Design Flaw of the Synthetic Control Method

Kuosmanen, Timo and Zhou, Xun and Eskelinen, Juha and Malo, Pekka

28 February 2021

Design Flaw of the Synthetic Control Method

Timo Kuosmanen* Xun Zhou† Juha Eskelinen‡ Pekka Malo§

March 2021

Abstract

Synthetic control method (SCM) identifies causal treatment effects by constructing a counterfactual treatment unit as a convex combination of donors in the control group, such that the weights of donors and predictors are jointly optimized during the pre-treatment period. This paper demonstrates that the true optimal solution to the SCM problem is typically a corner solution where all weight is assigned to a single predictor, contradicting the intended purpose of predictors. To address this inherent design flaw, we propose to determine the predictor weights and donor weights separately. We show how the donor weights can be optimized when the predictor weights are given, and consider alternative data-driven approaches to determine the predictor weights. Re-examination of the two original empirical applications to Basque terrorism and California's tobacco control program demonstrates the complete and utter failure of the existing SCM algorithms and illustrates our proposed remedies.

Keywords: Causal effects; Comparative case studies; Policy impact assessment; Treatment effect models

JEL Codes: C54; C61; C71

* *Corresponding author.* Department of Information and Service Management, Aalto University School of Business, 02150 Espoo, Finland. Email: timo.kuosmanen@aalto.fi, Tel.: +358947001.

† Chair for Agricultural Production and Resource Economics, Technical University of Munich, Alte Akademie 14, 85354 Freising, Germany and Department of Information and Service Management, Aalto University School of Business, 02150 Espoo, Finland. Emails: xun.zhou@tum.de; xun.zhou@aalto.fi.

‡ Department of Information and Service Management, Aalto University School of Business, 02150 Espoo, Finland. Email: juha.p.eskelinen@aalto.fi.

§ Department of Information and Service Management, Aalto University School of Business, 02150 Espoo, Finland. Email: pekka.malo@aalto.fi.

1 Introduction

During the past two decades, the synthetic control method (SCM) has emerged as a popular tool for estimating causal effects of policy interventions and programs in a comparative case study setting. Abadie and Gardeazabal (2003) originally introduced this innovative approach to examine the economic impacts of Basque terrorism. Abadie et al. (2010) further developed the statistical foundations of the method in their study of California's tobacco control program. Subsequently, SCM has been used in a large number of influential applications, including Acemoglu et al. (2016) (political connections), Cavallo et al. (2013) (natural disasters), Gobillon and Magnac (2016) (enterprise zones), Kleven et al. (2013) (taxation of athletes), and Abadie et al. (2015) (German reunification), among others. Recently, Cole et al. (2020) apply SCM to study the impact of the Covid-19 lockdown on air pollution and health in Wuhan, China. There is clearly large and growing interest in this approach: Athey and Imbens (2017) refer to SCM as "*arguably the most important innovation in the policy evaluation literature in the last 15 years.*"

Technically, SCM estimates the treatment effect by constructing a counterfactual of the treated unit using a convex combination of similar units not exposed to the treatment. The convex combination requires non-negative weights that sum to one to avoid extrapolation. The weights are determined to ensure that the treated unit and the synthetic control resemble each other as closely as possible prior to the treatment, both with respect to the outcome of interest and some observed economic predictors. Since there are typically multiple predictors, the predictors are also weighted using another set of non-negative weights. In practice, virtually all published SCM applications resort to the data-driven procedure where the weights of predictors and control units are jointly optimized to minimize the mean squared prediction error of the synthetic control over the pre-treatment period. Ability of the synthetic control to closely match the pre-treatment outcomes of the treated unit is frequently cited as a highly appealing feature of SCM.

Almost all empirical SCM studies apply the *Synth* algorithm described in Abadie et al. (2011), which is available for R, Matlab, and Stata. However, several recent studies report rather disturbing findings, suggesting that the synthetic control weights produced by *Synth* may be numerically unstable and suboptimal (e.g., Becker and Klößner, 2017, 2018; Becker et al., 2018; Klößner et al., 2018).[1] A related but even more serious concern is that the predictors often turn out to have little impact on the synthetic control, as noted by several authors (e.g., Ben-Michael et al., 2018; Doudchenko and Imbens, 2017; Kaul et al., 2015). This is a disturbing concern because the statistical properties of the SCM estimator critically depend on the ability of the synthetic control to reproduce the observed and unobserved characteristics of the treated unit (Abadie, 2019; Abadie et al., 2010). If most predictors are typically assigned negligibly small weights, then the ability of SCM to reproduce the observed characteristics and the latent factors is seriously compromised.

The recent study by Malo et al. (2020) sheds new light on the computational difficulties noted above. Developing the first explicit mathematical formulation of the standard SCM problem where the predictor weights and the donor weights are jointly optimized, Malo et al. (2020) argue that the original SCM problem is in fact a NP-hard bilevel optimization problem. The good news for SCM is that a unique optimum exists; the problem is solvable. Malo et al. (2020) develop an iterative algorithm based on Tykhonov regularization, which is guaranteed to converge to the true optimal solution. The bad news is that *Synth* and other SCM algorithms known in the literature generally fail to converge to the optimum. As a result, several thousands of SCM applications published thus far are based on more or less suboptimal weights, which may affect the qualitative conclusions.

The purpose of the present paper is not only to demonstrate the computational failure, but also provide constructive suggestions of how the problems identified could be addressed. Our specific contributions are three-fold:

1) As a motivating example, we demonstrate that numerical instability of the *Synth* algorithm occurs even in the two original SCM studies of Basque terrorism by Abadie and Gardeazabal (2003) and California's tobacco control program by Abadie et al. (2010). Specifically, we show that random reordering of the donors and predictors affects the *Synth* results. Klößner et al. (2018) have previously noted similar numerical instability, but they misleadingly attribute the problem to the cross-validation approach suggested by Abadie et al. (2015). We show that their diagnosis is false: numerical instability of

[1] Abadie et al. (2011), Footnote 16, acknowledge that "*Depending on the exact setup of the data there exist situations in which the objective function may contain local minima, so that (as is routinely the case in these types of problems) there is no analytical guarantee that the derivative-based algorithms routinely used by optim() (i.e., Nelder-Mead and BFGS) will converge to the global minimum.*"

Synth is an even more wide-spread problem, affecting also the original SCM setting.

2) Applying insights from game theory, we explain why the optimal solution of the SCM problem is typically a corner solution where all weight is assigned to a single predictor. This is also the case in the two original SCM applications to Basque terrorism Abadie and Gardeazabal (2003) and the California tobacco control program Abadie et al. (2010). In our interpretation, the numerical instability of SCM is a symptom, but the tendency towards corner solutions is the underlying design flaw of SCM, caused by the joint optimization of donor weights and predictor weights. Developing better algorithms to solve the NP-hard bilevel optimization problem does not suffice to address the root cause of the problem.

3) To address the design flaw, we propose to determine the predictor weights and donor weights in two separate stages. We develop a simple two-step algorithm to optimize the donor weights when the predictor weights are given *a priori*. This proves a non-trivial task, in fact, we find that the *Synth* algorithm fails to produce optimal donor weights even when the predictor weights are given by the user. We also briefly explore alternative data-driven approaches to determine the predictor weights. These include the use of regression-based weights, which are used as starting values for the *Synth* algorithm (Abadie et al., 2011) and the default option in the Matlab implementation of *Synth*. Another possibility is to apply equal weights to standardized predictors, analogous to Bloom and Van Reenen's (2007) approach to aggregate management survey indicators. The use of equal weights has also been considered in some empirical SCM studies (e.g., Bohn et al., 2014). We illustrate the application of the regression-based and uniform weights in the case of the two original SCM applications to Basque terrorism and California tobacco control program.

Interestingly, the recent methodological advances in the SCM literature have largely focused on the simplified setting where the additional predictors are omitted, which is rather peculiarly referred to as the "canonical SCM" (e.g., Doudchenko and Imbens, 2017; Ben-Michael et al., 2018; Powell, 2018; Ferman et al., 2018; Chernozhukov et al., 2020). We suspect the computational problems and the design flaw identified in this paper might help to explain the recent "canonization" of the simplified approach, which Abadie and Gardeazabal

(2003) considered to be “less appropriate”. Indeed, the statistical basis of SCM strongly rests on the empirical fit to the predictors (Abadie et al., 2010; Abadie, 2019). In the opposite extreme, Abadie and L'Hour (2020) focus on optimizing fit with respect to the predictors, without explicitly considering the fit to the pre-treatment outcomes. Despite the major advances of the recent SCM literature, the question of how to reconcile the trade-off between the fit with respect to the predictors versus the fit with respect to the pre-treatment outcomes remains a major unresolved issue. We will focus on the original SCM setting in this paper, emphasizing that the findings of our study have implications to more recent methodological advances in the literature. We hope that our results might help to establish the SCM approach on a more solid foundation: it is clearly wrong to use suboptimal weights that are artifacts of a computational failure.

The rest of the paper is organized as follows. Section 2 briefly introduces the original SCM method and empirically demonstrates the instability of the *Synth* algorithm by reexamining the two original SCM applications. Section 3 presents the SCM problem as a bilevel optimization problem, discusses its game theoretic interpretation, and explains why the optimum is typically a corner solution. Section 3.3 demonstrates empirically that the two original SCM applications both have corner solutions, and that the existing *Synth* and MSCMT (Multivariate Synthetic Control Method using Time Series) algorithms fail to converge to the optimum. Section 4 proposes a simple two-step approach to optimize the donor weights when the predictor weights are given *a priori*, explores alternative data-driven approaches to determine the predictor weights, and revisits the two original SCM applications to illustrate the proposed approaches. Section 5 presents our concluding remarks and discusses avenues for future research. Additional numerical illustrations and details of data processing are provided in Appendices A, B, and C, respectively. Documentation of the essential source code (in R) is provided in Appendix D to allow readers to reproduce the iterative algorithm to check for the feasibility of the unconstrained optimum and the possibility of corner solutions, and to reproduce our two-step approach to optimize the donor weights when the predictor weights are empirically determined. The latest updates to the code and the technical documentation are available online at the GitHub page: https://github.com/Xun90/SCM-Debug.git.

2 Synthetic control method

2.1 Preliminaries

To estimate causal effects in a comparative case study setting, the outcomes of the unit affected by an event or intervention (the treatment group) are compared with the outcomes of one or more unaffected units (the control group). The rationale behind the SCM method is to use the control group's outcome to approximate the counterfactual outcome of the treated group in the absence of treatment. To this end, SCM constructs a synthetic control as a convex combination of multiple control units. The weights that determine the synthetic control are chosen to best approximate the relevant characteristics of the treated unit during the pre-treatment period. The post-intervention outcomes for the synthetic control unit are then used to estimate the outcomes that would have been observed for the treated unit in the absence of the intervention.

Suppose we observe units $j = 1, \ldots, J + 1$, where the first unit is exposed to the intervention and the J remaining units are control units that can contribute to the synthetic control. The set of J control units is referred to as the pool of *donors*. The number of time periods prior to the treatment is denoted as T^{pre} and the number of time periods after the treatment as T^{post}. For the sake of clarity, we indicate vectors with bold lowercase font and matrices with bold capital letters. The outcome of interest of the treated unit is denoted by $\mathbf{y}$: column vectors $\mathbf{y}_1^{\text{pre}}$ and $\mathbf{y}_1^{\text{post}}$ with T^{pre} and T^{post} rows, respectively, refer to the time series of the pre-treatment and post-treatment outcomes. Similarly, matrices $\mathbf{Y}_0^{\text{pre}}$ and $\mathbf{Y}_0^{\text{post}}$ with J columns refer to the pre-treatment and post-treatment outcomes of the control group, respectively.

Ideally, the impact of treatment could be measured as

$$\boldsymbol{\alpha} = \mathbf{y}_1^{\text{post}} - \mathbf{y}_1^{\text{post,N}}, \tag{1}$$

where $\mathbf{y}_1^{\text{post,N}}$ refers to the counterfactual outcome that would occur if the unit was not exposed to the treatment. If one could observe the outcomes $\mathbf{y}_1^{\text{post,N}}$ in an alternative state of nature where the unit was not exposed to the treatment, then one could simply calculate the elements of vector $\boldsymbol{\alpha}$. The main challenge in the estimation of the treatment effect is that only $\mathbf{y}_1^{\text{post}}$ is observable, whereas the counterfactual $\mathbf{y}_1^{\text{post,N}}$ is not.

The goal of SCM is to construct a synthetic control group to estimate the counterfactual $\mathbf{y}_1^{\text{post,N}}$. The key idea of the SCM is to use the convex combination of the observed outcomes of the control units $\mathbf{Y}_0^{\text{post}}$ as an estimator of $\mathbf{y}_1^{\text{post,N}}$. Formally, the SCM estimator is defined as

$$\hat{\boldsymbol{\alpha}} = \mathbf{y}_1^{\text{post}} - \mathbf{Y}_0^{\text{post}}\mathbf{w}, \tag{2}$$

where the J elements of column vector $\mathbf{w}$ are non-negative and sum to one. The weights $\mathbf{w}$ characterize the synthetic control, that is, a counterfactual path of outcomes for the treated unit in the absence of treatment.

The main challenge is to determine the weights $\mathbf{w}$. The simplest approach considered by Abadie and Gardeazabal (2003) is to track the observed path of pre-treatment outcomes as closely as possible to minimize the mean squared prediction error (MSPE). That is, one could apply the weights $\mathbf{w}$ that solve the following constrained least squares problem

$$\begin{aligned}
&\min_{\mathbf{w}} L = \frac{1}{T^{\text{pre}}}(\mathbf{y}_1^{\text{pre}} - \mathbf{Y}_0^{\text{pre}}\mathbf{w})'(\mathbf{y}_1^{\text{pre}} - \mathbf{Y}_0^{\text{pre}}\mathbf{w}) \\
&\text{subject to} \\
&\mathbf{1}'\mathbf{w} = 1 \\
&\mathbf{w} \geq \mathbf{0}
\end{aligned} \tag{3}$$

For transparency, we write the constraints on weights explicitly throughout the paper to remind a reader that we are dealing with a constrained optimization problem. The constraints on weights $\mathbf{w}$ ensure that the synthetic control is a convex combination of the control units in the pool of donors. The fact that SCM does not involve extrapolation is considered as one of its greatest advantages over regression analysis (e.g., Abadie, 2019).

Note that if we relax the constraints on weights $\mathbf{w}$ in (3), then the unconstrained minimization problem reduces to the classic OLS problem without the intercept term. In that case, one could simply regress the time series $\mathbf{y}_1^{\text{pre}}$ on the parallel outcomes of the J donors in the control group, and set the weights $\mathbf{w}$ equal to the corresponding OLS coefficients. While the OLS problem has the well-known closed form solution that satisfies the first-order conditions, however, the optimal solution to the constrained least squares problem must be solved numerically. To remind a reader about this fact, we write explicitly the linear constraints on the weights $\mathbf{w}$ in (3). The constrained least squares problem can be efficiently solved by quadratic programming (QP) algorithms such as CPLEX, Gurobi, or CVXOPT, which are guaranteed to converge to the global optimum.

In addition to the outcome of interest, an integral part of SCM is to utilize additional K variables referred to as *predictors* (also known as growth factors, characteristics, or covariates), which are observed prior to the treatment or are unaffected by the treatment, and which can influence the evolution of outcomes. These predictors are denoted by a $(K \times 1)$ vector $\mathbf{x}_1$ and a $(K \times J)$ matrix $\mathbf{X}_0$, respectively.[2] Abadie et al. (2010) prove unbiasedness and consistency of the SCM in the ideal case where the synthetic control yields perfect fit to the predictors, that is, $\mathbf{x}_1 = \mathbf{X}_0 \times \mathbf{w}$. Abadie (2019) acknowledges that "*In practice, the condition* $\mathbf{x}_1 = \mathbf{X}_0 \times \mathbf{w}$ *is replaced by the approximate version* $\mathbf{x}_1 \approx \mathbf{X}_0 \times \mathbf{w}$*. It is important to notice, however, that for any particular data-set there are not ex-ante guarantees on the size of the difference* $\mathbf{x}_1 - \mathbf{X}_0 \times \mathbf{w}$*. When this difference is large, Abadie et al. (2010) recommend against the use of synthetic controls because of the potential for substantial biases.*" The previous quotation aptly highlights the critical importance of achieving a good fit with respect to predictors.

Since the K predictors do not necessarily have the same effect on the outcomes, Abadie and Gardeazabal (2003) introduce predictor weights using a $(K \times K)$ diagonal matrix $\mathbf{V}$. For notational convenience, we denote $\mathbf{V} = \text{diag}(\mathbf{v})$ where $\mathbf{v}$ is a vector of predictor weights that reflects their relative importance. The K elements of $\mathbf{v}$ must be non-negative[3] and are usually normalized to sum to unity.[4] The optimal choice of $\mathbf{v}$ satisfies the solution to the

[2] A common practice in SCM is to include some convex combinations of the pre-treatment outcomes also as predictors (see Abadie et al., 2010, 2015, for discussion). However, Kaul et al. (2015) demonstrate that including all pre-treatment outcomes as predictors is not a good idea because the predictors become completely redundant in that case.

[3] While Abadie et al. (2010) assume that the diagonal elements of $\mathbf{V}$ must be positive, a positive real number can be arbitrarily close to zero, and therefore, the distinction between positive and non-negative model variables has no real meaning in optimization unless one imposes some explicit lower bound, e.g., $V_{kk} \geq 0.01$. Becker and Klößner (2018) set a lower bound $V_{kk} \geq 0.00000001$, which is so low that it has no practical meaning.

[4] Of course, other normalizations are possible, but we here restrict attention to the most standard normalization that allows one to interpret the elements of $\mathbf{v}$ as shared weights that sum to one.

following problem

$$\begin{aligned}
&\mathbf{v}^\star = \arg\min_{\mathbf{v}} (\mathbf{x}_1 - \mathbf{X}_0\mathbf{w}^\star(\mathbf{v}))'(\mathbf{x}_1 - \mathbf{X}_0\mathbf{w}^\star(\mathbf{v})) \\
&\text{subject to} \\
&\mathbf{1}'\mathbf{v} = 1 \\
&\mathbf{v} \geq \mathbf{0}
\end{aligned} \tag{4}$$

To compute the optimal predictor weights $\mathbf{v}^\star$, most SCM studies use the *Synth* package described in Abadie et al. (2011), which is available for R, Matlab, and Stata. Unfortunately, the *Synth* package is numerically unstable and unreliable, as the following example demonstrates.

2.2 Numerical instability of *Synth*

Recently Klößner et al. (2018) reported a rather disturbing finding that simply reordering the donors can have a major effect on the *Synth* results. They attributed the problem to the cross-validation approach by Abadie et al. (2015), but this diagnosis is false: the numerical instability occurs even in the original SCM setting without cross-validation.

To demonstrate our claim, let us first revisit the original SCM application by Abadie et al. (2010) to California's tobacco control program using the original data and the standard R implementation of *Synth*.[5] We compare the *Synth* results obtained using the original ordering of predictors and donors used by Abadie et al. (2010) with those obtained by randomly reordering the donors and predictors. More specifically, we draw 1,000 random orderings of the rows of matrix $\mathbf{X}_0$, and another 1,000 random orderings of the columns of matrix $\mathbf{X}_0$, while retaining all other features of the original data and using the default settings of *Synth*. Obviously, such random reordering of either the rows or columns of the data matrix does not affect the true optimal solution to the SCM problem in any way. However, it does influence the results produced by the *Synth* package, as Table 1 demonstrates.

[5] We assume that the reader is familiar with the original SCM applications; we refer to Abadie and Gardeazabal (2003) and Abadie et al. (2010) for a more detailed description of the donors and predictors. The R implementation of *Synth* is described in Abadie et al. (2011) and the *Synth* R package is available at https://cran.r-project.org/web/packages/Synth/index.html. The R package contains the original data for the Basque terrorism application, while the original data for the application to California's tobacco control program are embedded in the Matlab implementation of *Synth* available at https://web.stanford.edu/~jhain/synthpage.html.

Table 1. Random reordering of either predictors or donors changes the *Synth* results in the application to California's tobacco control program.

	Original ordering	Random ordering of predictors		Random ordering of donors	
		Min.	Max.	Min.	Max.
w					
Utah	0.3432	0.3274	0.3432	0.3234	0.3432
Nevada	0.2358	0.2272	0.2358	0.2243	0.2358
Montana	0.1820	0.1820	0.2020	0.1820	0.2011
Colorado	0.1747	0.1605	0.1788	0.1627	0.1863
Connecticut	0.0624	0.0624	0.0752	0.0624	0.0773
v					
smoking 1975	0.4925	0.3700	0.4925	0.3512	0.5557
smoking 1980	0.3917	0.0252	0.3917	0.0005	0.3917
smoking 1988	0.0682	0.0579	0.1009	0.0481	0.1007
retail price	0.0312	0.0312	0.4191	0.0312	0.4003
beer consum.	0.0124	0.0124	0.0922	0.0124	0.0267
percent 15–19	0.0034	0.0034	0.2384	0.0034	0.2148
L_V	3.20908	3.18659	3.20908	3.14722	3.20908
L_W	0.00170	0.00165	0.00243	0.00137	0.00296

NOTE: Following the notation of *Synth*, "smoking" denotes cigarette sales per capita; "retail price" denotes average retail price of cigarettes; "beer consum." denotes beer consumption per capita; and "percent 15–19" denotes the percentage of the population aged 15–19.

The leftmost column of Table 1 presents the donor weights $\mathbf{w}$ and the predictor weights $\mathbf{v}$ sorted in the descending order, and values of the loss functions L_V and L_W (the loss functions will be formally introduced in Section 3) reported by the R implementation of *Synth* using the original ordering of donors and predictors. The donor and predictor weights have been rounded to the four decimal digit accuracy, and we only report those with the minimum weight of 0.001. We note that R version of *Synth* produces somewhat different weights than the Matlab version, and both weights differ from those reported in the original article (see Appendix A for details). For the sake of brevity, we here focus on the R implementation of *Synth*.

For comparison, the four rightmost columns of Table 1 report the corresponding minimum and maximum values obtained using the randomly reordered samples. The results of Table 1 clearly demonstrate that the *Synth* package is numerically unstable even in the context where it was originally designed. Random ordering of either predictors or donors affects all of the donor and predictor weights, but also the values of the loss function. The most extreme example is the second largest predictor weight (0.3917) for cigarette sales per capita 1980 of the original ordering, which decreases to 0.0252 or 0.0005 by just randomly reordering the predictors or donors, respectively. Interestingly, the value of the loss function L_V that the SCM problem aims to minimize can also decrease as a result of reordering the data, which directly implies that the donor and predictor weights reported by Abadie et al. (2010) cannot be the optimal solution to the SCM problem.

In the original SCM application to Basque terrorism by Abadie and Gardeazabal (2003), the *Synth* results for the Basque Country proved numerically stable in 1,000 random orderings of predictors or donors. However, in the placebo study of one of its key donors, Catalonia, we do find numerical instabilities in 1,000 random orderings of predictors or donors as reported in Table 2. Abadie and Gardeazabal (2003) devote the entire Section II.B to the placebo study of Catalonia, which forms an important piece of evidence to support the SCM method.[6] Note that Table 2 is organized similar to Table 1. All the donor or predictor weights are to some extent affected by simply randomly reordering the donors or predictors, respectively.

[6] Abadie and Gardeazabal (2003) justify the choice of Catalonia as follows: "*To conduct this 'placebo' study we chose Catalonia which was the region with the largest weight in the synthetic control for the Basque Country. In addition to being the region most similar to the Basque Country before terrorism in economic growth determinants (as measured using our methods), Catalonia resembles the Basque Country in many characteristics, some of which are not directly measured in our data.*"

The value of the loss function L_W can also decrease as a result of reordering the data. To gain a better understanding of why the *Synth* algorithm fails, we need to take a closer look at the SCM problem from the optimization perspective.

3 Design flaw of the SCM problem

3.1 Bilevel formulation

Abadie and Gardeazabal (2003) and Abadie et al. (2010) state the SCM problem implicitly. The recent study by Malo et al. (2020) develops the first explicit mathematical formulation of the standard SCM problem where the predictor weights and the donor weights are jointly optimized. They show that the SCM problem can be stated as the following optimistic bilevel optimization problem

$$\min_{\mathbf{v},\mathbf{w}} L_V = \frac{1}{T^{\text{pre}}}(\mathbf{y}_1^{\text{pre}} - \mathbf{Y}_0^{\text{pre}}\mathbf{w})'(\mathbf{y}_1^{\text{pre}} - \mathbf{Y}_0^{\text{pre}}\mathbf{w}) \tag{5}$$

subject to

$$\mathbf{w} = \operatorname{argmin} L_W = (\mathbf{x}_1 - \mathbf{X}_0\mathbf{w})'\operatorname{diag}(\mathbf{v})(\mathbf{x}_1 - \mathbf{X}_0\mathbf{w}) \tag{6}$$

$$\mathbf{1}'\mathbf{w} = 1$$

$$\mathbf{1}'\mathbf{v} = 1$$

$$\mathbf{w} \geq \mathbf{0}, \mathbf{v} \geq \mathbf{0}$$

For the sake of transparency, we state the linear constraints for weights $\mathbf{w}$ and $\mathbf{v}$ explicitly. Note that the feasible sets of $\mathbf{w}$ and $\mathbf{v}$ are standard simplexes whose vertices consist of the J and K standard unit vectors, respectively.

The explicit formulation of the optimization problem reveals that the SCM problem is far from trivial from the computational point of view. The minimization problem (6) referred to the lower-level problem, and problem (5) is called the upper-level problem; the SCM literature commonly uses the terms inner and outer problems, but the meaning is the same. The problem is solvable, when it is interpreted as an optimistic bilevel problem, but the global optimum is not necessarily unique. Unfortunately, the bilevel optimization problems are generally NP-hard (Hansen et al., 1992; Vicente et al., 1994). In particular, the hierarchical optimization structure can introduce difficulties such as non-convexity and disconnectedness (e.g., Sinha et al., 2013), which are also problematic in the present setting.

Table 2. Random reordering of either predictors or donors changes the *Synth* results in the placebo study of Catalonia in the original SCM application to Basque terrorism.

	Original ordering	Random ordering of predictors		Random ordering of donors	
		Min.	Max.	Min.	Max.
w					
Madrid	0.4350	0.4348	0.4352	0.4347	0.4352
Baleares	0.2716	0.2714	0.2719	0.2712	0.2719
Cantabria	0.2575	0.2565	0.2577	0.2564	0.2577
Asturias	0.0359	0.0356	0.0369	0.0356	0.0368
v					
gdpcap	0.5167	0.5162	0.5168	0.5162	0.5168
sec.agriculture	0.2817	0.2814	0.2820	0.2814	0.2820
sec.energy	0.0875	0.0874	0.0875	0.0874	0.0875
invest	0.0794	0.0777	0.0795	0.0779	0.0795
school.illit	0.0141	0.0127	0.0143	0.0128	0.0143
school.prim	0.0134	0.0129	0.0163	0.0129	0.0161
school.med	0.0050	0.0049	0.0056	0.0049	0.0056
school.high	0.0018	0.0016	0.0023	0.0016	0.0023
L_V	0.00031	0.00031	0.00031	0.00031	0.00031
L_W	0.02724	0.02623	0.02755	0.02629	0.02753

NOTE: Following the notation of *Synth*, "gdpcap" denotes real GDP per capita; "sec.agriculture" and "sec.energy" denote the sectoral shares of agriculture, forestry, and fishing, and energy and water, respectively; "invest" denotes investment ratio; and "school.illit", "school.prim", "school.med", and "school.high" denote the percentages of the working-age population that were illiterate, up to primary school education, with some high school, and with high school or above, respectively.

These observations can help at least partly to explain the numerical instability of SCM demonstrated in Section 2.2. The derivative-based general-purpose algorithms are simply ill-equipped for the task at hand. If the weights $\mathbf{w}, \mathbf{v}$ are arbitrarily determined by an algorithm that fails to converge to the optimum, then all the attractive theoretical properties of the estimator fly out of the window.

3.2 Game interpretation

To gain intuition, we find it helpful to consider the bilevel SCM problem (5)–(6) as a Stackelberg game where the upper-level problem characterizes the optimal strategy of a "leader" who determines $\mathbf{v}$ and the lower-level problem defines the optimal strategy of a "follower" who determines $\mathbf{w}$. The optimal solution to the bilevel optimization problem can then be interpreted as the mixed strategy Nash equilibrium of the game.

Consider first the optimal strategy of the follower. In the non-cooperative Nash equilibrium, the follower solves the following QP problem, taking the weights $\mathbf{v}^\star$ as given

$$\begin{aligned}
&\min L_W = (\mathbf{x}_1 - \mathbf{X}_0\mathbf{w})'\mathrm{diag}\,(\mathbf{v}^\star)(\mathbf{x}_1 - \mathbf{X}_0\mathbf{w}) \\
&\text{subject to} \\
&\mathbf{1}'\mathbf{w} = 1 \\
&\mathbf{w} \geq \mathbf{0}
\end{aligned} \tag{7}$$

The lower-level problem of the follower is straightforward. In contrast, the optimal strategy of the leader is much more complicated in the non-cooperative setting. The leader sets weights $\mathbf{v}$ to incentivize the follower to choose attractive weights $\mathbf{w}$ to minimize the upper-level loss function L_V. In general, it is well-known that the Nash equilibrium of the non-cooperative game is not unique, and not necessarily Pareto efficient (compare, e.g., with the classic Prisoner's Dilemma).

Of course, the SCM problem is not a game played by two independent agents: there is just one agent with the primary objective to minimize the upper-level loss function, subject to the lower-level problem taken as a constraint. Therefore, the SCM problem is more analogous to a coordination problem by a single social planner. To allow for coordination between the upper-level and the lower-level problems, we can rephrase the lower-level problem (7) as

$$\begin{aligned}
&\min_{\mathbf{w}} L_W = (\mathbf{x}_1 - \mathbf{X}_0\mathbf{w})'\mathrm{diag}\,(\mathbf{v}^\star)(\mathbf{x}_1 - \mathbf{X}_0\mathbf{w}) + \varepsilon(\mathbf{y}_1^{\mathrm{pre}} - \mathbf{Y}_0^{\mathrm{pre}}\mathbf{w})'(\mathbf{y}_1^{\mathrm{pre}} - \mathbf{Y}_0^{\mathrm{pre}}\mathbf{w}) \\
&\text{subject to} \\
&\mathbf{1}'\mathbf{w} = 1 \\
&\mathbf{w} \geq \mathbf{0}
\end{aligned} \tag{8}$$

where $\varepsilon > 0$ denotes an infinitesimally small non-Archimedean scalar (see Malo et al., 2020 for a more detailed discussion). Introducing the upper-level objective as a part of the lower-level QP problem in (8) makes a subtle but important difference compared to problem (7): the primary objective of both (7) and (8) is to minimize the loss function L_W with respect to the predictors. However, if there are alternate optima $\mathbf{w}^\star$ that minimize the loss function L_W, problem (8) chooses the best solution for the upper-level problem. This is an important missing link between the lower-level problem and the upper-level problem because, in general, there can be many alternate optima where the loss function goes to zero, $L_W = 0$. Recall that unbiasedness and consistency of the SCM estimator depend on the perfect match between the treated unit and the synthetic control with respect to the predictors (Abadie et al., 2010).

Next, consider the optimization of weights $\mathbf{v}$ in the cooperative setting. For the given weights $\mathbf{w}^\star$, suppose the leader assigns predictor weights $\mathbf{v}$ is to minimize the lower-level loss function, formally,

$$\begin{aligned}
&\min_{\mathbf{v}} L_W = (\mathbf{x}_1 - \mathbf{X}_0\mathbf{w}^\star)'\mathrm{diag}\,(\mathbf{v})(\mathbf{x}_1 - \mathbf{X}_0\mathbf{w}^\star) + \varepsilon(\mathbf{y}_1^{\mathrm{pre}} - \mathbf{Y}_0^{\mathrm{pre}}\mathbf{w}^\star)'(\mathbf{y}_1^{\mathrm{pre}} - \mathbf{Y}_0^{\mathrm{pre}}\mathbf{w}^\star) \\
&\text{subject to} \\
&\mathbf{1}'\mathbf{v} = 1 \\
&\mathbf{v} \geq \mathbf{0}
\end{aligned} \tag{9}$$

The rationale of this cooperative solution is the following. The leader chooses weights $\mathbf{v}$ to minimize the loss for the follower, and the follower reciprocates by choosing among the alternate optima for weights $\mathbf{w}$ that minimize the loss for the leader. The resulting solution is Pareto efficient, and it is also one of the Nash equilibria to the non-cooperative game. But without coordination, there is zero probability that the non-cooperative game would converge to the cooperative solution. The lack of an explicit link between the upper-level and the lower-level problems is one of the reasons why the existing SCM algorithms generally fail to converge to the optimum.

Observe that problem (9) is a linear programming (LP) problem since both the objective function and the constraints are linear functions of weights $\mathbf{v}$: recall that the feasible set of weights $\mathbf{v}$ is a standard simplex whose vertices are unit vectors, and note that we can equivalently write the objective function as $\mathbf{q}'\mathbf{v}$, where $\mathbf{q} = (\mathbf{x}_1 - \mathbf{X}_0\mathbf{w}^\star) \odot (\mathbf{x}_1 - \mathbf{X}_0\mathbf{w}^\star)$ and $\odot$ denotes the Hadamard product. The fundamental theorem of linear programming states that every feasible LP problem has an optimal solution in a zero-dimensional face (a vertex) of the feasible set (see, e.g., Tardella, 2011). This implies that, for given weights $\mathbf{w}^\star$, the optimal solution to problem (9) is always a corner solution where one of the elements of $\mathbf{v}$ is equal to one and all other elements are equal to zero. In other words, all weight is assigned to a single predictor, and all other predictors are left with zero weight. Since this is the optimal strategy to set weights $\mathbf{v}$ for any given eights $\mathbf{w}^\star$, the optimal solution to the SCM problem is typically a corner solution. We consider this tendency towards corner solutions as an inherent design flaw of the data-driven approach to set weights predictor weights $\mathbf{v}$.

Based on the previous discussion it might be tempting to assume the optimal $\mathbf{v}^\star$ must always be a corner solution. Since the weights $\mathbf{w}$ and $\mathbf{v}$ are jointly optimized, this is not necessarily the case, as the following counter-example demonstrates. For the sake of simplicity, assume there are only two predictors, two donors, and a single outcome. The data of the treated unit are $\mathbf{x}_1 = (4, 6)$, $y_1 = 5$. The two donors are $\mathbf{x}_A = (1, 1)$, $y_A = 1$ and $\mathbf{x}_B = (9, 9)$, $y_B = 9$, respectively. It is easy to verify that the optimal weights are $w_A = w_B = 0.5$ and $\mathbf{v} = (0.5, 0.5)$, which yield $L_V = 0$. In contrast, the corner solution $\mathbf{v} = (1, 0)$ implies donor weights $w_A = 5/8$, $w_B = 3/8$ and $L_V = 1$. Similarly, for $\mathbf{v} = (0, 1)$, we have $w_A = 3/8$, $w_B = 5/8$, and $L_V = 1$. This simple counter-example suffices to demonstrate that the optimal $\mathbf{v}^\star$ is not necessarily a corner solution.

To illustrate the prevalence of corner solutions in the SCM applications, we next revisit the two original SCM applications to Basque terrorism and the tobacco control program in California. In both applications, the optimal solution turns out to be a corner solution.

3.3 Comparison of *Synth*, MSCMT, and the global optimum

Applying the iterative algorithm proposed by Malo et al. (2020) to the data of the two original SCM applications to Basque terrorism (Abadie and Gardeazabal, 2003) and the California tobacco control program (Abadie et al., 2010), we empirically verify that the optimal solution in both cases is indeed a corner solution. The corner solution is found superior to the solutions obtained by *Synth* and the MSCMT algorithm proposed by Becker and Klößner (2018). This observation demonstrates that the existing SCM algorithms fail to find the optimal solution even in the two original applications of SCM, which are also used as illustrative examples for *Synth*.

We compare the results of the following three algorithms: the standard implementation of *Synth* described in Abadie et al. (2011),[7] the MSCMT package described in Becker and Klößner (2018), and the iterative algorithm proposed by Malo et al. (2020), which ensures the true global optimum.[8] Tables 3 and 4 report the donor weights ($\mathbf{w}$), the predictor weights ($\mathbf{v}$), and the loss function values of the upper-level problem (L_V) and the lower-level problem (L_W) estimated by different algorithms in R for the Basque terrorism application and the California tobacco control application, respectively. For convenience, Tables 3 and 4 are organized similar to Tables 1 and 2 above. We discuss the results of both tables in parallel.

Recall that the value of L_V measures how well the synthetic control matches the pre-treatment outcomes of the treated unit, and this is the upper-level objective to be minimized. In this respect, all algorithms come relatively close to the global optimum. Note that L_V depends on the measurement units of outcomes: for example, multiplying $\mathbf{y}_1^{\text{pre}}$ and $\mathbf{Y}_0^{\text{pre}}$ by 1 Thousand would increase L_V by a factor of 1 Million. Therefore, it is helpful to measure empirical fit with respect to the pre-treatment outcomes in terms of the coefficient of determination (R^2)—after all, the upper-level problem is just constrained least squares regression without intercept. Such a comparison reveals that the differences in empirical fit are rather marginal, the R^2 statistic varies between 0.96866 (*Synth*) to 0.98541 (optimum) in the Basque example and between 0.97518 (*Synth*) and 0.97878 (optimum) in the California

[7] In addition to the standard *Synth* command, we have also considered the genoud() option available in *Synth*, as noted in Abadie et al. (2011). However, the use of the genoud() option does not improve the matter; in fact, the solution is only worse.

[8] The R code to implement this algorithm is documented in Appendix D. The latest updates to the R code are available on the GitHub page: https://github.com/Xun90/SCM-Debug.git.

Table 3. Basque terrorism application revisited: donor weights, predictor weights, loss functions, and empirical fit by different algorithms.

	Synth	MSCMT	Optimum
w			
Catalonia	0.8508	0.6328	0.0000
Madrid	0.1492	0.1479	0.4405
Baleares	0.0000	0.2193	0.3700
La Rioja	0.0000	0.0000	0.1895
v			
Schooling of working age population (%)			
Illiterates	0.0156	0.0000	0
Up to primary school	0.0018	0.0000	0
With some high school	0.0442	0.0000	0
With high school or above	0.0341	0.0003	0
Investment ratio	0.0001	0.0003	0
Real GDP per capita	0.2010	0.9993	1
Sectoral shares (%)			
Agriculture, forestry, and fishing	0.0948	0.0000	0
Energy and water	0.0077	0.0000	0
Industry	0.1339	0.0000	0
Construction and engineering	0.0087	0.0000	0
Marketable services	0.0097	0.0000	0
Non-marketable services	0.1081	0.0000	0
Population density	0.3403	0.0000	0
L_V	0.00886	0.00429	0.00413
L_W	0.24670	0.00034	0.00000
R^2	0.96866	0.98485	0.98541

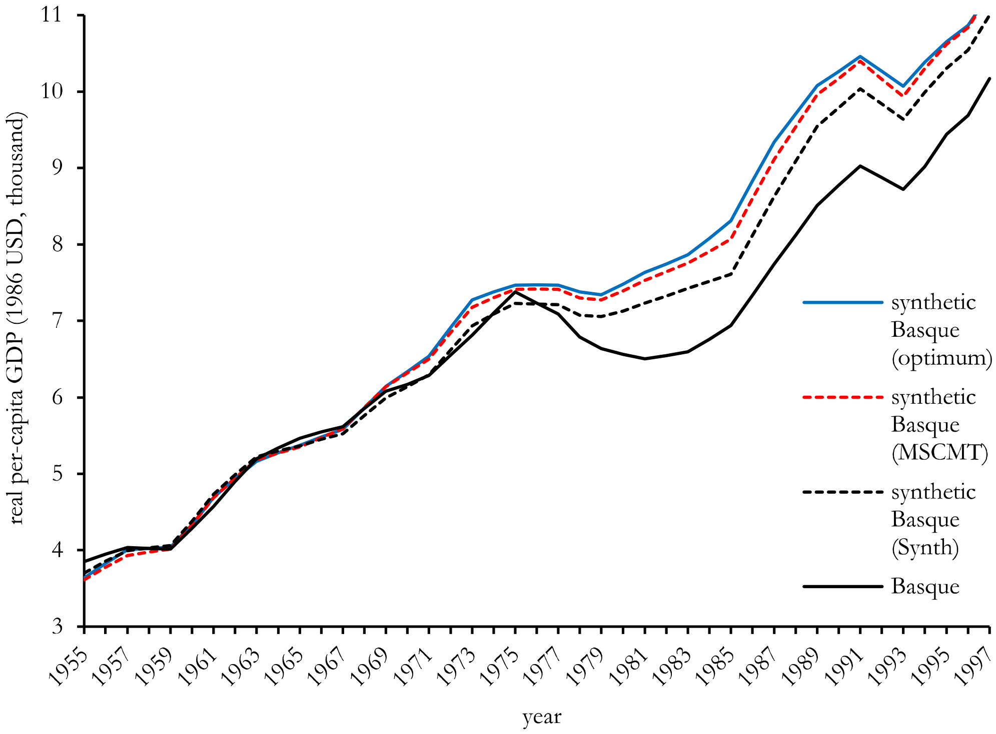
Table 4. California tobacco control application revisited: donor weights, predictor weights, loss functions, and empirical fit by different algorithms.

	Synth	MSCMT	Optimum
w			
Utah	0.3432	0.3351	0.3939
Nevada	0.2358	0.2356	0.2049
Montana	0.1820	0.2019	0.2318
Colorado	0.1747	0.1595	0.0148
Connecticut	0.0624	0.0679	0.1091
New Hampshire	0.0000	0.0000	0.0454
v			
Income per capita	0.0006	0.0000	0
Retail price of cigarettes	0.0312	0.3333	0
Population aged 15–19 (%)	0.0034	0.3333	0
Beer consumption per capita	0.0124	0.0000	0
Cigarette sales per capita 1988	0.0682	0.0000	0
Cigarette sales per capita 1980	0.3917	0.0000	1
Cigarette sales per capita 1975	0.4925	0.3333	0
L_V	3.20908	3.07666	2.74366
L_W	0.00170	0.00000	0.00000
R^2	0.97518	0.97621	0.97878

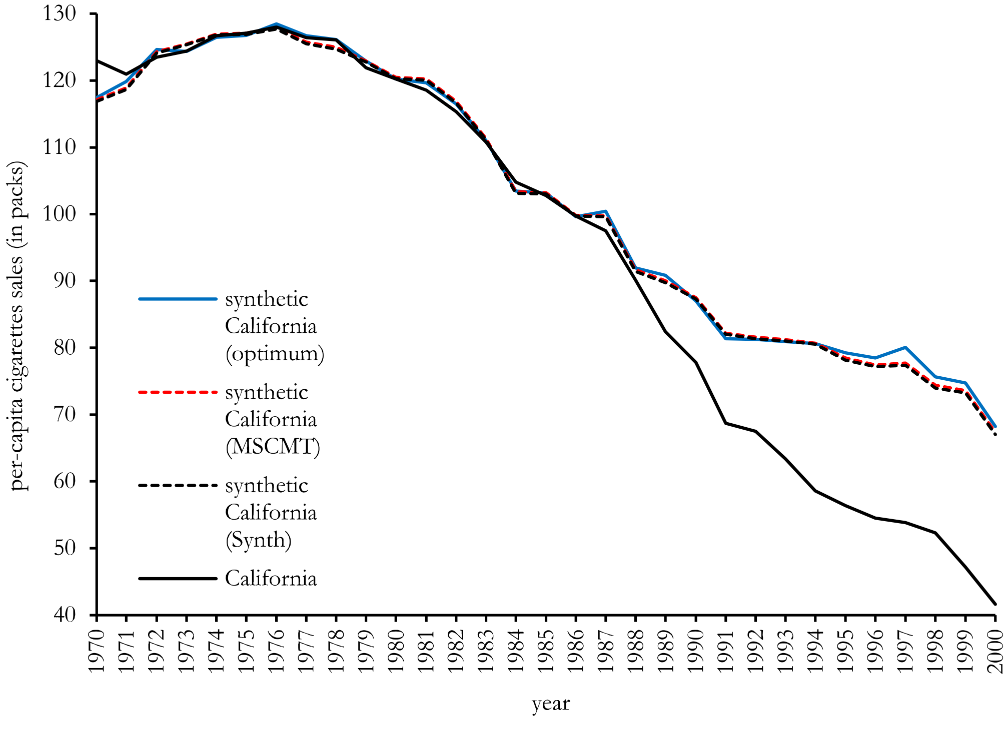
example. In contrast, the differences in weights $\mathbf{w}$ and $\mathbf{v}$ are rather dramatic. The results of Tables 3 and 4 help to illustrate that good empirical fit may be achieved with a wide variety of weights $\mathbf{w}$ and $\mathbf{v}$, but there is only one unique global optimum.

The loss function L_W measures how well the synthetic control matches the predictors $\mathbf{x}_1$. Minimization of L_W is the lower-level objective, but the consistency of SCM depends on the (nearly) perfect match with the predictors. In this regard, the relatively high value of L_W given by the standard *Synth* command in both applications indicates that *Synth* fails to converge to the global optimum. Furthermore, the MSCMT procedure greatly improves L_W, but the performance varies between the two empirical examples: L_W converges to the global optimum in the California case but not in the Basque case. In contrast, the value of L_W at the global optimum goes to zero, suggesting a perfect match in terms of the weighted predictors. However, this is an illusion because the optimal solution is a corner solution that assigns all weight to a single predictor: real per capita GDP in the Basque terrorism application and cigarette sales per capita in 1980 in the California tobacco control application (see Tables 3 and 4). The MSCMT algorithm comes close to the corner solution in the former application, but fails to converge to the corner solution in the latter. The *Synth* algorithm appears to use more balanced weights for predictors, however, note that *Synth* also assigns almost 90% of the predictor weight to cigarette sales per capita (the outcome variable) during two years of the pre-treatment period. Unfortunately, *Synth* fails to solve the optimization problem it is supposed to solve; its predictor weights are just artifacts of a computational failure. This is the design flaw that we intended to demonstrate by these two empirical examples.

Of course, the most important piece of information for SCM are the donor weights $\mathbf{w}$, which are used to form the synthetic control. As noted above, a marginal improvement in the empirical fit leads to rather dramatic changes in the composition of the synthetic control. Consider first the synthetic control for Basque. The *Synth* algorithm identifies Catalonia and Madrid as the benchmarks, with 85% weight assigned to Catalonia. The solution found by the MSCMT algorithm reassigns 22 percentage points of Catalonia's weight to the Balearic Islands, maintaining the weight of Madrid. In sharp contrast, the global optimum assigns no weight to Catalonia, whereas the largest weights are assigned to Madrid (44%) and the Balearic Islands (37%), but also the neighboring region of La Rioja enters the synthetic control with the 19% weight. Consider next the synthetic control for California. *Synth* and



(a) Basque terrorism



(b) California's tobacco control program

Fig. 1. The impact of suboptimal **w** weights on the evolution of synthetic controls.

MSCMT yield almost the same donor weights despite their different estimates of the loss function values. However, the global optimum reassigns nearly all of Colorado's weight and 4 percentage points of Nevada's weight to Utah (consolidating as the largest weighting state), Montana, Connecticut, and New Hampshire (a new state entering the synthetic California).

Figure 1 illustrates the impact of suboptimal donor weights on the evolution of the synthetic Basque (panel 1a) and the synthetic California (panel 1b). Fortunately the qualitative conclusions of these two original and highly influential applications remain, but the suboptimal weights lead to a lower treatment effect in both cases, particularly in the Basque terrorism application. We stress that the globally optimal weights minimize the MSPE of the pre-treatment outcomes $\mathbf{y}_1^{\text{pre}}$, but there is no guarantee that the weights are optimal to minimize the MSPE of the counterfactual because the good empirical fit to pre-treatment outcomes was achieved by disregarding all predictors except for one. We compare the solutions produced by the *Synth* and MSCMT algorithms to the global optimum just to illustrate the computational failure, but the practical use of this global optimum is not the approach that we advocate.

4 Alternative data-driven approaches

4.1 Optimizing donor weights when predictor weights are given

In the previous section we found that the original SCM problem is solvable, but unfortunately, the solution is not nice. In light of the arguments presented in the previous section, we would strongly recommend the users of SCM to determine the predictor weights $\mathbf{v}$ separately, before optimizing the donor weights $\mathbf{w}$.

In this sub-section we develop a simple iterative procedure to compute the optimal weights $\mathbf{w}$ when the predictor weights $\mathbf{v}^\star$ are given *a priori*. Malo et al. (2020) previously consider this problem, suggesting to solve problem (8) such that the non-Archimedean ε is gradually decreased towards zero. In practice, it is difficult to ensure that ε is sufficiently close to zero to give the priority to the lower-level objective function L_W, but high enough to achieve coordination with the upper-level objective L_V. To operationalize the theoretical idea of Malo et al. (2020), we propose to optimize the weights $\mathbf{w}$ using the following two-step procedure when the predictor weights $\mathbf{v}^\star$ are predetermined:

Step 1: Solve the QP problem

$$\begin{aligned}
&\min_{\mathbf{w}} L_W = (\mathbf{x}_1 - \mathbf{X}_0\mathbf{w})'\mathrm{diag}\,(\mathbf{v}^\star)(\mathbf{x}_1 - \mathbf{X}_0\mathbf{w})\\
&\text{subject to}\\
&\mathbf{1}'\mathbf{w} = 1\\
&\mathbf{w} \geq \mathbf{0}
\end{aligned}$$

Step 2: Given the optimal $L_W^\star$ from Step 1, solve the convex programming problem

$$\begin{aligned}
&\min_{\mathbf{w}} L_V = (\mathbf{y}_1^{\text{pre}} - \mathbf{Y}_0^{\text{pre}}\mathbf{w})'(\mathbf{y}_1^{\text{pre}} - \mathbf{Y}_0^{\text{pre}}\mathbf{w})\\
&\text{subject to}\\
&(\mathbf{x}_1 - \mathbf{X}_0\mathbf{w})'\mathrm{diag}\,(\mathbf{v}^\star)(\mathbf{x}_1 - \mathbf{X}_0\mathbf{w}) = L_W^\star\\
&\mathbf{1}'\mathbf{w} = 1\\
&\mathbf{w} \geq \mathbf{0}
\end{aligned}$$

Breaking the problem into two separate stages allows to eliminate the non-Archimedean ε in (8). In Step 1 we minimize the lower-level objective function L_W, and its optimal value is subsequently inserted as a constraint to the optimization problem in Step 2. This establishes an explicit link between the upper-level and the lower-level objectives. The two-step procedure explicitly considers the possibility of alternate optima in Step 1. Since the *Synth* algorithm does not take the possibility of alternate optima into account, there is no guarantee that it finds the optimal donor weights $\mathbf{w}$ even when the predictor weights $\mathbf{v}$ are defined by the user (see Appendix B for a numerical demonstration). In the next sub-sections we explore and demonstrate alternative data-driven strategies to determine the weights $\mathbf{v}$ empirically.

Before proceeding to the predictor weights, it is worth to note the recent study by Abadie and L'Hour (2020), which similarly takes the predictor weights $\mathbf{v}$ as given. The authors deviate from the original SCM approach in that they focus solely on the lower-level objective of optimizing the fit with respect to the predictors, ignoring the upper-level objective of optimizing the fit with respect to the pre-treatment outcomes. The authors introduce an additional penalty to minimize the sum of pairwise matching discrepancies, which ensures that the optimal donor weights are unique in this new setting. The additional penalty term to improve matching is a valuable extension, which could be readily combined with the developments of our study. However, omitting the upper-level objective function would typically result as poor fit to the pre-treatment outcomes. Of course, one might incorporate pre-treatment outcomes among the predictors, but this would quite dramatically change the logic of the original SCM. In mathematical terms, the original bilevel optimization problem would then become a multi-objective optimization problem where the weights $\mathbf{v}$ govern the relative importance assigned to the empirical fit to the pre-treatment outcomes and the fit to the additional predictors, respectively.

4.2 Panel regression approach to determine predictor weights

There are several possibilities to set weights $\mathbf{v}$ based on empirical data. Both Abadie and Gardeazabal (2003) and Abadie et al. (2010) discuss the possibility to use subjectively determined weights $\mathbf{v}$. The default option of the Stata implementation of the *Synth* package is to use regression-based weights $\mathbf{v}$, which are also used as starting values in the R and Matlab implementation of *Synth* (see Abadie et al., 2011). In this sub-section we similarly resort to a regression-based approach, but propose some modifications to the *Synth* approach.

If panel data of predictors $\mathbf{X}$ are available, we propose to first estimate the equation

$$y_{jt}^{\text{pre}} = \mu + \mathbf{x}_{jt}'\boldsymbol{\beta} + \gamma_j + \varepsilon_{jt} \qquad j = 1, 2, \ldots, J+1;\; t = 1, 2, \ldots, T^{\text{pre}}. \tag{10}$$

Model (10) can be estimated by standard fixed effects (FE) or random effects (RE) panel data regression. Note that the FE estimator cannot be used when there are time-invariant predictors. The original SCM application to Basque terrorism, to be revisited below, does include some time-invariant predictors. Therefore, we will resort to the RE estimator below, assuming that the random effects γ_j are uncorrelated with the predictors.

Given estimated coefficients $\hat{\boldsymbol{\beta}}$, we propose to assign weights $\mathbf{v}$ based the absolute values of the parameter estimates, that is

$$v_k = |\hat{\beta}_k| \Big/ \sum_{j=1}^{K} |\hat{\beta}_j|. \tag{11}$$

We note that the *Synth* algorithm uses the squared values of the parameter estimates to assign weights $\mathbf{v}$. By using the absolute values rather than squared values, one achieves a more equal balance between different predictors.

Having optimized the predictor weights, we apply the two-step procedure proposed in Section 4.1 to optimize the donor weights. Given the optimal donor weights $\mathbf{w}^{\star}$, we estimate the counterfactual as

$$\mathbf{y}_1^{\mathrm{N}} = \mathbf{Y}_0\mathbf{w}^{\star} + (\hat{\gamma}_1 - \hat{\boldsymbol{\gamma}}_0'\mathbf{w}^{\star}). \tag{12}$$

Note that the random effects γ_j were not taken into account in the optimization of the donor weights. Therefore, we utilize the estimated random effects to implement the standard bias correction, following Ben-Michael et al. (2018) and Ferman et al. (2018).

We next illustrate the regression-based approach outlined above by reexamining the original SCM application to Basque terrorism. Imputing the missing values by suitable methods (see Appendix C for details), we obtain panel data for most of the predictors during the pre-treatment period. In the RE panel regression to set weights $\mathbf{v}$, we excluded the real GDP per capita, the percentage of the illiterate working-age population, and the sectoral share of non-marketable services to avoid perfect collinearity. Table 5 reports the RE estimates of predictor coefficients and the empirical $\mathbf{v}$ weights determined by equation (11) for the Basque example. The percentage of the working age population with some high school and the sectoral share of marketable services are found to be statistically significant predictors. Together with the percentage of the working age population with high school or higher education, those two significant predictors are the three most influential predictors that receive more than 70% weight. On the other hand, the empirical $\mathbf{v}$ weights are relatively balanced among the other predictors, except for population density, which is attributed less than 1% weight. In addition, the overall empirical fit of the RE panel regression is 0.8808, with the between and within effects being 0.8734 and 0.9277, respectively. Note that 78% of the unexplained variation of the outcome is attributed to the random effects and that the random effects are statistically significant.

Given the empirically set $\mathbf{v}$ weights, we next determine the optimal $\mathbf{w}$ weights to construct the synthetic Basque by using the two-step procedure described in Section 4.1. The donor weight is assigned to Cantabria (79.9%), Catalonia (12.4%), and Madrid (7.7%). Interestingly, Cantabria enters the synthetic control with a large weight. Cantabria is a neighboring region to the Basque Country, but it was not included in any of the the three synthetic controls considered in Section 3.3. However, it was one of the components that construct the synthetic controls for Catalonia considered in the placebo study of Section 2.2.

Table 5. Predictor coefficients and empirical predictor weights for the Basque example.

Predictors	Coefficients	Robust standard errors	Empirical **v**
Schooling of working age population (%)			
Up to primary school	0.0397	0.0264	0.0532
With some high school	0.2567***	0.0527	0.3439
With high school or above	0.2126	0.2275	0.2848
Investment ratio	-0.0085	0.0068	0.0114
Sectoral shares (%)			
Agriculture, forestry, and fishing	0.0150	0.0335	0.0201
Energy and water	0.0196	0.0389	0.0262
Industry	0.0446	0.0368	0.0598
Construction and engineering	-0.0477	0.0715	0.0639
Marketable services	0.1007**	0.0397	0.1349
Population density	-0.0014	0.0016	0.0019
Intercept	-5.7426**	2.9123	
R^2: within = 0.9277, between = 0.8734, overall = 0.8808			
$\sigma_{\hat{\gamma}} = 0.2062$, $\sigma_{\hat{\varepsilon}} = 0.1099$, $\rho = 0.7789$ (fraction of variance due to γ_i)			

NOTE: * $p \leq 0.10$; ** $p \leq 0.05$; *** $p \leq 0.01$.

Figure 2 illustrates the impact of the alternative strategy to set $\mathbf{v}$ on the evolution of the synthetic Basque. The time series start from 1960, which is the first year in the panel model. Note that the absolute RE weights approach with bias-correction yields notably better fit to the pre-treatment outcomes than the "canonical" SCM that does not use any predictors, which is exactly the same as the "global optimum" considered in Section 3.3 obtained by assigning all weight to a single predictor. The synthetic Basque based on the absolute RE weights still identifies the treatment effect of Basque terrorism on real GDP per capita. However, the treatment effect is considerably smaller than the "canonical" synthetic control that does not use any predictors. The treatment effect disappears by the mid-1990s. This example illustrates that appropriate use of the predictors does influence the results, and can potentially affect the qualitative conclusions.

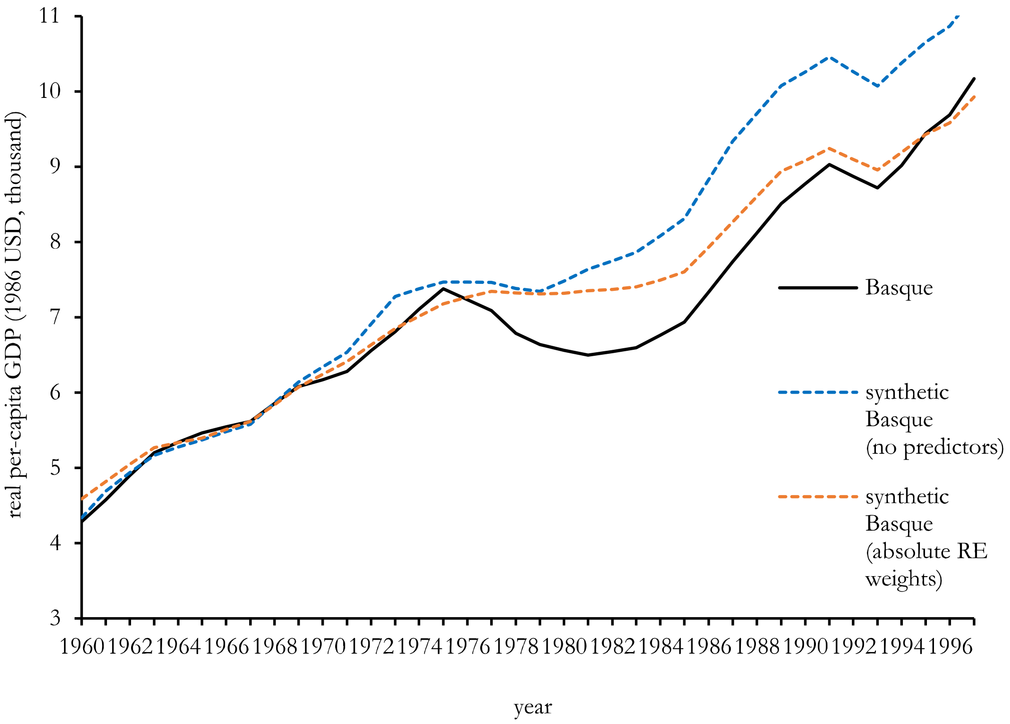


Fig. 2. The impact of alternative approaches on the evolution of synthetic Basque.

One of the key assumptions of any treatment effect model is that the control group is not exposed to the treatment. This assumption does not, strictly speaking, hold in the present application because a significant proportion of Euskadi Ta Askatasuna (ETA)'s terrorism activity took place in other regions, including Madrid and Catalonia, which have large weight in the synthetic control. Abadie and Gardeazabal (2003) indicate that 69% of deaths attributed to terrorism occurred in the Basque Country, which directly implies that almost one third of deaths occurred in the regions that form the donor pool. Further, the specification of the pre-treatment and post-treatment periods (before and after 1970, respectively) could be debated. ETA was founded in 1968 and there were three victims during the pre-treatment period, but only one victim during the first three years of the post-treatment period. The difference between the actual outcome and the counterfactual synthetic control becomes evident from the year 1975 onwards, which matches perfectly with the death of Dictator Franco and the transition towards democracy. While we do not intend to deny the economic cost of ETA's terrorism, perhaps at least some part of the observed treatment effect may be attributed to the economic transition from Franco's dictatorship to democracy, which had varying effects across different regions of Spain. Of course, ETA's terrorism is also closely related to this historical context, but ETA's terrorism did not cause the major political regime shift in Spain.

4.3 Uniform weights to standardized predictors

Suitable panel data are not always available for the purposes of SCM. The original application to California's tobacco control program is one example of such application. Another possibility would be to apply uniform $\mathbf{v}$ weights when panel data for the predictors are simply unavailable. In this approach, we propose to first standardize the predictors as

$$z_{ik} = (x_{ik} - \bar{\mathbf{x}}_k)) \;/\; \mathrm{std}(\mathbf{x}_k).$$

and subsequently apply equal weights $v_k = 1/K$ to the standardized predictors. By doing so, all predictors will count, and the weights are invariant to rescaling or changing the units of measurement.

We next illustrate the application of uniform $\mathbf{v}$ weights by revisiting the California tobacco control application. The donor weights are obtained by applying the two-stage procedure proposed in Section 4.1. This yields the following optimal donor weights: Colorado (62.6%), Connecticut (27.8%), Texas (6.5%), and Utah (3.2%). Colorado was included in the synthetic control in the examples of Sections 2.2 and 3.3, but the use of standardized uniform predictor weights notably increases its weight. In contrast, Utah was previously assigned the largest weight, but in the present analysis it gets only 3.2% weight.

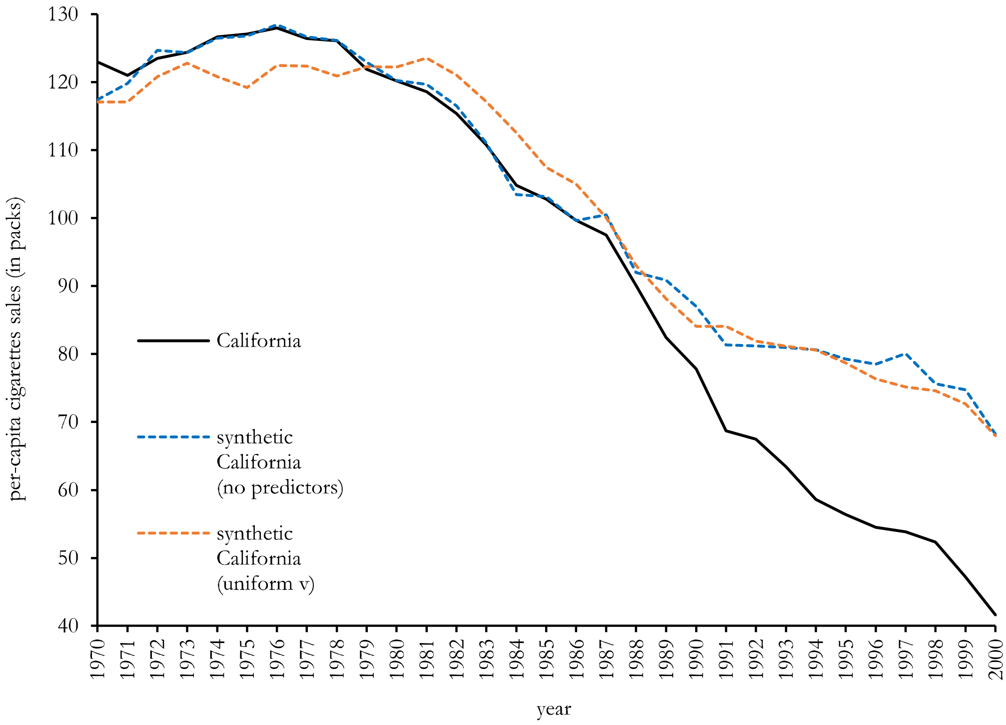


Fig. 3. The impact of alternative approaches on the evolution of synthetic California.

Figure 3 illustrates the impact of the uniform $\mathbf{v}$ on the evolution of the synthetic California. Note that in this example the uniform $\mathbf{v}$ approach leads to worse empirical fit to the pre-treatment outcomes than the "canonical" SCM that does not use any predictors. There is a trade-off: when we put more emphasis on optimizing the empirical fit with respect to predictors $\mathbf{X}$, then the fit with respect to pre-treatment outcomes $\mathbf{y}$ is likely to deteriorate, and vice versa. In our interpretation, Figure 3 is a useful illustration of why focusing solely on optimizing the fit with respect to predictors, ignoring the pre-treatment outcomes, is not necessarily a viable solution. In many applications, the good pre-treatment fit of *Synth* is to some extent illusion because it tends to put negligibly small weight to many predictors.

However, it is reassuring to find that the post-treatment outcomes of the synthetic California based on uniform $\mathbf{v}$ are very similar to those of the "canonical" synthetic California. Therefore, the use of predictors mainly affects the pre-treatment fit, but not so much the post-treatment. One would be mainly interested in the post-treatment effect, so this would help to support the empirical finding that there was indeed impact. In fact, we suggest that one could examine a range of alternative $\mathbf{v}$ weights for testing robustness of the treatment effect (as an additional tool, in addition to the placebo trials and statistical tests that are already known in the literature).

In summary, the main point of Section 4 is to demonstrate that alternative data-driven approaches to determine the weights $\mathbf{v}$ are available. The empirical comparisons above demonstrate that the introduction of empirically determined $\mathbf{v}$ weights presents a viable remedy to the ill-designed *Synth* algorithm. While the relative merits of the alternative approaches clearly warrant further research, in light of the problems discussed in Sections 2 and 3, we strongly recommend that the suboptimal weights produced by *Synth* should not be used.

5 Conclusions

SCM has proved a highly appealing approach to estimate causal treatment effects in a comparative case study setting, as a large number of published applications clearly demonstrate. Unfortunately, the computational difficulties caused by joint optimization of the donor weights and the predictor weights cast serious doubts on the reliability of this method. The purpose of this paper was to demonstrate the inherent design flaw of SCM, but also propose a constructive solution. The contributions of our paper are three-fold.

Firstly, we revisited the two original SCM applications to illustrate that arbitrary re-ordering of the rows or columns of the data can change the results produced by the *Synth* algorithm. Klößner et al. (2018) have previously pointed out a similar problem, but their diagnosis was misleading. Our results demonstrate that the numerical instability of *Synth* is an even more serious issue, affecting the original SCM setting. However, numerical instability is merely a symptom, not the root cause of the problem.

Secondly, we examined the explicit bilevel optimization formulation of the SCM problem by Malo et al. (2020). By applying insights from the game theory, we explained why the optimal solution to the SCM problem is typically a corner solution where all weight is assigned to a single predictor and all other predictors become redundant. This is also the case in the two original SCM applications. We stress that development of a better computational algorithm is not the solution that we advocate because it does not help to address the root cause of the problem. While the optimal solution is not always a corner solution, the computational complexity of the NP-hard bilevel optimization can explain why *Synth* and other SCM algorithms generally fail to converge to the optimum.

Thirdly, we proposed to address the root cause of the problem by estimating the predictor weights and the donor weights separately. We first proposed a simple two-step procedure to optimize the donor weights for a given set of predictor weights. We then explored two simple empirical strategies to determine the predictor weights using panel data regression or applying uniform weights to standardized predictors.

We illustrated the application of the proposed data-driven approaches to determine the predictor weights by revising the two original SCM applications to Basque terrorism and the California tobacco control program. Our results demonstrate that alternative empirical strategies to determine the predictor weights are readily available and, in our view, yield meaningful results. While further research is clearly needed, there is no excuse to apply suboptimal weights that are just artifacts of a computational failure.

This study opens several important avenues for future research, both empirical and methodological studies. From the empirical point of view, the findings of our paper call for systematic replication of the published SCM studies to examine whether and to what extent the use of suboptimal weights produced by *Synth* has affected the qualitative conclusions. Becker and Klößner (2017) is an excellent example of such a replication study. We hope that the qualitative results of the influential SCM studies prove robust to the optimization errors that are evidently present, but this remains to be tested empirically. Our replication of the two original applications of SCM showed that the suboptimal weights yield somewhat different results than the optimal ones, but fortunately the qualitative conclusions of these two studies remain. The source code and documentation provided in Appendix D and the online supplementary material can be easily adapted to other data sets for replication purposes.

From the methodological point of view, while we strongly recommend the users of the classic SCM to determine the predictor weights *a priori*, we do not consider the joint optimization of the predictor weights and the donor weights entirely hopeless. However, the loss function to be minimized requires careful reconsideration to ensure that the optimal solution is meaningful for the intended purposes of using the predictors, and that the problem remains computationally tractable. In this respect, utilizing the structural similarity of SCM with the benefit-of-the-doubt weighting (e.g., Cherchye et al., 2007) could provide useful insights. It would also be helpful to establish more detailed practical guidelines regarding what kind of variables are suitable predictors for SCM. At present, many SCM studies include a mixed set of predictors expressed in levels, logs, differences, and percentage growth rates, which appears potentially problematic. Finally, we hope that the insights of our paper might contribute to further integration of SCM with other estimation approaches such as the difference-in-differences, panel data regression, and machine learning; several recent studies such as Doudchenko and Imbens (2017), Xu (2017), Amjad et al. (2018), Arkhangelsky et al. (2018), Ben-Michael et al. (2018), Abadie and L'Hour (2020), and Chernozhukov et al. (2020) have made impressive progress in this direction. Again, we hope that the results of the present study can help to strengthen the foundation of SCM.

Beyond SCM and treatment effect models, we hope that the findings of our paper would help to stimulate further discussion about the importance of corner solutions, and the role of optimization in economics more broadly. It is standard in economics to assume away the corner solutions, for the sake of convenience, so much so that the corner solutions appear to be a blind spot. However, relevant problems are not always well-behaved: we have shown that the SCM problem is NP-hard with a strong tendency towards corner solutions.

For a long time, optimization theory was central to economics as the economic theory was heavily based on the idea of rational agents making optimal decisions. However, behavioral economists have successfully challenged the rational paradigm during the recent decades. In our interpretation, the design flaw of SCM discussed in this paper is intimately related to the declined status of optimization in economics. Our results strongly suggest that the SCM paradigm has placed too much faith on a black-box algorithm. In light of the growing interest in artificial intelligence, machine learning, and similar techniques, we hope that this study might serve as a healthy reminder to the economics profession about the possible risks associated with the replacement of rigorous optimization by black-box algorithms.

References

Abadie, A. (2019). Using Synthetic Controls: Feasibility, Data Requirements, and Methodological Aspects. *Journal of Economic Literature*, forthcoming.

Abadie, A., Diamond, A., & Hainmueller, J. (2010). Synthetic Control Methods for Comparative Case Studies: Estimating the Effect of California's Tobacco Control Program. *Journal of the American Statistical Association*, *105*(490), 493–505.

Abadie, A., Diamond, A., & Hainmueller, J. (2011). Synth: An R package for synthetic control methods in comparative case studies. *Journal of Statistical Software*, *42*(13), 1–17.

Abadie, A., Diamond, A., & Hainmueller, J. (2015). Comparative Politics and the Synthetic Control Method. *American Journal of Political Science*, *59*(2), 495–510.

Abadie, A., & Gardeazabal, J. (2003). The Economic Costs of Conflict: A Case Study of the Basque Country. *American Economic Review*, *93*(1), 113–132.

Abadie, A., & L'Hour, J. (2020). A Penalized Synthetic Control Estimator for Disaggregated Data. *Work. Pap., Mass. Inst. Technol., Cambridge, MA.*

Acemoglu, D., Johnson, S., Kermani, A., Kwak, J., & Mitton, T. (2016). The value of connections in turbulent times: Evidence from the United States. *Journal of Financial Economics*, *121*(2), 368–391.

Amjad, M., Shah, D., & Shen, D. (2018). Robust synthetic control. *Journal of Machine Learning Research*, *19*(1), 802–852.

Arkhangelsky, D., Athey, S., Hirshberg, D. A., Imbens, G. W., & Wager, S. (2018). Synthetic Difference in Differences. *arXiv preprint arXiv:1812.09970.*

Athey, S., & Imbens, G. W. (2017). The state of applied econometrics: Causality and policy evaluation. *Journal of Economic Perspectives*, *31*(2), 3–32.

Becker, M., & Klößner, S. (2017). Estimating the economic costs of organized crime by synthetic control methods. *Journal of Applied Econometrics*, *32*(7), 1367–1369.

Becker, M., & Klößner, S. (2018). Fast and reliable computation of generalized synthetic controls. *Econometrics and Statistics*, *5*, 1–19.

Becker, M., Klößner, S., & Pfeifer, G. (2018). Cross-validating synthetic controls. *Economics Bulletin*, *38*(1), 603–609.

Ben-Michael, E., Feller, A., & Rothstein, J. (2018). The Augmented Synthetic Control Method. *arXiv preprint arXiv:1811.04170.*

Bloom, N., & Van Reenen, J. (2007). Measuring and Explaining Management Practices Across Firms and Countries. *The Quarterly Journal of Economics*, *122*(4), 1351–1408.

Bohn, S., Lofstrom, M., & Raphael, S. (2014). Did the 2007 legal Arizona workers act reduce the state's unauthorized immigrant population? *Review of Economics and Statistics*, *96*(2), 258–269.

Cavallo, E., Galiani, S., Noy, I., & Pantano, J. (2013). Catastrophic natural disasters and economic growth. *Review of Economics and Statistics*, *95*(5), 1549–1561.

Cherchye, L., Moesen, W., Rogge, N., & Puyenbroeck, T. V. (2007). An introduction to 'benefit of the doubt' composite indicators. *Social Indicators Research*, *82*(1), 111–145.

Chernozhukov, V., Wuthrich, K., & Zhu, Y. (2020). An Exact and Robust Conformal Inference Method for Counterfactual and Synthetic Controls. *arXiv preprint arXiv:1712.09089v8*.

Cole, M. A., Elliott, R. J., & Liu, B. (2020). The Impact of the Wuhan Covid-19 Lockdown on Air Pollution and Health: A Machine Learning and Augmented Synthetic Control Approach. *Environmental and Resource Economics*, *76*(4), 553–580.

Doudchenko, N., & Imbens, G. W. (2017). Balancing, Regression, Difference-In-Differences and Synthetic Control Methods: A Synthesis. *arXiv preprint arXiv:1610.07748*.

Ferman, B., Pinto, C., & Possebom, V. (2018). Cherry Picking with Synthetic Controls. *MPRA Paper 85138*.

Gobillon, L., & Magnac, T. (2016). Regional policy evaluation: Interactive fixed effects and synthetic controls. *Review of Economics and Statistics*, *98*(3), 535–551.

Hansen, P., Jaumard, B., & Savard, G. (1992). New Branch-and-Bound Rules for Linear Bilevel Programming. *SIAM Journal on Scientific and Statistical Computing*, *13*(5), 1194–1217.

Kaul, A., Klößner, S., Pfeifer, G., & Schieler, M. (2015). Synthetic Control Methods: Never Use All Pre-Intervention Outcomes Together With Covariates. *MPRA Paper 83790*.

Kleven, H. J., Landais, C., & Saez, E. (2013). Taxation and international migration of superstars: Evidence from the European football market. *American Economic Review*, *103*(5), 1892–1924.

Klößner, S., Kaul, A., Pfeifer, G., & Schieler, M. (2018). Comparative politics and the synthetic control method revisited: A note on Abadie et al. (2015). *Swiss Journal of Economics and Statistics*, *154*(1), 11.

Malo, P., Eskelinen, J., Zhou, X., & Kuosmanen, T. (2020). Computing Synthetic Controls Using Bilevel Optimization. *MPRA Paper 104085*.

Powell, D. (2018). Imperfect Synthetic Controls: Did the Massachusetts Health Care Reform Save Lives? *SSRN Electronic Journal*.

Sinha, A., Malo, P., & Deb, K. (2013). Efficient Evolutionary Algorithm for Single-Objective Bilevel Optimization. *arXiv preprint arXiv:1303.3901*.

Tardella, F. (2011). The fundamental theorem of linear programming: Extensions and applicationsy. *Optimization*, *60*(1-2), 283–301.

Vicente, L., Savard, G., & Júdice, J. (1994). Descent approaches for quadratic bilevel programming. *Journal of Optimization Theory and Applications*, *81*(2), 379–399.

Xu, Y. (2017). Generalized synthetic control method: Causal inference with interactive fixed effects models. *Political Analysis*, *25*(1), 57–76.

Supplementary materials

Appendix A The donor weights reported by Abadie et al. (2010) are not replicable

In this appendix we try to replicate the original donor weights of the synthetic California reported by Abadie et al. (2010) using the data and codes provided by the authors. We stress that the results of the R and Matlab implementations of *Synth* reported below are based on exactly the same data set provided by the Matlab implementation, which is available at https://web.stanford.edu/~jhain/synthpage.html.

Table A1 presents the donor weights **w** obtained by the R and Matlab implementations of *Synth*, and those reported by the original article Abadie et al. (2010). The donor weights by *Synth*/R and *Synth*/Matlab have been rounded to the three decimal digit accuracy following the style of Abadie et al. (2010), and we only report those states assigned with a positive weight by Abadie et al. (2010).

The comparison of the donor weights reveals that the results produced by *Synth* differ across the computational platforms even in terms of the most important piece of information for SCM. The most extreme example is the third largest donor weight for the state of Montana, which increases from 0.182 (*Synth*/R) to 0.202 (*Synth*/Matlab) just by switching from one software to another. Further, both weights differ from the value of 0.199 for Montana reported by Abadie et al. (2010).

We also note that the original data provided by the Matlab implementation claims that it uses the "percentage of state population aged 15–19" as one of the predictors. However, Abadie et al. (2010) define this predictor as "the percentage of state population aged 15–24".

In conclusion, despite our best efforts, we find it impossible to replicate (in the narrow sense) the donor weights reported by Abadie et al. (2010) using the data and computational codes provided by the authors. In light of the numerical instabilities that were demonstrated in Section 2.2, this result is not surprising. Note that the donor weights reported by Abadie et al. (2010) do fit withing the range of values reported in Table 1, obtained by randomly reordering the rows and columns of the data matrix $\mathbf{X}_0$. In that sense, we do manage to successfully replicate the donor weights reported in the original study.

Table A1. California tobacco control application revisited: donor weights by different implementations of *Synth* and Abadie et al. (2010)

	Synth/R	*Synth*/Matlab	Abadie et al. (2010)
w			
Utah	0.343	0.335	0.334
Nevada	0.236	0.235	0.234
Montana	0.182	0.202	0.199
Colorado	0.175	0.160	0.164
Connecticut	0.062	0.068	0.069

Appendix B Demonstrating the failure of ***Synth*** when the predictor weights are given ***a priori***

This appendix demonstrates numerically that the *Synth* algorithm does not always find the optimal donor weights $\mathbf{w}$ even when the user has predefined the predictor weights. Table B1 reports the results obtained in one of the robustness checks to the re-examination of the California tobacco control application reported in Section 4.3. We emphasize that the predictor weights $\mathbf{v}$ reported at the bottom part of the table should be interpreted as subjective weights given by the user.

We have inserted exactly the same set of predictor weights $\mathbf{v}$ to both *Synth* and our two-step algorithm proposed in Section 4.1. Table B1 demonstrates that the donor weights $\mathbf{w}$ produced by *Synth* differ substantially from the optima ones. Note that the donor weights $\mathbf{w}$ influence both the upper-level and lower-level loss functions L_V and L_W. The comparison of the loss functions reveals that the results produced by *Synth* are far from optimal in this case. This example demonstrates that *Synth* is not reliable even in the drastically simplified case where the user has prespecified the predictor weights by some other method.

Table B1. California tobacco control application revisited: *Synth* fails to find the optimal donor weights $\mathbf{w}^\star$ even when the predictor weights $\mathbf{v}$ are defined by the user.

	Synth	Optimum		*Synth*	Optimum
$\mathbf{w}$			$\mathbf{w}$		
Alabama	0.0021	0.0000	Nevada	0.1842	0.2324
Arkansas	0.0018	0.0000	New Hampshire	0.0057	0.0000
Colorado	0.0117	0.1193	New Mexico	0.0034	0.0000
Connecticut	0.0036	0.0169	North Carolina	0.0311	0.0000
Delaware	0.0018	0.0000	North Dakota	0.0147	0.0613
Georgia	0.0018	0.0000	Ohio	0.0019	0.0000
Idaho	0.0213	0.0404	Oklahoma	0.0050	0.0000
Illinois	0.0042	0.0000	Pennsylvania	0.0028	0.0000
Indiana	0.0017	0.0000	Rhode Island	0.0017	0.0000
Iowa	0.0042	0.0000	South Carolina	0.0017	0.0000
Kansas	0.0036	0.0000	South Dakota	0.0045	0.0000
Kentucky	0.0000	0.0000	Tennessee	0.0017	0.0000
Louisiana	0.0036	0.0000	Texas	0.0064	0.0000
Maine	0.0022	0.0000	Utah	0.3315	0.3302
Minnesota	0.0057	0.0000	Vermont	0.0024	0.0000
Mississippi	0.0025	0.0000	Virginia	0.0018	0.0000
Missouri	0.0019	0.0000	West Virginia	0.0026	0.0000
Montana	0.3088	0.1995	Wisconsin	0.0037	0.0000
Nebraska	0.0051	0.0000	Wyoming	0.0059	0.0000
$\mathbf{v}$			$\mathbf{v}$		
income	0.0000	0.0000	smoking 1988	0.0296	0.0296
retail price	0.0005	0.0005	smoking 1980	0.5082	0.5082
percent 15–19	0.0008	0.0008	smoking 1975	0.4604	0.4604
beer consumption	0.0005	0.0005			
L_V	6.89385	5.16664	L_W	0.00013	0.00007

NOTE: Following the notation of *Synth*, "income" denotes personal income per capita; "retail price" denotes average retail price of cigarettes; "percent 15–19" denotes the percentage of the population aged 15–19; "beer consumption" denotes beer consumption per capita; and "smoking" denotes cigarette sales per capita.

Appendix C Imputation of missing values

The *Synth* R package contains the original data for the Basque terrorism application. This data set contains incomplete panel data for the predictors across different regions in the pre-treatment period (1960–1969) (see Abadie et al., 2011 for more details). To implement the panel regression approach described in Section 4.2 to determine predictor weights in the Basque terrorism application, it is necessary to impute the missing values by suitable methods.

For the six sectoral share predictors (i.e., "sec.agriculture", "sec.energy", "sec.industry", "sec.construction", "sec.services.venta", and "sec.services.nonventa"), panel data are available for odd years only (1961, 1963, . . ., 1969). We replaced the missing values in even years from 1962 through 1968 with the mean of the data of two adjacent years. We then estimated a linear time trend by regressing the values of 1961–1969, and used the predicted value for the year 1960.

For the four schooling predictors (i.e., "school.illit", "school.prim", "school.med", and "school.high") and the predictor "investment ratio", panel data are available only for the years 1964–1969. Again, we estimated a linear time trend by regressing the values of 1964–1969, and used the predicted values for the years 1960–1963.

Finally, the predictor "population density" was observed only in the year 1969. Since the population density usually changes very slowly, in the absence of better data, we used the observed value of population density in the year 1969 throughout the pre-treatment period 1960–1969.

Appendix D R code

In this appendix we provide the essential R code to help the reader to reproduce our empirical results or adapt the code for their own applications. The latest updates to the code and the technical documentation are available at GitHub: https://github.com/Xun90/SCM-Debug.git. We assume the reader is familiar with the *Synth* R package (see Abadie et al., 2011 for an introduction), and suggest the use of the dataprep() function provided in *Synth* to pre-process the data.

Step 1: Load necessary R packages.

```
library("Synth")      #Synth package
#The two QP solvers used by "Synth" are employed here for a direct comparison with "Synth"
library(kernlab)      #QP solver 1: ipop
library(LowRankQP)    #QP solver 2: LowRankQP, whose results are reported in this study
library(lpSolve)      #LP solver
library(matrixcalc)   #for matrix calculations
```

Step 2: Re-examine the *Synth* results for the California tobacco control application with 1,000 random reorderings of predictors.

```
##loop on 1000 random orders
lossV <- matrix(0, 1000, 1)
lossW <- matrix(0, 1000, 1)
W <- matrix(0, 38, 1000)
V <- matrix(0, 7, 1000)
C <- matrix(0, 7, 1000)
set.seed(42)
for (i in 1:1000){
  row <- sample(nrow(X0))
  C[,i] <- row
  dataprep.out$X0 <- X0[row,]
  dataprep.out$X1 <- as.matrix(X1[row,])
  synth.out <- synth(data.prep.obj = dataprep.out, method = "BFGS")
  lossV[i,] <- synth.out$loss.v
  lossW[i,] <- synth.out$loss.w
  W[,i] <- synth.out$solution.w
  sorted <- cbind(row, t(synth.out$solution.v))
  sorted <- sorted[order(sorted[,"row"]),]
  V[,i] <- sorted[,2]}
```

Step 3: Re-examine the *Synth* results for the California tobacco control application with 1,000 random reorderings of donors.

```
##loop on 1000 random orders
lossV <- matrix(0, 1000, 1)
lossW <- matrix(0, 1000, 1)
W <- matrix(0, 38, 1000)
V <- matrix(0, 7, 1000)
C <- matrix(0, 38, 1000)
set.seed(42)
for (i in 1:1000){
  column <- sample(ncol(X0))
  C[,i] <- column
  dataprep.out$X0 <- X0[,column]
```

```
  dataprep.out$Z0 <- Y0pre[,column]
  synth.out <- synth(data.prep.obj = dataprep.out, method = "BFGS")
  lossV[i,] <- synth.out$loss.v
  lossW[i,] <- synth.out$loss.w
  V[,i] <- t(synth.out$solution.v)
  sorted <- cbind(column, synth.out$solution.w)
  sorted <- sorted[order(sorted[,"column"]),]
  W[,i] <- sorted[,2]}
```

Step 4: Implement the iterative algorithm proposed by Malo et al. (2020) to check for the feasibility of the unconstrained optimum and the possibility of corner solutions.

```
scm.corner <- function(Y1pre,Y0pre,X1,X0){
  ##step1
  Tpre <- dim(Y0pre)[1]
  nDonors <- dim(Y0pre)[2]
  #QP setup
  c1 <- -t(Y0pre) %*% Y1pre
  H1 <- t(Y0pre) %*% Y0pre
  A1 <- matrix(rep(1,nDonors), ncol = nDonors)
  b1 <- 1
  r1 <- 0
  l1 <- matrix(rep(0,nDonors), nrow = nDonors)
  u1 <- matrix(rep(1,nDonors), nrow = nDonors)
  #run QP
  step1_ipop <- ipop(c = c1, H = H1, A = A1, b = b1, l = l1, u = u1, r = r1,
                     margin = 0.0005, maxiter = 1000, sigf = 7, bound = 10) #QP_Solver1
  step1_lowr <- LowRankQP(Vmat = H1, dvec = c1, Amat = A1, bvec = b1, uvec = u1,
                          method = "LU") #QP_Solver2
  W_ipop <- matrix(step1_ipop@primal, nrow = nDonors)
  W_lowr <- step1_lowr$alpha
  L1_ipop <- (t(Y1pre) %*% Y1pre)/Tpre + 2/Tpre * (t(c1) %*% W_ipop
                                          + 0.5 * t(W_ipop) %*% H1 %*% W_ipop)
  L1_lowr <- (t(Y1pre) %*% Y1pre)/Tpre + 2/Tpre * (t(c1) %*% W_lowr
                                          + 0.5 * t(W_lowr) %*% H1 %*% W_lowr)
  ##step2
  #normalize X - Synth
  nvarsV <- dim(X0)[1]
  big.dataframe <- cbind(X0, X1)
  divisor <- sqrt(apply(big.dataframe, 1, var))
  scaled.matrix <- t(t(big.dataframe) %*% ( 1/(divisor)
                                    * diag(rep(dim(big.dataframe)[1], 1)) ))
  X0.scaled <- scaled.matrix[,c(1:(dim(X0)[2]))]
  if(is.vector(X0.scaled)==TRUE)
  {X0.scaled <- t(as.matrix(X0.scaled))}
  X1.scaled <- scaled.matrix[,dim(scaled.matrix)[2]]
  #LP setup
  f.obj_ipop <- (X1.scaled - X0.scaled %*% W_ipop)^2
  f.obj_lowr <- (X1.scaled - X0.scaled %*% W_lowr)^2
  f.con <- rbind(rep(1,nvarsV), diag(x = 1, nrow = nvarsV))
  f.dir <- c("=", rep(">=",nvarsV))
  f.rhs <- c(1, rep(0,nvarsV))
  #run LP
  step2_ipop <- lp ("min", f.obj_ipop, f.con, f.dir, f.rhs)
  step2_lowr <- lp ("min", f.obj_lowr, f.con, f.dir, f.rhs)
  V_ipop <- step2_ipop$solution
  V_lowr <- step2_lowr$solution
  L2_ipop <- step2_ipop$objval
  L2_lowr <- step2_lowr$objval

  scm.corner.out <- list(W = cbind(W_ipop,W_lowr), V = cbind(V_ipop,V_lowr),
                         Lv = c(L1_ipop,L1_lowr), Lw = c(L2_ipop,L2_lowr))
  return(scm.corner.out)}
```

Step 5: Implement the two-step procedure described in Section 4.1. This implementation is currently a hybrid of Section 4.1 and Malo et al. (2020). Since there are currently no reliable solvers in R for the second-stage convex programming problem, we solve the non-Archimedean problem (8) iteratively, decreasing ε towards zero until the objective function reaches the optimal solution of the first-stage QP problem.

```
two.step.iterative <- function(Y1pre,Y0pre,X1.scaled,X0.scaled,SV){
  #SV - predictor weights defined by the user
  ##Solve non-Archimedean problem (8)
  Tpre <- dim(Y0pre)[1]
  nDonors <- dim(Y0pre)[2]
  #QP setup
  A <- matrix(rep(1,nDonors), ncol = nDonors)
  b <- 1
  r <- 0
  l <- matrix(rep(0,nDonors), nrow = nDonors)
  u <- matrix(rep(1,nDonors), nrow = nDonors)
  #Loop on 10 epsilon values (0.1^1 ... 0.1^10) to find the best performer
  L_upper = matrix(0, 10, 2)
  L_lower = matrix(0, 10, 2)
  W_ipop = matrix(0, nDonors, 10)
  W_lowr = matrix(0, nDonors, 10)
  for (i in 1:10){
    eps <- 0.1^(i) #epsilon - penalty term
    c <- (-t(X0.scaled) %*% diag(SV1) %*% X1.scaled) - eps * t(Y0pre) %*% Y1pre
    H <- t(X0.scaled) %*% diag(SV1) %*% X0.scaled + eps * t(Y0pre) %*% Y0pre
    #run QP
    QP_ipop <- ipop(c = c, H = H, A = A, b = b, l = l, u = u, r = r,
                    margin = 0.0005, maxiter = 1000, sigf = 7, bound = 10) #QP_Solver1
    QP_lowr <- LowRankQP(Vmat = H, dvec = c, Amat = A, bvec = b, uvec = u,
                         method = "LU") #QP_Solver2
    W_ipop[,i] <- matrix(QP_ipop@primal, nrow = nDonors)
    W_lowr[,i] <- QP_lowr$alpha
    L_upper[i,1] <- 1/Tpre * t(Y1pre - Y0pre %*% W_ipop[,i]) %*%
      (Y1pre - Y0pre %*% W_ipop[,i])
    L_upper[i,2] <- 1/Tpre * t(Y1pre - Y0pre %*% W_lowr[,i]) %*%
      (Y1pre - Y0pre %*% W_lowr[,i])
    L_lower[i,1] <- t(X1.scaled - X0.scaled %*% W_ipop[,i]) %*% diag(SV1) %*%
      (X1.scaled - X0.scaled %*% W_ipop[,i])
    L_lower[i,2] <- t(X1.scaled - X0.scaled %*% W_lowr[,i]) %*% diag(SV1) %*%
      (X1.scaled - X0.scaled %*% W_lowr[,i])}
  ##Use the first step of the two-step procedure in Section 4.1 to determine epsilon
  c1 <- (-t(X0.scaled) %*% diag(SV) %*% X1.scaled)
  H1 <- t(X0.scaled) %*% diag(SV) %*% X0.scaled
  #run QP
  QP_ipop1 <- ipop(c = c1, H = H1, A = A, b = b, l = l, u = u, r = r,
                   margin = 0.0005, maxiter = 1000, sigf = 7, bound = 10) #QP_Solver1
  QP_lowr1 <- LowRankQP(Vmat = H1, dvec = c1, Amat = A, bvec = b, uvec = u,
                        method = "LU") #QP_Solver2
  W_ipop1 <- matrix(QP_ipop1@primal, nrow = nDonors)
  W_lowr1 <- QP_lowr1$alpha
  W1 <- cbind(W_ipop1, W_lowr1)
  obj_left <- t(X1.scaled) %*% diag(SV) %*% X1.scaled
  Lw_ipop1 <- obj_left + 2 * (t(c1) %*% W_ipop1 + 1/2 * t(W_ipop1) %*% H1 %*% W_ipop1)
  Lw_lowr1 <- obj_left + 2 * (t(c1) %*% W_lowr1 + 1/2 * t(W_lowr1) %*% H1 %*% W_lowr1)
  Lw1 <- c(Lw_ipop1, Lw_lowr1)

  two.step.iterative.out <- list(W = cbind(W_ipop,W_lowr), W1 = W1, V = SV,
                                 L_upper = L_upper, L_lower = L_lower, Lw1 = Lw1)
  return(two.step.iterative.out)}
```