heiser[0000−0001−7057−3160], Mari-Liis Allikivi[0000−0002−1019−3454], and Meelis Kull (✉)[0000−0001−9257−595X]

Institute of Computer Science, University of Tartu, Tartu, Estonia
{mari-liis.allikivi, meelis.kull}@ut.ee*

Abstract. Minimizing expected loss measured by a proper scoring rule, such as Brier score or log-loss (cross-entropy), is a common objective while training a probabilistic classifier. If the data have experienced dataset shift where the class distributions change post-training, then often the model's performance will decrease, over-estimating the probabilities of some classes while under-estimating the others on average. We propose unbounded and bounded general adjustment (UGA and BGA) methods that transform all predictions to (re-)equalize the average prediction and the class distribution. These methods act differently depending on which proper scoring rule is to be minimized, and we have a theoretical guarantee of reducing loss on test data, if the exact class distribution is known. We also demonstrate experimentally that, when in practice the class distribution is known only approximately, there is often still a reduction in loss depending on the amount of shift and the precision to which the class distribution is known.

Keywords: Multi-class classification · Proper scoring rule · Dataset shift · Classifier calibration · Classifier adjustment

1 Introduction

Classical supervised machine learning is built on the assumption that the joint probability distribution that features and labels are sourced from does not change during the life cycle of the predictive model: from training to testing and deployment. However, in reality this assumption is broken more often than not: medical diagnostic classifiers are often trained with an oversampling of disease-positive instances, surveyors are often biased to collecting labelled samples from certain segments of a population, user demographics and preferences change over time on social media and e-commerce sites, etc.

While these are all examples of dataset shift, the nature of these shifts can be quite different. There have been several efforts to create taxonomies of dataset shift [14, 11]. The field of *transfer learning* offers many methods of learning models for scenarios with awareness of the shift during training. However, often the shift is not yet known during training and it is either too expensive or even impossible to retrain once the shift happens. There are several reasons for it:

* T. Heiser can be reached at teddyheiser@google.com.

original training data or training infrastructure might not be available; shift happens so frequently that there is no time to retrain; the kind of shift is such that without having labels in the shifted context there is no hope of learning a better model than the original.

In this work we address multi-class classification scenarios where training a classifier for the shifted deployment context is not possible (due to any of the above reasons), and the only possibility is to post-process the outputs from an existing classifier that was trained before the shift happened. To succeed, such post-processing must be guided by some information about the shifted deployment context. In the following, we will assume that we know the overall expected class distribution in the shifted context, at least approximately. For example, consider a medical diagnostic classifier of disease sub-types, which has been trained on the cases of country A, and gets deployed to a different country B. It is common that the distribution of sub-types can vary between countries, but in many cases such information is available. So here many labels are available but not the feature values (country B has data about sub-types in past cases, but no diagnostic markers were measured back then), making training of a new model impossible. Still, the model *adjustment* methods proposed in this paper can be used to adjust the existing model to match the class distribution in the deployment context. As another example, consider a bank's fraud detection classifier trained on one type of credit cards and deployed to a new type of credit cards. For new cards there might not yet be enough cases of fraud to train a new classifier, but there might be enough data to estimate the class distribution, that is the prevalence of fraud. The old classifier might predict too few or too many positives on the new data, so it must be adjusted to the new class distribution.

In many application domains, including the above examples of medical diagnostics and fraud detection, it is required that the classifiers would output confidence information in addition to the predicted class. This is supported by most classifiers, as they can be requested to provide for each instance the class probabilities instead of a single label. For example, the feed-forward neural networks for classification typically produce class probabilities using the final soft-max layer. Such confidence information can then be interpreted by a human expert to choose the action based on the prediction, or feeded into an automatic cost-sensitive decision-making system, which would use the class probability estimates and the mis-classification cost information to make cost-optimal decisions. Probabilistic classifiers are typically evaluated using *Brier score* or *log-loss* (also known as *squared error* and *cross-entropy*, respectively). Both measures belong to the family of proper scoring rules: measures which are minimized by the true posterior class probabilities produced by the Bayes-optimal model. Proper losses also encourage the model to produce calibrated probabilities, as every proper loss decomposes into *calibration loss* and *refinement loss* [9].

Our goal is to improve the predictions of a given model in a shifted deployment context, using the information about the expected class distribution in this context, without making any additional assumptions about the type of dataset shift. The idea proposed by Kull et al. [9] is to take advantage of a prop-

erty that many dataset shift cases share: a difference in the classifier's average prediction and the expected class distribution of the data. They proposed two different *adjustment procedures* which transform the predictions to re-equalise the average prediction with the expected class distribution, resulting in a theoretically guaranteed reduction of Brier score or log-loss. Interestingly, it turned out that different loss measures require different adjustment procedures. They proved that their proposed *additive adjustment* (additively shifting all predictions, see Section 2 for the definitions) is guaranteed to reduce Brier score, while it can increase log-loss in some circumstances. They also proposed *multiplicative adjustment* (multiplicatively shifting and renormalising all predictions) which is guaranteed to reduce log-loss, while it can sometimes increase Brier score. It was proved that if the adjustment procedure is *coherent* with the proper loss (see Section 2), then the reduction of loss is guaranteed, assuming that the class distribution is known exactly. They introduced the term *adjustment loss* to refer to the part of calibration loss which can be eliminated by adjustment. Hence, adjustment can be viewed as a weak form of calibration. In the end, it remained open: (1) whether for every proper scoring rule there exists an adjustment procedure that is guaranteed to reduce loss; (2) is there a general way of finding an adjustment procedure to reduce a given proper loss; (3) whether this reduction of loss from adjustment materializes in practice where the new class distribution is only known approximately; (4) how to solve algorithm convergence issues of the multiplicative adjustment method; (5) how to solve the problem of additive adjustment sometimes producing predictions with negative 'probabilities'.

The contributions of our work are the following: (1) we construct a family called BGA (Bounded General Adjustment) of adjustment procedures, with one procedure for each proper loss, and prove that each BGA procedure is guaranteed to reduce the respective proper loss, if the class distribution of the dataset is known; (2) we show that each BGA procedure can be represented as a convex optimization task, leading to a practical and tractable algorithm; (3) we demonstrate experimentally that even if the new class distribution is only known approximately, the proposed BGA methods still usually improve over the unadjusted model; (4) we prove that the BGA procedure of log-loss is the same as multiplicative adjustment, thus solving the convergence problems of multiplicative adjustment; (5) we construct another family called UGA (Unbounded General Adjustment) with adjustment procedures that are dominated by the respective BGA methods according to the loss, but are theoretically interesting by being coherent to the respective proper loss in the sense of Kull et al. [9], and by containing the additive adjustment procedure as the UGA for Brier score.

Section 2 of this paper provides the background for this work, covering the specific types of dataset shift and reviewing some popular methods of adapting to them. We also review the family of proper losses, i.e. the loss functions that adjustment is designed for. Section 3 introduces the UGA and BGA families of adjustment procedures and provides the theoretical results of the paper. Section 4 provides experimental evidence for the effectiveness of BGA adjustment in

practical settings. Section 5 concludes the paper, reviewing its contributions and proposing open questions.

2 Background and Related Work

2.1 Dataset Shift and Prior Probability Adjustment

In supervised learning, dataset shift can be defined as any change in the joint probability distribution of the feature vector X and label Y between two data generating processes, that is $\mathbb{P}_{old}(X, Y) \neq \mathbb{P}_{new}(X, Y)$, where $\mathbb{P}_{old}$ and $\mathbb{P}_{new}$ are the probability distributions before and after the shift, respectively. While the proposed adjustment methods are in principle applicable for any kind of dataset shift, there are differences in performance across different types of shift. According to Moreno-Torres et al [11] there are 4 main kinds of shift: covariate shift, prior probability shift, concept shift and other types of shift. *Covariate shift* is when the distribution $\mathbb{P}(X)$ of the covariates/features changes, but the posterior class probabilities $P(Y|X)$ do not. At first, this may not seem to be of much interest since the classifiers output estimates of posterior class probabilities and these remain unshifted. However, unless the classifier is Bayes-optimal, then covariate shift can still result in a classifier under-performing [14]. Many cases of covariate shift can be modelled as sample selection bias [8], often addressed by retraining the model on a reweighted training set [15, 13, 7].

Prior probability shift is when the prior class probabilities $\mathbb{P}(Y)$ change, but the likelihoods $\mathbb{P}(X|Y)$ do not. An example of this is down- or up-sampling of the instances based on their class in the training or testing phase. Given the new class distribution, the posterior class probability predictions can be modified according to Bayes' theorem to take into account the new prior class probabilities, as shown in [12]. We will refer to this procedure as the *Prior Probability Adjuster* (PPA) and the formal definition is as follows:

$$\text{PPA:} \qquad \mathbb{P}_{new}(Y=y|X) = \frac{\mathbb{P}_{old}(Y=y|X)\mathbb{P}_{new}(Y=y)/\mathbb{P}_{old}(Y=y)}{\sum_{y'} \mathbb{P}_{old}(Y=y'|X)\mathbb{P}_{new}(Y=y')/\mathbb{P}_{old}(Y=y')}$$

In *other types of shift* both conditional probability distributions $\mathbb{P}(X|Y)$ and $\mathbb{P}(Y|X)$ change. The special case where $\mathbb{P}(Y)$ or $\mathbb{P}(X)$ remains unchanged is called *concept shift*. Concept shift and other types of shift are in general hard to adapt to, as the relationship between X and Y has changed in an unknown way.

2.2 Proper Scoring Rules and Bregman Divergences

The best possible probabilistic classifier is the Bayes-optimal classifier which for any instance X outputs its true posterior class probabilities $\mathbb{P}(Y|X)$. When choosing a loss function for evaluating probabilistic classifiers, it is then natural to require that the loss would be minimized when the predictions match the correct posterior probabilities. Loss functions with this property are called proper scoring rules [5, 10, 9]. Note that throughout the paper we consider multi-class

classification with k classes and represent class labels as one-hot vectors, i.e. the label of class i is a vector of $k-1$ zeros and a single 1 at position i.

Definition 1 (Proper Scoring Rule (or Proper Loss)). *In a k-class classification task a loss function $f : [0,1]^k \times \{0,1\}^k \to \mathbb{R}$ is called a* proper scoring rule *(or* proper loss*), if for any probability vectors $p, q \in [0,1]^k$ with $\sum_{i=1} p_i = 1$ and $\sum_{i=1} q_i = 1$ the following inequality holds:*

$$\mathbb{E}_{Y\sim q}[f(q,Y)] \leq \mathbb{E}_{Y\sim q}[f(p,Y)]$$

where Y is a one-hot encoded label randomly drawn from the categorical distribution over k classes with class probabilities represented by vector q. The loss function f is called strictly proper *if the inequality is strict for all $p \neq q$.*

This is a useful definition, but it does not give a very clear idea of what the geometry of these functions looks like. Bregman divergences [4] were developed independently of proper scoring rules and have a constructive definition (note that many authors have the arguments p and q the other way around, but we use this order to match proper losses).

Definition 2 (Bregman Divergence). *Let $\phi : \Omega \to \mathbb{R}$ be a strictly convex function defined on a convex set $\Omega \subseteq \mathbb{R}^k$ such that ϕ is differentiable on the relative interior of Ω, $ri(\Omega)$. Denoting the dot product by $\langle \cdot, \cdot \rangle$, the Bregman divergence $d_\phi : ri(\Omega) \times \Omega \to [0, \infty)$ is defined as*

$$d_\phi(p,q) = \phi(q) - \phi(p) - \langle q - p, \nabla\phi(p) \rangle$$

Previous works [1] have shown that the two concepts are closely related. Every Bregman divergence is a strictly proper scoring rule and every strictly proper scoring rule (within an additive constant) is a Bregman divergence. Best known functions in these families are *squared Euclidean distance* defined as $d_{SED}(\mathbf{p}, \mathbf{q}) = \sum_{j=1}^{d}(p_j - q_j)^2$ and *Kullback-Leibler-divergence* $d_{KL}(\mathbf{p}, \mathbf{q}) = \sum_{j=1}^{d} q_j \log \frac{q_j}{p_j}$. When used as a scoring rule to measure loss of a prediction against labels, they are typically referred to as *Brier Score* d_{BS}, and *log-loss* d_{LL}, respectively.

2.3 Adjusted Predictions and Adjustment Procedures

Let us now come to the main scenario of this work, where dataset shift of unknown type occurs after a probabilistic k-class classifier has been trained. Suppose that we have a test dataset with n instances from the post-shift distribution. We denote the predictions of the existing probabilistic classifier on these data by $p \in [0,1]^{n\times k}$, where p_{ij} is the predicted class j probability on the i-th instance, and hence $\sum_{j=1}^{k} p_{ij} = 1$. We further denote the hidden actual labels in the one-hot encoded form by $y \in \{0,1\}^{n\times k}$, where $y_{ij} = 1$ if the i-th instance belongs to class j, and otherwise $y_{ij} = 0$. While the actual labels are hidden, we assume that the overall class distribution $\pi \in [0,1]^k$ is known, where $\pi_j = \frac{1}{n}\sum_{i=1}^{n} y_{ij}$.

The following theoretical results require π to be known exactly, but in the experiments we demonstrate benefits from the proposed adjustment methods also in the case where π is known approximately. As discussed in the introduction, examples of such scenarios include medical diagnostics and fraud detection. Before introducing the adjustment procedures we define what we mean by *adjusted predictions*.

Definition 3 (Adjusted Predictions). *Let $p \in [0,1]^{n \times k}$ be the predictions of a probabilistic k-class classifier on n instances and let $\pi \in [0,1]^k$ be the actual class distribution on these instances. We say that* predictions p are adjusted on this dataset*, if the average prediction is equal to the class proportion for every class j, that is $\frac{1}{n}\sum_{i=1}^{n} p_{ij} = \pi_j$.*

Essentially, the model provides adjusted predictions on a dataset, if for each class its predicted probabilities on the given data are on average neither under- nor over-estimated. Note that this definition was presented in [9] using random variables and expected values, and our definition can be viewed as a finite case where a random instance is drawn from the given dataset.

Consider now the case where the predictions are not adjusted on the given test dataset, and so the estimated class probabilities are on average too large for some class(es) and too small for some other class(es). This raises a question of whether the overall loss (as measured with some proper loss) could be reduced by shifting all predictions by a bit, for example with additive shifting by adding the same constant vector ε to each prediction vector $p_{i\cdot}$. The answer is not obvious as in this process some predictions would also be moved further away from their true class. This is in some sense analogous to the case where a regression model is on average over- or under-estimating its target, as there also for some instances the predictions would become worse after shifting. However, additive shifting still pays off, if the regression results are evaluated by mean squared error. This is well known from the theory of linear models where mean squared error fitting leads to an intercept value such that the average predicted target value on the training set is equal to the actual mean target value (unless regularisation is applied). Since Brier score is essentially the same as mean squared error, it is natural to expect reduction of Brier score after additive shifting of predictions towards the actual class distribution. This is indeed so, and [9] proved that *additive adjustment* guarantees a reduction of Brier score. Additive adjustment is a method which adds the same constant vector to all prediction vectors to achieve equality between average prediction vector and actual class distribution.

Definition 4 (Additive Adjustment). Additive adjustment *is the function $\alpha_+ : [0,1]^{n \times k} \times [0,1]^k \to [0,1]^{n \times k}$ which takes in the predictions of a probabilistic k-class classifier on n instances and the actual class distribution π on these instances, and outputs adjusted predictions $a = \alpha_+(p, \pi)$ defined as $a_{i\cdot} = p_{i\cdot} + (\varepsilon_1, \ldots, \varepsilon_k)$ where $a_{i\cdot} = (a_{i1}, \ldots, a_{ik})$, $p_{i\cdot} = (p_{i1}, \ldots, p_{ik})$, and $\varepsilon_j = \pi_j - \frac{1}{n}\sum_{i=1}^{n} p_{ij}$ for each class $j \in \{1, \ldots, k\}$.*

It is easy to see that additive adjustment procedure indeed results in adjusted predictions, as $\frac{1}{n}\sum_{i=1}^{n} a_{ij} = \frac{1}{n}\sum_{i=1}^{n} p_{ij} + \varepsilon_j = \pi_j$. Note that even if the

original predictions p are probabilities between 0 and 1, the additively adjusted predictions a can sometimes go out from that range and be negative or larger than 1. For example, if an instance i is predicted to have probability $p_{ij} = 0$ to be in class j and at the same time on average the overall proportion of class j is over-estimated, then $\varepsilon_j < 0$ and the adjusted prediction $a_{ij} = \varepsilon_j$ is negative. While such predictions are no longer probabilities in the standard sense, these can still be evaluated with Brier score. So it is always true that the overall Brier score on adjusted predictions is lower than on the original predictions, $\frac{1}{n} d_{BS}(a_{i\cdot}, y_{i\cdot}) \leq \frac{1}{n} d_{BS}(p_{i\cdot}, y_{i\cdot})$, with equality only when the original predictions are already adjusted, $a = p$. Note that whenever we mention the guaranteed reduction of loss, it always means that there is no reduction in the special case where the predictions are already adjusted, since then adjustment has no effect.

Additive adjustment is just one possible transformation of unadjusted predictions into adjusted predictions, and there are infinitely many other such transformations. We will refer to these as *adjustment procedures*. If we have explicitly required the output values to be in the range $[0, 1]$ then we use the term *bounded adjustment procedure*, otherwise we use the term *unbounded adjustment procedure*, even if actually the values do not go out from that range.

Definition 5 (Adjustment Procedure). Adjustment procedure *is any function* $\alpha : [0,1]^{n \times k} \times [0,1]^k \to [0,1]^{n \times k}$ *which takes as arguments the predictions* p *of a probabilistic* k*-class classifier on* n *instances and the actual class distribution* π *on these instances, such that for any* p *and* π *the output predictions* $a = \alpha(p, \pi)$ *are adjusted, that is* $\frac{1}{n} \sum_{i=1}^{n} a_{ij} = \pi_j$ *for each class* $j \in \{1, \ldots, k\}$.

In this definition and also in the rest of the paper we assume silently, that p contains valid predictions of a probabilistic classifier, and so for each instance i the predicted class probabilities add up to 1, that is $\sum_{j=1}^{k} p_{ij} = 1$. Similarly, we assume that π contains a valid class distribution, with $\sum_{j=1}^{k} \pi_j = 1$.

Definition 6 (Bounded Adjustment Procedure). *An adjustment procedure* $\alpha : [0,1]^{n \times k} \times [0,1]^k \to [0,1]^{n \times k}$ *is* bounded*, if for any* p *and* π *the output predictions* $a = \alpha(p, \pi)$ *are in the range* $[0, 1]$*, that is* $a_{ij} \in [0, 1]$ *for all* i, j.

An example of a bounded adjustment procedure is the *multiplicative adjustment* method proposed in [9], which multiplies the prediction vector componentwise with a constant weight vector and renormalizes the result to add up to 1.

Definition 7 (Multiplicative Adjustment). Multiplicative adjustment *is the function* $\alpha_* : [0,1]^{n \times k} \times [0,1]^k \to [0,1]^{n \times k}$ *which takes in the predictions of a probabilistic* k*-class classifier on* n *instances and the actual class distribution* π *on these instances, and outputs adjusted predictions* $a = \alpha_*(p, \pi)$ *defined as* $a_{ij} = \frac{w_j p_{ij}}{z_i}$*, where* $w_1, \ldots, w_k \geq 0$ *are real-valued weights chosen based on* p *and* π *such that the predictions* $\alpha_*(p, \pi)$ *would be adjusted, and* z_i *are the renormalisation factors defined as* $z_i = \sum_{j=1}^{k} w_j p_{ij}$.

As proved in [9], the suitable class weights $w_1, \ldots, w_k$ are guaranteed to exist, but finding these weights is a non-trivial task and the algorithm based on

coordinate descent proposed in [9] can sometimes fail to converge. In the next Section 3 we will propose a more reliable algorithm for multiplicative adjustment.

It turns out that the adjustment procedure should be selected depending on which proper scoring rule is aimed to be minimised. It was proved in [9] that Brier score is guaranteed to be reduced with additive adjustment and log-loss with multiplicative adjustment. It was shown that when the 'wrong' adjustment method is used, then the loss can actually increase. In particular, additive adjustment can increase log-loss and multiplicative adjustment can increase Brier score. A sufficient condition for a guaranteed reduction of loss is *coherence* between the adjustment procedure and the proper loss corresponding to a Bregman divergence. Intuitively, coherence means that the effect of adjustment is the same across instances, where the effect is measured as the difference of divergences of this instance from any fixed class labels j and j'. The definition is the following:

Definition 8 (Coherence of Adjustment Procedure and Bregman Divergence [9]). *Let $\alpha : [0,1]^{n\times k} \times [0,1]^k \to [0,1]^{n\times k}$ be an adjustment procedure and d_ϕ be a Bregman divergence. Then α is called to be coherent with d_ϕ if and only if for any predictions p and class distribution π the following holds for all $i = 1,\dots,n$ and $j, j' = 1,\dots,k$:*

$$(d_\phi(a_{i\cdot}, c_j) - d_\phi(p_{i\cdot}, c_j)) - (d_\phi(a_{i\cdot}, c_{j'}) - d_\phi(p_{i\cdot}, c_{j'})) = const_{j,j'}$$

where $const_{j,j'}$ is a quantity not depending on i, and where $a = \alpha(p,\pi)$ and c_j is a one-hot vector corresponding to class j (with 1 at position j and 0 elsewhere).

The following result can be seen as a direct corollary of Theorem 4 in [9].

Theorem 9 (Decomposition of Bregman Divergences [9]). *Let d_ϕ be a Bregman divergence and let $\alpha : [0,1]^{n\times k} \times [0,1]^k \to [0,1]^{n\times k}$ be an adjustment procedure coherent with d_ϕ. Then for any predictions p, one-hot encoded true labels $y \in \{0,1\}^{n\times k}$ and class distribution π (with $\pi_j = \frac{1}{n}\sum_{i=1}^n y_{ij}$) the following decomposition holds:*

$$\frac{1}{n}\sum_{i=1}^{n} d_\phi(p_{i\cdot}, y_{i\cdot}) = \frac{1}{n}\sum_{i=1}^{n} d_\phi(p_{i\cdot}, a_{i\cdot}) + \frac{1}{n}\sum_{i=1}^{n} d_\phi(a_{i\cdot}, y_{i\cdot}) \tag{1}$$

Proof. The proofs and source code are in the Online Supplementary[1].

Due to non-negativity of d_ϕ this theorem gives a guaranteed reduction of loss, that is the loss on the adjusted probabilities a (average divergence between a and y) is less than the loss on the original unadjusted probabilities (average divergence between p and y), unless the probabilities are already adjusted ($p = a$). As additive adjustment can be shown to be coherent with the squared Euclidean distance and multiplicative adjustment with KL-divergence [9], the respective guarantees of loss reduction follow from Theorem 9.

[1] Proofs, code: `https://github.com/teddyheiser/Shift_Happens_ECML_PKDD_2019`

3 General Adjustment

Our main contribution is a family of adjustment procedures called BGA (Bounded General Adjustment). We use the term 'general' to emphasise that it is not a single method, but a family with exactly one adjustment procedure for each proper loss. We will prove that every adjustment procedure of this family is guaranteed to reduce the respective proper loss, assuming that the true class distribution is known exactly. To obtain more theoretical insights and answer the open questions regarding coherence of adjustment procedures with Bregman divergences and proper losses, we define a weaker variant of BGA called UGA (Unbounded General Adjustment). As the name says, these methods can sometimes output predictions that are not in the range $[0, 1]$. On the other hand, the UGA procedures turn out to be coherent with their corresponding divergence measure, and hence have the decomposition stated in Theorem 9 and also guarantee reduced loss. However, UGA procedures have less practical value, as each UGA procedure is dominated by the respective BGA in terms of reductions in loss. We start by defining the UGA procedures, as these are mathematically simpler.

3.1 Unbounded General Adjustment (UGA)

We work here with the same notations as introduced earlier, with p denoting the $n \times k$ matrix with the outputs of a k-class probabilistic classifier on a test dataset with n instances, and y denoting the matrix with the same shape containing one-hot encoded actual labels. We denote the unknown true posterior class probabilities $\mathbb{P}(Y|X)$ on these instances by q, again a matrix with the same shape as p and y.

Our goal is to reduce the loss $\frac{1}{n}\sum_{i=1}^{n} d_\phi(p_{i\cdot}, y_{i\cdot})$ knowing the overall class distribution π, while not having any other information about labels y. Due to the defining property of any proper loss, the expected value of this quantity is minimised at $p = q$. As we know neither y nor q, we consider instead the set of all possible predictions Q_π that are adjusted to π, that is $Q_\pi = \left\{ a \in \mathbb{R}^{n\times k} \;\middle|\; \frac{1}{n}\sum_{i=1}^{n} a_{i,j} = \pi_j,\ \sum_{j=1}^{k} a_{i,j} = 1 \right\}$. Note that here we do not require $a_{ij} \geq 0$, as in this subsection we are working to derive unbounded adjustment methods which allow predictions to go out from the range $[0, 1]$.

The question is now whether there exists a prediction matrix $a \in Q_\pi$ that is better than p (i.e. has lower divergence from y) regardless of what the actual labels y are (as a sidenote, y also belongs to Q_π). It is not obvious that such a exists, as one could suspect that for any a there exists some bad y such that the original p would be closer to y than the 'adjusted' a is.

Now we will define UGA and prove that it outputs adjusted predictions $a^\star$ that are indeed better than p, regardless of what the actual labels y are.

Definition 10 (Unbounded General Adjuster (UGA)). *Consider a k-class classification task with a test dataset of n instances, and let d_ϕ be a Bregman divergence. Then the* unbounded general adjuster corresponding to d_ϕ *is*

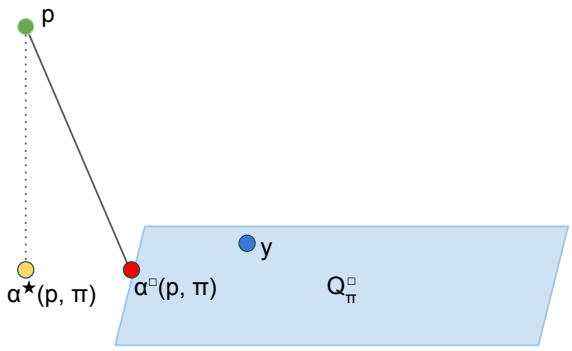

Fig. 1. A schematic explanation with $\alpha^\star(p,\pi)$ of UGA and $\alpha^\square(p,\pi)$ of BGA.

the function $\alpha^\star : \mathbb{R}^{n\times k} \times \mathbb{R}^k \to \mathbb{R}^{n\times k}$ *defined as follows:*

$$\alpha^\star(p,\pi) = \arg\min_{a\in Q_\pi} \frac{1}{n}\sum_{i=1}^{n} d_\phi(p_i., a_i.)$$

The definition of UGA is correct in the sense that the optimisation task used to define it has a unique optimum. This is because it is a convex optimisation task, as will be explained in Section 3.3. Intuitively, Q_π can be thought of as an infinite hyperplane of adjusted predictions, also containing the unknown y. The original prediction p is presumably not adjusted, so it does not belong to Q_π. UGA essentially 'projects' p to the hyperplane Q_π, in the sense of finding a in the hyperplane which is closest from p according to d_ϕ, see the diagram in Figure 1.

The following theorem guarantees that the loss is reduced after applying UGA by showing that UGA is coherent with its Bregman divergence.

Theorem 11. *Let* $\alpha^\star$ *be the unbounded general adjuster corresponding to the Bregman divergence* d_ϕ*. Then* $\alpha^\star$ *is coherent with* d_ϕ*.*

The next theorem proves that UGA is actually the one and only adjustment procedure that decomposes in the sense of Theorem 9. Therefore, UGA coincides with additive and multiplicative adjustment on Brier score and log-loss, respectively.

Theorem 12. *Let* d_ϕ *be a Bregman divergence, let* p *be a set of predictions, and* π *be a class distribution over* k *classes. Suppose* $a \in Q_\pi$ *is such that for any* $y \in Q_\pi$ *the decomposition of Eq.(1) holds. Then* $a = \alpha^\star(p,\pi)$*.*

As explained in the example of additive adjustment (which is UGA for Brier score), some adjusted predictions can get out from the range $[0, 1]$. It is clear that a prediction involving negative probabilities cannot be optimal. In the following section we propose the Bounded General Adjuster (BGA) which does not satisfy the decomposition property but is guaranteed to be at least as good as UGA.

3.2 Bounded General Adjustment

For a given class distribution π, let us constrain the set of all possible adjusted predictions Q_π further, by requiring that all probabilities are non-negative:

$$Q_\pi^\square = \{a \in Q_\pi \mid a_{i,j} \geq 0 \text{ for } i = 1, \ldots, n \text{ and } j = 1, \ldots, k\}$$

We now propose our bounded general adjuster (BGA), which outputs predictions within $Q_\pi^\square$.

Definition 13 (Bounded General Adjuster (BGA)). *Consider a k-class classification task with a test dataset of n instances, and let d_ϕ be a Bregman divergence. Then the* bounded general adjuster corresponding to d_ϕ *is the function $\alpha^\square : [0,1]^{n\times k} \times [0,1]^k \to [0,1]^{n\times k}$ defined as follows:*

$$\alpha^\square(p, \pi) = \operatorname*{arg\,min}_{a \in Q_\pi^\square} \frac{1}{n} \sum_{i=1}^{n} d_\phi(p_{i\cdot}, a_{i\cdot})$$

Similarly as for UGA, the correctness of BGA is guaranteed by the convexity of the optimisation task, as shown in Section 3.3. BGA solves almost the same optimisation task as UGA, except that instead of considering the whole hyperplane Q_π it finds the closest a within a bounded subset $Q_\pi^\square$ within the hyperplane. Multiplicative adjustment is the BGA for log-loss, because log-loss is not defined at all outside the $[0, 1]$ bounds, and hence the UGA for log-loss is the same as the BGA for log-loss. The following theorem shows that there is a guaranteed reduction of loss after BGA, and the reduction is at least as big as after UGA.

Theorem 14. *Let d_ϕ be a Bregman divergence, let p be a set of predictions, and π be a class distribution over k classes. Then for any $y \in Q_\pi^\square$ the following holds:*

$$\begin{aligned}
&\sum_{i=1}^{n} (d_\phi(p_{i\cdot}, y_{i\cdot}) - d_\phi(a_{i\cdot}^\square, y_{i\cdot})) \\
&\geq \sum_{i=1}^{n} d_\phi(p_{i\cdot}, a_{i\cdot}^\square) \geq \sum_{i=1}^{n} d_\phi(p_{i\cdot}, a_{i\cdot}^\star) = \sum_{i=1}^{n} (d_\phi(p_{i\cdot}, y_{i\cdot}) - d_\phi(a_{i\cdot}^\star, y_{i\cdot}))
\end{aligned}$$

Note that the theorem is even more general than we need and holds for all $y \in Q_\pi^\square$, not only those y which represent label matrices. A corollary of this theorem is that the BGA for Brier score is a new adjustment method dominating over additive adjustment in reducing Brier score. In practice, all practitioners should prefer BGA over UGA when looking to adjust their classifiers. Coherence and decomposition are interesting from a theoretical perspective but from a loss reduction standpoint, BGA is superior to UGA.

3.3 Implementation

Both UGA and BGA are defined through optimisation tasks, which can be shown to be convex. First, the objective function is convex as a sum of convex functions (Bregman divergences are convex in their second argument [2]). Second, the equality constraints that define Q_π are linear, making up a convex set. Finally, the inequality constraints of $Q_\pi^\square$ make up a convex set, which after intersecting with Q_π remains convex. These properties are sufficient [3] to prove that both the UGA and BGA optimisation tasks are convex.

UGA has only equality constraints, so Newton's method works fine with it. For Brier score there is a closed form solution [9] of simply adding the difference between the new distribution and the old distribution for every set of k probabilities. BGA computations are a little more difficult due to inequality constraints, therefore requiring interior point methods [3]. While multiplicative adjustment is for log-loss both BGA and UGA at the same time, it is easier to calculate it as a UGA, due to not having inequality constraints.

4 Experiments

4.1 Experimental Setup

While our theorems provide loss reduction guarantees when the exact class distribution is known, this is rarely something to be expected in practice. Therefore, the goal of our experiments was to evaluate the proposed adjustment methods in the setting where the class distribution is known approximately. For loss measures we selected Brier score and log-loss, which are the two most well known proper losses. As UGA is dominated by BGA, we decided to evaluate only BGA, across a wide range of different types of dataset shift, across different classifier learning algorithms, and across many datasets. We compare the results of BGA with the prior probability adjuster (PPA) introduced in Section 2, as this is to our knowledge the only existing method that inputs nothing else than the predictions of the model and the shifted class distribution. As reviewed in [17], other existing transfer learning methods need either the information about the features or a limited set of labelled data from the shifted context.

To cover a wide range of classifiers and datasets, we opted for using OpenML [16], which contains many datasets, and for each dataset many runs of different learning algorithms. For each run OpenML provides the predicted class probabilities in 10-fold cross-validation. As the predictions in 10-fold cross-validation are obtained with 10 different models trained on different subsets of folds, we compiled the prediction matrix p from one fold at a time. From OpenML we downloaded all user-submitted sets of predictions for both binary and multiclass (up to eight classes) classification tasks, restricting ourselves to tasks with the number of instances in the interval of $[2000, 1000000]$. Then we discarded every dataset that included a predicted score outside the range $(0, 1)$. To emphasize, we did not include runs which contain a 0 or a 1 anywhere in the predictions, since log-loss becomes infinite in case of errors with full confidence. We discarded

datasets with less than 500 instances and sampled datasets with more than 1000 instances down to 1000 instances. This left us with 590 sets of predictions, each from a different model. These 590 sets of predictions come from 59 different runs from 56 different classification tasks. The list of used datasets and the source code for running the experiments is available in the Online Supplementary at `https://github.com/teddyheiser/Shift_Happens_ECML_PKDD_2019`.

Shifting. For each dataset we first identified the majority class(es). After sorting the classes by size decreasingly, the class(es) $1, \ldots, m$ were considered as majority class(es), where j was the smallest possible integer such that $\pi_1 + \cdots + \pi_m > 0.5$. We refer to other class(es) as minority class(es). We then created 4 variants of each dataset by artificially inducing shift in four ways. Each of those shifts has a parameter $\varepsilon \in [0.1, 0.5]$ quantifying the amount of shift, and ε was chosen uniformly randomly and independently for each adjustment task.

The first method induces prior probability shift by undersampling the majority class(es), reducing their total proportion from $\pi_1 + \cdots + \pi_m$ to $\pi_1 + \cdots + \pi_m - \varepsilon$. The second method induces a variety of concept shift by selecting randomly a proportion ε of instances from majority class(es) and changing their labels into uniformly random minority class labels. The third method induces covariate shift by deleting within class m the proportion ε of the instances with the lowest values of the numeric feature which correlates best with this class label. The fourth method was simply running the other three methods all one after another, which produces an other type of shift.

Approximating the New Class Distribution. It is unlikely that a practitioner of adjustment would know the exact class distribution of a shifted dataset. To investigate this, we ran our adjustment algorithms on our shifted datasets with not only the exact class distribution, but also eight 'estimations' of the class distribution obtained by artificially modifying the correct class distribution $(\pi_1, \ldots, \pi_k)$ into $(\pi_1 + \delta, \ldots, \pi_m + \delta, \pi_{m+1} - \delta', \ldots, \pi_k - \delta'$, where δ was one of eight values $+0.01, -0.01, +0.02, -0.02, +0.04, -0.04, +0.08, -0.08$, and δ' was chosen to ensure that the sum of class proportions adds up to 1. If any resulting class proportion left the [0,1] bounds, then the respective adjustment task was skipped.In total, we recorded results for 17527 adjustment tasks resulting from combinations of dataset fold, shift amount, shift method, and estimated class distribution.

Adjustment. For every combination of shift and for the corresponding nine different class distribution estimations, we adjusted the datasets/predictions using the three above-mentioned adjusters: Brier-score-minimizing-BGA, log-loss-minimizing-BGA, and PPA. PPA has a simple implementation, but for the general adjusters we used the CVXPY library [6] to perform convex optimization. For Brier-score-minimizing-BGA, the selected method of optimization was OSQP (as part of the CVXPY library). For log-loss-minimizing-BGA, we used the ECOS optimizer with the SCS optimizer as backup (under rare conditions the optimizers could numerically fail, occurred 30 times out of 17527). For both

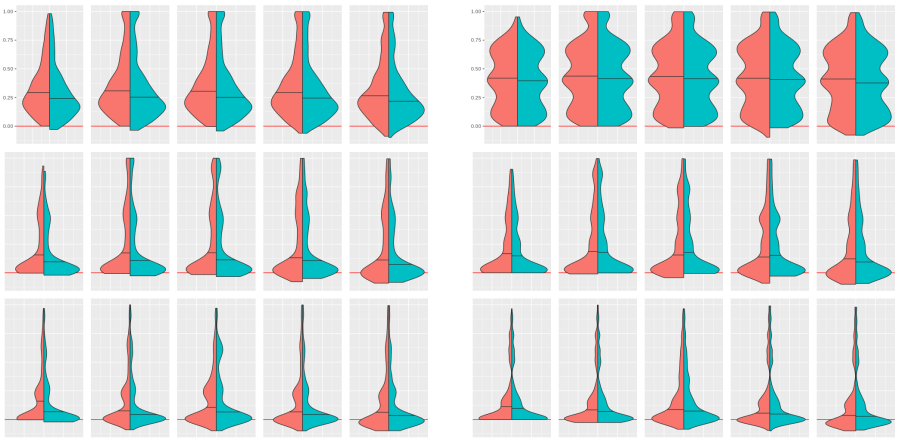

Fig. 2. The reduction in Brier score (left figure) and log-loss (right figure) after BGA adjustment (left side of the violin) and after PPA adjustment (right side of the violin). The rows correspond to different amounts of shift (with high shift at the top and low at the bottom). The columns correspond to amount of induced error in class distribution estimation, starting from left: 0.00, 0.01, 0.02, 0.04 and 0.08.

Brier score and log loss, we measured the unadjusted loss and the loss after running the dataset through the aforementioned three adjusters.

4.2 Results

On different datasets the effects of our shifting procedures vary and thus we have categorized the shifted datasets into 3 equal-sized groups by the amount of squared Euclidean distance between the original and new class distributions (high, medium and low shift). Note that these are correlated to the shift amount parameter ε, but not determined by it. Figures 2 and 3 both visualise the loss reduction after adjustment in proportion to the loss before adjustment. In these violin plots the part of distributions above 0 stands for reduction of loss and below 0 for increased loss after adjustment. For example, proportional reduction value 0.2 means that 20% of the loss was eliminated by adjustment. The left side of the left-most violins in Figure 2 show the case where BGA for Brier score is evaluated on Brier score (with high shift at the top row and low at the bottom). Due to guaranteed reduction in loss the left sides of violins are all above 0. In contrast, the right side of the same violins shows the effect of PPA adjustment, and PPA can be seen to sometimes increase the loss, while also having lower average reduction of loss (the horizontal black line marking the mean is lower). When the injected estimation error in the class distribution increases (next 4 columns of violins), BGA adjustment can sometimes increase the loss as well, but is on average still reducing loss more than PPA in all of the violin plots. Similar patterns of results can be seen in the right subfigure of Figure 2, where

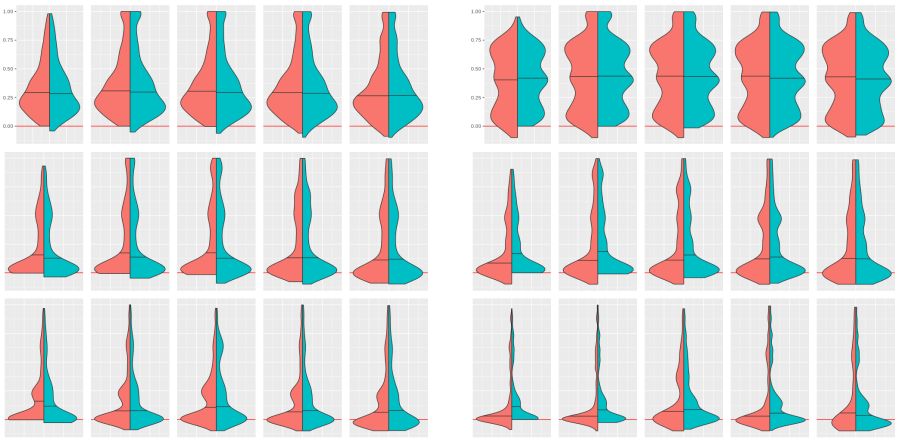

Fig. 3. The reduction in Brier score (left figure) and log-loss (right figure) after BGA adjustment to reduce Brier score (left side of the violin) and after BGA to reduce log-loss (right side of the violin). The rows correspond to different amounts of shift (high at the top and low at the bottom). The columns correspond to amount of induced error in class distribution estimation, starting from left: 0.00, 0.01, 0.02, 0.04 and 0.08.

BGA for log-loss is compared with PPA, both evaluated on log-loss. The mean proportional reduction of loss by BGA is higher than by PPA in 13 out of 15 cases. The bumps in some violins are due to using 4 different types of shift.

Figure 3 demonstrates the differences between BGA aiming to reduce Brier score (left side of each violin) and BGA to reduce log loss (right side of each violin), evaluated on Brier score (left subfigure) and log-loss (right subfigure). As seen from the right side of the leftmost violins, BGA aiming to reduce the wrong loss (log-loss) can actually increase loss (Brier score), even if the class distribution is known exactly. Therefore, as expected, it is important to adjust by minimising the same divergence that is going to be used to test the method.

5 Conclusion

In this paper we have constructed a family BGA of adjustment procedures aiming to reduce any proper loss of probabilistic classifiers after experiencing dataset shift, using knowledge about the class distribution. We have proved that the loss is guaranteed to reduce, if the class distribution is known exactly. According to our experiments, BGA adjustment to an approximated class distribution often still reduces loss more than prior probability adjustment.

Acknowledgments

This work was supported by the Estonian Research Council under grant PUT1458.

References

1. Banerjee, A., Guo, X., Wang, H.: On the optimality of conditional expectation as a Bregman predictor. IEEE Trans. on Information Theory **51**(7), 2664–2669 (2005)
2. Bauschke, H.H., Borwein, J.M.: Joint and separate convexity of the Bregman distance. In: Studies in Computational Mathematics, vol. 8, pp. 23–36. Elsevier (2001)
3. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge Univ. Press (2004)
4. Bregman, L.M.: The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. USSR Computational Mathematics and Mathematical Physics **7**(3), 200–217 (1967)
5. Dawid, A.P.: The geometry of proper scoring rules. Annals of the Institute of Statistical Mathematics **59**(1), 77–93 (2007)
6. Diamond, S., Boyd, S.: CVXPY: A Python-embedded modeling language for convex optimization. Journal of Machine Learning Research **17**(83), 1–5 (2016)
7. Gretton, A., Smola, A.J., Huang, J., Schmittfull, M., Borgwardt, K.M., Schölkopf, B.: Covariate shift by kernel mean matching. Dataset shift in machine learning pp. 131–160 (2009)
8. Hein, M.: Binary classification under sample selection bias. Dataset Shift in Machine Learning (J. Candela, M. Sugiyama, A. Schwaighofer and N. Lawrence, eds.). MIT Press, Cambridge, MA pp. 41–64 (2009)
9. Kull, M., Flach, P.: Novel decompositions of proper scoring rules for classification: Score adjustment as precursor to calibration. In: Joint European Conf. on Machine Learning and Knowledge Discovery in Databases. pp. 68–85. Springer (2015)
10. Merkle, E.C., Steyvers, M.: Choosing a strictly proper scoring rule. Decision Analysis **10**(4), 292–304 (2013)
11. Moreno-Torres, J.G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N.V., Herrera, F.: A unifying view on dataset shift in classification. Pattern Recognition **45**(1), 521–530 (2012)
12. Saerens, M., Latinne, P., Decaestecker, C.: Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. Neural Comp. **14**(1), 21–41 (2002)
13. Shimodaira, H.: Improving predictive inference under covariate shift by weighting the log-likelihood function. J. of Stat. Planning and Inference **90**(2), 227–244 (2000)
14. Storkey, A.: When training and test sets are different: characterizing learning transfer. Dataset shift in machine learning pp. 3–28 (2009)
15. Sugiyama, M., Krauledat, M., Müller, K.R.: Covariate shift adaptation by importance weighted cross validation. Journal of Machine Learning Research **8**(May), 985–1005 (2007)
16. Vanschoren, J., van Rijn, J.N., Bischl, B., Torgo, L.: OpenML: Networked science in machine learning. SIGKDD Explorations **15**(2), 49–60 (2013)
17. Weiss, K., Khoshgoftaar, T.M., Wang, D.: A survey of transfer learning. Journal of Big Data **3**(1), 9 (2016)