

UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Theodore James Thibault Heiser

Dataset Shift and the Adjustment of
Probabilistic Classifiers

Master's Thesis (30 ECTS)

Supervisor: Meelis Kull, PhD

Tartu 2018

Dataset Shift and the Adjustment of Probabilistic Classifiers

Abstract: Classification is the machine learning problem of assigning a class to a given instance of data defined by a set of features. Probabilistic classification is the stricter problem of assigning probabilities to each possible class given an instance, indicating the classifiers confidence in that class being correct for the given instance.

The underlying assumption of classical machine learning is that any instance used to train or test the classifier is sampled independently and identically distributed from the same joint probability distribution of features and labels. This, however, is a very unlikely situation in real world applications, as the distribution of data frequently changes over time. The change in the distribution of data between the time of training the classifier and a future point in the classifier's life cycle (testing, deployment, etc.) is known as dataset shift.

In this thesis, a novel procedure is presented which improves the performance of a probabilistic classifier experiencing any pattern of shift that causes the class distribution to change, a property most patterns of shift share. This new technique is based off of adjustment, the process of matching the probabilistic classifier's expected output to the class distribution of the data. In previous works it has been shown that adjustment can be used to reduce expected loss for mean squared error and KL divergence. These two loss functions are a part of a wider family of loss functions called proper scoring rules.

The proposed novel procedure is termed general adjustment, since it reduces expected loss for all proper scoring rules. It comes in two varieties, unbounded and bounded. Unbounded general adjustment gives results equivalent to the previously described adjustment procedures for mean squared error and KL divergence. Bounded general adjustment is a further refinement, reducing expected loss as much or more than its unbounded form. Both are convex minimization tasks, and therefore computationally efficient to compute.

The results of a series of experiments show that bounded general adjustment reduces loss in a practical setting, where the exact value of the new class distribution may not be known. Even with moderate error in the estimation of the new class distribution, bounded general adjustment still reduces loss in most cases.

Keywords: machine learning, statistical learning, transfer learning, dataset shift, probabilistic classification, calibration, adjustment, proper scoring rules, Bregman divergence

CERCS: P176 - Artificial intelligence

Andmenihe ja tõenäosuslike klassifitseerijate kohandamine

Lühikokkuvõte: Klassifitseerimine on masinõppe ülesanne, kus igale andmepunktile tuleb tema tunnuste põhjal määrata klass. Tõenäosuslik klassifitseerimine on kitsam ülesanne, kus kõikidele võimalikele klassidele tuleb määrata iga andmepunkti puhul tõenäosus, mis näitaks klassifitseerija enesekindlust andmepunktile antud klassi määramisel.

Klassikalises masinõppes eeldatakse, et kõik andmepunktid, mida kasutatakse klassifitseerija treenimiseks või testimiseks on valitud sõltumatult ja samast tunnuste ja märgendite ühisjaotusest. See on aga päriselulistest rakendustes väga ebatõenäoline, kuna sageli andmete jaotus muutub aja jooksul. Muutust andmete jaotuses klassifitseerija treenimise ja hilisema rakendamise vahel tuntakse kui andmenihet.

Antud töös pakutakse välja uus meetod mistahes selliste tõenäosuslike klassifitseerijate töö parandamiseks, mille puhul on andmetes klassijaotust muutev nihe - omadus, mis on enamikel andmenihetel. Välja pakutud meetod baseerub kohandamise protsessil, mille käigus sobitatakse tõenäosusliku klassifitseerija oodatav väljund andmete klassijaotusega. Varasemas töös on näidatud, et kohandamine vähendab oodatavat kahju keskmise ruutvea ja KL-divergentsi puhul. Need kaks kaofunktsiooni on osa laiemast funktsioonide perest, mida kutsutakse puhasteks skoorireegliteks.

Välja pakutud protseduuri kutsume edaspidi üldiseks kohandamiseks, kuna see vähendab oodatavat kahju kõikide puhaste skoorireeglite korral. Üldisel kohandamisel on kaks variatsiooni: piiramata ja piiratud. Piiramata üldine kohandamine annab keskmise ruutvea ja KL-divergentsi korral sama tulemuse nagu juba eksisteerivad kohandamise protseduurid. Piiratud üldine kohandamine on täiendus, mis vähendab oodatavat kahju vähemalt sama palju või rohkem kui piiramata versioon. Mõlemad meetodid lahenduvad kui kumerad minimiseerimisülesanded ning on seega arvutuslikult efektiivsed.

Ekspriimentide tulemused näitavad, et piiratud üldine kohandamine vähendab kahju praktilistes olukordades, kus uue andmejaotuse klassijaotus ei pruugi olla täpselt teada. Isegi mõõduka veaga hinnatud klassijaotuse korral suudab piiratud üldine kohandamine enamikel juhtudel kahju vähendada.

Võtmesõnad: masinõpe, statistiline õpe, siirdeõpe, andmenihe, tõenäosuslik klassifitseerimine, kalibreerimine, kohandamine, puhtad skoorireeglid, Bregmani divergents

CERCS: P176 - Tehisintellekt

Contents

1	Introduction	5
2	Shift Happens	7
2.1	Causes of Shift	8
2.2	Patterns of Shift	10
2.3	Correction of Shift	16
3	Loss and Divergence	19
3.1	From Proper Scoring Rules to Bregman Divergences	19
3.2	Some Examples	21
3.3	Bonus Properties of Bregman Divergence	25
3.4	Decompositions	26
4	Adjusting Generalized	30
4.1	Unbounded General Adjustment	30
4.2	Bounded General Adjustment	35
4.3	Implementing Adjustment	38
5	Experimental Results	41
5.1	Setup	41
5.2	Results	43
6	Conclusion	51
	References	56
	Appendix	57
	II. Licence	57

1 Introduction

Classification is the task of labelling a given instance of data with a set of values, called "features", as its appropriate "class". As an example, consider the task of labelling a description of an animal as a cat or a dog. The description of a given animal (ear shape, weight, diet, etc) are the features, whereas the class is either a cat or a dog. A classifier is a computer program which performs this task.

Classifiers are typically created through supervised learning. The classifier model is trained by an algorithm and a set of pre-labelled data. A well trained model should be reasonably adept at predicting the correct class of an unlabelled instance of data afterwards. A probabilistic classifier goes a step further in this task, and given an instance of data, instead of an all-or-nothing prediction for a single class, it outputs a probability vector of length k , for the k possible classes in the task. The value at a given index in this vector represents the classifier's confidence that its corresponding class is the true class of the given instance. The features and the class can be described as random variables, X and Y respectively. Each instance of data can then be understood as being drawn from a joint probability distribution relating the random variables, $\mathbb{P}(X, Y)$. A probabilistic classifier, in these terms, is supposed to output the correct posterior probabilities, $\mathbb{P}(Y|X = x)$, for a given set of features, x .

The core assumption forming the basis for classification is that both the training data and the testing/deployment data are sampled independently and identically distributed (i.i.d.) from the same probability distribution, $\mathbb{P}_{train}(X, Y) = \mathbb{P}_{test}(X, Y)$. However, in real world settings, this assumption is broken more often than not: surveyors are often biased to collecting data from certain segments of a population, medical diagnostic classifiers are typically trained with an oversampling of the disease-positive class, user demographics and preferences change over time on e-commerce and social media sites, etc. This change in the probability distribution is referred to as *dataset shift*, and it comes in many forms [Moreno-Torres et al., 2012, Kull and Flach, 2014, Storkey, 2009]. Many methods over the past several decades have been proposed to correct for shift, taking advantage of the properties of specific patterns to do so, yet none are without their drawbacks. Sometimes their requirements are difficult to accommodate in practice [Gama et al., 2014], require retraining of the model [Sugiyama et al., 2007], or are unsuitable for probabilistic classifiers [Provost and Fawcett, 2001].

But what does correcting for shift exactly mean? For probabilistic classifiers, the goal is to output the correct posterior probability for each class of any given instance. To measure the performance of such a model a loss function is needed, particularly one which has its expected output minimized only when the model's prediction of the posterior probability is correct. Loss functions with this property are called strictly proper scoring rules [Epstein, 1969, Dawid, 2007] which are

closely related to another family of functions, Bregman divergences [Bregman, 1967, Banerjee et al., 2005a]. Two proper scoring rules are in especially common use: Brier score and log-loss. Correcting a model in these terms means to reduce expected loss, reducing the divergence between the predicted probabilities and the true probabilities.

A wide variety of shift patterns, including some that currently do not have well-known correction methods, share the trait of having their class distributions change, $\mathbb{P}_{train}(Y) \neq \mathbb{P}_{test}(Y)$. Typically, when the class distribution changes the expected output of probabilistic classifier fails to change with it. The process of correcting this mismatch was introduced in "Novel Decompositions of Proper Scoring Rules for Classification: Score Adjustment as Precursor to Calibration" [Kull and Flach, 2015] and termed *adjustment*.

In that same paper, it was proved that performing "coherent" adjustment, a procedure fulfilling a particular requirement, on a set of predictions will reduce the expected loss across that set. It showed that coherent adjustment procedures exist for two loss functions, mean squared error (MSE) and KL divergence, but they were not without issues. The adjusted probabilities from the MSE adjustment method could take values under zero and above one, clearly nonsensical values if interpreted as probabilities, and the proposed iterative algorithm to adjust for KL divergence was unreliable, often failing to converge. Several open questions remained at the end of the paper. Does coherent adjustment exist for all proper scoring rules? Is there a reliable algorithm to compute adjusted scores? Can predictions be adjusted so that their output can be interpreted as probabilities and still have their expected loss reduced? Is adjustment useful in a realistic setting?

This thesis answers those questions by introducing two novel adjustment procedures: unbounded general adjustment (UGA) and bounded general adjustment (BGA). UGA provides coherent adjustment for any proper scoring rule, and is equivalent to the previously proposed adjustment procedures when used for MSE and KL divergence. BGA guarantees at least as much reduction of expected loss as UGA, and has the benefit of only outputting probabilities between zero and one. Both are easy to compute, as they both can be implemented with convex optimizers.

In Section 2, we will look more in depth to dataset shift, classifying and identifying different patterns and reviewing popular techniques for correcting for shift. In Section 3, we review proper scoring rules, connecting them to another class of functions, Bregman divergences, giving common examples, deriving some properties, and reviewing the decomposition of loss presented in [Kull and Flach, 2015]. In Section 4, UGA and BGA are defined and their theorems proved. In Section 5, the results of a series of experiments are displayed and analyzed. In Section 6, the thesis is concluded.

2 Shift Happens

The goal of a probabilistic classifier is to, given instance of data x and a set of k classes y , correctly output the posterior probability $\mathbb{P}(y_j|x)$ for each class y_j with $j \in [1, \dots, k]$. A joint probability distribution $\mathbb{P}(X, Y)$, with X being the random variable representing the features/covariates and Y being the random variable representing the class labels, can be broken up into four components: the evidence $\mathbb{P}(X)$, the prior probability $\mathbb{P}(Y)$, the likelihood $\mathbb{P}(X|Y)$, and the posterior probability $\mathbb{P}(Y|X)$. Bayes' theorem gives the relation between these four probability distributions, $\mathbb{P}(Y = y_i|X) = \frac{\mathbb{P}(X|Y=y_i)\mathbb{P}(Y=y_i)}{\mathbb{P}(X)} = \frac{\mathbb{P}(X|Y=y_i)\mathbb{P}(Y=y_i)}{\sum_{j \in \text{classes}} \mathbb{P}(X|Y=y_j)\mathbb{P}(Y=y_j)}$, for a given class y_i . A classifier which predicts the true posterior probability distribution, $\mathbb{P}(Y|X = x)$, for every instance x is referred to as a Bayes' optimal classifier.

One of the underlying principles of classical statistical learning is that all the instances given to a model during training, testing, and deployment are drawn independently and identically distributed (i.i.d.) from the same joint distribution. Dataset shift describes a change in the distribution, breaking the i.d. portion of the i.i.d. assumption. This is a pervasive problem in real world applications of machine learning [Ditzler et al., 2015].

Definition 1 (Dataset Shift [Moreno-Torres et al., 2012]). *Define X as the random variable representing the values of the features/covariates, Y as the random variable representing the different classes, and $\mathbb{P}(X, Y)$ as the joint probability distribution that relates them. Dataset shift is*

$$\mathbb{P}_{\text{train}}(X, Y) \neq \mathbb{P}_{\text{test}}(X, Y),$$

where $\mathbb{P}_{\text{train}}$ represents the distribution at the time of training and \mathbb{P}_{test} represents the distribution at testing, deployment, or some later time.

Shifting distributions are pervasive in practice, almost any practical application of classification is going to see the distribution it is operating upon change, so the question of how to change the model to correct for shift and improve performance is motivated. Correcting for dataset shift is a subfield of transfer learning, which is more broadly defined and can involve changing the learning task entirely.

Definition 2 (Transfer Learning [Pan and Yang, 2010]). *Define domain $\mathcal{D} = \{X, \mathbb{P}(X)\}$ and task $\mathcal{T} = \{Y, f(\cdot)\}$, where X , $\mathbb{P}(X)$, and Y are as described above, and $f(\cdot)$ is the probabilistic function that maps from X to Y which can be understood as $\mathbb{P}(Y|X)$. Given a source domain \mathcal{D}_S and learning task \mathcal{T}_S , a target domain \mathcal{D}_T and learning task \mathcal{T}_T , transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in \mathcal{D}_T using the knowledge in domain \mathcal{D}_S and \mathcal{T}_S , where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$.*

Dataset shift in this context is the subset of transfer learning problems where $X_S = X_T$ and $Y_S = Y_T$. Only their probability distributions change. The set of features do not change and no new classes are added.

In Section 2.1, a brief introduction to why and how shift occurs is given. In Section 2.2, a categorization of the patterns of shift is presented. In Section 2.3, some current methods of correcting shift are reviewed.

2.1 Causes of Shift

The actual causes of shift are far too varied to go over individually, and there does not seem to be a large consensus on how to categorize them. In the previously mentioned "Unifying View" paper [Moreno-Torres et al., 2012], the authors divide shift into either being the result of a non-stationary environment or of sample selection bias. In "Patterns of Dataset Shift" [Kull and Flach, 2014], the authors introduce two generators of shift, the "context" and the "sampling bias", also divide the variables into the observed and hidden versions. Denote X and Y as the observed variables and X^H and Y^H as the underlying/hidden variables. An example of how these variables may differ: X may represent the measured temperature of a process at a chemical plant, whereas X^H would represent the actual temperature. A sensor making such a measurement may be sensitive to unstable/high-heat environments and the recorded temperature may lose accuracy over time. The recorded data will see the data shift, while in reality the temperatures for a given process have not shifted in such a way. The authors then present a graphical framework showing that the context can act on both the observed and hidden variables and that the sampling bias affects only the observed variable.

In this section, a simpler model is presented with sampling selection bias included in the context. In this model, causes of shift can be broken into two main categories, shift affecting the observed variables and shift affecting the underlying variables. Figure 1 is a graphical representation of this difference. The underlying variables, X^H and Y^H , are influencing the observed values, X and Y , while the context can shift the hidden variables (the dotted red arrows) or the observed variables (the dashed blue arrows).

Shift of the Observed Variables The observed features and class are not necessarily representative of the actual distribution, $\mathbb{P}(X^H, Y^H)$. The observed features are a random variable X dependent on X^H and the observed labels are Y dependent on Y^H . They are from the distribution $\mathbb{P}(X, Y) = g(\mathbb{P}(X^H, Y^H))$, where g is the function that applies bias, inaccuracies, and/or noise. The shift of these variables is represented by the context (the origin of shift) affecting X and Y with the dashed blue arrows in Figure 1.

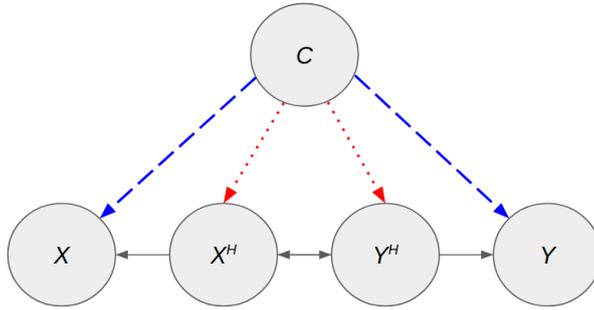


Figure 1. Representation of the context C shifting the random variables. The dashed blue arrows indicate shift of observed variables. The dotted red arrows indicate shift of underlying/hidden variables.

Sample selection bias can cause shift if the bias changes from training to testing. Ideally, a data collector should sample data uniformly, or at least with the same bias (identically), from the distribution, but this is often not the case. Labelled data is often easier to collect from some parts of the distribution than others. The following two conditions are sufficient and necessary to classify a shift as sample selection bias [Hein, 2009].

$$\begin{aligned} \text{support condition: } & \mathbb{P}_{test}(x) > 0 \Rightarrow \mathbb{P}_{train}(x) > 0 \\ \text{selection condition: } & \mathbb{P}_{test}(x, y) > 0 \Rightarrow \mathbb{P}_{train}(x, y) > 0 \end{aligned}$$

The first condition states that if a particular set of features x can be drawn from the testing distribution, then it can also be drawn from the training distribution. The second condition states that if an instance of features x with a certain class y can be sampled from the testing distribution, then that class y can also be drawn from the training distribution for that given x .

Of course, in reality a collection process can have so much bias that regions of the distribution go unsampled in the training data. Technically this would fall outside of these conditions, while still clearly being the result of bias by the data collector. However, these conditions are necessary for a family of shift correcting methods that target sample selection bias. They will be reviewed in Section 2.3.

The distribution of observational variables can also differ from the underlying variables through inaccurate measurements. Collecting data is hardly ever a perfect process, and noise and uncalibrated instrumentation can misrepresent our data. If this noise rate changes or the instrumentation collecting the data becomes more or less calibrated between training and testing, then shift can arise. For example, the

previously mentioned temperature measuring sensor becoming less accurate over time.

Shift of the Hidden Variables In the above case, reasonable solutions might simply involve changing the sampling methods to collect data or repairing/improving the instrumentation doing the data collection. The underlying variables can change too though. Shift of these variables is represented by the context (the origin of shift) affecting X^H and Y^H with the dotted red arrows in figure 1.

This is usually the result of non-stationary environments or when a model trained in one environment is applied to a new environment. The demographics of people using a service might change over the years, customer preferences might change season to season, a business might enter a new country’s market and attempt to reapply their old model.

Although never truly caused by sampling bias, in some cases this shift can be modelled as sample selection bias if the sufficient and necessary conditions as described above are met.

2.2 Patterns of Shift

This section follows the terminology set by the paper "A Unifying View on Dataset Shift in Classification" [Moreno-Torres et al., 2012], describing four ways to intuitively group shift: prior probability shift, covariate shift, concept shift, and other types of shift. Figure 2 represents the data of a yet-to-be-shifted example distribution which will be compared against shifted versions when the different patterns of shift are introduced. The data was generated equally from two 2-D Gaussian distributions. One represented the positive classes centered at coordinate $(x1, x2) = (-3, 0)$ with a variance of 2 in each direction, and the other represented the negative class centered at coordinate $(x1, x2) = (3, 0)$ with the same variance.

Prior Probability Shift Prior shift is when the likelihood stays the same, but the prior probability/class distribution changes.

$$\begin{aligned}\mathbb{P}_{train}(Y) &\neq \mathbb{P}_{test}(Y) \\ \mathbb{P}_{train}(X|Y) &= \mathbb{P}_{test}(X|Y)\end{aligned}$$

Since the class distribution is guaranteed to change, it is a clear candidate for adjustment. Figure 3 shows an example of what prior shift might look like on our toy dataset in Figure 2. In both figures, 7000 instances have been sampled, but in Figure 3 the positive class was down-sampled. The class distribution has moved from a 50:50 positive-to-negative ratio in Figure 2 to a 15:85 positive-to-negative ratio in Figure 3. The reader can easily see that in Figure 2 instances with $x1 \approx 0.0$

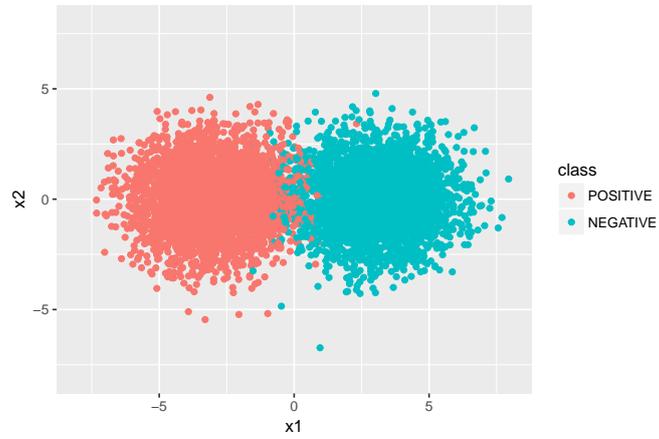


Figure 2. 7000 instances sampled from the original / training distribution. The positive class is at 50%.

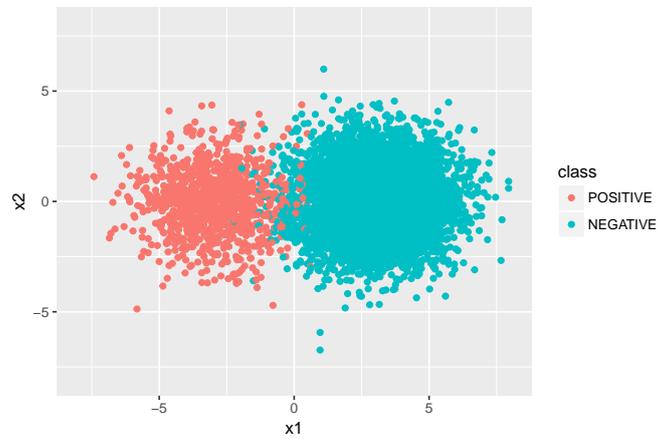


Figure 3. 7000 instances sampled from the prior shifted distribution. The positive class is at 15%.

have $P_{train}(+1|X_{x1=0}) \approx 0.5$, the corresponding posterior probability in Figure 3 would be much lower.

Often, this shift is caused by a non-stationary environment. For example, a program that tries to classify clothing items posted on online stores will likely run into more shorts and sandals in the summer and more coats and gloves in the winter. The description of a given piece of clothing is the X variable and its type/category is the Y variable. The descriptions for a given type of clothing $\mathbb{P}(X|Y)$ will not change, but the distribution of clothing types $\mathbb{P}(Y)$ will.

Sometimes, the model designer will often purposefully induce prior shift with sampling bias. This a very common approach in trying to solve the imbalanced class problem [Sun et al., 2009], where practitioners often apply down/up-sampling to make up for imbalanced data when one class is smaller than the other(s) [Japkowicz and Stephen, 2002].

As a side note, the authors of the "Unifying View" [Moreno-Torres et al., 2012] paper make a clear distinction between causal direction of a given problem being either $X \rightarrow Y$ or $Y \rightarrow X$. They then claim that only $Y \rightarrow X$ problems can experience prior probability shift. The author of this thesis does not believe such a requirement is necessary.

Take for example the $X \rightarrow Y$ problem of assessing a patient's probability of developing lung cancer from their risk factors, such as if they smoke, if they are exposed to radon, if they have a family history of cancer. The risk factors are the X variable and developing lung cancer is the Y variable. Clearly the risk factors are affecting the risk of developing lung cancer, not the other way around. However, it is completely plausible that a machine learning practitioner would up-sample the instances of individuals who were diagnosed with lung cancer during training. Cancer is quite rare, so balancing the dataset might help a with training a model.

Because of counter-examples like this and for other reasons, this author chooses to not make the same distinction as the referenced paper.

Covariate Shift Covariate shift is when the posterior probability stays the same, but the evidence/covariate distribution changes.

$$P_{train}(X) \neq P_{shift}(X)$$

$$P_{train}(Y|X) = P_{shift}(Y|X)$$

This might strike the reader as strange, since the probabilistic classifier is attempting to output the posterior probability, so why worry about it? In the case of the Bayes' optimal classifier, there is no concern, but this is rarely the case for a classifier in practice. The covariates might get over-sampled in part of the distribution that was not learned well, or covariates might appear with values unseen in the test distribution. A model retrained on the new distribution

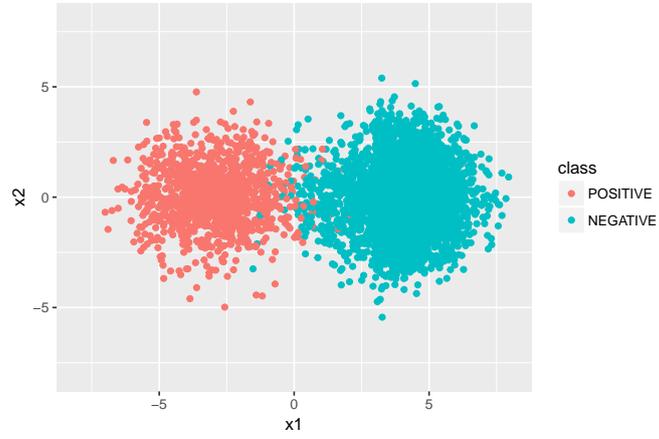


Figure 4. 7000 instances sampled from the covariate shifted distribution. The positive class is at 15%.

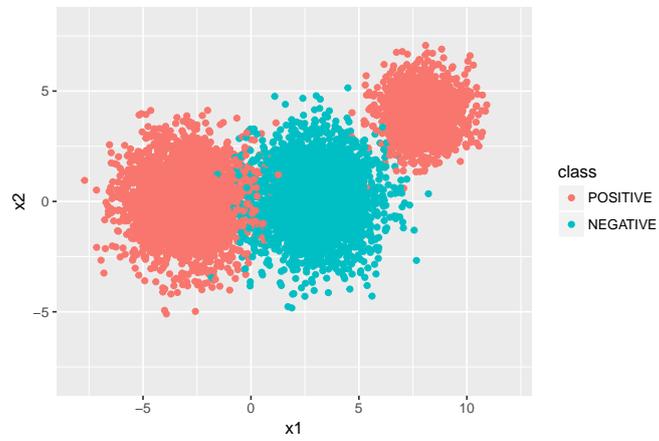


Figure 5. 7000 instances sampled from the covariate shifted distribution. The positive class is at 60%.

would likely perform better than the old model, while neither of them imitate the posterior distribution perfectly.

Undersampling by the data collector is a common cause of covariate shift. For example, a prestigious psychology department might try to learn behaviors from its student body and apply its predictions to the general public. The students are more likely to be younger and wealthier than the general public though.

Covariate shift can be further split up into two more categories, sample selection bias shift and shift with new data outside of the training set's domain.

The former is when the shift can be modelled as sample selection bias (as described in Section 2.1). Correcting for this type of shift is usually addressed with reweighting of the labelled training instances (which will be described in Section 2.3). Figure 4 shows an example of this subvariety. Here we have 7000 instances sampled from a distribution that down-samples points where $x_1 < 3$. The class distribution becomes a 15:85 positive-to-negative ratio. Note that if instances with $x_2 < 0$ were down-sampled instead, the class distribution would not change.

The type of shift with new data outside of the training set's domain is more problematic, as this new region of features was not learned during training. Often the only reasonable solution is collection of data from the new region and retraining the model. In Figure 5 data is seen in a new region. This changes the class distribution to a 40:60 positive-to-negative ratio.

Concept Shift For concept shift, a conditional probability (the likelihood or posterior) changes, but the distribution that they are conditioned on (the prior or evidence, respectively) doesn't change.

Concept shift is defined as:

$$P_{train}(X) = P_{test}(X)$$
$$P_{train}(Y|X) \neq P_{test}(Y|X).$$

or

$$P_{train}(Y) = P_{test}(Y)$$
$$P_{train}(X|Y) \neq P_{test}(X|Y).$$

This is when the "concept", what is being learned, is changing. Consider Figure 6 for an example of the former and Figure 7 for an example of the latter. These are challenging types of shift, although relatively uncommon. Typically, these cases are addressed with adaptive learning techniques, a form of online learning to correct for a changing environment.

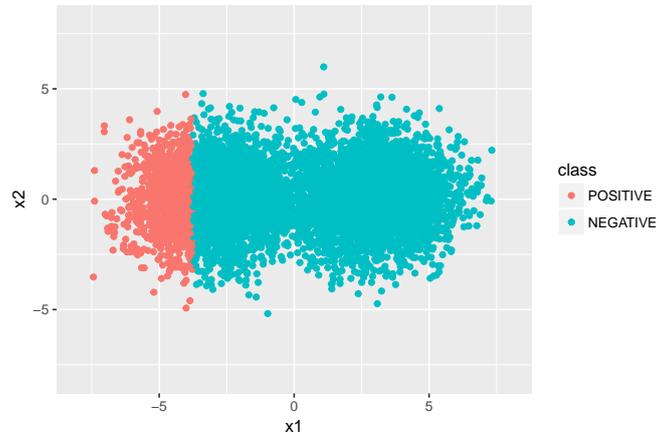


Figure 6. 7000 instances sampled from the concept shifted (first definition) distribution. The positive class is at 15%.

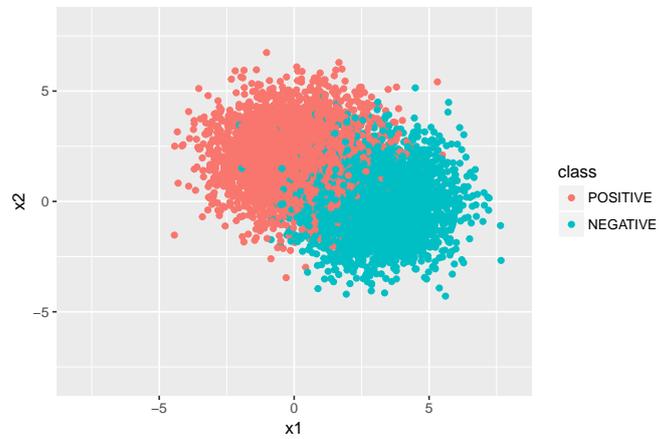


Figure 7. 7000 instances sampled from the concept shifted (second definition) distribution. The positive class is at 50%.

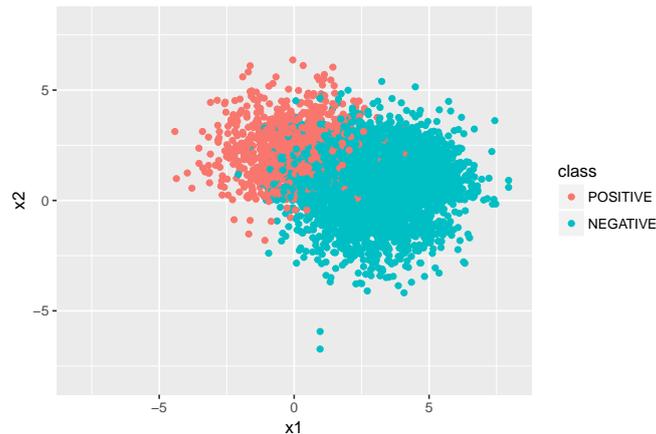


Figure 8. 7000 instances sampled from one of many possible other shifted distributions. The positive class is at 15%.

Other Types of Shift This describes the remaining types of shift that have not been covered. Look to Figure 8 for an illustration of one possible shifted set.

Because all components of the distribution are changed, it is a difficult problem to adapt for. Online adaptive learning techniques can still be an option, although those have extra requirements that might make them impractical in many situations. If the other type of shift is not too severe, the classifiers performance may not worsen too much and the original model is still useful.

This may sound like an exotic shift, but it can be common in real world settings. For example, this can simply be the result of sample selection bias. Undersampling one region of the features and undersampling a class is a realistic setting, and it falls under this category. In such a case, reweighting is an option.

2.3 Correction of Shift

Correcting shift has many proposed solutions. Three main groupings of correction methods are listed below. Methods to correct for sample selection bias and prior probability shift try to take advantage of specific properties of the type of shift being experienced, whereas adaptive learning tries to learn any type of shift resulting from a non-stationary environment by continuously updating its model.

Addressing Sampling Selection Bias If the shift can be modelled as sample selection bias as described in Section 2.1, many methods exist to correct for it. The primary way this type of situation is resolved is by reweighting the training set and retraining the model. Specific approaches include importance weighted

cross validation [Sugiyama et al., 2007], maximum weighted log-likelihood estimate [Shimodaira, 2000], and kernel mean matching [Gretton et al., 2009].

Prior Probability Shift Specific Solutions Identifying the new class distribution is an emerging field in its own right. First formally defined just over a decade ago [Forman, 2005], quantification is the learning task of estimating the class distribution, $\mathbb{P}(Y)$ [González et al., 2017]. The methods can be broken into three types: ① classify, count, and correct for bias [Forman, 2008], ② classify with models trained on quantification loss functions [Milli et al., 2013, Barranquero et al., 2013], and ③ distribution matching [Saerens et al., 2002].

With a fairly accurate estimation of the new class distribution, it has been proposed [Saerens et al., 2002] that the old prior probability term in Bayes’ formula can be swapped out for the new one, adjusting the predictions. In this thesis, this procedure is referred to as prior probability adjustment (PPA) and the formal definition is as follows:

Definition 3 (Prior Probability Adjustment (PPA)).

$$\mathbb{P}_{test}(Y|X) = \frac{\mathbb{P}_{test}(Y)}{\mathbb{P}_{train}(Y)} \frac{\mathbb{P}_{train}(X|Y)}{\sum \frac{\mathbb{P}_{test}(Y)}{\mathbb{P}_{train}(Y)} \mathbb{P}_{train}(Y|X)}$$

This does adjust a Bayes’ optimal classifier perfectly to output the corrected posterior probabilities of the prior probability shifted distribution. However, being able to learn a Bayes’ optimal classifier is uncommon in practical settings. A very poor classifier can be easily shown to perform even worse when PPA is applied, so it is unclear how PPA affects the typical classifier.

Other methods include using an iterative expectation maximization (EM) algorithm if the new class distribution is unknown [Saerens et al., 2002] and using reweighting techniques, since prior probability shift can be modelled as a case of sample selection bias. Some suggest creating a robust classifier that performs well under various class distributions by using AUC as the loss function [Provost and Fawcett, 2001]. However, AUC is not a proper scoring rule [Byrne, 2015], so it should not be used for training probabilistic classifiers. Proper scoring rules will be thoroughly explained in the Section 3.

Adaptive Learning Environments that shift relatively slowly over time can also be adapted to if the right infrastructure is in place. Adaptive learning works in an incremental or online fashion, where the model is continually updated even while in production [Gama et al., 2014]. This is only really practical for classification applications where the practitioner is able to automate retrieving the correct label after the model makes a prediction. Creating such a pipeline to perform online

learning is not always viable, but if online adaptive learning is applied then it can handle a wide range of shift, especially those from a non-stationary environment.

All online adaptive models ① make a prediction for a given instance, ② receive the true label of the instance after some time, and then ③ update their model. The details of this procedure vary from model to model. Some differences between these models include how they weight older data, their data forgetting mechanisms, if they can identify when shift is occurring, etc.

3 Loss and Divergence

One of the core features of any supervised machine learning problem is the use of a loss (also known as risk) function to evaluate a model's performance. In classification, a loss function takes the output of the model and the label for a given instance as parameters, and then returns a real number as output. The lower number indicates a better result, with zero indicating the model matching the instance's label. The goal of any model is to minimize the expected loss over the joint distribution which generates the data. Since the joint distribution is unknown to the machine learning practitioner, the expected loss is approximated by the empirical loss averaged over a set of data.

Loss functions for probabilistic classifiers take in a model's posterior probability prediction for each class of a given instance and its true label. This can be understood as taking two vectors of equal length with each index representing a different class. Each prediction will have a value between zero and one subject to all entries summing to one, while the true label will have a single one at its appropriate index. In most applications, a given set of features/covariates will usually not be deterministically associated with a single class, implying that the correct posterior probability doesn't have a one at any index. This means a perfect/Bayes' optimal classifier will indeed generate at least some loss for nearly every instance. Intuitively then, it would be ideal if the expected loss over the whole probability distribution would be at a minimum when the model correctly predicts the posterior probability distribution for a given instance.

This intuition leads to the definition of proper scoring rules, the topic covered in this section. In Section 3.1, proper scoring rules and the closely related Bregman divergences will be defined and connected. In Section 3.2, a few common proper scoring rules / Bregman divergences will be given as examples. In Section 3.3, two properties of Bregman divergence will be derived which will be needed to introduce our general adjusters. In Section 3.4, a decomposition of proper scoring rules [Kull and Flach, 2015] will be reviewed.

3.1 From Proper Scoring Rules to Bregman Divergences

As mentioned previously, for a probabilistic classifier to hope to perform its task of outputting correct posterior probabilities, it must be trained with a loss function that is minimized when the predictions match the correct posterior probabilities. Loss functions with this property are called proper scoring rules (PSRs) [Dawid, 2007, Merkle and Steyvers, 2013, Kull and Flach, 2015].

Definition 4 (Proper Scoring Rule). *Given a loss function f and a random variable representing the label Y , we define f to be a proper scoring rule if, for any prediction*

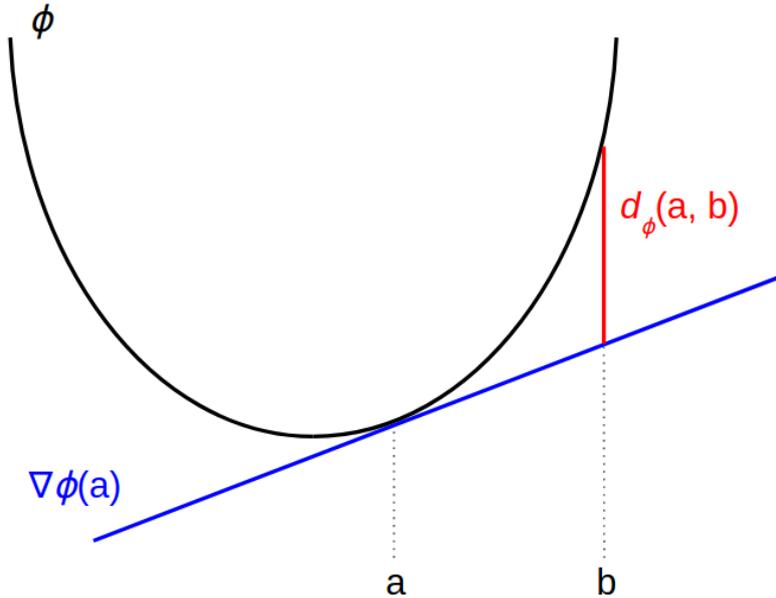


Figure 9. Graphical representation of a Bregman divergence d_ϕ from point a to point b measured on a strictly convex function ϕ . The divergence is the difference in the value of ϕ and the tangent $\nabla\phi(a)$ at the point b .

of the probability distribution of labels p and the true probability distribution of labels q , the following inequality holds

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

and strictly proper if it is a strict inequality when $p \neq q$.

This is a straightforward definition, but it does not give any intuition of what the geometry of these functions look like. On the other hand, Bregman divergences, also called Bregman loss functions (BLFs) when used as a loss measure, are a family of functions with a very geometric definition [Bregman, 1967, Banerjee et al., 2005b]. Consult Figure 9 for a graphical representation.

Definition 5 (Bregman Divergence). *Let $\phi : S \mapsto \mathbb{R}$ be a strictly convex function defined on a convex set $S \subseteq \mathbb{R}^k$ such that ϕ is differentiable on the relative interior of S , $ri(S)$. The Bregman divergence $d_\phi : ri(S) \times S \mapsto [0, \infty)$ is defined as*

$$d_\phi(a, b) = \phi(b) - \phi(a) - \langle b - a, \nabla\phi(a) \rangle.$$

The optimality and exhaustive properties of BLFs were proven in "On the Optimality of Conditional Expectation as a Bregman Predictor" [Banerjee et al., 2005a]. Simply stated, the optimality property states that all BLFs fulfill the requirements of being PSRs, and the exhaustive property states that all twice differentiable PSRs are BLFs (within an additive constant).

Theorem 6 (Optimality Property of Bregman Divergence). *Let $\phi : \mathbb{R}^d \mapsto \mathbb{R}$ be a strictly convex differentiable function, and let d_ϕ be the corresponding BLF. Let Y be an arbitrary random variable taking values in \mathbb{R}^d for which both $\mathbb{E}[Y]$ and $\mathbb{E}[\phi(Y)]$ are finite, and let X be a random variable indicating an event that gives partial information about Y , then*

$$\arg \min_{P \in \mathbb{R}^d} \mathbb{E}[d_\phi(P, Y)|X] = \mathbb{E}[Y|X].$$

Theorem 7 (Exhaustive Property of Bregman Divergence). *Let $F : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ be a nonnegative function such that $F(y; y) = 0, \forall y \in \mathbb{R}^d$ for $d \geq 1$. Assume that $F(x, y)$ is twice continuously differentiable. For all random variables Y taking value in \mathbb{R}^d , if $\mathbb{E}[Y]$ is the unique minimizer of $\mathbb{E}[F(x, Y)]$ over all constants $x \in \mathbb{R}^d$, i.e.,*

$$\arg \min_{x \in \mathbb{R}^d} \mathbb{E}[F(x, Y)] = \mathbb{E}[Y],$$

then $F(x; y) = d_\phi(x, y)$ for some strictly convex and differentiable function $\phi : \mathbb{R}^d \mapsto \mathbb{R}$.

BLFs will be used through the rest of the thesis, because their definition is more suited for the derivations and theorems presented in later sections. But keep in mind that they are PSRs too.

3.2 Some Examples

In this section, some common strictly proper scoring rules / Bregman divergences are presented. The most popular of these are mean squared error and KL divergence. These BLFs are summarized in Table 1.

Squared Euclidean Distance / Mean Squared Error / Brier Score Perhaps the most well known and used of the examples BLFs. It is referred to as squared Euclidean distance when used as a divergence measure, or mean squared error/Brier score [Brier, 1950] when used as a loss measure. Refer to the example shown in Figure 10.

$$d_{BS}(\mathbf{x}, \mathbf{y}) = \|\mathbf{y} - \mathbf{x}\|^2 = \sum_{j=1}^d (y_j - x_j)^2$$

Name	Domain, S	Base Function, $\phi(x)$	Definition, $d_\phi(x, y)$
Squared Euclidean Distance	\mathbb{R}^d	\mathbf{x}^2	$\ \mathbf{y} - \mathbf{x}\ ^2$
KL-Divergence	d-simplex	$\sum_{j=1}^d x_j \log x_j$	$\sum_{j=1}^d y_j \log \frac{y_j}{x_j}$
Mahalanobis Distance	\mathbb{R}^d	$\mathbf{x}^T \Sigma \mathbf{x}$	$(\mathbf{y} - \mathbf{x})^T \Sigma (\mathbf{y} - \mathbf{x})$
Itakura-Saito Distance	$\mathbb{R}_{>0}$	$-\log x$	$\frac{y}{x} - \log \frac{y}{x} - 1$

Table 1. Example Bregman Divergences

Squared Euclidean distance has the property of symmetry, $d_\phi(\mathbf{x}, \mathbf{y}) = d_\phi(\mathbf{y}, \mathbf{x})$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, which is not required by and almost always lacking in the family of Bregman divergences. Its range is $[0, 2]$, so some choose to multiply by a coefficient of $\frac{1}{2}$ to the sum, making the range $[0, 1]$.

When the second class is implicit in the binary-class case, this special form that is often used.

$$d_{BS}(\mathbf{x}, \mathbf{y}) = (y - x)^2 + ((1 - y) - (1 - x))^2 = 2(y - x)^2$$

KL Divergence / Logarithmic Loss Kullback-Leibler divergence [Kullback and Leibler, 1951] is also very well known and used in practice. It is closely related to the concept of Shannon entropy [Shannon, 1948], since ϕ is its negative form. When used as a loss function, it is often referred to as logarithmic loss (log loss). Refer to the example shown in Figure 11.

$$d_{KL}(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^d y_j \log \frac{y_j}{x_j}$$

Note that as the prediction approaches 0 or 1, the loss goes to infinity. A practitioner should not use this loss measure if the model can output all-or-nothing predictions for a single class.

When the second class is implicit in the binary-class case, this special form that is often used.

$$d_{KL}(\mathbf{x}, \mathbf{y}) = y \log \frac{y}{x} + (1 - y) \log \frac{(1 - y)}{(1 - x)}$$

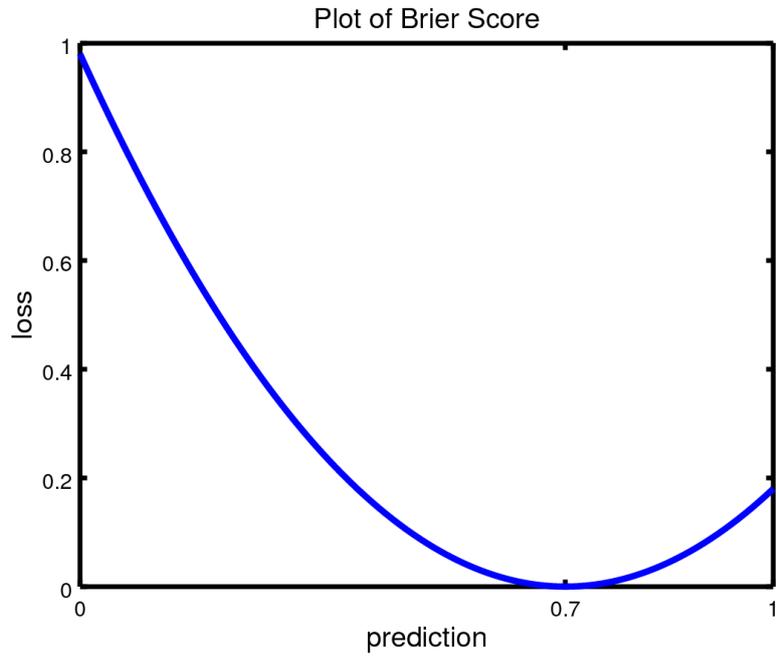


Figure 10. $d_{BS}(a, b)$ when $b_+ = 0.7$, with respect to the prediction, a_+ .

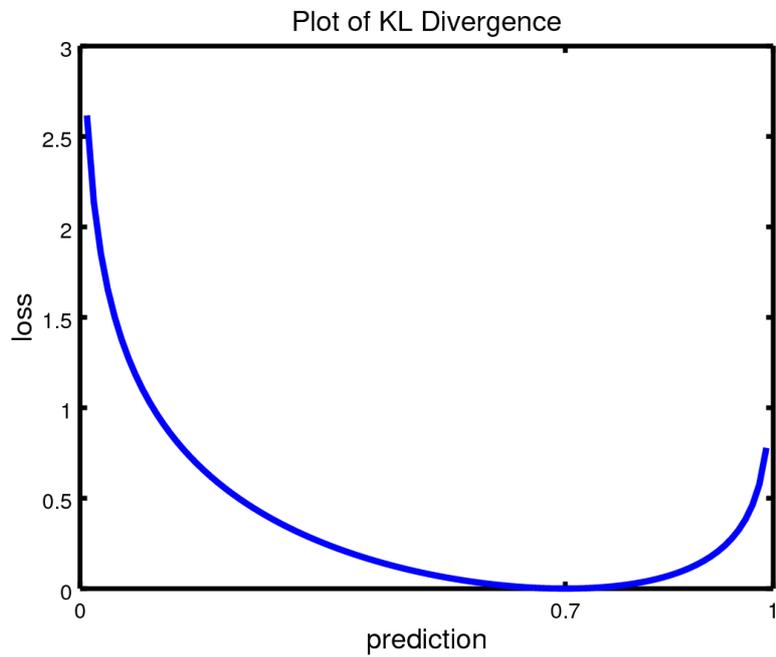


Figure 11. $d_{KL}(a, b)$ when $b_+ = 0.7$, with respect to the prediction, a_+ .

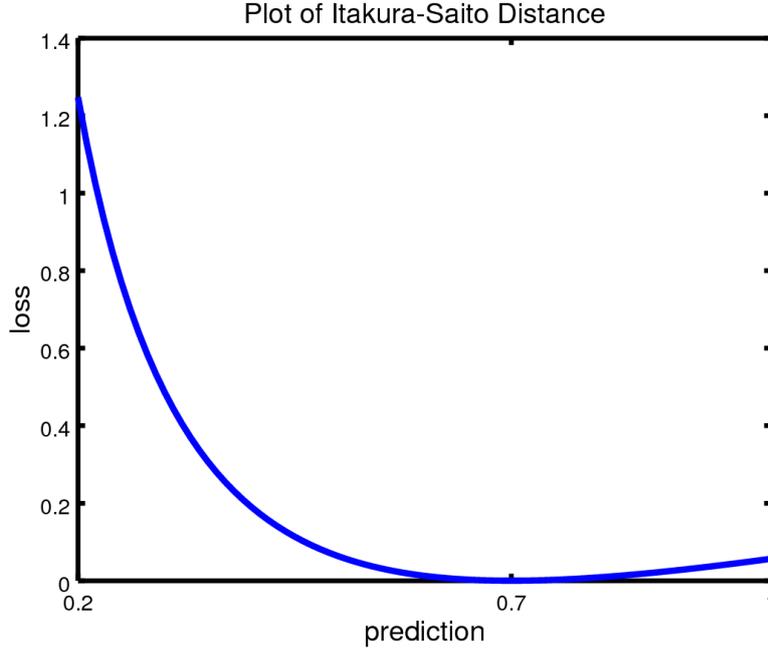


Figure 12. $d_{IS}(a, b)$ when $b_+ = 0.7$, with respect to the prediction, a_+ . The domain is only shown for values in $[0.2, 1]$, since the function grows so rapidly near $a_+ = 0$ that the concavity of the function can not be clearly seen.

Mahalanobis Distance Introduced in [Mahalanobis, 1936], Mahalanobis distance is a popular generalized distance to use in clustering, classification, and other fields of machine learning [Cayton, 2008]. Σ can be any covariance matrix. The Mahalanobis Distance can be considered a generalization of squared Euclidean distance, since they are equivalent when Σ is set to the identity matrix.

$$d_{MD}(\mathbf{x}, \mathbf{y}) = (\mathbf{y} - \mathbf{x})^T \Sigma (\mathbf{y} - \mathbf{x})$$

Itakura-Saito Distance Introduced in [Itakura, 1968], Itakura-Saito distance is frequently used in audio and speech processing [Gray et al., 1980]. Refer to the example shown in Figure 12.

$$d_{IS}(x, y) = \frac{y}{x} - \log\left(\frac{y}{x}\right) - 1$$

Note how as x approaches 0, loss goes to infinity.

3.3 Bonus Properties of Bregman Divergence

In this section, two properties of Bregman divergence are stated that are necessary deriving our general adjusters. Since they were not explicitly written out in the reviewed literature, they are included with proofs.

Proposition 8. *Given $\phi : S \mapsto \mathbb{R}$ and its corresponding d_ϕ ,*

$$\nabla_b d_\phi(a, b) = \nabla\phi(b) - \nabla\phi(a),$$

where ∇_b notates the gradient taken with respect to only the variables of b .

Proof. By the definition of Bregman divergence,

$$d_\phi(a, b) = \phi(b) - \phi(a) - \langle b - a, \nabla\phi(a) \rangle.$$

Taking ∇_b of each side and simplification gives:

$$\begin{aligned} \nabla_b d_\phi(a, b) &= \nabla_b(\phi(b) - \phi(a) - \langle b - a, \nabla\phi(a) \rangle) \\ &= \nabla_b\phi(b) - \nabla_b\phi(a) - \nabla_b\langle b - a, \nabla\phi(a) \rangle \\ &= \nabla\phi(b) - \nabla_b\langle b - a, \nabla\phi(a) \rangle \\ &= \nabla\phi(b) - \nabla_b\langle b, \nabla\phi(a) \rangle \\ &= \nabla\phi(b) - \nabla_b(b_1 \frac{\partial}{\partial a_1} \phi(a) + \dots + b_k \frac{\partial}{\partial a_k} \phi(a)) \\ &= \nabla\phi(b) - (\frac{\partial}{\partial a_1} \phi(a), \dots, \frac{\partial}{\partial a_k} \phi(a)) \\ &= \nabla\phi(b) - \nabla\phi(a) \end{aligned}$$

□

Proposition 9. *Given $d_\phi : S \times \text{ri}(S) \mapsto \mathbb{R}$, $a, b \in \text{ri}(S)$ and $c \in S$,*

$$d_\phi(a, c) - d_\phi(b, c) = \langle c - b, \nabla_b d_\phi(a, b) \rangle + d_\phi(a, b).$$

Proof. Simplifying from the definition of Bregman divergence gives:

$$\begin{aligned} d_\phi(a, c) - d_\phi(b, c) &= (\phi(c) - \phi(a) - \langle c - a, \nabla\phi(a) \rangle) - (\phi(c) - \phi(b) - \langle c - b, \nabla\phi(b) \rangle) \\ &= \phi(b) - \phi(a) + \langle c - b, \nabla\phi(b) \rangle - \langle c - a, \nabla\phi(a) \rangle \end{aligned}$$

Using the previous theorem to rewrite the third term yields:

$$\begin{aligned} &= \phi(b) - \phi(a) + \langle c - b, \nabla_b d_\phi(a, b) + \nabla\phi(a) \rangle - \langle c - a, \nabla\phi(a) \rangle \\ &= \phi(b) - \phi(a) + \langle c - b, \nabla_b d_\phi(a, b) \rangle + \langle c - b, \nabla\phi(a) \rangle - \langle c - a, \nabla\phi(a) \rangle \\ &= \phi(b) - \phi(a) + \langle c - b, \nabla_b d_\phi(a, b) \rangle - \langle b, \nabla\phi(a) \rangle + \langle a, \nabla\phi(a) \rangle \\ &= \phi(b) - \phi(a) + \langle c - b, \nabla_b d_\phi(a, b) \rangle - \langle b - a, \nabla\phi(a) \rangle \\ &= \langle c - b, \nabla_b d_\phi(a, b) \rangle + \phi(b) - \phi(a) - \langle b - a, \nabla\phi(a) \rangle \\ &= \langle c - b, \nabla_b d_\phi(a, b) \rangle + d_\phi(a, b) \end{aligned}$$

□

3.4 Decompositions

To reduce loss, it is important to understand its sources. It is well understood that a portion of loss is irreducible. Even a Bayes' optimal classifier will gain at least some small amount of loss from each instance measured by a BLF, unless the posterior probability is deterministic (exactly zero or one). It has also been shown that BLF loss can be broken into pre- and post-calibration loss [DeGroot and Fienberg, 1983]. Calibration is defined as follows.

Definition 10 (Calibrated Predictions). *Let (X, Y) be random variables representing features and labels for a k -class classification task (where $Y = (Y_1, \dots, Y_k)$ and $Y_i = 1$ for class i and 0 otherwise), and f be a probabilistic classifier. Denote a prediction by the classifier as $S = f(X)$. The prediction S is calibrated if*

$$S_j = \mathbb{E}[Y_j|S_j] \text{ for } j = 1, \dots, k.$$

A calibrated probability is one which is true at the given rate across all instances that share the same probability. Consider the task of forecasting rain one day in advance. A weather forecaster might predict a 50% chance of rain the following day. Over a number of years, that forecaster will make such a prediction many times. If in approximately 50% of all those cases it did actually rain, then those predictions are well-calibrated. If the forecaster has such a property for not just predictions of 50%, but all different probabilities, then the forecaster is a well-calibrated model.

This forecaster may not be a Bayes' optimal classifier though. Half of those predictions for a 50% chance of rain might actually have $\mathbb{P}(\text{rain}|X) = 0.7$, while the other half have $\mathbb{P}(\text{rain}|X) = 0.3$. Overall they occur 50% of the time, but they were grouped together incorrectly. This grouping loss is another source in the overall loss.

To calibrate a model, a calibration function is used which maps the probabilities output by the classifier to new calibrated probabilities. Two instances with the same original probability get mapped to the same calibrated probability. In practice, isotonic regression [Barlow, 1972] and Platt scaling [Platt et al., 1999] are the two most popular calibration procedures. They require a calibration set of data separate from the training and testing sets to calculate their mapping from uncalibrated to calibrated probabilities.

In the "Novel Decompositions" paper [Kull and Flach, 2014], the concept of "adjusted" scores was introduced. This is a "poor man's" calibration in a way. Instead of having small groups of scores that share the same value of S set equal to $\mathbb{P}(Y|S)$, adjusted predictions simply have the class distribution as their overall expected value.

Definition 11 (Adjusted Predictions). *Let (X, Y) be random variables representing features and labels for a k -class classification task (where $Y = (Y_1, \dots, Y_k)$ and*

$Y_i = 1$ for class i and 0 otherwise), f be a probabilistic classifier, and d_ϕ be a BLF. Denote a prediction by the classifier as $S = f(X)$. The prediction S is adjusted if

$$\mathbb{E}[S_j] = \mathbb{E}(Y_j) \text{ for } j = 1, \dots, k.$$

What this means is that the average prediction for each class by the classifier should match the class distribution. Not being adjusted was shown to be another source of loss, but clearly all adjusted predictions are not better than some unadjusted predictions. For instance, a constant model that always outputs the class distribution $\mathbb{P}(Y)$ is not helpful at all and in most realistic situations will have quite high expected loss. On the other hand, a reasonably trained naive Bayes classifier will likely have much better performance, even if it is unadjusted. So it raises the question of how to properly adjust a model.

The authors of that paper [Kull and Flach, 2015] proposed the following two adjustment procedures.

Definition 12 (Additive Adjustment). *Given a prediction $s \in \mathbb{R}^k$, the adjustment procedure $\alpha_+ : \mathbb{R}^k \mapsto \mathbb{R}^k$ is defined as*

$$\alpha_+(s) = (s_1 + b_1, \dots, s_k + b_k),$$

where $b = \mathbb{E}[Y] - \mathbb{E}[S]$.

Definition 13 (Multiplicative Adjustment). *Given a prediction $s \in \mathbb{R}^k$, the adjustment procedure $\alpha_* : \mathbb{R}^k \mapsto \mathbb{R}^k$ is defined as*

$$\alpha_*(s) = \left(\frac{w_1 s_1}{\sum_{j=1}^k w_j s_j}, \dots, \frac{w_k s_k}{\sum_{j=1}^k w_j s_j} \right),$$

where $w \in \mathbb{R}^k$ is a set of weights (which have been proven to exist [Kull and Flach, 2015]).

Finding the weights for multiplicative adjustment is a non-trivial task and the proposed iterative algorithm to calculate the adjusted scores often fails to converge. On top of that, it is apparent that additive adjustment can give some adjusted predictions values outside of the range $[0, 1]$, which are nonsensical values if interpreted as probabilities.

Despite these problems, additive adjustment was shown to reduce expected Brier score and and multiplicative adjustment was shown to reduce expected logarithmic loss. This is visible in Table 2. The table represents a toy dataset from a binary-classification problem. The displayed predictions are for the positive class, as the values of the negative class do not need to be explicitly stated. There are eight different possible values for X , with the probability of sampling each value

\mathbf{X}	$\mathbb{P}(\mathbf{X})$	\mathbf{S}	\mathbf{A}_+	\mathbf{C}	\mathbf{Q}
(0, 0, 0)	0.1	0.10	0.20	0.20	0.10
(0, 0, 1)	0.1	0.10	0.20	0.20	0.30
(0, 1, 0)	0.1	0.20	0.30	0.40	0.40
(0, 1, 1)	0.1	0.20	0.30	0.40	0.40
(1, 0, 0)	0.1	0.50	0.60	0.65	0.50
(1, 0, 1)	0.1	0.50	0.60	0.65	0.70
(1, 1, 0)	0.2	0.50	0.60	0.65	0.70
(1, 1, 1)	0.2	0.95	1.05	0.85	0.85
expected value		0.45	0.55	0.55	0.55
expected BS		0.2175	0.2075	0.1965	0.1915

Table 2. Toy dataset and example predictions S . A_+ represents the additive adjusted predictions, C represents the calibrated predictions, Q represents the true posterior probabilities. Expected BS indicates the expected loss from using Brier score as the loss measure.

given by $\mathbb{P}(X)$. The original predictions S become additive adjusted predictions A_+ , which become calibrated predictions C , and then finally become the Bayes' optimal predictions $Q = \mathbb{P}(Y|X)$. The expected Brier score is clearly decreasing as the predictions move closer to Q , the values of the minimum expected loss.

These two adjustment procedures can reduce expected loss because they have the property of coherence with their respective loss function. Coherent adjustment is a difficult concept, but the intuitive understanding is that α applies the same adjustment to each set of predictions with respect to the loss function, regardless of the specific predictions made.

Definition 14 (Coherent Adjustment). *Let (X, Y) be random variables representing features and labels for a k -class classification task, f be a probabilistic classifier, d_ϕ be a BLF, and α be an adjustment procedure, meaning that $\mathbb{E}[A] = \mathbb{E}[Y]$ where $A = \alpha(f(X))$. Then α is called to be coherent with d_ϕ if and only if the following quantity is a constant (not a random variable), depending on i, j only:*

$$d_\phi(A, e_i) - d_\phi(A, e_j) - d_\phi(S, e_i) + d_\phi(S, e_j) = \text{const}_{i,j}$$

where e_m is a vector of length k with 1 at position m and 0 everywhere else.

The following theorem proves that coherent adjustment reduces expected loss. More specifically, it proves that the reduction of expected loss is equal to the divergence from the original predictions to the adjusted predictions. It can be said that the loss of the original predictions can be decomposed by the adjusted predictions. This is also true for calibrated predictions and Bayes' optimal predictions.

Theorem 15 (Decomposition of BLFs [Kull and Flach, 2015]). *Let (X, Y) be random variables representing features and labels for a k -class classification task (where $Y = (Y_1, \dots, Y_k)$ and $Y_i = 1$ for class i and 0 otherwise), f be a probabilistic classifier, d_ϕ be a BLF, and α be an adjustment procedure coherent with d_ϕ . Denote $S = f(X)$, $A = \alpha(S)$, $C_j = \mathbb{E}[Y_j|S]$, and $Q_j = \mathbb{E}[Y_j|X]$ for $j = 1, \dots, k$. Then for any subsequence V_1, V_2, V_3 of the random variables S, A, C, Q, Y the following holds:*

$$\mathbb{E}[d_\phi(V_1, V_3)] = \mathbb{E}[d_\phi(V_1, V_2)] + \mathbb{E}[d_\phi(V_2, V_3)]$$

If a practitioner wants to reduce loss and is trying to decide between adjusting or calibrating their predictions, the obvious choice is still calibration. From Theorem 15, it is clear that calibration reduces loss more than adjustment, and from their definitions we know a calibrated dataset is also adjusted. But the previously mentioned calibration methods need a sizeable sample of data to work. This might not be very easily attainable for a practitioner, especially in instances of dataset shift, but getting a roughly approximate class distribution estimate might be much easier. That is all that is required for adjustment.

There are remaining problems with adjustment though. ① The previously proposed algorithm for multiplicative adjustment is unreliable, ② the only BLFs with known coherent adjustment procedures are Brier score and logarithmic loss, ③ and additive adjustment's results are not interpretable as probabilities.

In the next section, a novel adjustment procedure called unbounded general adjustment (UGA) is introduced which solves the first two problems. It provides a coherent adjustment for all BLFs, and is equivalent with additive and multiplicative adjustment for Brier score and logarithmic loss, respectively. Implementation is possible with convex optimization, allowing adjusted scores to be calculated efficiently and reliably. Afterwards, bounded general adjustment (BGA) is introduced which solves the third problem. It lacks coherence, so it can not provide decomposable results like in Theorem 15, but it keeps predictions in the $[0, 1]$ bounds and reduces expected loss at least as much as UGA.

4 Adjusting Generalized

In the previous section, adjustment was introduced and coherent adjustment was defined. It has been shown that a coherent adjustment will reduce expected loss. It has only been shown that coherent adjustments exist for two proper scoring rules, additive adjustment for Brier score and multiplicative adjustment for logarithmic loss. Only the algorithm for additive adjustment is reliable, as the previously proposed [Kull and Flach, 2015] iterative algorithm to compute the multiplicative adjustment often fails to converge. In addition, additive adjustment results can exist outside the $[0, 1]$ bounds that probabilities should exist in.

In Section 4.1, unbounded general adjustment (UGA) will be introduced, which is equivalent to additive and multiplicative adjustment when used in both of those settings. It is also shown that only one set of predictions is an output of coherent adjustment for a given set of predictions. In Section 4.2, bounded general adjustment (BGA) is introduced. This adjustment is not coherent for most BLFs, but its reduction of expected loss is higher than UGA in the cases where it is not coherent. In addition, BGA is guaranteed to give output that is in the range $[0, 1]$ and is therefore interpretable as probabilities. In Section 4.3, implementing the UGA and BGA functions is shown to be a convex optimization problem and easily solvable by standard convex optimizers.

4.1 Unbounded General Adjustment

Denote the output of a probabilistic classifier on a dataset as $p \in \mathbb{R}^{n \times k}$, which represents a set of k predictions for each of the n instances which can be represented as a matrix. Each instance $i \in [1, \dots, n]$ has a matching row p_i , where each entry $p_{i,j}$ is a real number indicating the probability of that instance being class $j \in [1, \dots, k]$.

In this framework, the goal of the classifier is to output $p = q$, where q is the corresponding set of true posterior probabilities, minimizing our expected loss. At first, without any additional information given, the value of q is completely unknown, so we can represent the possible values as the set $\mathbb{R}^{n \times k}$. This set of possible values can be restricted with equality constraints ensuring that all rows in any possible $q' \in \mathbb{R}^{n \times k}$ should sum to one: $\sum_{j=1}^k q'_{i,j} = 1 \forall i \in [1, \dots, n]$. Denote this restricted set as Q . The reader might think at this point that inequality constraints should be added so that every entry/probability in each possible $q' \in Q$ is between 0 and 1. Ignore this requirement for now, since these constraints keep UGA from being coherent. This requirement will be added when we introduce BGA in the next subsection.

Denote the column averages/class distribution of q to be $\pi \in \mathbb{R}^k$, which is the only additional piece of information needed to do any kind of adjustment. To consider only values in Q that are adjusted to π , add another equality constraint

ensuring any possible $q' \in Q$ has a class distribution of π : $\frac{1}{n} \sum_{i=1}^n q'_{i,j} = \pi_j \forall j \in [1, \dots, k]$. Denote this subset of Q as Q^* .

The goal of a coherent adjuster, is to find an $a \in Q^*$ such that $\sum_{i=1}^n (d_\phi(p_i, a_i) + d_\phi(a_i, q'_i)) = \sum_{i=1}^n d_\phi(p_i, q'_i) \forall q' \in Q^*$. This means that, for all possible values of $q' \in Q^*$, the loss decomposes from p to a and from a to q' . The question becomes, does there exist a point $a \in Q^*$ that meets this requirement for any given p , π , and d_ϕ . If so, what is the procedure to find it? The answers to these questions are not obvious, as one could suspect that for all values of a , there could exist a value of q' such that the original p would be closer to q' than the adjusted a is. And even if the former question is true, there is little indication of how it would be calculated.

What Theorem 17 shows is that an adjusted point, $a^* \in Q^*$, exists for any given p , π , and d_ϕ , such that a^* satisfies the above requirement. The adjustment procedure, α^* , used to calculate this value is termed unbounded general adjustment (UGA) and is defined as follows.

Definition 16 (Unbounded General Adjuster (UGA)). *Let $d_\phi : \text{ri}(S) \times S \mapsto [0, \text{inf})$ be our Bregman divergence, p be a set of predictions, and π is the class distribution. Given these we define our unbounded generalized adjustment function $\alpha^* : \mathbb{R}^{n \times k} \times \mathbb{R}^k \mapsto \mathbb{R}^{n \times k}$:*

$$\alpha^*(p, \pi) = \arg \min_{a \in \mathbb{R}^{n \times k}} \sum_{i=1}^n d_\phi(p_i, a_i) \text{ s.t. } \sum_{j=1}^k a_{i,j} = 1 \forall i \in [1, \dots, n]$$

$$\frac{1}{n} \sum_{i=1}^n a_{i,j} = \pi_j \forall j \in [1, \dots, k]$$

also written as

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

where $Q^* = \{a \in \mathbb{R}^{n \times k} \mid \sum_{j=1}^k a_{i,j} = 1 \forall i \in [1, n] \text{ and } \frac{1}{n} \sum_{i=1}^n a_{i,j} = \pi_j \forall j \in [1, k]\}$.

UGA finds the point in Q^* with the minimum average divergence from p . In a way a^* is the projection of p to closest point (measured by d_ϕ) on the set Q^* . Figure 13 gives a visual explanation of the relationship.

The following theorem guarantees that a^* is indeed the point that meets the above requirement. It says that UGA guarantees an expected reduction of divergence equal to the expected divergence from the original predictions to the adjusted predictions.

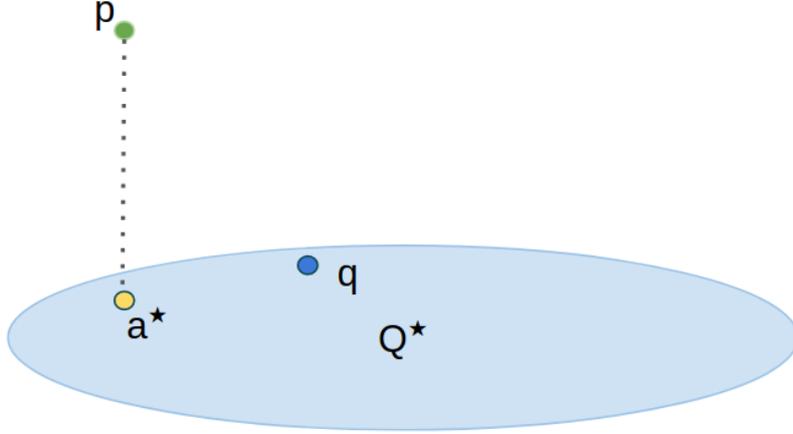


Figure 13. Graphical representation of α^* mapping a set of predictions p to UGA predictions a^* , the "nearest" point in Q^* .

Theorem 17. Given $p \in P$ and π . Let $a^* = \alpha(p, \pi)$.

$$\sum_{i=1}^n (d_\phi(p_i, q_i) - d_\phi(a_i^*, q_i)) = \sum_{i=1}^n (d_\phi(p_i, a_i^*))$$

Proof. From Proposition 9, we can write

$$\sum_{i=1}^n (d_\phi(p_i, q_i) - d_\phi(a_i^*, q_i)) = \sum_{i=1}^n (\langle q_i - a_i^*, \nabla_{a_i^*} d_\phi(p_i, a_i^*) \rangle + d_\phi(p_i, a_i^*))$$

If we can prove that

$$\sum_{i=1}^n \langle q_i - a_i^*, \nabla_{a_i^*} d_\phi(p_i, a_i^*) \rangle = 0$$

then the proof will be complete. So we begin by using the method of Lagrange multipliers to define what each $\nabla_{a_i^*} d_\phi(p_i, a_i^*)$ is for each $i \in [1, n]$. We rewrite the function α^* that solves its minimization problem as a Lagrangian function, F . Keep note our Lagrangian function will have $n \times k$ variables from a , n variables from our first constraint, and k variables from our second constraint. These are input to the function as a vector.

$$F(a, \theta, \lambda) = \sum_{i=1}^n d_\phi(p_i, a_i) + \sum_{i=1}^n \theta_i (1 - \sum_{j=1}^k a_{i,j}) + \sum_{j=1}^k \lambda_j (n\pi_j - \sum_{i=1}^n a_{i,j})$$

In this form, $a = a^*$ when

$$\nabla F(a, \theta, \lambda) = \mathbf{0}.$$

Let's expand the gradient.

$$\nabla F(a, \theta, \lambda) = (\nabla_a F(a, \theta, \lambda), \nabla_\theta F(a, \theta, \lambda), \nabla_\lambda F(a, \theta, \lambda))$$

Let's expand the first term. For simplicity's sake, we will represent ∇_a as a matrix, so we can show the partial derivatives in a way that mimics the variables location on the $n \times k$ matrix of a .

$$\begin{aligned} \nabla_a F(a, \theta, \lambda) &= \begin{bmatrix} \frac{\partial}{\partial a_{1,1}} F(a, \theta, \lambda) & \dots & \frac{\partial}{\partial a_{1,k}} F(a, \theta, \lambda) \\ \dots & \dots & \dots \\ \frac{\partial}{\partial a_{n,1}} F(a, \theta, \lambda) & \dots & \frac{\partial}{\partial a_{n,k}} F(a, \theta, \lambda) \end{bmatrix} \\ &= \begin{bmatrix} -\theta_1 + -\lambda_1 + \frac{\partial}{\partial a_{1,1}} d_\phi(p_1, a_1) & \dots & -\theta_1 + -\lambda_k + \frac{\partial}{\partial a_{1,k}} d_\phi(p_1, a_1) \\ \dots & \dots & \dots \\ -\theta_n + -\lambda_1 + \frac{\partial}{\partial a_{n,1}} d_\phi(p_n, a_n) & \dots & -\theta_n + -\lambda_k + \frac{\partial}{\partial a_{n,k}} d_\phi(p_n, a_n) \end{bmatrix} \end{aligned}$$

We can now see that for each entry (i, j) in $\nabla_a F(a, \theta, \lambda)$ to equal 0 at the minimum $a = a^*$, then the following is true.

$$\frac{\partial}{\partial a_{i,j}} d_\phi(p_i, a_i^*) = \theta_i + \lambda_j$$

$$\nabla_{a_i} d_\phi(p_i, a_i^*) = (\theta_i + \lambda_1, \dots, \theta_i + \lambda_k)$$

We can write out

$$\begin{aligned} \sum_{i=1}^n \langle q_i - a_i^*, \nabla_{a_i^*} d_\phi(p_i, a_i^*) \rangle &= \sum_{i=1}^n \sum_{j=1}^k (q_{i,j} - a_{i,j}^*) (\theta_i + \lambda_j) \\ &= \sum_{i=1}^n \sum_{j=1}^k (q_{i,j} - a_{i,j}^*) \theta_i + \sum_{i=1}^n \sum_{j=1}^k (q_{i,j} - a_{i,j}^*) \lambda_j \\ &= \sum_{i=1}^n \theta_i \sum_{j=1}^k (q_{i,j} - a_{i,j}^*) + \sum_{j=1}^k \lambda_j \sum_{i=1}^n (q_{i,j} - a_{i,j}^*) \end{aligned}$$

We know from the constraints that each row and column of $q - a^*$ sums to 0. So it's clear that

$$\sum_{i=1}^n \theta_i \sum_{j=1}^k (q_{i,j} - a_{i,j}^*) + \sum_{j=1}^k \lambda_j \sum_{i=1}^n (q_{i,j} - a_{i,j}^*) = 0.$$

□

Theorem 18 shows that UGA satisfies the conditions of coherence presented in Definition 14. This means that it works in the decomposition framework presented in the previous section.

Theorem 18. *Let $d_\phi : \text{ri}(S) \times S \mapsto [0, \infty)$ be a Bregman divergence and let α^* be the unbounded general adjuster corresponding to d_ϕ . Then α^* is coherent with d_ϕ .*

Proof. For an adjustment procedure to be coherent, the following equation must be satisfied.

$$d_\phi(a_x, e_i) - d_\phi(a_x, e_j) - d_\phi(p_x, e_i) + d_\phi(p_x, e_j) = \text{const}_{i,j}$$

We can just use the definition of divergence and properties of vectors to get the equation into a new form.

$$\begin{aligned} \text{const}_{i,j} &= d_\phi(a_x, e_i) - d_\phi(a_x, e_j) - d_\phi(p_x, e_i) + d_\phi(p_x, e_j) \\ &= \phi(e_i) - \phi(a_x) - \langle e_i - a_x, \nabla\phi(a_x) \rangle - \phi(e_j) + \phi(a_x) - \langle e_j - a_x, \nabla\phi(a_x) \rangle - \\ &\quad \phi(e_i) + \phi(p_x) - \langle e_i - p_x, \nabla\phi(p_x) \rangle + \phi(e_j) - \phi(p_x) - \langle e_j - p_x, \nabla\phi(p_x) \rangle \\ &= \langle e_j - a_x, \nabla\phi(a_x) \rangle - \langle e_i - a_x, \nabla\phi(a_x) \rangle - \langle e_j - p_x, \nabla\phi(p_x) \rangle + \langle e_i - p_x, \nabla\phi(p_x) \rangle \\ &= \langle e_j - e_i, \nabla\phi(a_x) \rangle - \langle e_j - e_i, \nabla\phi(p_x) \rangle \\ &= \langle e_j - e_i, \nabla\phi(a_x) - \nabla\phi(p_x) \rangle \end{aligned}$$

From Proposition 8, we know

$$\langle e_j - e_i, \nabla\phi(a_x) - \nabla\phi(p_x) \rangle = \langle e_j - e_i, \nabla_{a_x} d_\phi(p_x, a_x) \rangle.$$

We know from the proof of Theorem 17 that $\nabla_{a_x^*} d_\phi(p_x, a_x^*)$ is defined by the sum of two variables that depend only on i and j , θ and λ . That means $\text{const}_{i,j} = \langle e_j - e_i, \nabla_{a_x^*} d_\phi(p_x, a_x^*) \rangle$ only depends on i and j matching the definition of coherence. \square

The following theorem proves that only one coherent adjusted point exists for any given p , π , and d_ϕ . This implies that the results of UGA used with Brier score will equal the results of additive adjustment, and the results of UGA with logarithmic loss will equal the results of multiplicative adjustment.

Theorem 19. *Let $d_\phi : \text{ri}(S) \times S \mapsto [0, \infty)$ be a Bregman divergence, let p be a set of predictions, let π be a class distribution, and let a be a set of adjusted predictions such that for any $q \in Q_\pi$ the following holds:*

$$\sum_{i=1}^n (d_\phi(p_i, q_i) - d_\phi(a_i, q_i)) = \sum_{i=1}^n (d_\phi(p_i, a_i))$$

Then $a = \alpha^(p, \pi)$.*

Proof. Assume that $a \neq \alpha^*(p, \pi)$ and take the case where $q = \alpha^*(p, \pi)$. We can rewrite the theorem's equality to

$$\sum_{i=1}^n d_\phi(p_i, q_i) = \sum_{i=1}^n (d_\phi(p_i, a_i) + d_\phi(a_i, q_i)).$$

We know that $\sum_{i=1}^n d_\phi(p_i, q_i) < \sum_{i=1}^n d_\phi(p_i, a_i)$ by the definition of α^* and $\sum_{i=1}^n d_\phi(a_i, q_i) > 0$ by the definition of Bregman divergence. Therefore, $\sum_{i=1}^n d_\phi(p_i, q_i) < \sum_{i=1}^n (d_\phi(p_i, a_i) + d_\phi(a_i, q_i))$, giving us a contradiction. \square

In this subsection, it has been shown that a unique coherent adjustment exists for every BLF and that this adjustment is UGA.

4.2 Bounded General Adjustment

In the previous subsection it was proved that coherent adjustment is possible for all BLFs, and that it can be considered a minimization problem. But for many BLFs, such as Brier score, Itakura-Saito distance, and Mahalanobis distance, the resulting scores can be nonsensical if interpreted as probabilities, as the probabilities will sometimes be outside the $[0, 1]$ bounds. This might be OK, if the practitioner simply wants to reduce BLF loss and doesn't want to interpret the results as probabilities. If that is the case though, then using a PSR/BLF is not necessary and may not be the best choice of a loss function. If the practitioner does want interpretable probabilities, then the results of UGA need to be improved. This motivates bounded general adjustment (BGA).

For a given class distribution π , let us constrain the set of all possible adjusted predictions Q^* further, by requiring that all probabilities are non-negative: $Q^\circ = \{a \in Q^* \mid a_{i,j} \geq 0 \forall (i, j) \in [1, n] \times [1, k]\}$ or $a_{i,j} \geq 0 \forall (i, j) \in [1, n] \times [1, k]$. Denote the Q with this inequality constraints and the original two equality constraints as Q° . We now propose our BGA procedure.

Definition 20 (Bounded General Adjustment (BGA)). *Let $d_\phi : \text{ri}(S) \times S \mapsto [0, \infty)$ be our Bregman divergence, p be a set of predictions, and π is the class distribution. Given these we define our bounded generalized adjustment function $\alpha^\circ : [0, 1]^{n \times k} \times [0, 1]^k \mapsto [0, 1]^{n \times k}$:*

$$\begin{aligned} \alpha^\circ(p, \pi) = \arg \min_a \sum_{i=1}^n d_\phi(p_i, a_i) \quad \text{s.t.} \quad & \sum_{j=1}^k a_{i,j} = 1 \quad \forall i \in [1, \dots, n] \\ & \frac{1}{n} \sum_{i=1}^n a_{i,j} = \pi_j \quad \forall j \in [1, \dots, k] \\ & a_{i,j} \geq 0 \quad \forall (i, j) \in [1, \dots, n] \times [1, \dots, k] \end{aligned}$$

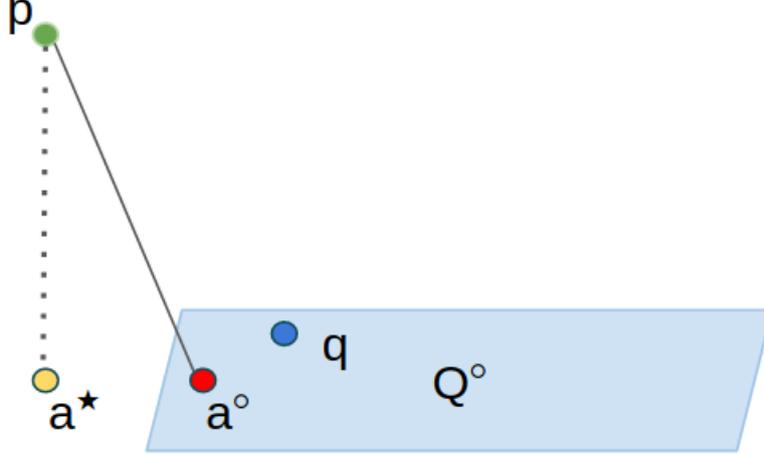


Figure 14. Graphical representation of α° mapping a set of predictions p to UGA predictions a° , the "nearest" point in Q° . Q° is sharply bounded in $[0, 1]^{n \times k}$ and is therefore smaller than Q^* in Figure 13. In this case a^* is outside the bounds of Q° , so a° is more "distant".

also written as

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

This adjuster is not coherent, but it is, from a practical viewpoint, even better than UGA. Figure 14 gives a visual explanation of the relationship. Theorem 22 proves that the reduction of expected loss by BGA is equal to or greater than that by UGA. To prove that theorem, the following lemma is needed.

Lemma 21. *Let $a^* = \alpha_{\text{bounded}}(p, \pi)$. Then*

$$\sum_{i=1}^n \langle q_i - a_i^*, \nabla_a^* d_\phi(p_i, a_i^*) \rangle \geq 0.$$

Proof. This is pretty much like the proof of Theorem 17, except we use the Karush-Kuhn-Tucker method instead of the method of Lagrange multipliers. We write our KKT function as follows, using the ψ variables to represent the inequality constraints.

$$F(a, \theta, \lambda, \psi) = \sum_{i=1}^n d_\phi(p_i, a_i) + \sum_{i=1}^n \theta_i (1 - \sum_{j=1}^k a_{i,j}) + \sum_{j=1}^k \lambda_j (n\pi_j - \sum_{i=1}^n a_{i,j}) + \sum_{i=1}^n \sum_{j=1}^k \psi_{i,j} (-a_{i,j})$$

The minimum where $a = a^\circ$ is when

$$\nabla F(a, \theta, \lambda, \psi) = \mathbf{0}.$$

Let's expand the gradient.

$$\nabla F(a, \theta, \lambda, \psi) = (\nabla_a F(a, \theta, \lambda), \nabla_\theta F(a, \theta, \lambda), \nabla_\lambda F(a, \theta, \lambda), \nabla_\psi F(a, \theta, \lambda))$$

Let's expand the first term. Like previously, we will represent ∇_a as a matrix.

$$\begin{aligned} \nabla_a F(a, \theta, \lambda) &= \begin{bmatrix} \frac{\partial}{\partial a_{1,1}} F(a, \theta, \lambda) & \dots & \frac{\partial}{\partial a_{1,k}} F(a, \theta, \lambda) \\ \dots & \dots & \dots \\ \frac{\partial}{\partial a_{n,1}} F(a, \theta, \lambda) & \dots & \frac{\partial}{\partial a_{n,k}} F(a, \theta, \lambda) \end{bmatrix} \\ &= \begin{bmatrix} -\theta_1 - \lambda_1 - \psi_{1,1} + \frac{\partial}{\partial a_{1,1}} d_\phi(p_1, a_1) & \dots & -\theta_1 - \lambda_k - \psi_{1,k} + \frac{\partial}{\partial a_{1,k}} d_\phi(p_1, a_1) \\ \dots & \dots & \dots \\ -\theta_n - \lambda_1 - \psi_{n,1} + \frac{\partial}{\partial a_{n,1}} d_\phi(p_n, a_n) & \dots & -\theta_n - \lambda_k - \psi_{n,k} + \frac{\partial}{\partial a_{n,k}} d_\phi(p_n, a_n) \end{bmatrix} \end{aligned}$$

We can now see that for each entry (i, j) in $\nabla_a F(a, \theta, \lambda, \psi)$ to equal 0, the following must be true.

$$\begin{aligned} \frac{\partial}{\partial a_{i,j}} d_\phi(p_i, a_i) &= \psi_{i,j} + \theta_i + \lambda_j \\ \nabla_{a_i} d_\phi(p_i, a_i^*) &= (\psi_{i,1} + \theta_i + \lambda_1, \dots, \psi_{i,k} + \theta_i + \lambda_k) \end{aligned}$$

We can write out

$$\begin{aligned} \sum_{i=1}^n \langle q_i - a_i^*, \nabla_{a_i^*} d_\phi(p_i, a_i^*) \rangle &= \sum_{i=1}^n \sum_{j=1}^k (q_{i,j} - a_{i,j}^*) (\psi_{i,j} + \theta_i + \lambda_j) \\ &= \sum_{i=1}^n \sum_{j=1}^k \psi_{i,j} (q_{i,j} - a_{i,j}^*) + \sum_{i=1}^n \sum_{j=1}^k (q_{i,j} - a_{i,j}^*) (\theta_i) + \\ &\quad \sum_{i=1}^n \sum_{j=1}^k (q_{i,j} - a_{i,j}^*) (\lambda_j) \end{aligned}$$

We know from the earlier proof of Theorem 17 that the last two terms equal 0, which leaves us

$$\sum_{i=1}^n \langle q_i - a_i^*, \nabla_{a_i^*} d_\phi(p_i, a_i^*) \rangle = \sum_{i=1}^n \sum_{j=1}^k \psi_{i,j} (q_{i,j} - a_{i,j}^*)$$

Now let's look at what each $\psi_{i,j}$ actually is. The KKT conditions require that each $\psi_{i,j} \geq 0$ and that $\psi_{i,j}(-a_{i,j}) = 0$. This implies that the only times that $\psi_{i,j} \neq 0$ is

when $a_{i,j} = 0$ in which case $\psi_{i,j} \geq 0$.

In our double sum, we only have to be concerned with the terms that have an $a_{i,j} = 0$ since all the other terms will be 0 since $\psi_{i,j}$ would be 0. $q_{i,j} - a_{i,j} > 0$ in these cases since $q_{i,j} \geq 0$ by the constraint. $q_{i,j} - a_{i,j} \geq 0$ and $\psi_{i,j} \geq 0$, so $\sum_{i=1}^n \sum_{j=1}^k \psi_{i,j}(q_{i,j} - a_{i,j}^*) \geq 0$. \square

Theorem 22. *Let $d_\phi : \text{ri}(S) \times S \mapsto [0, \text{inf})$ be our Bregman divergence, let p be a set of predictions, let π be the class distribution that defines Q° , let $a^* = \alpha^*(p, \pi)$, and let $a^\circ = \alpha^\circ(p, \pi)$. Then for any $q \in Q^\circ$ the following holds:*

$$\sum_{i=1}^n (d_\phi(p_i, q_i) - d_\phi(a^\circ, q_i)) \geq \sum_{i=1}^n d_\phi(p_i, a_i^\circ) \geq \sum_{i=1}^n d_\phi(p_i, a_i^*) = \sum_{i=1}^n (d_\phi(p_i, q_i) - d_\phi(a^*, q_i))$$

Proof. Writing out the difference and using Lemma 21 gives us

$$\begin{aligned} \sum_{i=1}^n (d_\phi(p_i, q_i) - d_\phi(a_i^\circ, q_i)) &= \sum_{i=1}^n (\langle a_i^\circ - q_i, \nabla_a d_\phi(p_i, a_i^\circ) \rangle + d_\phi(p_i, a_i^\circ)) \\ &= \sum_{i=1}^n \langle a_i^\circ - q_i, \nabla_a d_\phi(p_i, a_i^\circ) \rangle + \sum_{i=1}^n d_\phi(p_i, a_i^\circ) \\ &\geq \sum_{i=1}^n d_\phi(p_i, a_i^\circ) \end{aligned}$$

We know $\sum_{i=1}^n d_\phi(p_i, a_i^\circ) \geq \sum_{i=1}^n d_\phi(p_i, a_i^*)$ since a^* and a° are either equal or a° would have been chosen over a^* in α^* 's minimization task.

The equality at the end comes from Theorem 17. \square

This theorem says that our adjustment function guarantees an expected reduction of divergence and that reduction will be at least as much as $\sum_{i=1}^n d_\phi(p_i, a_i^\circ)$. This actually produces even a better reduction in loss than UGA.

Consider Table 3, an extension of the earlier Table 2. The predictions adjusted by BGA using Brier score as the loss function are marked by $\mathbf{A}_{\circ,BS}$. Agreeing with Theorem 22, the expected loss of $\mathbf{A}_{\circ,BS}$ is less than that of \mathbf{A}_+ , which is adjusted by UGA with Brier score/additive adjustment.

4.3 Implementing Adjustment

The previous two sections make strong guarantees about correctness of the results of the minimization functions, but global optimization problems can often be intractable in practice. Luckily, both UGA and BGA happen to be convex minimization problems, a subset of optimization problems that are computationally

\mathbf{X}	$\mathbb{P}(\mathbf{X})$	\mathbf{S}	\mathbf{A}_+	$\mathbf{A}_{\circ,BS}$	\mathbf{C}	\mathbf{Q}
(0, 0, 0)	0.1	0.10	0.20	0.2125	0.20	0.10
(0, 0, 1)	0.1	0.10	0.20	0.2125	0.20	0.30
(0, 1, 0)	0.1	0.20	0.30	0.3125	0.40	0.40
(0, 1, 1)	0.1	0.20	0.30	0.3125	0.40	0.40
(1, 0, 0)	0.1	0.50	0.60	0.6125	0.65	0.50
(1, 0, 1)	0.1	0.50	0.60	0.6125	0.65	0.70
(1, 1, 0)	0.2	0.50	0.60	0.6125	0.65	0.70
(1, 1, 1)	0.2	0.95	1.05	1.0000	0.85	0.85
expected value		0.45	0.55	0.55	0.55	0.55
expected BS		0.2175	0.2075	0.2031	0.1965	0.1915

Table 3. Updated toy dataset and example predictions from Table 2. Note that $A_{\circ,BS}$ has been added representing the predictions adjusted by BGA with respect to Brier score. Their expected loss is lower than that of A_+ .

efficient and feature only one local minimum so the correctness of the output is assured. To be a convex optimization problem, both the constrained domain/search space and the objective function have to be convex [Boyd and Vandenberghe, 2004].

Q^* is defined by linear equality constraints, so it is a convex search space. Q° has the same equality constraints as Q^* and an extra set of inequality constraints, which forces Q° to be within $[0, 1]^{n \times k}$. This can be represented as the intersection of Q^* and $[0, 1]^{n \times k}$. Because $[0, 1]^{n \times k}$ is also clearly convex and convexity is preserved in the intersection of convex sets, Q° is a convex set too.

For UGA and BGA, both are minimizing the same function, the sum of Bregman divergences with constant first arguments. This sum is a convex function because positive weighted sums of convex functions preserve convexity and each individual Bregman divergence is convex with regard to their second argument [Banerjee et al., 2005b]. So the second requirement of having a convex objective function is satisfied for both of them.

UGA only has equality constraints, so Newton’s method works fine with it. For Brier score UGA, additive adjustment (Definition 12) is its closed form solution.

BGA computations are a little more difficult since they involve inequality constraints, making interior point methods necessary [Boyd and Vandenberghe, 2004]. Interior point methods involve transforming the objective using a soft

indicator function that acts as the barrier defined by the inequality constraints. This modified function can then be solved for by applying Newton's method.

It's worth noting that KL divergence is naturally bounded in $[0, 1]$, since KL divergence goes to infinity at those bounds. This means BGA is equivalent to UGA for KL divergence, and Newton's methods can be used directly instead of interior point methods.

5 Experimental Results

The theorems in Section 4 gave promising results, but they only hold when the correct class distribution is known. In practice, having such knowledge is the exception, not the rule. To demonstrate that BGA works in a real world setting, a series of experiments were run.

In Section 5.1, a description of the experimental setup is given. In Section 5.2, the results of the experiments are presented and discussed. The results compare the performance of BGA for Brier score and for log loss against each other, and against PPA, another adjustment procedure introduced in Definition 3.

5.1 Setup

The experiments began by downloading the datasets from OpenML [Vanschoren et al., 2013] using their R library API [R Core Team, 2015, Casalicchio et al., 2017, RStudio Team, 2016]. Both binary and multiclass datasets were downloaded with corresponding user-submitted sets of predictions. Tasks were restricted to have a number of instances in the interval of [2000, 1000000] and with 8 or fewer classes. Models that only made all-or-nothing predictions (a probability of 1 for one of the classes) were discarded, and predictions of 0 were changed to 0.00001 and predictions of 1 were changed to 0.99999 for the remaining models, so they could be used to measure logarithmic loss too. Sometimes an error was thrown while downloading, in which case that file was skipped. Each set of predictions was divided by their cross-validation fold and each fold was saved in its own file. The first 1500 by lexicographical order were used in the experiments.

Every combination of three different shifting methods were then applied at four different rates: 10%, 30%, 50%, and 70%. We performed the shift on either the majority class or, in the case of no single majority class present, the two or more largest classes that made up a majority.

The first method imitates prior probability shift by undersampling the majority class(es) by the given rate. The second method imitates a variety of concept shift by flipping the class label of the majority class(es) to the minority class(es) by the given rate. The third method imitates covariate shift by undersampling the overall data set based on a real-valued feature. To encourage this to result in a notably changed class distribution, the Pearson correlation coefficient between each real-valued feature and each of the majority classes was measured. Then, the data was sorted on the feature with the largest coefficient and the beginning instances were skipped by the given rate. When two of the above methods are combined it produces an other type of shift.

The resulting set of predictions and shifted labels had their average Brier score and log loss measured. If the pre-adjusted set of predictions had a Brier score or

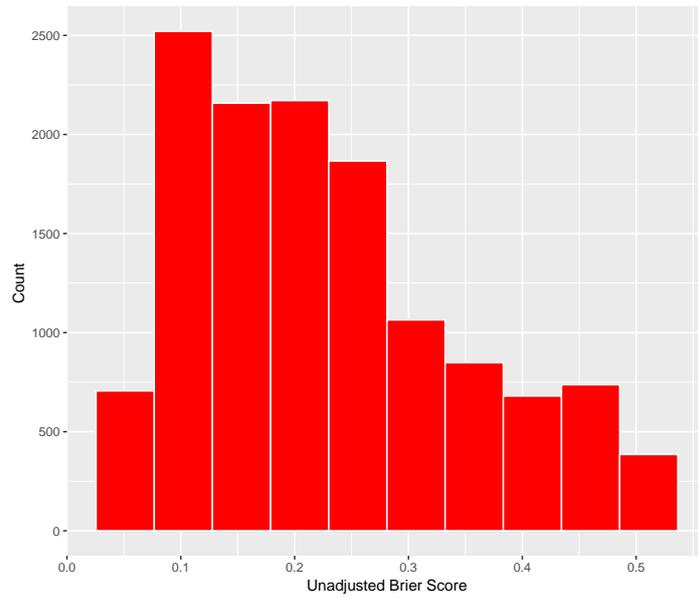


Figure 15. Histogram showing the distribution of the pre-adjusted Brier scores of the valid set of predictions.

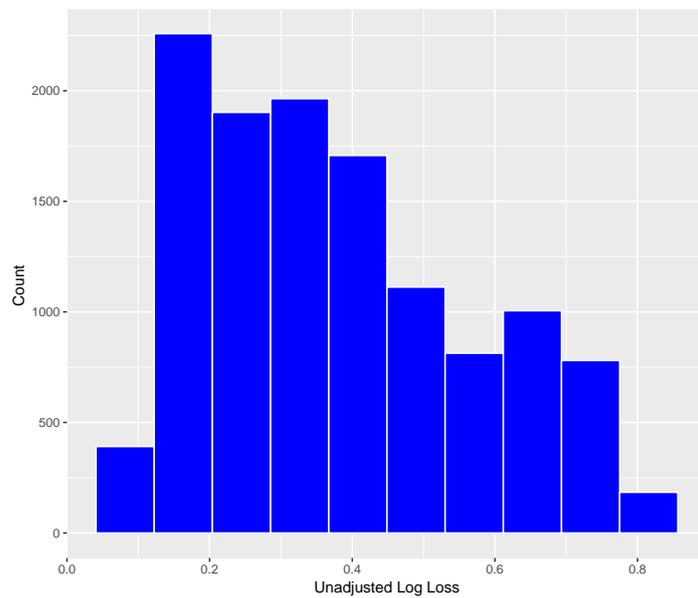


Figure 16. Histogram showing the distribution of the pre-adjusted log losses of the valid set of predictions.

log loss of under 0.02 then they were discarded, since adjustment can not help much if the model is already high-performing. If the Brier score was above 0.5 or the log loss was above 0.8, then those predictions were also discarded for being too poor. If mean squared error between the original class distribution and the shifted class distribution was under 0.02 then those predictions were also discarded for not shifting enough. The estimated class distribution could not put any class under 0.1% or above 99.9% Also, sets of predictions with less than 200 instances were discarded.

Then, the models were adjusted using BGA with Brier Score, BGA with log loss, and PPA, with varying error rates in the new class distribution estimate: 0.00, 0.01, 0.02, 0.04, 0.06, and 0.08. The majority class estimations were increased by the error rate, while the minority class estimations shared equally in decreasing so the estimate summed to one. With each error-induced class distribution, the predictions were adjusted using the three above-mentioned adjusters. BGA was implemented using the CVXPY library [Diamond and Boyd, 2016]. For both Brier score and log loss, the resulting losses were recorded.

About the Brier Score Experiment Data Overall, there were 13135 sets of predictions used for the experiments judged by Brier score. These are from the various shifts performed on 1228 models from 204 separate datasets. By adding the different error rates, 76096 data points were collected. Figure 15 shows the histogram of the datasets binned by the value of their pre-adjusted Brier score.

About the Log Loss Experiment Data Overall, there were 12116 sets of predictions used for the experiments judged by log loss. These are from the various shifts performed on 1039 models from 176 separate datasets. By adding the different error rates, 70383 data points were collected. Figure 16 shows the histogram of the datasets binned by the value of their pre-adjusted log loss.

5.2 Results

Three groups of experiments were run. The first was to check if using BGA for the corresponding loss function is indeed the best choice in practice. The theory work suggests that improvement is only guaranteed when BGA is used for the corresponding loss function, but maybe in practice one adjuster tends to be better than the other? The second was to compare BGA with the corresponding loss function to PPA in the setting of prior probability shift, the pattern of shift PPA is recommended for. The third was to compare BGA with the corresponding loss function to PPA in shifted settings not including prior probability shift.

In each experiment, the results were split into 18 groups and represented using split violin graphs. Each column of graphs represents a different error rate going left-to-right: 0.00, 0.01, 0.02, 0.04, 0.06, and 0.08. The rows represent the amount of shift the datasets experienced, with respect to the MSE between the old class distribution and the new one. The bottom row represents the bottom third, the middle row represents middle third, and the top row represents top third of the set of predictions with regard to the amount of shift. The y-axis in each violin plot represents the amount of loss reduced after adjustment, proportional to the loss of the original unadjusted predictions.

Comparison Between BGA for Brier Score and Log Loss In Figure 17, it is visible that when using Brier score as the loss measure, BGA using Brier score as the minimization objective tends to do better than BGA using log loss. With each distribution using Brier score pushed higher up the y-axis, indicating that loss was reduced more.

Figure 18 confirms that using the corresponding loss measure with BGA is the proper choice. When using Log loss as the loss measure, BGA with Brier score makes a large portion of the models perform worse than being unadjusted. Although, when the error is very high, both forms of BGA perform poorly.

Comparison Between BGA and PPA for Prior Probability Shift The same trends are seen when measuring with Brier score in Figure 19 and with log loss in Figure 20. With absolute certainty in the class distribution, BGA tends to outperform PPA, but this advantage disappears with just the slightest error (1 or 2%). With high error, PPA still performs relatively well, while BGA often increases loss.

Comparison Between BGA and PPA for Everything Else When it comes to non-prior probability shift and being measured with Brier score, Figure 21 shows that BGA is clearly the way to go. In all cases, BGA outperforms PPA by far. This improvement is especially profound in the most shifted datasets. Even with high error, the average reduction of loss is by nearly 40%.

This same trend is seen to a much less degree when measuring with log loss. In most cases it reduces loss more, but just slightly. With high error and low shift, it can do substantially worse.

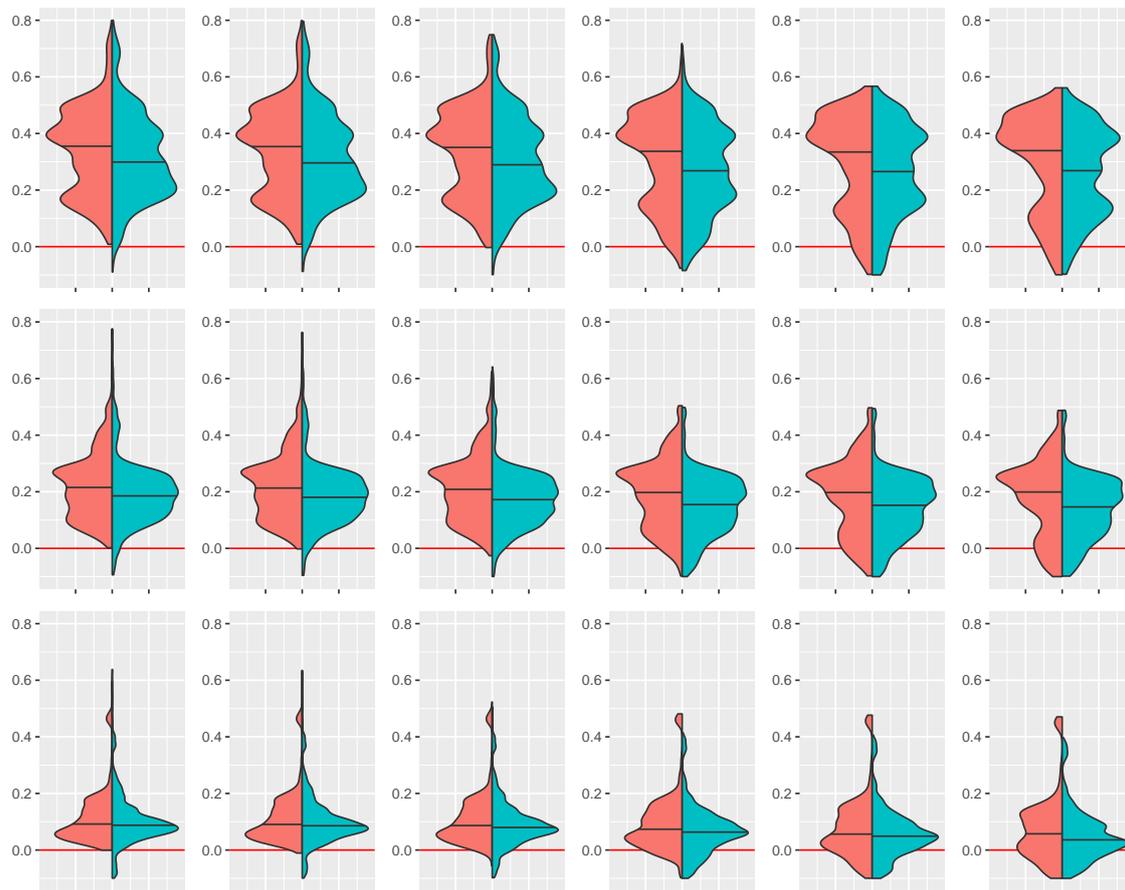


Figure 17. The reduction of Brier score by BGA for Brier score (left side of the violin) and BGA for log loss (right side of the violin). The rows correspond to different amounts of shift (with the most shifted third on top and the least shifted third on bottom). The columns correspond to amount of induced error in class distribution estimation, starting from left: 0.00, 0.01, 0.02, 0.04, 0.06, and 0.08.

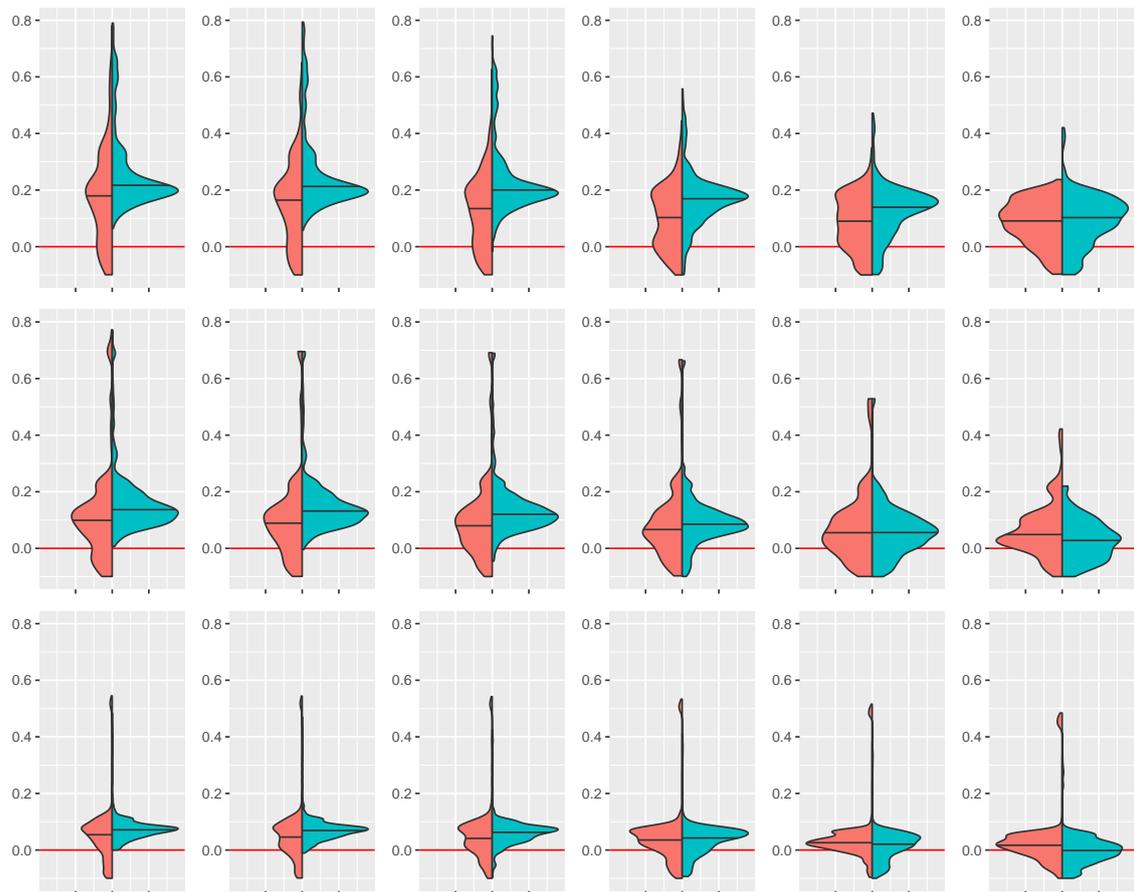


Figure 18. The reduction of log loss by BGA for Brier score (left side of the violin) and BGA for log loss (right side of the violin). The rows correspond to different amounts of shift (with the most shifted third on top and the least shifted third on bottom). The columns correspond to amount of induced error in class distribution estimation, starting from left: 0.00, 0.01, 0.02, 0.04, 0.06, and 0.08.

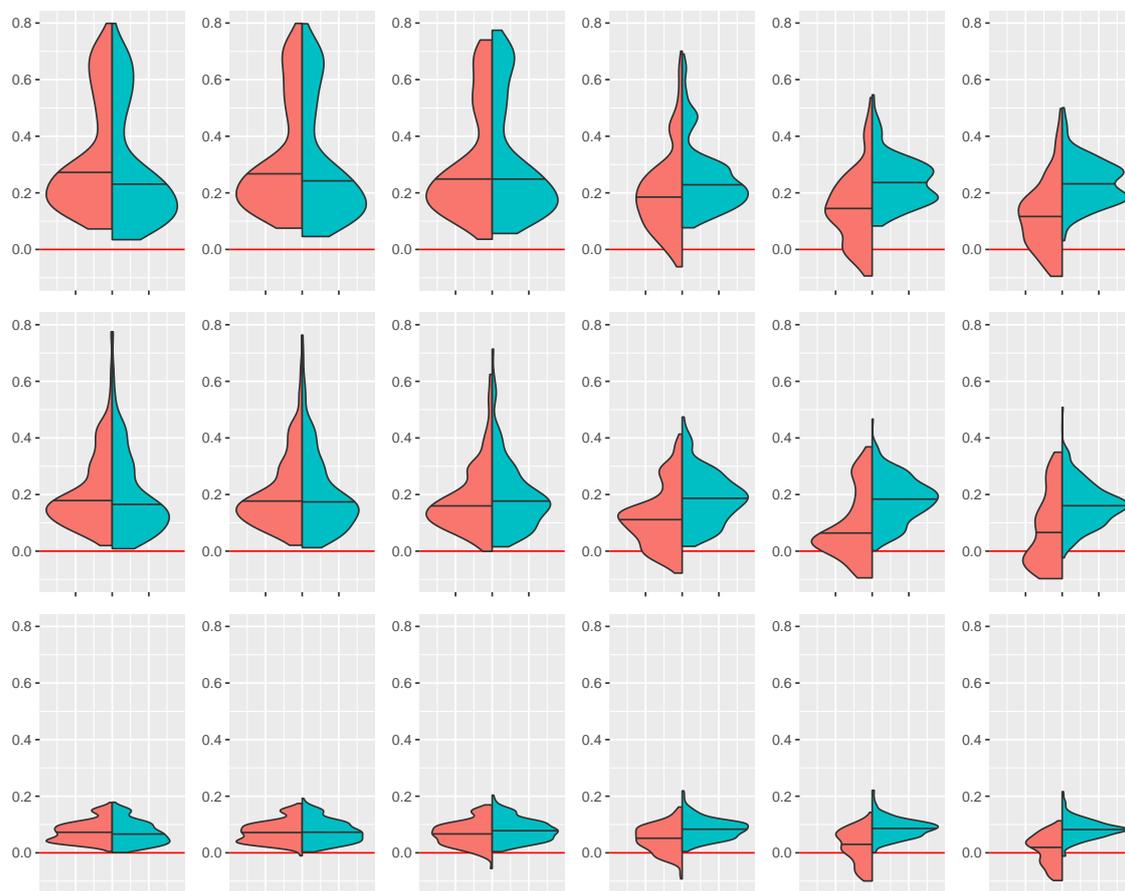


Figure 19. The reduction of Brier score by BGA for Brier score (left side of the violin) and PPA (right side of the violin). The rows correspond to different amounts of shift (with the most shifted third on top and the least shifted third on bottom). The columns correspond to amount of induced error in class distribution estimation, starting from left: 0.00, 0.01, 0.02, 0.04, 0.06, and 0.08. Only prior probability shift cases.

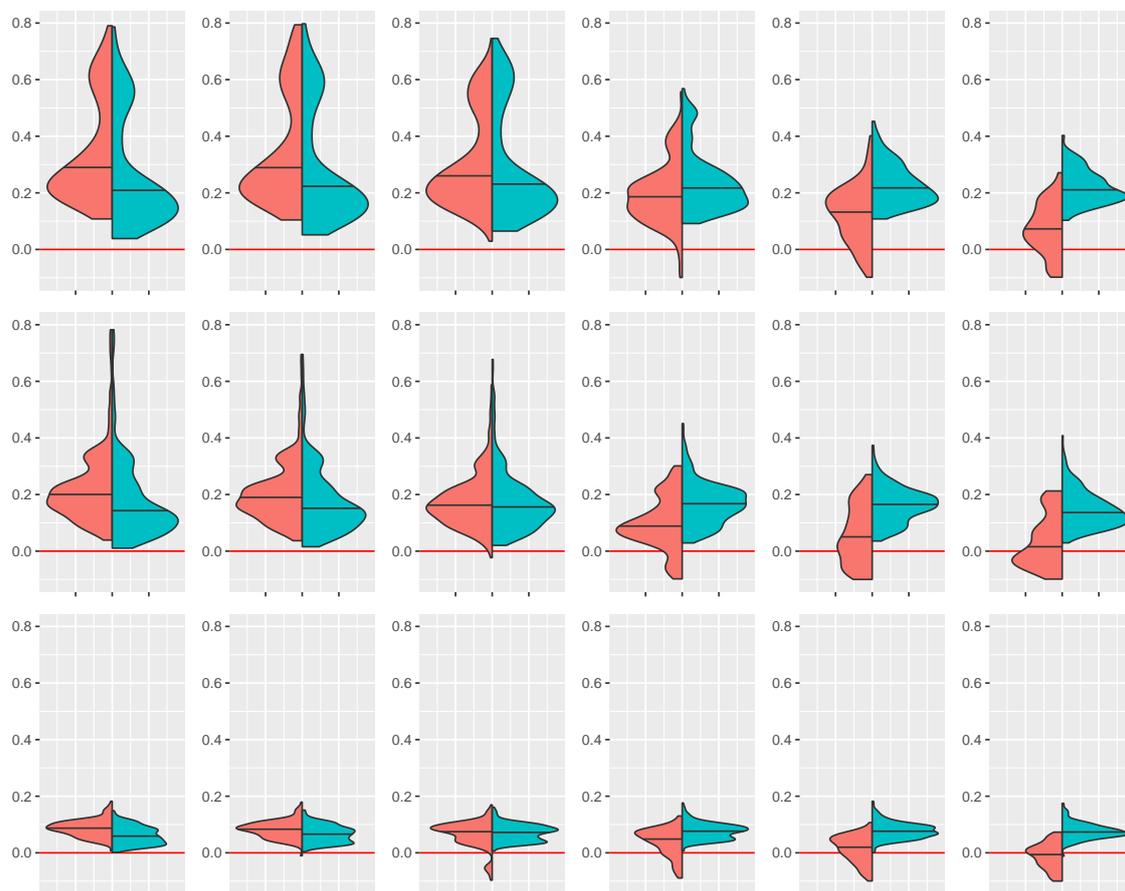


Figure 20. The reduction of log loss by BGA for log loss (left side of the violin) and PPA (right side of the violin). The rows correspond to different amounts of shift (with the most shifted third on top and the least shifted third on bottom). The columns correspond to amount of induced error in class distribution estimation, starting from left: 0.00, 0.01, 0.02, 0.04, 0.06, and 0.08. Only prior probability shift cases.

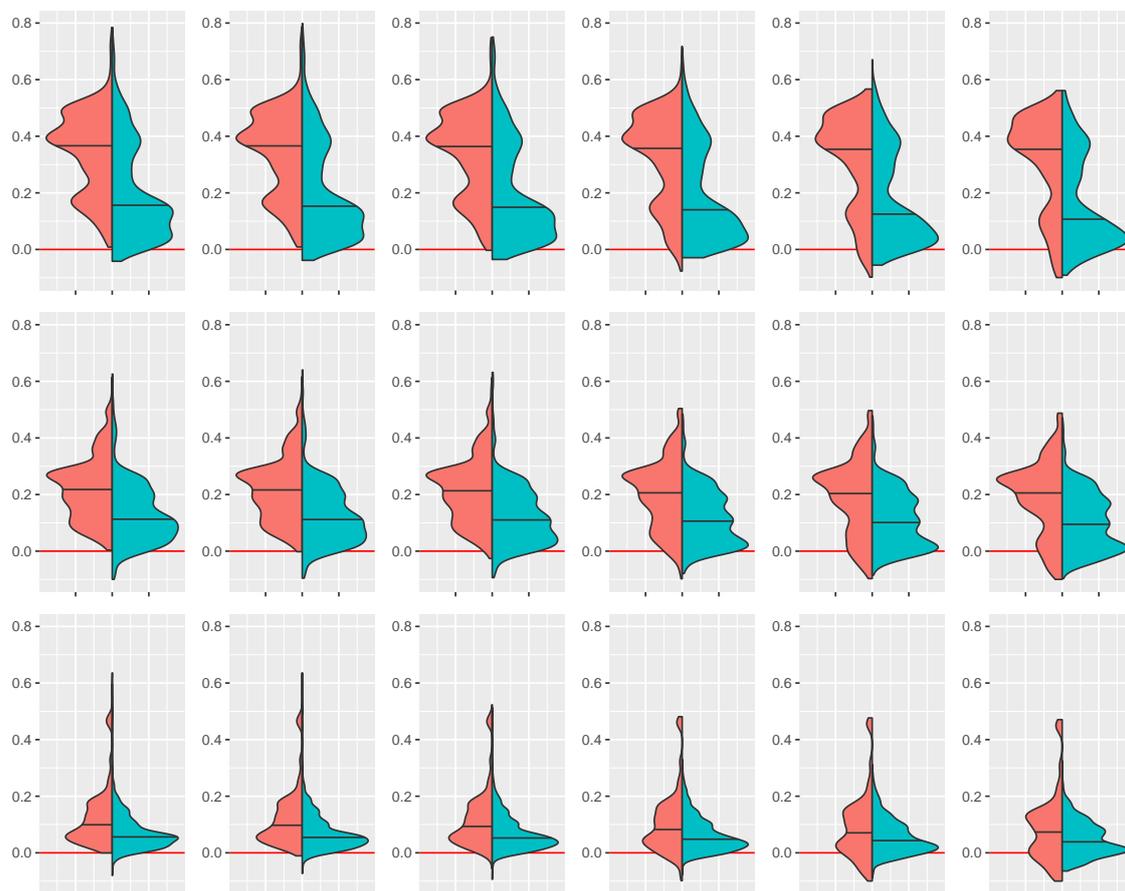


Figure 21. The reduction of Brier score by BGA for Brier score (left side of the violin) and PPA (right side of the violin). The rows correspond to different amounts of shift (with the most shifted third on top and the least shifted third on bottom). The columns correspond to amount of induced error in class distribution estimation, starting from left: 0.00, 0.01, 0.02, 0.04, 0.06, and 0.08. Only non-prior probability shift cases.

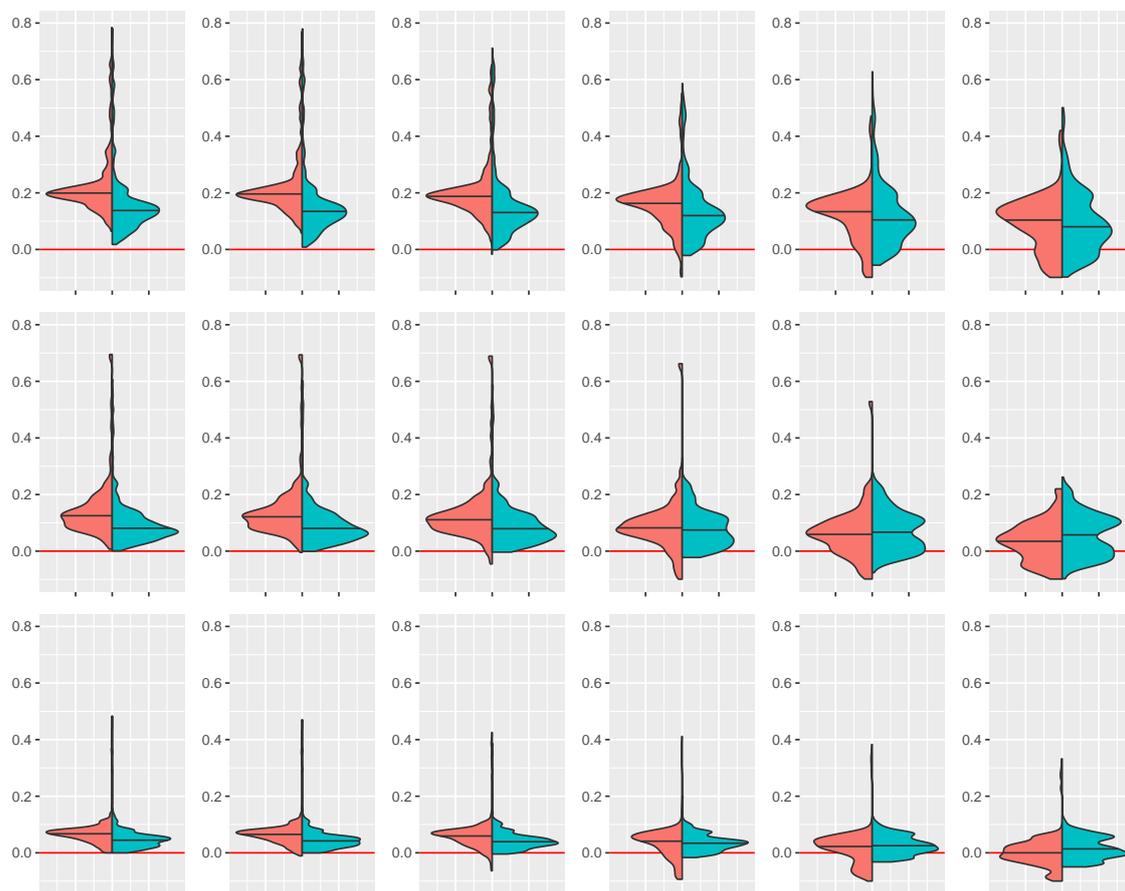


Figure 22. The reduction of log loss by BGA for log loss (left side of the violin) and PPA (right side of the violin). The rows correspond to different amounts of shift (with the most shifted third on top and the least shifted third on bottom). The columns correspond to amount of induced error in class distribution estimation, starting from left: 0.00, 0.01, 0.02, 0.04, 0.06, and 0.08. Only non-prior probability shift cases.

6 Conclusion

Dataset shift is a challenging and pervasive problem in the real world, and in recent years many researchers have started paying more attention to this issue. Probabilistic classifiers often see their performance deteriorate when shift happens, and a frequent side effect is that they become unadjusted, meaning that the expected output of the classifier no longer matches the class distribution of the data.

It has been previously shown [Kull and Flach, 2015] that adjusting a set of predictions from a probabilistic classifier in a "coherent" fashion, will reduce their expected loss. "Coherent adjustment" was only demonstrated to exist for two loss functions, mean squared error and KL divergence. The adjusted results of the former were frequently outside the range $[0, 1]$, nonsensical if interpreted as probabilities. The iterative algorithm proposed to calculate adjusted results for the latter was unreliable, often failing to converge.

This thesis presented two new adjustment procedures, unbounded general adjustment (UGA) and bounded general adjustment (BGA). UGA is a coherent adjustment that works with all proper scoring rules, a family of loss functions that includes not only mean squared error and KL divergence but all loss functions that minimize expected output at the true posterior probabilities. BGA improves upon UGA by ensuring the adjusted results are interpretable as probabilities. It is not coherent, but its reduction of expected loss is equal to or greater than that of UGA. They are both defined through minimization tasks that can be easily implemented with convex optimization algorithms.

A series of experiments were run to test the effectiveness of BGA in practice. Predictions of probabilistic classifiers were collected from OpenML [Vanschoren et al., 2013], an open database of datasets, learning tasks, and results. Various patterns of shift to various degrees of severity were applied on the models. The performance of the unadjusted model, BGA with squared Euclidean distance, BGA with KL divergence, and PPA were measured with MSE and KL divergence. This was done with various degrees of error in the estimation of the new class distribution.

Many popular classifiers, such as naive Bayes' classifiers and neural networks, will not necessarily be well-adjusted to even the probability distribution of the training set. Perhaps there are situations where adjustment might give better results than calibration? Adjustment might be useful when a large enough calibration set is not attainable. Also, adjustments effects were only explored in regard to probabilistic classifiers and proper scoring rules. Many classifiers have no intention of outputting interpretable probabilities, outputting some real numbered "score" instead as an abstract measure of confidence. These are referred to as scoring classifiers, like support vector machines. The goal for these is to make an appropriate cutoff at

some score, and give an all-or-nothing class prediction to the user. These use other loss measures: accuracy, AUC, F-score, etc. Adjustment, or a variation of it, may have some practical use in this arena as well.

References

- [Banerjee et al., 2005a] Banerjee, A., Guo, X., and Wang, H. (2005a). On the optimality of conditional expectation as a bregman predictor. *IEEE Transactions on Information Theory*, 51(7):2664–2669.
- [Banerjee et al., 2005b] Banerjee, A., Merugu, S., Dhillon, I. S., and Ghosh, J. (2005b). Clustering with bregman divergences. *Journal of machine learning research*, 6(Oct):1705–1749.
- [Barlow, 1972] Barlow, R. E. (1972). Statistical inference under order restrictions; the theory and application of isotonic regression. Technical report.
- [Barranquero et al., 2013] Barranquero, J., González, P., Díez, J., and Del Coz, J. J. (2013). On the study of nearest neighbor algorithms for prevalence estimation in binary problems. *Pattern Recognition*, 46(2):472–482.
- [Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.
- [Bregman, 1967] Bregman, L. M. (1967). 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.
- [Brier, 1950] Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3.
- [Byrne, 2015] Byrne, S. (2015). Empirical auc for evaluating probabilistic forecasts. *arXiv preprint arXiv:1508.05503*.
- [Casalicchio et al., 2017] Casalicchio, G., Bossek, J., Lang, M., Kirchhoff, D., Kerschke, P., Hofner, B., Seibold, H., Vanschoren, J., and Bischl, B. (2017). Openml: An r package to connect to the machine learning platform openml. *Computational Statistics*, 32(3):1–15.
- [Cayton, 2008] Cayton, L. (2008). Fast nearest neighbor retrieval for bregman divergences. In *Proceedings of the 25th international conference on Machine learning*, pages 112–119. ACM.
- [Dawid, 2007] Dawid, A. P. (2007). The geometry of proper scoring rules. *Annals of the Institute of Statistical Mathematics*, 59(1):77–93.
- [DeGroot and Fienberg, 1983] DeGroot, M. H. and Fienberg, S. E. (1983). The comparison and evaluation of forecasters. *The statistician*, pages 12–22.

- [Diamond and Boyd, 2016] Diamond, S. and Boyd, S. (2016). CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5.
- [Ditzler et al., 2015] Ditzler, G., Roveri, M., Alippi, C., and Polikar, R. (2015). Learning in nonstationary environments: A survey. *IEEE Computational Intelligence Magazine*, 10(4):12–25.
- [Epstein, 1969] Epstein, E. S. (1969). A scoring system for probability forecasts of ranked categories. *Journal of Applied Meteorology*, 8(6):985–987.
- [Forman, 2005] Forman, G. (2005). Counting positives accurately despite inaccurate classification. In *European Conference on Machine Learning*, pages 564–575. Springer.
- [Forman, 2008] Forman, G. (2008). Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery*, 17(2):164–206.
- [Gama et al., 2014] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):44.
- [González et al., 2017] González, P., Castaño, A., Chawla, N. V., and Coz, J. J. D. (2017). A review on quantification learning. *ACM Computing Surveys (CSUR)*, 50(5):74.
- [Gray et al., 1980] Gray, R., Buzo, A., Gray, A., and Matsuyama, Y. (1980). Distortion measures for speech processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):367–376.
- [Gretton et al., 2009] Gretton, A., Smola, A. J., Huang, J., Schmittfull, M., Borgwardt, K. M., and Schölkopf, B. (2009). Covariate shift by kernel mean matching. *Dataset shift in machine learning*, pages 131–160.
- [Hein, 2009] Hein, M. (2009). 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, pages 41–64.
- [Itakura, 1968] Itakura, F. (1968). Analysis synthesis telephony based on the maximum likelihood method. In *The 6th international congress on acoustics, 1968*, pages 280–292.
- [Japkowicz and Stephen, 2002] Japkowicz, N. and Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449.

- [Kull and Flach, 2014] Kull, M. and Flach, P. (2014). Patterns of dataset shift. In *First International Workshop on Learning over Multiple Contexts (LMCE) at ECML-PKDD*.
- [Kull and Flach, 2015] Kull, M. and Flach, P. (2015). Novel decompositions of proper scoring rules for classification: Score adjustment as precursor to calibration. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 68–85. Springer.
- [Kullback and Leibler, 1951] Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.
- [Mahalanobis, 1936] Mahalanobis, P. C. (1936). On the generalized distance in statistics. National Institute of Science of India.
- [Merkle and Steyvers, 2013] Merkle, E. C. and Steyvers, M. (2013). Choosing a strictly proper scoring rule. *Decision Analysis*, 10(4):292–304.
- [Milli et al., 2013] Milli, L., Monreale, A., Rossetti, G., Giannotti, F., Pedreschi, D., and Sebastiani, F. (2013). Quantification trees. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 528–536. IEEE.
- [Moreno-Torres et al., 2012] Moreno-Torres, J. G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N. V., and Herrera, F. (2012). A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530.
- [Pan and Yang, 2010] Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.
- [Platt et al., 1999] Platt, J. et al. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.
- [Provost and Fawcett, 2001] Provost, F. and Fawcett, T. (2001). Robust classification for imprecise environments. *Machine learning*, 42(3):203–231.
- [R Core Team, 2015] R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- [RStudio Team, 2016] RStudio Team (2016). *RStudio: Integrated Development Environment for R*. RStudio, Inc., Boston, MA.
- [Saerens et al., 2002] Saerens, M., Latinne, P., and Decaestecker, C. (2002). Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural computation*, 14(1):21–41.

- [Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423.
- [Shimodaira, 2000] Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244.
- [Storkey, 2009] Storkey, A. (2009). When training and test sets are different: characterizing learning transfer. *Dataset shift in machine learning*, pages 3–28.
- [Sugiyama et al., 2007] Sugiyama, M., Krauledat, M., and MÅzller, K.-R. (2007). Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(May):985–1005.
- [Sun et al., 2009] Sun, Y., Wong, A. K., and Kamel, M. S. (2009). Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(04):687–719.
- [Vanschoren et al., 2013] Vanschoren, J., van Rijn, J. N., Bischl, B., and Torgo, L. (2013). Openml: Networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60.

Appendix

I. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Theodore James Thibault Heiser**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:
 - 1.1 reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and
 - 1.2 make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

of my thesis

Dataset Shift and the Adjustment of Probabilistic Classifiers

supervised by Meelis Kull

2. I am aware of the fact that the author retains these rights.
3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, 28.05.2018