Raiders of the Loss Function

Jose Alvarez Cabrera

Peter Song

Minerva Schools at KGI

Abstract

Genetic Matching (GenMatch) is an automated covariate balance matching algorithm that, by default, relies on the lexical optimization of a vector of balance statistics produced by t-tests and KS-tests. Although it is becoming an increasingly popular tool for researchers as they attempt to find causal effects in observational studies, little to no research has been done on alternatives that improve the details of the default optimization algorithm and practitioners may not question this enough. We devised two original alternative loss functions that aim to enhance the optimization procedure of the GenMatch function in the Matching package for R. We replicated the setting and used the same example data sets as by Diamond & Sekhon (2013) to assess the performance of our original loss functions against the default. One of our original loss functions consistently showed encouraging results.

I. Introduction

What comprises the quality of an automated procedure greatly is *what* it optimizes, and *how* it optimizes it. When GenMatch was first proposed by Diamond & Sekhon (DS), the *what* was a breakthrough. At that time, even prominent researchers had not critically assessed with enough scrutiny the limitations of Mahalanobis distance matching and propensity score matching (e.g., Dehejia & Wahba (1999), Smith & Todd (2005)), let alone explored a superior generalization of both[1]. Rosenbaum and Rubin (1983; 1984; 1985) and Rubin (1997) had shown that propensity score matching should asymptotically balance the covariates if the correct propensity score model was used. But the reality was often far from the ideal, as limitations of

[1] To be precise, Rosenbaum & Rubin (1985) did suggest including the propensity score as an individual covariate in Mahalanobis distance matching, but did not propose the further use of weights.

sample size and nonexact matching easily screwed it all up (Diamond & Sekhon, 2013; Rosenbaum & Rubin, 1985). Also, both in normal propensity score matching and Mahalanobis distance matching, it is not obvious (if not close to impossible) to know how to deterministically determine either the *true* propensity score model or the *optimal* way to measure the distance between units. All this, plus the suboptimal basic standards to measure covariate balance that used to be accepted across the literature, made matching procedures fall far from their full potential.

GenMatch got the *what* right by generalizing these methods cleverly, by proposing "a range of distance metrics to find the particular measure that optimizes postmatching covariate balance" (Diamond & Sekhon, 2013); and offered an smart solution to the *how* by automatizing the search of the optimal weights through a genetic search algorithm that *used not only t-tests to measure the level of balance on the covariates, but also KS-tests* that better accounted for differences on the overall shape of the paired distributions. It is hard to believe today how long debates about the limits of matching procedures (e.g., Dehejia & Wahba (1999) vs Smith & Todd (2005)) were largely hindered because of the terrible basic standards that used to be accepted to measure covariate balance. The inclusion of KS-tests and the demonstration of improved performance at assessing the level of balance (Diamond & Sekhon 2013) were indeed a simple yet valuable addition.

But an automated procedure is only as reliable as to the extent to what its optimization algorithm is able to consistently find the optimal solution. For proving how misguided the old

methods were and how better GenMatch was, many details of the *how* devised by DS were way more than enough.

However, arguably little justification was given by DS on why the default loss function was chosen (i.e., lexical optimization of the minimum p-value from t-tests and KS-tests) (Diamond & Sekhon, 2013; Sekhon, 2011). As acknowledged by DS, there is no consensus on the best way or statistical tests to measure covariate balance or an expectation from them for the default tests to be necessarily the best. We reviewed the recent literature, and it seems that neither theorists nor practitioners have questioned this enough.[2]

The focus of our research is to enhance the default loss function (i.e., alternatives to performing lexical optimization of the vector of p-values produced by the default statistical tests). We do recognize that the exploration of alternatives to the default fitness function (i.e., alternatives to paired t-tests and KS-tests) are also worth trying - *if not even more* - but think that it is best to explore enhancements to the *how* in GenMatch one step at a time.

II. Methods

The documentation from the Matching package for R (Sekhon, 2011) details how to provide a different value for the loss function. The user specifies a function that receives a vector of statistics (e.g., p-values) as input and outputs a *loss value* to optimize (or a vector of). By default, the "loss" parameter is valued to 1, which corresponds to the function *sort()* (i.e., lexical

[2] Among the observational studies that use genetic matching that we reviewed, only Frey's *The economic burden of schizophrenia in Germany: A population-based retrospective cohort study using genetic matching* went beyond relying on the defaults to assess balance, by showing a visualization of a Q-Q plot. The rest merely discuss the parameters chosen, and even less justify the usage of the defaults - e.g., Lindlbauer's "*Changes in technical efficiency after quality management certification*" and Wood's "*Safety evaluation of continuous green T intersections*."

optimization of a vector is performed), but it also may be valued to the secondary default of 2, which corresponds to *min()* (i.e., direct optimization of the highest min p-value)[3]. Any other specified value is interpreted as (and expected to be) a reference to a function defined by the user.

To explore whether taking into consideration *only* the highest minimum p-value is misguided, we devised a series of original loss-functions that consider the entire vector of statistics *and* vary the algorithm to provide a *loss value*. For example, *my.loss.function.9* takes the weighted sum of the vector of balance statistics[4], where the heavier weights are given to the smallest values, with quadratic decrements. We will expand on this and other functions in section III.

To assess the performance of our series of alternative loss functions, we compared their matching estimates to the ones yielded by the default loss function used by DS (2013), relative to the experimental benchmarks, on each of the different versions of the canonical National Supported Work (NSW) demonstration data set used by Lalonde (1986) - i.e., the original Lalonde sample (Lalonde, 1986), the Dehejia & Wabba (DW) sample (Dehejia & Wabba, 1999), and the early random assignment (RA) sample (Smith & Todd, 2005). We used the CPS-1 observational "fake" controls to do the matching[5]; see DS (2013) for further details.

---

[3] Originally, this was the way that GenMatch() was implemented (a loss function based on *min()*) (Sekhon, 2011). Although we may recall that lexical optimization, in fact, still optimizes the highest minimum p-value, unless there is a tie with the second highest minimum (in which case it would try to optimize the third, fourth,... highest minimum as long as the given statistic has a different value that breaks the tie).

[4] We also considered the plain sum, but it did not yield promising results. This was expected, as taking into as much consideration the highest p-values in the vector of statistics is not likely to be helpful; as, often, such values correspond to the covariates that are easy to balance (and, in fact, already are).

[5] DS (2013) also used PSED-1, an observational data set that produced that generally worse estimates. We recognize that using it would allow an opportunity for further testing and research.

If any of our loss functions is better, we would expect its estimates to be consistently closer to the benchmark. The reasoning is that a better loss function should yield a more effective optimization procedure, which in turn, *if it is possible*, should yield a matched data set with a higher balance. Matched datasets with higher degrees of balance are assumed to enable observational estimates that are closer to the experimental benchmarks. Hence, what we mean with higher degrees of balance is not necessarily only evaluated based on the obtained after matching minimum p-value, as primarily done by DS (2013)[6].

The next section summarizes two alternative loss functions that execute an alternative algorithm to produce a loss value, taking as input of balance statistics a vector of p-values produced by paired KS-tests and t-tests proposed by DS (2013).

III. Original Alternative Loss Functions

A. Weighted Sum of Balance Statistics, with Polynomial Decrements

We propose a weighted sum of the entire vector of balance statistics as a fitness measure. We propose a sum because this statistic produces a score that takes into account the contribution of all the balance statistics. We use the weighted sum because, in general, we agree that small improvements on the covariates that are more difficult to balance ought to be taken into greater consideration than improvements on the covariates that are easy to balance.

[6] We recall that DS (2013) may have never claimed to believe that such measure is an ultimate measurement of the degree of balance. In fact, they emphasize that there is no consensus in the literature on how to measure covariate balance (both in the sense of the test(s) used (e.g., t-tests and KS-tests) and the primary summary statistic (e.g., minimum p-value) to rely on). Still, DS (2013) have largely relied on it as a reliable point of reference for comparing the balance obtained across multiple matched data sets.

We tested different variations of this idea. *my.loss.function.6* is the plain sum of p-values (without weights); *my.loss.function.7* is the weighted sum with linear decrements of the weights as higher p-values are summed; *my.loss.function.9* is the weighted sum with quadratic decrements, and *my.loss.function.10* is the weighted sum with cubic decrements. Among these variations, quadratic decrements worked the best (**i.e., *my.loss.function.9***).

B. Weighted Sum of Differences, with Polynomial Decrements

A second family of alternative loss functions we propose is based on a different key idea. Instead of summing the vector of balance statistics to determine how good the overall balance is, we add up how much each balance deviates from the ideal. In other words, if a p-value is 0.25, the corresponding difference is 1 - 0.25 = 0.75, i.e., how far such balance statistic is from the ideal level of balance for the respective covariate, which is 1 (in which case the difference would be 1-1 = 0).

The main advantage of this approach is that it effectively allows to *ignore completely* the covariates on which GenMatch has already achieved great balance. This is desirable since it allows the optimization to focus *only* on the covariates with balance statistics that are "far" from one.

Similar to the previous family of alternative loss functions based on the direct sum of the values in the vector, the sum of differences performed better as a weighted sum of differences. In particular, as a weighted sum of differences with quadratic decrements on the weights (**i.e., *my.subtraction.loss.function.3***).

The next section discusses the performance of *my.loss.function.9* (quadratically weighted sum of balance statistics) and *my.subtraction.func.3* (quadratically weighted sum of differences), relative to the default loss function in GenMatch within the context of the different versions of the Lalonde data set.

IV. **Matching Results**

TABLE 1- Matching Results (DW Sample)

Data	**Method**	**Balance Measure**	**Point Estimate**
DW Sample (Benchmark)	Experiment		$1,794
CPS-1	GenMatch - default loss function (reported by DS)	Highest min p.value = 0.21	$1,734
CPS-1	GenMatch - default loss function *(replicated)*	Highest min p.value = 0.28	$1,470
CPS-1	GenMatch - *my.loss.function.9*	Highest min p.value = 0.14	$1,957
CPS-1	GenMatch - *my.subtraction.loss.func.3*	Highest min p.value = 0.21	$1,857

TABLE 2 - Matching Results (Early RA Sample)

Data	**Method**	**Balance Measure**	**Estimate**
Early RA Sample (Benchmark)	Experiment		$2,748
CPS-1	GenMatch - default loss function (reported by DS)	Highest min p.value = 0.46	$1,631
CPS-1	GenMatch - default loss function *(replication)*	Highest min p.value = 0.31	$1,330
CPS-1	GenMatch - *my.loss.function.9*	Highest min p.value = 0.012	$1,237
CPS-1	GenMatch - *my.subtraction.loss.func.3*	Highest min p.value = 0.30	$1,710

TABLE 3 - Matching Results (Original Lalonde Sample)

Data	**Method**	**Balance Measure**	**Estimate**
Lalonde Sample (Benchmark)	Experiment		$886
CPS-1	GenMatch - default loss function (reported by DS)	Highest min p.value = 0.23	$281
CPS-1	GenMatch - default loss function *(replication)*	Highest min p.value = 0.22	$87
CPS-1	GenMatch - *my.loss.function.9*	Highest min p.value = 0.12	$182
CPS-1	GenMatch - *my.subtraction.loss.func.3*	Highest min p.value = 0.30	$260

TABLES 1-3. Matching results induced by the different loss functions for each of the three subsamples of the Lalonde data set and their balance measures.

My.subtraction.loss.func.3 (i.e., quadratically weighted sum of differences) reached a better estimate (i.e., closer to the benchmark) and a higher minimum p-value than *my.loss.function.9* (i.e., quadratically weighted sum of balance statistics) in the three subsamples of the Lalonde data set.

Compared to the results induced by the default loss function reported by DS, *my.subtraction.loss.func.3* did not do poorly. For the DW subsample, it attained an estimate of $1,857 in comparison to DS's reported $1,734 (relative to an experimental benchmark of $1,794) (see Table 1). For the early RA subsample, it attained an estimate of $1,710 in comparison to DS's reported $1,631 (relative to a benchmark of $2,748); even with a considerably lower after matching minimum p-value (see Table 2). For the original Lalonde sample, it attained an estimate of $260 in comparison to DS's reported of $281 (relative to a benchmark of $886) (see Table 3). We recall that this is in comparison to the matching estimates *reported* by DS. A more fair comparison for *my.subtraction.loss.func.3* would be to contrast its results against the ones induced by the default loss function under the same exact settings of our replication. In other words, to compare them where we were able to make sure that only the 'loss' parameter varied and every other nuance in the code was held constant.

Thereby, under the setting of our replication, the estimates induced by *my.subtraction.loss.func.3* outperformed the ones induced by the default loss function on each of the three subsamples (see Table 1-3), even when the obtained after matching minimum p-value was lower. The more radical difference occurred in the original Lalonde sample, where *my.subtraction.loss.func.3* induced an estimate of $260 while the default loss function induced $87 (Table 3). We also judge the other differences to be substantial (see Tables 1-3).

The results also highlight what we recalled before: the minimum p-value is a very limited measure to judge covariate balance[7]. Across tables 1-3, it is apparent that the accuracy of the estimate is not consistent with how high this balance statistic was.

V. Conclusion & Areas of Further research

We do not claim the previous results to be a breaking point to decide whether *my.subtraction.loss.func.3* is a better loss function (or not). It is possible that *my.subtraction.loss.func.3* is definitely more efficient than the default, as it was able to perform very well with arguably modest initial parameters (e.g., "population.size = 250", "max.generations = 75"), whereas the default was not. But assurance of greater efficiency does not necessarily mean assurance of more effectivity. We believe that further testing is needed.

Recalling that the minimum p-value is a balance statistic of limited insight, a future improvement to this research design may be to not only report the after matching minimum p-value as a statistic to inform the level of balance but also to provide the difference between the bias adjusted and non bias adjusted point estimates (i.e., the bias). Better matching should, in general, reduce bias.

Even more, loss functions combined with extensions to the fitness function that consider not only (or not at all[8]) p-values produced by paired t-tests and KS-tests, but rather the bias reduction itself may also be promising.

---

[7] This is currently the default primary metric reported by the MatchBalance() function in the Matching package in R.

[8] There also exist alternatives to the t-test that do not need to assume normality. Sawilowsky (2005) discusses the use of nonparametric alternatives to the t-test (e.g., Wilcoxon Mann-Whitney test); particularly on the context of testing for shifts in location of the distributions - which is the main concern that t-tests aim to address within the context of GenMatch.
In addition, Sekhon (2011) discusses a range of metrics based on the use eQQ differences as an alternative to the KS-test.

References

Austin, P. (2009). Balance Diagnostics for Comparing the Distribution of Baseline Covariates between Treatment Groups in Propensity Score Matched Samples. *Statistics in Medicine.*

Dehejia, R. & Wahba, S. (1999). Practical Propensity Score Matching: A Reply to Smith and Todd. *Journal of Econometrics*.

Diamond, A. & Sekhon, J.S. (2013). Genetic matching for estimating causal effects: A general multivariate matching package for achieving balance in observational studies. *Review of Economics and Statistics*, 95(3), 932-945.

Lalonde, R. (1986). Evaluating the Econometric Evaluations of Training Programs with Experimental Data. *American Economic Review.*

Sawilowsky, S. S. (2005). Misconceptions leading to choosing the t-test over the Wilcoxon Mann-Whitney U test for shift in location parameter. Journal of Modern Applied Statistical Methods, 4(2), 598-600.

Sekhon, J. S. (2011). Multivariate and propensity score matching software with automated balance optimization: the matching package for R.

Smith, J. & Petra E., Todd. (2005). Reconciling Conflicting Evidence on the Performance of Propensity Score Matching Methods. *AEA Papers and Proceedings.*

Rosenbaum, P., & Rubin, D. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika, 70*(1). doi:10.2307/2335942

Rosenbaum, P. & Rubin, D. (1984). Reducing Bias in Observational Studies Using Subclassification on the Propensity Score. *Journal of the American Statistical Association*

Rosenbaum, P., & Rubin, D. (1985). Constructing a control group using multivariate matched sampling methods that incorporate the propensity score. The American Statistician, 39(1), 33-38

Rubin, D. B. (1997). Estimating causal effects from large data sets using propensity scores. *Annals of internal medicine*, *127*(8_Part_2), 757-763.

Appendix A

GitHub Repository: https://github.com/josealvarez97/The-Raiders-of-the-Loss-Function

Which includes:

- A README.md with an overview that walks the reader around the replication files and other relevant documents.
- **An explicit list and comprehensive of 5 .R replication files, with an accompanying description (Recommended).**
- Data files (both in .txt and .dta format) used to produce the three different subsamples of the Lalonde data set.
- **The code that defines *my.loss.function.9* and *my.subtraction.loss.func.3*.**