CBPS: a Stata command to implement Covariate Balancing Propensity Score

Filip Premik*

Abstract

A dual nature of propensity score manifests itself in being both a conditional probability of treatment assignment and a covariate balancing score. The standard approach in propensity score estimation exploits the former feature leaving balancing properties to be checked after estimation. Imai and Ratkovic (2014) exploit both, proposing an estimator that automatically balances the conditional distribution of covariates (CBPS, covariate balancing propensity score). Obtained through a GMM estimation, CBPS is a convenient way to obtains propensity score estimates or weights to be used in subsequent estimations. Monte Carlo experiments confirm its good performance in reducing bias of treatment effects estimates. This paper reviews the method and introduces Stata user-written package `CBPS` which implements the estimator.

1 Introduction

Conditional independence assumption is required in a vast majority of non-experimental treatment evaluations. Rosenbaum and Rubin (1983) showed that if it is satisfied while conditioning on a set of covariates, say X_i, then it is also satisfied conditional on a single-valued function of these covariates. Since this seminal paper, the literature has flourished,

*The author gratefully acknowledges the support of the Polish National Science Centre (grant #2016/23/N/HS4/03637).

exploiting the single-valued fuction in the form of the conditional probability of treatment assignment, called in the literature the propensity score. Propensity score is of a particular interest for the matchng estimators, as it overcomes the curse of dimensionality. Additionally, researchers find it useful to produce weights balancing distributions of observations.

However, misspecification in the propensity score model might lead to severely biased estimates of treatment effects (Smith and Todd, 2005). Since in vast majority of applications the correct model is unknown, evaluating the quality of the treatment effect estimators remains a challenge. Given a successful estimation, distributions of covariates in the treated and non-treated cells of propensity score should be statistically equal. Any significant discrepancy might indicate either the misspecification of probability model or a failure of CIA assumption (Caliendo and Kopeinig, 2008). Therefore one of the suggested checks of the quality of propensity score estimates is a covariate balance check (Dehejia and Wahba, 2002). It relies on distinguishing narrow intervals of propensity score values and comparing the mean and variance of the covariates between the treated and untreated units within these intervals.

Imai and Ratkovic (2014) propose to systematically exploit the moments of the distribution of covariates for the treated and untreated units. They show theoretically that GMM estimation of the propensity score imposes balance on the conditional distributions. They exploit the dual nature of propensity score which is both the conditional probability of treatment assignment and covariate balancing score. Since this estimator is based on moments of the distribution, the functional form is not relevant for identification, making the estimator robust to model misspecification. The estimator is called Covariate Balancing Propensity Score (CBPS).

The major applications of the CBPS estimator concern obtaining propensity score estimates or weights that may be used in subsequent steps of estimation, e.g. matching analysis or weighted regressions. It makes CBPS a useful tool in many fields of applied econometrics. In this paper we focus on causal framework. The paper is structured as follows. Section 2 provides theoretical background of CBPS estimator, section 3 gives the details of Stata user-written package `CBPS` that implements CBPS, section 4 describes saved results of the estimation commands, section 5 provides some empirical examples, section 6 concludes.

2 Covariate Balancing Propensity Score

In this section a brief note on CBPS derivation is delivered (for more details see Imai and Ratkovic, 2014). Depending on which treatment effect is the parameter of interest, researchers may choose between average treatment effect (ATE) and average treatment effect on the treated (ATT) version of CBPS estimation.

We use the following notation. Assume a sample with N observations. T_i denotes a binary treatment assignment for a unit i and $\sum_i T_i = N_1$. X_i is a K dimensional vector of covariates, $F_\beta(X_i)$ and $f_\beta(X_i)$ are respectively cumulative and probability distribution functions.

2.1 ATE

We start with the version of CBPS that reweighs both treated and control subpopulations. It is particularly relevant if ATE is the effect of main interest.

Let g be a measurable function of X_i. Consider a set of moment conditions:

$$\mathbb{E}\Big[\frac{T_i g(X_i)}{F_\beta(X_i)}\Big] = \mathbb{E}\Big[\frac{(1-T_i)g(X_i)}{1-F_\beta(X_i)}\Big] \tag{1}$$

Conditionally on X_i it is satisfied for any $g(\cdot)$, as $\mathbb{E}[T_i|X_i] = F_\beta(X_i)$ (i.e. propensity score is a conditional probability of treatment assignment). It does not depend on any functional form for $F_\beta(X_i)$ and assures proper balance even if the conditional independence assumption does not hold, improving balance of observed X_i regardless of the unobserved variables (Imai and Ratkovic, 2014). Hence, the desired robustness to the functional form misspecification is assured. Additionally, one may perceive it as a balancing condition which balances weighted distributions of $g(\cdot)$ in treated and non-treated subpopulations (i.e. propensity score is a balancing score).

The easiest way to think about the CBPS is to take $g(X_i) = X_i$ and by that construct a weight which would make averages of covariates in treated and control groups equal. Rewriting equation (1):

$$\mathbb{E}\Big[\frac{T_i - F_\beta(X_i)}{F_\beta(X_i)(1-F_\beta(X_i))}X_i\Big] = 0 \tag{2}$$

Equation (2) provides the set of K moment conditions sufficient to just-identify K parameters of the propensity score function. Additionally, it defines also a balancing weight:

$$w_\beta(T_i, X_i) = \frac{T_i - F_\beta(X_i)}{F_\beta(X_i)(1 - F_\beta(X_i))}.$$

Given the appropriate moment conditions, Imai and Ratkovic (2014) define CBPS in terms of standard GMM framework:

$$\hat{\beta}^{GMM} = \arg\min_{\beta} \quad \mathbb{E}[g_\beta(T_i, X_i) \cdot W_\beta(T_i, X_i) \cdot g_\beta(T_i, X_i)] \tag{3}$$

where $g_\beta(T_i, X_i) = \frac{T_i - F_\beta(X_i)}{F_\beta(X_i)(1-F_\beta(X_i))} X_i$ is a vector of moment conditions of interest, $W_\beta(T_i, X_i)$ is weighting matrix.

$$W_\beta(T_i, X_i) = \mathbb{E}[g_\beta(T_i, X_i) g_\beta(T_i, X_i)'].$$

W depends among others on β, leading to a so called continuous updating GMM (Hansen et al., 1996). Despite clear advantages of that method, it may lead to severe numerical difficulties throughout the estimation process. To address this problem, `CPBS` implements also a a two-step estimator. Sample weighting matrix W is calculated using preliminary $\hat{\beta}^{MLE}$ obtained in the first step maximum likelihood estimation (depending on the functional form chosen, logit or probit model) and remains fixed in subsequent the GMM estimation. In turn, the covariance matrix of moments is obtained using the same formula as weighting matrix, $\Sigma_\beta(X_i, T_i) = \mathbb{E}[g_\beta(T_i, X_i) g_\beta(T_i, X_i)']$ However, here the already estimated $\hat{\beta}^{GMM}$ is used.

$\hat{\beta}^{GMM}$ has asymptotically normal distribution with covariance matrix:

$$\frac{1}{N}(G'WG)^{-1}G'WSWG(G'WG)^{-1}.$$

2.2 ATT

If the effect of interest is ATT, a researcher wants to adjust the distribution of the control group such that it matches average characteristics of treated units. Consider a population moment equality:

$$\mathbb{E}\Big[T_i g(X_i)\Big] = \mathbb{E}\Big[\frac{F_\beta(X_i)(1 - T_i) g(X_i)}{1 - F_\beta(X_i)}\Big] \tag{4}$$

Conditionally on X_i it is satisfied for all measurable functions $g(\cdot)$. Define $g(X_i) = X_i$. It provides K moment conditions of the form:

$$\mathbb{E}\Big[\frac{T_i - F_\beta(X_i)}{1 - F_\beta(X_i)} X_i\Big] = 0. \qquad (5)$$

which is sufficient to just-identify K parameters of the propensity score function $F_\beta(X_i)$. The balancing weights are given by:

$$w_\beta(T_i, X_i) = \frac{N}{N_1}\frac{T_i - F_\beta(X_i)}{1 - F_\beta(X_i)}.$$

The ATT version of CBPS is a GMM estimator which exploits moment conditions from the equation (5) with a weighting matrix

$$W_\beta(T_i, X_i) = \mathbb{E}[g_\beta(T_i, X_i) g_\beta(T_i, X_i)'].$$

2.3 Overidentification

Imai and Ratkovic (2014) emphasize close relation between ATE-version CBPS moment conditions and the maximum likelihood framework of standard binary probability models. Setting $g(X_i) = f_\beta(X_i)$ in equation (1) one obtains immediately the first order conditions of a maximum likelihood estimand. Therefore, it is possible to impose additional moment conditions that would overidentify the model:

$$\mathbb{E}\Big[\frac{T_i f_\beta(X_i)}{F_\beta(X_i)}\Big] = \mathbb{E}\Big[\frac{(1 - T_i) f_\beta(X_i)}{1 - F_\beta(X_i)}\Big] \qquad (6)$$

Accordingly, the set of moment conditions for ATE and ATT is given by :

$$g_\beta(T_i, X_i)^{ATE} = \begin{bmatrix} \frac{T_i - F_\beta(X_i)}{F_\beta(X_i)(1 - F_\beta(X_i))} f_\beta(X_i) \\ \frac{T_i - F_\beta(X_i)}{F_\beta(X_i)(1 - F_\beta(X_i))} X_i \end{bmatrix} = 0 \qquad (7)$$

$$g_\beta(T_i, X_i)^{ATT} = \begin{bmatrix} \frac{T_i - F_\beta(X_i)}{F_\beta(X_i)(1 - F_\beta(X_i))} f_\beta(X_i) \\ \frac{N}{N_1}\frac{T_i - F_\beta(X_i)}{1 - F_\beta(X_i)} X_i \end{bmatrix} = 0 \qquad (8)$$

The rest of estimation follows the steps described in previous subsections. However, the number of moment conditions is now $2K$, whereas the number of parameters to estimate is only K. Overidentified estimators are supposed to be more efficient in large samples at the cost of increased biased in finite samples (Imai and Ratkovic, 2014). Additionally, weights obtained from overidentified model do not balance perfectly the conditional distributions, which is why they may not be of primary interest for further use. Nevertheless, overidentified model enables a researcher to perform Sargan-like test of overidentifying restrictions, which is basically a specification test. A rejection of the null may suggest that conditional independence assumption does not hold e.g. because of omitted covariates or a failure to control for heterogeneity among units.

2.4 Balancing properties

Imai and Ratkovic (2014) cite two measures of covariate imbalance. The overall covariate imbalance (Rosenbaum and Rubin, 1985) is calculated as follows:

$$\left[\Big(\frac{1}{N}\sum_{i=1}^{N} w_{\beta \hat{GMM}}(T_i, X_i)X_i\Big)'\Big(\frac{1}{N}\sum_{i=1}^{N} X_i X_i'\Big)^{-1}\Big(\frac{1}{N}\sum_{i=1}^{N} w_{\beta \hat{GMM}}(T_i, X_i)X_i\Big)\right]^{.5} \tag{9}$$

and the standardized bias for treated units:

$$\left[\Big(\frac{1}{N}\sum_{i=1}^{N} w_{\beta \hat{GMM}}(T_i, X_i)X_i\Big)'\Big(\frac{1}{N_1}\sum_{i=1}^{N} T_i X_i X_i'\Big)^{-1}\Big(\frac{1}{N}\sum_{i=1}^{N} w_{\beta \hat{GMM}}(T_i, X_i)X_i\Big)\right]^{.5} \tag{10}$$

Postestimation command `CBPS_imbalance` calculates both of them. The overall and treated units imbalance statistics should equal zero for ATE and ATT just-identified estimates respectively. Therefore, they constitute a simple check if the optimization algorithm has found a proper minimum. They might also serve to verify whether imposing overidentifying restrictions has not been rejected by the data.

3 Stata implementation

The package to estimate CBPS consists of `CBPS` main command and two postestimation functions `predict` and `CBPS_imbalance`.

3.1 CBPS function

syntax

`CBPS` *depvar* [*indepvars*] [*if*] [*in*] [`, ate att logit probit over traceoff outputoff`

`optimization_technique(`*string*`) evaluator_type(`*string*`) ptol(#) vtol(#) nrtol(#)`]

main options

`ate` calculates the propensity score using ATE formulas.

`att` calculates the propensity score using ATT formulas (default option).

`logit` uses logistic cdf as a functional form for the propensity score (default option).

`probit` uses normal cdf as a functional form for the propensity score.

`over` adds overidentifying restrictions based on ML first order conditions.

display options

`traceoff` suppresses the iteration details of optimizing algorithm.

`outputoff` suppresses coefficient table.

programmer's options

`optimization_technique(`*string*`)` sets the optimization technique. See `optimize()` for more details; default is `optimization_technique("bgfs")`.

`evaluator_type(`*string*`)` sets the evaluator type for optimization. See `optimize()` for more details; default is `evaluator_type("gf1")`.

`ptol(#)`, `vtol(#)` and `nrtol(#)` set tolerance values for convergence. See `optimize()` for more details; default values are 1e-6, 1e-7 and 1e-8 respectively.

Note that the result of the overidentification test is delivered in `CBPS` any time a user estimates an overidentified model.

3.2 Predict after CBPS

syntax

`predict` saves scores after `CBPS` estimation. It may be either estimated propensity score or balancing weight.

`predict` *varname* [*if*] [*in*] [, `replace pscore bscore`]

options

`replace` overwrites *varname* if existed previously.

`pscore` calculates the estimated propensity score (default option).

`bscore` calculates the estimated balancing weights.

3.3 CBPS_imbalance

`CBPS_imbalance` calculates measures of covariate imbalance.

syntax

`CBPS_imbalance`

Scalars

`e(N)`	number of observations
`e(N1)`	number of treated observations
`e(N0)`	number of control observations
`e(J)`	J-Stat in χ^2 overid. test (added by `over` option)
`e(Jdf)`	degrees of freedom in the overid. test (added by `over` option)
`e(Jpval)`	p-val. in the overid. test (added by `over` option)
`e(i_ate)`	measure of overall imbalance (added by `CBPS_imbalance` command)
`e(i_att)`	measure of imbalance for the treated units (added by `CBPS_imbalance` command)

Macros

`e(cmd)`	`CBPS`
`e(cmdline)`	command as typed
`e(depvar)`	name of treatment assignment variable
`e(link)`	functional form chosen for propensity score
`e(predict)`	program used to implement `predict`
`e(properties)`	`b V`
`e(type)`	`ate` or `att`

Matrices

`e(b)`	coefficient vector
`e(V)`	covariance matrix of estimators

Functions

`e(sample)`	marks estimation sample

4 Saved results

5 Examples

This section contains two examples on how to use the `CBPS` command. They do not aim to prove superiority or inferiority of CBPS performance comparing to other estimators. For a discussion on these issues see Imai and Ratkovic (2014).

5.1 Graduate admission data

We begin with an example of data on graduate studies admissions. The file `admit.dta` may be easily downloaded at:

```
. use https://stats.idre.ucla.edu/stat/stata/dae/binary.dta, clear
```

The dependent variable is a binary indicator of candidate's admission to a graduate program (*admit*). We use candidate's GRE result, GPA and the rank of college they attended as covariates. The college rank is a qualitative variable, but `CBPS` fully supports factors. We suppress iteration output using option `traceoff`. Standard output of `CBPS` contains estimated parameters along with significance tests:

```
. CBPS admit gre gpa ib4.rank, traceoff
Covariate Balancing Propensity Score estimation
```

admit	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
admit						
gre	.0020149	.0012249	1.64	0.100	-.0003858	.0044156
gpa	.8082846	.3700951	2.18	0.029	.0829115	1.533658
rank						
1	1.568305	.4225452	3.71	0.000	.740132	2.396479
2	.8746031	.3677326	2.38	0.017	.1538605	1.595346
3	.2098293	.3939625	0.53	0.594	-.5623229	.9819815
_cons	-5.407959	1.140942	-4.74	0.000	-7.644165	-3.171753

Comparing the just-/over-identified models may be used to verify the specification test. Comparing ATE/ATT may be used to analyze the sources of the treatment effects. The following code produces desired result using `esttab` command:

```
. qui: CBPS admit gre gpa ib4.rank, ate

. qui: CBPS_imbalance

. qui: estimates store jate

. qui: CBPS admit gre gpa ib4.rank, over ate

. qui: CBPS_imbalance
```

```
. qui: estimates store oate

. qui: CBPS admit gre gpa ib4.rank

. qui: CBPS_imbalance

. qui: estimates store jatt

. qui: CBPS admit gre gpa ib4.rank, over

. qui: CBPS_imbalance

. qui: estimates store oatt

. esttab jate oate jatt oatt , mtitles("exact ate" "over ate" "exact att" "over att") sfmt
> (3) keep(gre) scalars(i_ate i_att J Jpval)
```

	(1)	(2)	(3)	(4)
	exact ate	over ate	exact att	over att
admit				
gre	0.00262*	0.00189*	0.00201	0.00224*
	(2.19)	(2.23)	(1.64)	(2.34)
N	400	400	400	400
i_ate	0.000	0.100	0.067	0.059
i_att	0.073	0.062	0.000	0.031
J		2.085		1.421
Jpval		0.912		0.965

```
t statistics in parentheses
* p<0.05, ** p<0.01, *** p<0.001
```

The imbalance statistics equal zero for estimates from just-identified models indicating numerical convergence of the optimizer. Overidentified models pass the specification test for any conventional confidence level.

5.2 Causal inference with the LaLonde data

Another example demonstrates applications of `CBPS` in a broader context. We consider causal effects of job training program introduction on earnings using well known LaLonde (1986) data from National Supported Work (NSW) with additional control observations from Current Population Survey (CPS) provided by Dehejia and Wahba (1999, 2002). Since the

NSW data is experimental, randomization assures all the assumptions necessary for the causal inference validity to hold even for conventional estimators consistent. Therefore, estimates on NSW subsample would constitute a benchmark for estimates on CPS combined with NSW sample (Dehejia and Wahba, 1999).

The pooled sample contains 16437 observations, out of which 445 come from the original NSW experiment and includes 185 treated individuals. We model the treatment assignment using pretreatment covariates. We are interested in the pre- and post-treatment difference.

First, we analyze ATE. We compare the results from a conventional regression of ΔY_i on treatment variable (OLS) and the full range of covariates with estimates from a regression of ΔY_i on the treatment variable weighted by balancing score produced by `CBPS` (WLS). The following code estimates just-identified ATE version of `CBPS` along with imbalance statistics and generates balancing score.

```
. * NSW sample
. local covs age education black hispanic married nodegree re74

. CBPS treat `covs´ if nsw, ate traceoff outputoff

. CBPS_imbalance
Total average covariate imbalance after cbps: 1.7e-04
Average covariate imbalance on the treated after cbps: .047

. predict bscore_nsw, p r

.
. * Polled sample
. CBPS treat `covs´, ate outputoff traceoff optimization_technique("nr")

. CBPS_imbalance
Total average covariate imbalance after cbps: 7.9e-05
Average covariate imbalance on the treated after cbps: 39.6

. predict bscore_all, p r
```

Average covariance imbalance on the treated using NSW sample amounts to 0.047 whereas the same statistic on the pooled sample is 39.6. Notably, in terms of the treated units the imbalance is by magnitudes higher in the latter sample. This result is expected, as CPS subsample comes from different survey and so describes a different population than NSW. Dehejia and Wahba (2002) provide more details on that issue.The overall balance is minimized to zero by applying the ATE.

```
. local covs age education black hispanic married nodegree re74

. reg dy treat `covs´ if nsw, noheader notable

. estimates store reg_nsw

. qui: reg dy treat [pw=bscore_nsw] if nsw

. estimates store wreg_nsw

. reg dy treat `covs´, noheader notable

. estimates store reg_all

. qui: reg dy treat [pw=bscore_all]

. estimates store wreg_all

. esttab reg_nsw wreg_nsw reg_all wreg_all, mtitles("NSW (OLS)" "NSW (WLS)" "pooled (OLS)"
>  "pooled (WLS)") keep(treat) nonumbers
```

	NSW (OLS)	NSW (WLS)	pooled (OLS)	pooled (WLS)
treat	1375.2*	1754.3*	1485.4**	1672.8*
	(2.05)	(2.44)	(2.60)	(2.29)
N	445	445	16437	16437

t statistics in parentheses
* p<0.05, ** p<0.01, *** p<0.001

The first two columns present results on NSW sample, the following two on the pooled sample. Estimates using CBPS-based weights are numerically higher on both samples. However, as the precision is pretty low, weighted regression (wls) estimates lie in the 95% confidence intervals of standard regression (ols). CBPS weighting leads to similar results as using covariates in the conventional regression. Moreover, adding CPS observations seems not to make affect much the performance of estimators.

Second, we analyze ATT. We obtain it using Heckman et al. (1997) difference-in-difference matching estimator. It is a two-step procedure. First, we estimate the propensity score. The results from `CBPS` will be compared to standard `logit` propensity score. Second, we apply the `psmatch2` command by Leuven et al. (2015) to run propensity score matching on ΔY_i. Since the purpose of this example is mainly to show how to apply the function, we choose only one specification of the estimator - nearest-neighbor matching estimator with .005 caliper - without justifying this choice from the econometric point of view. In addition, for ease of exposition, we use a wrapper to `psmatch2` command called `psmatch3`, which produces results with desired names. The code for this wrapper is attached in the appendix.

```
. local covs age education black hispanic married nodegree re74

. qui: logit treat `covs´ if nsw

. CBPS_imbalance
Total average covariate imbalance after logit: 5.3e-03
Average covariate imbalance on the treated after logit: .045

. predict lscore_nsw if nsw, p
(15992 missing values generated)

. qui: logit treat `covs´

. CBPS_imbalance
Total average covariate imbalance after logit: .725
Average covariate imbalance on the treated after logit: .036

. predict lscore_all, p

. CBPS treat `covs´ if nsw, traceoff outputoff

. CBPS_imbalance
Total average covariate imbalance after cbps: .033
Average covariate imbalance on the treated after cbps: 1.0e-04

. predict pscore_nsw if nsw, p r

. CBPS treat `covs´, outputoff traceoff optimization_technique("nr")

. CBPS_imbalance
Total average covariate imbalance after cbps: .743
Average covariate imbalance on the treated after cbps: 7.3e-06

. predict pscore_all, p r

. psmatch3 lscore_nsw if nsw

. estimates store logit_nsw

. psmatch3 pscore_nsw if nsw

. estimates store cbps_nsw

. psmatch3 lscore_all

. estimates store logit_all

. psmatch3 pscore_all

. estimates store cbps_all
```

```
. esttab logit_nsw cbps_nsw logit_all cbps_all, mtitles("NSW(logit)" "NSW(CBPS)" "pooled(l
> ogit)" "pooled(CBPS)") nonumbers keep(att)
```

	NSW(logit)	NSW(CBPS)	pooled(log~)	pooled(CBPS)
att	1140.6***	1085.6***	1319.4***	1112.5***
	(38.17)	(36.87)	(44.12)	(37.82)
N	445	445	16437	16437

t statistics in parentheses
* p<0.05, ** p<0.01, *** p<0.001

As noticed before, NSW estimates constitute a benchmark for estimates on the pooled sample. Since this is an *experimental* sample, any estimator should not suffer problems resulting from nonrandom selection. In fact, one may observe that results do not differ significantly regardless of which way of estimating the propensity score is chosen. However, once we add observations from a different survey (CPS), the estimate using logit propensity score departs quite a lot from the benchmark level whereas CBPS-based estimate is rather indistinguishable between the two samples. This result suggests that the DID estimator using CBPS is robust to using a imbalanced data sets.

6 Conclusion

CBPS estimator provides is a convenient way to deal with unbalanced data. Potential applications concern a broad scope of non-experimental issues. It is of particular use in causal inference, assisting in obtaining a reliable counterfactual distribution. CBPS estimates of the treatment effects are by construction perfectly balanced. The balancing weights may easily be used for any further analysis. Being robust to misspecification, CBPS estimates may limit the degree of bias in comparison with standard estimators using precalculated weights or propensity score (Imai and Ratkovic, 2014).

This paper introduces Stata user-written package `CBPS` to estimate Covariate Balancing Propensity Score. The syntax allows to obtain both the average treatment effect and the average treatment effect on treated. The syntax stores the estimated scores, they may be recovered through `predict` command.

References

Caliendo, M. and S. Kopeinig (2008): "Some practical guidance for the implementation of propensity score matching," *Journal of economic surveys*, 22, 31–72.

Dehejia, R. H. and S. Wahba (1999): "Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs," *Journal of the American Statistical Association*, 94, 1053–1062.

——— (2002): "Propensity score-matching methods for nonexperimental causal studies," *Review of Economics and Statistics*, 84, 151–161.

Hansen, L. P., J. Heaton, and A. Yaron (1996): "Finite-sample properties of some alternative GMM estimators," *Journal of Business & Economic Statistics*, 14, 262–280.

Heckman, J. J., H. Ichimura, and P. E. Todd (1997): "Matching as an econometric evaluation estimator: Evidence from evaluating a job training programme," *The review of economic studies*, 64, 605–654.

Imai, K. and M. Ratkovic (2014): "Covariate balancing propensity score," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76, 243–263.

LaLonde, R. J. (1986): "Evaluating the econometric evaluations of training programs with experimental data," *The American economic review*, 604–620.

Leuven, E., B. Sianesi, et al. (2015): "PSMATCH2: Stata module to perform full Mahalanobis and propensity score matching, common support graphing, and covariate imbalance testing," *Statistical Software Components*.

Rosenbaum, P. R. and D. B. Rubin (1983): "The central role of the propensity score in observational studies for causal effects," *Biometrika*, 70, 41–55.

——— (1985): "Constructing a control group using multivariate matched sampling methods that incorporate the propensity score," *The American Statistician*, 39, 33–38.

Smith, J. A. and P. E. Todd (2005): "Does matching overcome LaLonde's critique of nonexperimental estimators?" *Journal of Econometrics*, 125, 305–353.

Appendix

code of the command psmatch3

```
. capture program drop psmatch3

. program define psmatch3, eclass
  1.         syntax varlist [if]
  2.         marksample touse
  3.         qui : psmatch2 treat `if´ , outcome(dy) pscore(`1´) common  caliper(.005)  ne
> ighbor(1)
  4.         scalar v_att=r(seatt)
  5.         scalar r_att=r(att)
  6.         matrix b=r_att
  7.         matrix V=v_att
  8.         matrix colnames b="att"
  9.         matrix colnames V="att"
 10.         matrix rownames V="att"
 11.         local n=e(N)
 12.         eret post b V, obs(`n´) esample(`touse´)
 13.         eret local cmd "psmatch3"
 14. end
```