Continuous Weight Balancing

Daniel J. Wu and Avoy Datta
danjwu@cs.stanford.edu, avoy.datta@stanford.edu
Departments of Computer Science and Electrical Engineering
Stanford University

Abstract

We propose a simple method by which to choose sample weights for problems with highly imbalanced or skewed traits. Rather than naively discretizing regression labels to find binned weights, we take a more principled approach – we derive sample weights from the transfer function between an estimated source and specified target distributions. Our method outperforms both unweighted and discretely-weighted models on both regression and classification tasks.

Motivation

Real-world datasets are heterogenous and frequently skewed. Imbalanced datasets will generally produce imbalanced models. These models can encode dataset-specific biases, and perform disproportionately better on highly represented data. There is a rich body of literature that aims to mitigate the negative effects of dataset imbalance; in practice, however, class balancing via reweighting is sufficient for many tasks. By virtue of being simple to understand, easy to use, and effective, class reweighting is a well-worn wrench in the machine learning toolbox.

In this work, we consider a similarly simple and easy-to-use strategy to mitigate a broader type of dataset skew – imbalance of a continuous trait. These continuous traits need not just be labels; datasets may also be biased along the axis of some continuous feature or metadata. Correcting against these biases within datasets is a key step towards developing robust and unbiased models.

Kernel Density Estimates

Kernel Density Estimation is a well-known method to evaluate the probability density of a random variable given some observed samples. Formally, let $x_1, x_2, ..., x_n$ be univariate samples drawn i.i.d. from a distribution with some density $f(x)$ at any given point x. We approximate this function f with the kernel density estimator

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

where K is some non-negative kernel function, and $h > 0$ is the bandwidth parameter which smooths the resultant estimate. By convenience, the standard normal density function ϕ is commonly used as the kernel function K.

The bandwidth parameter h encodes a trade-off between the KDE's bias and variance; a common heuristic is Scott's rule, which, for a dataset of size n with dimensionality d, sets the bandwidth h as $h = n^{\frac{-1}{d+4}}$.

Models

We experiment with 3 classes of models: random forests, linear/logistic regression, and shallow neural networks. Our random forests use 100 estimators, and our neural networks are an ensemble of 10 shallow networks, each with two hidden layers containing 64 and 16 nodes, respectively. We use ReLU activations and apply dropout ($p = 0.5$).

Evaluation

We compare our continuous weights against *no weighting* and *discretized weighting*. We evaluate the R2 score for regression and AUROC for binary classification, on an out-of-sample subset of the data. Since our goal is to enable models to do well on underrepresented subsets of our data, we evaluate model performance on underrepresented subsets of the out-of-sample data. For the California housing dataset, we examine higher housing prices (> 2), while for the heart disease dataset we examine lower age groups (< 60).

Method

Our method is a simple four-step procedure, shown in Figure 1.

- **Choose a weight trait:** A weight trait is some continuous variable that captures an important feature of each data point; this is the variable we would like to weight based off of.
- **Approximate the source distribution:** We approximate the empirical distribution of weight traits with a normal kernel density estimate, with bandwidth set by Scott's rule. This produces a smoothed estimate of the underlying data distribution, which is particularly useful in case of sparsely sampled or highly skewed traits.
- **Determine a target distribution:** In this step, we determine the ideal distribution of the weight trait. Generally, this target distribution can be specified by some characteristics of the problem setting or dataset. For example, it may be prudent to reweight traits to match the population distribution, or to capture some notion of importance.
- **Determine weights:** We find a set of weights which transforms the source distribution into the target distribution. Formally, for a dataset $\{x_1, \ldots, x_n\}$ with corresponding traits $\{t_1, \ldots, t_n\}$, and an approximated source probability density f_S and target probability density f_T, we calculate the corresponding weights as $w_i = \frac{f_T(t_i)}{f_S(t_i)}$.

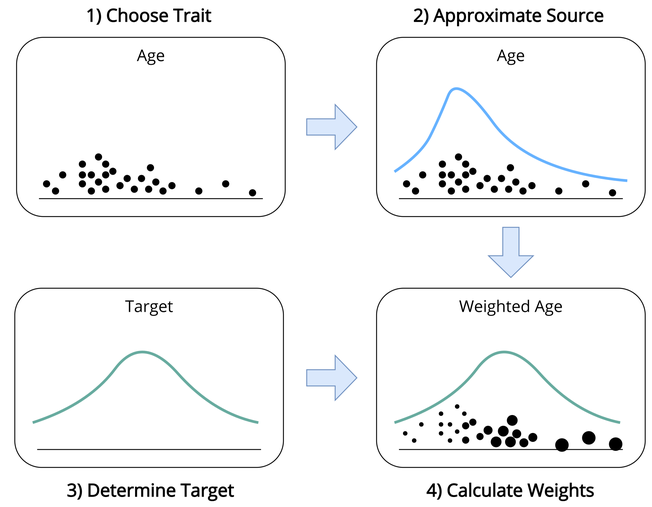


Figure 1: An illustration of continuous weighting in action.

Results

Model	No weights	Discrete	CWB
California Housing (R2 score)			
Random Forest regression	0.5156	0.5042	**0.5226**
Linear Regression	0.0476	0.2892	**0.3501**
Fully-connected network	0.4496	0.4237	**0.5066**
Heart Disease (AUROC)			
Random Forest classification	0.9554	0.9593	**0.9602**
Logistic Regression	0.9261	0.9223	**0.9242**
Fully-connected network	**0.9375**	0.9267	0.9324

Table 1: Experiments on both datasets using continuous weight balancing (**CWB**), discrete balancing (**discrete**) and no weights. Metrics are reported on target out-of-sample subsets; patients under 60 for the heart disease dataset, and prices above 2 for the housing dataset.

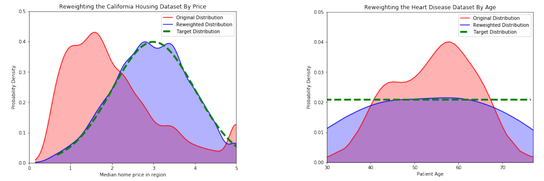
Key Contributions

1. We outline a method which approximates the underlying distribution of the continuous trait, and then chooses sample weights to adjust this distribution to an arbitrary target distribution.
2. We demonstrate the performance of our method on two canonical datasets – the California housing prices dataset and the heart disease dataset – and with three classes of models – regression, random forest, and shallow neural networks.
3. We provide an open-source and modular implementation of our method (https://github.com/Daniel-Wu/Continuous-Weight-Balancing).

Data

We run our experiments on two canonical datasets – one regression, and one classification. The California Housing dataset [2] contains the median housing prices of Californian census block groups in the 1990 census. It consists of 20,640 data points, where each data point contains 8 numeric attributes of houses in that block, and an accompanying median home price. For this dataset, we simply use the target variable – the natural log of median housing prices – as the weight trait. The dataset is skewed right, and we use a unit normal target distribution centered at 3 (Figure 2a).

The Cleveland Heart Disease Database [1], contains the clinical information of 303 patients undergoing angiography. each with 14 attributes. The target is a binary label indicating presence of heart disease. This dataset tests the limits of our method, as it is both extremely small and highly skewed. For this dataset we use age, which is slightly skewed left, as the weight trait, and we use a uniform target distribution (Figure 2b).



(a) Housing data reweighted to $\mathcal{N}(3, 1)$ **(b)** Heart data reweighted to $\mathcal{U}(29, 77)$

Figure 2: Our continuous reweighting method applied to our two datasets.

Discussion

In this work, we described a framework for continuous weight balancing, and assessed the performance of one simple way of doing so. While our method showed reasonable results, there are still many open questions about the best way to find continuous weights. What is the best way to approximate the source distribution? What classes of target distributions lead to the best performance, particularly with a given loss function? How do we strike the balance between skewed models, and models which memorize specific examples due to high magnitude weights? We hope that these and other questions may be answered in future work, in order to enable the development of robust models on skewed datasets.

References

[1] R. Detrano, A. Janosi, W. Steinbrunn, M. Pfisterer, J.-J. Schmid, S. Sandhu, K. H. Guppy, S. Lee, and V. Froelicher. International application of a new probability algorithm for the diagnosis of coronary artery disease. *The American Journal of Cardiology*, 64(5):304–310, 1989.

[2] R. K. Pace and R. Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297, 1997.