Measuring Data Drift with the Unstable Population Indicator

Marcel R. Haas [a,b,*] and Lisette Sibbald [c,d,b]
[a] *Public Health and Primary Care, Leiden University Medical Center, Albinusdreef 2, The Netherlands*
E-mail: datascience@marcelhaas.com; ORCID: https://orcid.org/0000-0003-2581-8370
[b] *Business Intelligence, University of Amsterdam, Spui 21, 1012WX Amsterdam, The Netherlands*
[c] *Department of Methodology and Statistics, Tilburg University, Prof. Cobbenhagenlaan 125, 5037 DB Tilburg, The Netherlands*
E-mail: L.Sibbald@tilburguniversity.edu; ORCID: https://orcid.org/0000-0000-0000-0002
[d] *Department of Cognitive Neuropsychology, Tilburg University, Prof. Cobbenhagenlaan 125, 5037 DB Tilburg, The Netherlands*

Abstract. Measuring data drift is essential in machine learning applications where training is necessarily done on strictly different samples than later scoring. The Kullback-Leibler divergence is a common measure of shifted probability distributions, for which discretized versions are invented to deal with binned or categorical data. We present the Unstable Population Indicator, a robust, flexible and numerically stable, discretized implementation of Jeffrey's divergence, along with an implementation in a `Python` package that can deal with continuous, discrete, ordinal and nominal data in a variety of popular data types. We show the numerical and statistical properties in controlled experiments. It is not advised to employ a common cut-off to distinguish stable from unstable populations, but rather to let that cut-off depend on the use case.

Keywords: data drift, data shift, machine learning, kl-divergence

1. Introduction

In studies where populations are compared, it is often required to quantify the change of the population from one sample to the next. The term population is used in the broad sense here: it can be a group of people, but it can also refer to a general ensemble of elements, about which we have data that will be statistically investigated. We will refer to populations in the broad sense throughout this article, and will use the term "entities" to denote the elements making up the the population (e.g. the people in a group).

Changes in populations can be interesting in their own right [1, 2], but they can also be a necessary prerequisite for modeling a particular property of the population in question or its entities [3, 4]. Common use cases are machine learning models that have been trained on a training data set that has been collected in the past (the "historical" data) while it is currently running in production, predicting properties of interest on "current" data, that may or may not still be representative of the training data [5, 6]. Usually, the model will be re-trained every once in a while to accommodate the model running in an ever-changing world. It is useful to know how dissimilar your population can become from your training population in either the target variable or any one of the features employed by the predictive model.

*Corresponding author. E-mail: datascience@marcelhaas.com.

Whether to retrain or not, or even whether to make any prediction at all can, and probably *should* be dependent on a measure of similarity between training target or model features and the properties of the data set the model faces in production. This requires a quantitative measure of data drift (also know as data shift or concept drift) between populations. This drift can, and if possible *should* be measured on both predictor and target values, contrary to what is sometimes used [7, 8]. Such a measure should be well-behaved, i.e. it should give numerical values that actually mean something, no matter how subtle or extreme the data shift is and it should be able to contrast between changes that are perhaps not important and changes that really require action from the data scientist. Note that this decision, whether or not to take action, will very much depend on the use case of the model in question, and can never depend only on the numerical value of a measure of data drift.

When comparing two populations, many measures of (dis-)similarity exist [9]. Some work exclusively for continuous data, some only for categorical data and many are commonly employed in niche branches of applications of machine learning. Some care about the source of the drift, while some don't [4]. Software implementations do exist, especially for the more "classical" examples, which are commonly included in statistical packages available for many programming languages.

In this paper we introduce yet another such measure and the question may rise why this would be necessary. Our new data drift quantification is a small modification to a popularly used existing parameter, the Population Stability Index [PSI, 10, 11]. We describe the PSI and our new implementation, the Unstable Population Indicator, in some detail, while connecting it to its underlying information theoretical foundation. We take away some numerical nuisances in the implementation and we release an implementation in `Python` that is stable, flexible and fast. Therefore, with just our UPI, users can identify data drift in target or feature variables, which can be continuous, discrete, ordinal or nominal in nature and may come in a variety of different popular data formats.

We will start with a brief outline of relative entropy-based measures of dissimilarity from information theory in Section 2, which will give the reader an understanding of how previous and our new implementation fit in the landscape of divergence measures. After that, we will introduce our new measure, the Unstable Population Indicator in Section 3. The numerical and statistical properties of the UPI are discussed in Section 4 and a short summary of a publicly released `Python` implementation is given in Section 5. We will conclude in Section 6.

2. Relative entropy based divergence between probability distributions

Quantitative comparisons of distribution functions are valuable for a great variety of reasons. They are mostly based on Information Theory, and in particular on the Shannon entropy [12], which is a measure of the "disorder" of a system based on the number of possible configuration of entities in a population that would result in the observed distribution function. For a discrete random variable X with possible values $x_i = x_1, x_2, ..., x_n$ (respectively, a continuous variable X), all occurring with probability $P(x_i)$ (respectively, of which the probability densities are described by a PDF $P(x)$), the Shannon entropy [13, 14] is given by

$$H(X) = -\sum_{i=1}^{n} P_i(x_i) \cdot \log(P_i(x_i)) \tag{1}$$

$$H(X) = -\int_{-\infty}^{\infty} P(x) \cdot \log(P(x))\,\mathrm{d}x \tag{2}$$

The units of this entropy are set by the base of the logarithm. In this paper we take the natural logarithm, which measures entropy in "nats" (base 2 would give "bits" or "shannons", and base 10 would give "dits", "bans" or "hartleys"). In words, this entropy describes the PDF-averaged negative log probability mass (or density) and is thus the expectation value of that probability mass or density. The normalization for the expectation values would after all be $\sum_{i=1}^{n} P_i(x_i) = 1$ and $\int_{-\infty}^{\infty} P(x)\,\mathrm{d}x = 1$, respectively.

Adding information (or: "learning something" about X) should make X better-determined, which lowers the entropy and increases the average probability mass/density.

2.1. Symmetrized Kullback-Leibler divergence

A change from one distribution function to another can be quantified by the *relative* entropy. The relative entropy of one distribution function P to another distribution function Q is known as the Kullback-Leibler divergence [15], which is given by

$$D_{\mathrm{KL}}(P \,||\, Q) = \int_{-\infty}^{\infty} p(x) \cdot \ln\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x \tag{3}$$

where $p(x)$ and $q(x)$ are the probability densities of P and Q. KL-divergence is not a metric, as it is neither symmetric nor does it obey the triangle inequality and is the expectation of the logarithmic difference between the probabilities P and Q. The asymmetry can be understood: when shifting the mean of a Gaussian, while leaving the standard deviation the same, the average probability density does not change, even though the PDF does, so information needs to be added about the random variable X. Shifting the same Gaussian back to its original position requires the same amount of information added (not subtracted): one added information twice so the total KL-divergence is non-zero, even though we are back at the same distribution function.

A symmetrized version measures the relative entropy from P to Q and adds the relative entropy from Q to P, which has become known as the Jeffrey's divergence. For discrete probability density functions (or, rather, probability mass functions), this can be written as

$$D_{\text{Jeffrey's, discr.}} = \sum_{x \in \chi} (P(x) - Q(x)) \cdot \ln\left(\frac{P(x)}{Q(x)}\right) \tag{4}$$

where the summation is over the same probability space χ: $P(x) = 0$ implies $Q(x) = 0$.

As this quantity is defined on a discrete probability space, it works for binned continuous variables as well as for categorical variables and it is not necessary for the space χ to be continuous.

2.2. The Population Stability Index

The Jeffrey's divergence, in simple implementations, is popularly known as the Population Stability Index [PSI, 10, 11]. Another such measure is the Population Accuracy Index [PAI, 16], but that definition seems to gain less traction. The PSI measures the amount of change in a population, based on a single (continuous, discrete or categorical) variable and measures the change of the fraction of entities at several possible values for the value of that variable. This is completely equivalent to the Jeffrey's divergence in Eq. 4 and usually written as

$$\mathrm{PSI} = \sum_{\text{bins, } i} (f_{1,i} - f_{0,i}) \cdot \ln\left(\frac{f_{1,i}}{f_{0,i}}\right) \tag{5}$$

where $f_{0,i}$ and $f_{1,i}$ are the fraction of the entities in bin or category i in the original and new population, respectively.

In many implementations, like described in a variety of online blog posts[1], but also e.g. in [11, 17], PSI is used on both categorical as well as on discretized versions of continuous variables. Often, 10 or 20 bins are chosen without too much justification. Note how the definition of PSI does not take into account any order in the bins, nor distances between the bins, which makes the measure equally suitable for categorical/nominal data, but interestingly this is rarely done. Besides, many posts and papers suggest the same, uninformed cut-off values for the PSI as a distinction between stable and shifted or drifted data sets. What counts as an important shift in your data should be strongly use case dependent and investigated on a per-feature basis.

2.3. The relation with chi-squared tests

For categorical data, the χ^2-test is often used to determine whether the distribution in question (the current or new data set) is significantly different from the original data set. It is a method from the frequentist realm of hypothesis testing where some null-hypothesis (e.g. no change) is contrasted with an alternative (e.g. a change), where it can be questioned how the null in this example can ever be true in practice. Significance levels need to be hand-picked (just going for $P < 0.05$ would be rather naive, given the large variation in data set sizes occurring in practice), resulting in either a significant detection of change, or not [e.g. 18]. As we will motivate below, a hard cut-off is not advised. Given the definition of the statistic, $\chi^2 = \sum(O - E)/E^2$, where the sum is over all independent categories, with O and E being the counts in the *O*bserved (or: new) vs *E*xpected (or: original) data sets, respectively, the same numerical issues as with the original definitions of PSI exist.

3. The Unstable Population Indicator

It will seem obvious that the evaluation of the logarithm in Eq.5 is impossible or meaningless when either $f_{0,i} = 0$ or $f_{1,i} = 0$, which is what occurs when a bin or category is un-populated in either the original or new population, respectively, as also noted by [7]. In practice, this may happen fairly often, depending on the use case for the PSI.

3.1. Defining UPI

We therefore propose a low-impact numerical adaptation to the original formulation of the PSI, and release a more flexible implementation of it. As the Index is close to zero for stable populations and higher for unstable populations, we name our new index the *Unstable Population Indicator*: UPI.

In order to solve the numerical nuisances in the evaluation of the logarithm, we add a fraction to every bin or category, equal to one extra entity in the combination of both populations, which is saying we divide one extra entity equally over all bins, in both the original and the new population. In well-populated bins, this difference is marginal, while in (almost) empty bins or categories the difference may be larger. Adding one to a logarithm to ensure positive evaluation is common practice, even though the effect differs strongly between almost empty and very well filled bins. The justification is that for most use cases, a population should be sufficiently large that adding one entity to the total of the populations,

[1] See, e.g., this medium post, this github page, this SAS paper and this post on Machine Learning+

n_{tot}, should make little difference (i.e. $1 \ll n_{\text{tot}}$) and that by smearing out that one extra entity over all bins (i.e. a flat distribution) results in a very small change in the population distribution; one that shouldn't be picked up by doing this analysis in the first place. We will add some more numerical justification below.

Our implementation conserves the symmetry of Jeffrey's divergence, which would have been broken by only adding (parts of) elements to either one of the population. Often, the algorithms following this investigation have been trained on the original population, which therefore determines the existing bins or categories. This could make it attractive to only add elements to the new population, when empty bins or categories arise. For many applications, though, having a symmetric distance measure is convenient, and in our implementation we lose nothing by ensuring just that.

For the UPI, we arrive at the following expression:

$$\text{UPI} = \sum_{\text{bins, } i} (f_{1,i} - f_{0,i}) \cdot \ln\left(\frac{f_{1,i} + 1/n_{\text{tot}}}{f_{0,i} + 1/n_{\text{tot}}}\right) \tag{6}$$

where n_{tot} is the number of entities in the original and new populations together and N_{bins} is the number of bins or categories.

For very large populations ($n \to \infty$) this reduces to the expression for the PSI, but including the extra terms in the logarithm solves the problem of empty bins or categories.

3.2. *UPI in higher dimensions*

UPI is defined for the comparison of two one-dimensional probability mass or density functions. Essentially, going to higher dimensions can be done by counting bins of unique combinations of all values in all dimensions. Typically, this is going to result in ever more sparsely filled bins. We have not implemented such a higher-dimensional version (potentially, if demand is there, this is left for future work). One trivial way to extend to higher dimensions and capture the drift of a data set in many dimensions simultaneously is to add the separate dimensions like distance measures in a Euclidean space. UPI is symmetric, positive definite and obeys the triangle inequality (as long as the same two populations are compared along all dimensions; not doing so makes the calculation meaningless) and is thus a distance metric. The total multi-dimensional UPI_{tot} is then

$$\text{UPI}_{\text{tot}}^2 = \sum_{\text{dim}} \text{UPI}_d^2 \tag{7}$$

where UPI_d is UPI as defined in Eq. 6 for every dimension separately and the sum is over all dimensions.

4. Numerical and statistical properties of the UPI

In this section we compare UPI, PSI and their underlying theoretical divergences to measure data drift for (samples of) known distribution functions, in order to illustrate the numerical and statistical behavior of such distance measures.

4.1. Sampling from known continuous distribution functions

As a numerical experiment we will start by calculating the UPI and PSI for randomly sampled distribution functions, for which we also know the actual KL-divergence analytically.

As a base population to compare with, we sample 1000 values from a Gaussian distribution with a mean of zero and a standard deviation of one. We then draw a second population with either 50, 100 or 1000 samples from Gaussians that have a standard deviation of one and means (0, 0.5, 1, 1.5, 2, 2.5). We then bin these using `numpy`'s `auto` setting (which we supply with both populations together).

The KL-divergence between two Guassian PDFs can be calculated analytically, and is given by

$$D_{\mathrm{KL}}(P\,||\,Q) = \ln\frac{\sigma_Q}{\sigma_P} + \frac{\sigma_P^2 + (\mu_P - \mu_Q)^2}{2\sigma_Q^2} - \frac{1}{2} \tag{8}$$

which for $\sigma_P = \sigma_Q = 1$ and $\mu_P = 0$ simply reduces to $D_{\mathrm{KL}}(P\,||\,Q) = \mu_Q^2/2$ and for the symmetrized divergence under the same conditions just to μ_Q^2. We show boxplots of UPI as a function of the offset between the distributions for 100 realizations of the set-up above in Fig. 1. The binning scheme results in very many empty bins for the second population. In fact, only 4 in 1800 simulations have a PSI value that can be calculated, all other realisations had empty bins in either population. This is in part a result of the binning scheme and we come back to this below. It is, nevertheless, an obvious sign that using the original implementation of PSI comes with severe drawbacks when used on continuous data, for which the binning scheme will need to be fine-tuned by hand. UPI gets rid of this problem and is compared to the analytical symmetrized KL-divergence with the stars in the upper panels of Fig. 1. As can be seen, the correspondence between UPI and KL-divergence is very good, especially for large sample sizes and large offsets.

To get around the problems of the binning scheme and to mimic the use of the data drift measures for discrete variables, we bin all of the data into bins with edges at (-∞, -1, 0, 1, ∞), which ensures well-filled bins at least for population 1. Population 2 is than binned into the same bins and the UPI and PSI are measured for those bins (or categories, if preferred) and displayed in the lower panels of Fig. 1. Here we also include the probability mass based KL-divergence for completeness. The grey bars show the fraction of the samples for which PSI is still undefined because of empty bins. For small samples and large offsets this can still be very common. As long as they are both well-determined, the measures largely agree. Mass-based KL-divergence is asymmetric and is therefore more often finite than PSI.

4.2. Measuring data drift in probability mass functions

In a second experiment we create some artificial categorical distributions and show how the values for UPI develop as a function of the difference in bin distributions. Specifically, we create a three-category data set. In the first experiment, the baseline population has a completely flat distribution of 3 $\times$ 300 entities per category. For the second population, while leaving the first category ('A') untouched, we gradually move all of the items from category 'B' to category 'C'. In Fig. 2, left panels, we show what values UPI takes (upper panel) and what the distributions over the categories look like before, halfway and at the end of the process (lower panel). For comparison, we also plot the value for UPI for distributions of 20 objects per category, initially.

PSI values are almost identical to UPI values, except that they are undefined for situations in which either of the categories is empty (which occurs in either 'Original' or 'Final' states).

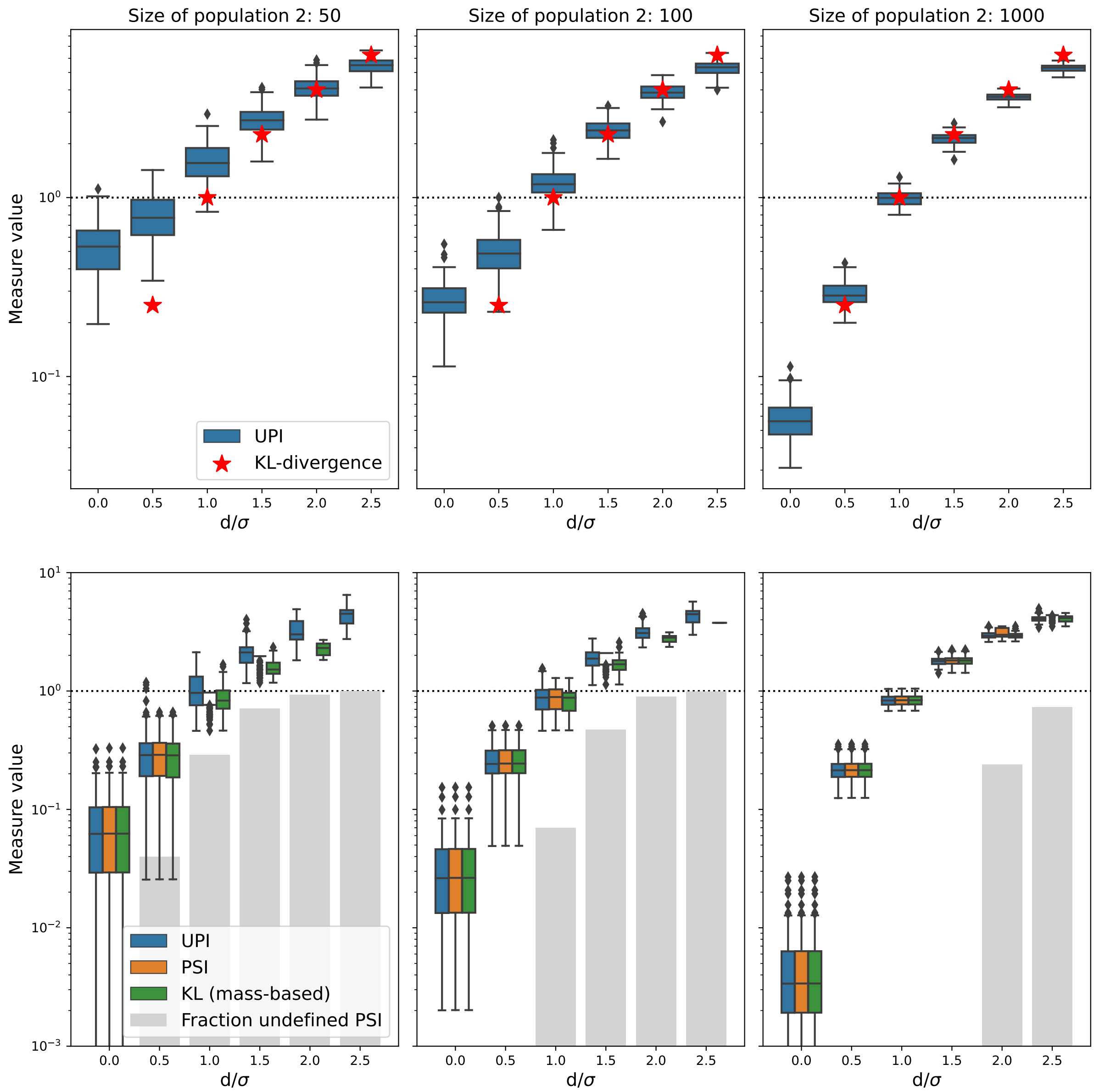

Fig. 1. The values for different measures of data shift in numerical experiments in which populations are randomly drawn from a Gaussian distribution function. One population always consist of 1000 entities, the second has 50, 100 or 1000 entities (the facet columns) from a Gaussian with the same standard deviation, but shifted by a number of standard deviations, indicated on the horizontal axis. The vertical axis indicates the value of either of the data shift measures, as indicated by the colors. The reference line helps guide the eye and shows that a value of 1 roughly corresponds to a difference of 1 standard deviation, but the relation between the value of the shift measure and the distance between the Gaussians is non-linear. The upper panels are created with variable binning using NUMPY's AUTO setting, and the stars indicate the exact KL-divergence for two shifted Gaussians (which is 0 for $d/\sigma = 0$). The lower panels use bins with bin edges at $(-\infty, -1, 0, 1, \infty)$ to ensure the PSI has mostly finite values). The fraction of populations that have undetermined PSI values (see text for more details) is indicated with the grey bars.

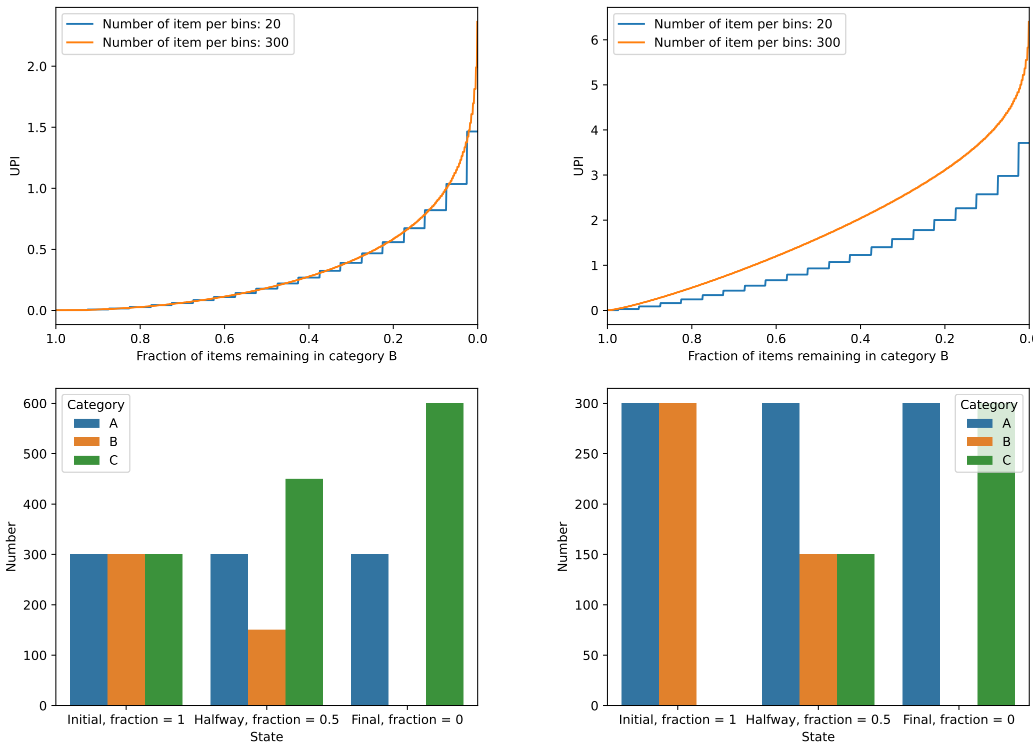

Fig. 2. Values for UPI for categorical distributions in which items from Category ‘B’ are gradually moved into Category ‘C’, while leaving Category ‘A’ untouched. In the left panels, ‘A’, ‘B’ and ‘C’ start out uniformly while in the right panels ‘C’ starts out empty. The evolution of UPI with the remaining fraction in Category ‘B’ is shown in the upper panels for 20 and 300 items per filled category, respectively. The distribution over categories is shown in the lower panel, where the ‘Original’ state of Category ‘B’ is identical to the distribution of the population that is compared to.

As another example, we repeat the above experiment, except that category ‘C’ (that receives items) starts out empty, and gradually takes over the elements from category ‘B’ (which starts with the same number of elements as category ‘A’ and ends up empty, like before). Note that the total number of elements in the population is smaller in this experiment and in the right panel of Fig. 2 we show that in this case, the value of UPI is more sensitive to the total size of the populations as well. An important difference is that the values for UPI are in general much higher than in the the previous exercise, which is explained by the larger relative difference in the populations displayed in the lower right panel (as compared to the lower left panel).

4.3. A cut-off value for UPI to determine significant data drift

It would be tempting to use one value as a cut-off above which populations are “different” and below which they are “the same”. This is common [8, 10], but uninformed practice, also when using the conventional PSI. As is shown in Fig. 1 the values for UPI and PSI vary smoothly with increasing difference in underlying populations, for both variable bins and fixed bins. What counts as an "important" difference depends on the use case and should be evaluated whenever UPI or PSI is used in practice. Even for the same distribution function shapes, with the same difference in peak location, a cut-off value will also depend on the two (and relative) sample sizes. Furthermore, as can be seen from the extent of the bar plots in Fig. 1, there is a sizable spread in UPI values for different random realizations of the same two underlying distribution functions. In Fig. 2 we show that depending on how relative categories are filled in categorical data, rather different choices may be made as cut-off values for UPI, which again should be largely informed by the use case.

5. A `Python` implementation

With this paper, we release a `Python 3` package to calculate the UPI for two distributions of data. The package also allows to calculate the original PSI and the one-directional mass-based KL-divergence. The package is called unstable_populations and can be installed from PyPI via "`pip install unstable_populations`". It allows continuous data as well as categorical (and, thus, ordinal and nominal) data. It is possible to feed the function with binned continuous data, or let the function do the binning of it, based on `numpy`'s `histogram` methods. The populations that are to be compared can be supplied as `list`, `tuple`, `dictionary`, `numpy.ndarray`, `pandas.Series` and `pandas.DataFrame`. With these features it is more flexible and more generic than previous packages to calculate (a subset of) these population stability parameters.

The documentation for the functions is included with the packages. The package comes with an extensive set of test cases, demonstrating the flexibility of the package in terms of input.

The main repository for the package, where issues can be raised (and pull requests submitted) and where a notebook can be found that reproduces the figures in this paper is located at GitHub.

6. Conclusions

Popular methods to measure the difference between two distributions functions, either of continuous or categorical nature, are based on the concept of relative entropy. Many such measures are implemented in a variety of packages or downloadable functions, but very often come with numerical nuisances that render them less than flexible in everyday use.

We introduce the Unstable Population Indicator, and release a `Python` package that implements it for continuous and discrete data sets in a variety of popular data types alongside. The definition of UPI is very close to that of Jeffrey's Divergence, which itself a symmetrized version (and hence a metric) of the Kullback-Leibler Divergence, i.e. the relative entropy between two distribution functions. With that, the definition is well-motivated by Information Theory, and we explained how to get from the entropy of a distribution to our definition of data shift, also known as data drift.

We have shown that UPI is a smoothly increasing function of data drift, regardless of the underlying distributions being categorical or continuous in nature and that it is largely insensitive to the binning scheme of a continuous variable. This smoothly increasing behavior leads us to strongly advise against the use of a "rule-of-thumb" like cut-off value between distributions that show no significant shift, versus those that do. On a case-by-case basis, users should evaluate what values of UPI are acceptable and which indicate that further action (e.g. re-training a machine learning model) is necessary.

The released code is easy to obtain, install and use and is fully open source.

Acknowledgments

The code and this paper use `Python 3` [19], `numpy` [20], `pandas` [21, 22], `matplotlib` [23] and `seaborn` [24]. The authors thank Joris Huese for valuable and inspiring discussions during the onset of this study and Robert Zwitser for the freedom to pursue it. Dan Foreman-Mackey and Adrian Price-Whelan are acknowledged for technical help and Bluesky for taking over the role of what once was Twitter for enabling such help.

References

[1] G.L. Poe, K.L. Giraud and J.B. Loomis, Computational Methods for Measuring the Difference of Empirical Distributions, *Econometric Modeling: Agriculture* (2005).
[2] G.I. Webb, L.K. Lee, B. Goethals and F. Petitjean, Analyzing Concept Drift and Shift from Sample Data, *Data Mining and Knowledge Discovery* **32** (2018), 1179–1199.
[3] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer and N.D. Lawrence (eds), *Dataset Shift in Machine Learning*, The MIT Press, 2008. ISBN 978-0-262-17005-5. doi:10.7551/mitpress/9780262170055.001.0001. https://direct.mit.edu/books/book/3841.
[4] M. Kull and P.A. Flach, Patterns of Dataset Shift, in: *Learning over Multiple Contexts, at ECML 2014*, 2014.
[5] K. Nelson, G. Corbin, M. Anania, M. Kovacs, J. Tobias and M. Blowers, Evaluating Model Drift in Machine Learning Algorithms, in: *2015 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, 2015, pp. 1–8. doi:10.1109/CISDA.2015.7208643.
[6] S. Ackerman, E. Farchi, O. Raz, M. Zalmanovici and P. Dube, Detection of Data Drift and Outliers Affecting Machine Learning Model Performance over Time, *ArXiv* **abs/2012.09258** (2020).
[7] A. Becker and J. Becker, Dataset Shift Assessment Measures in Monitoring Predictive Models, in: *International Conference on Knowledge-Based Intelligent Information & Engineering Systems*, 2021.
[8] F.M. Polo, R. Izbicki, E.G. Lacerda, J.P. Ibieta-Jimenez and R. Vicente, A Unified Framework for Dataset Shift Diagnostics, *ArXiv* **abs/2205.08340** (2022).
[9] I. Goldenberg and G.I. Webb, Survey of Distance Measures for Quantifying Concept Drift and Shift in Numeric Data, *Knowledge and Information Systems* (2018), 1–25.
[10] B. Yurdakul and J. Naranjo, Statistical Properties of the Population Stability Index, *Journal of Risk Model Validation* **14**(4) (2007), 89–100. doi:10.21314/JRMV.2020.227.
[11] B. Yurdakul, Statistical Properties of Population Stability Index (PSI), PhD thesis, Western Michigan University, 2018.
[12] J. Lin, Divergence Measures Based on the Shannon Entropy, *IEEE Transactions on Information Theory* **37**(1) (1991), 145–151. doi:10.1109/18.61115.
[13] C.E. Shannon, A Mathematical Theory of Communication, *The Bell System Technical Journal* **27**(3) (1948), 379–423. doi:10.1002/j.1538-7305.1948.tb01338.x.
[14] C.E. Shannon, A Mathematical Theory of Communication, *The Bell System Technical Journal* **27**(4) (1948), 623–656. doi:10.1002/j.1538-7305.1948.tb00917.x.
[15] S. Kullback and R.A. Leibler, On Information and Sufficiency, *The Annals of Mathematical Statistics* **22**(1) (1951), 79–86. doi:10.1214/aoms/1177729694.
[16] R. Taplin and C. Hunt, The Population Accuracy Index: A New Measure of Population Stability for Model Monitoring, *Risks* **7**(2) (2019). doi:10.3390/risks7020053. https://www.mdpi.com/2227-9091/7/2/53.
[17] G. Karakoulas, Empirical Validation of Retail Credit-Scoring Models, *The RMA Journal* (2004), 56–60.
[18] D.M. Sharpe, Your Chi-Square Test Is Statistically Significant: Now What?., *Practical Assessment, Research and Evaluation* **20** (2015), 1–10.
[19] G. Van Rossum and F.L. Drake, *Python 3 Reference Manual*, CreateSpace, Scotts Valley, CA, 2009. ISBN 1441412697.
[20] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J.F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke and T.E. Oliphant, Array Programming with NumPy, *Nature* **585**(7825) (2020), 357–362. doi:10.1038/s41586-020-2649-2.
[21] T. pandas development team, pandas-dev/pandas: Pandas, *Zenodo* (2020). doi:10.5281/zenodo.3509134.
[22] Wes McKinney, Data Structures for Statistical Computing in Python, in: *Proceedings of the 9th Python in Science Conference*, Stéfan van der Walt and Jarrod Millman, eds, 2010, pp. 56–61. doi:10.25080/Majora-92bf1922-00a.
[23] J.D. Hunter, Matplotlib: A 2D Graphics Environment, *Computing in Science & Engineering* **9**(3) (2007), 90–95. doi:10.1109/MCSE.2007.55.

[24] M.L. Waskom, seaborn: Statistical Data Visualization, *Journal of Open Source Software* **6**(60) (2021), 3021. doi:10.21105/joss.03021.