

RNN-Based Counterfactual Prediction

Jason Poulos

University of California, Berkeley

Abstract

This paper proposes an alternative to the synthetic control method (SCM) for estimating the effect of a policy intervention on an outcome over time. Recurrent neural networks (RNNs) are used to predict counterfactual time-series of treated units using only the outcomes of control units as model inputs. The proposed method does not rely on pre-intervention covariates to construct the synthetic control and is consequently less susceptible to p -hacking. RNNs are also capable of handling multiple treated units and can learn nonconvex combinations of control units. In placebo tests, RNNs outperform the SCM in predicting the post-intervention time-series of control units, while yielding a comparable proportion of false positives. The RNN-based approach contributes to a new generation of data-driven machine learning techniques such as matrix completion and the Lasso for generating counterfactual predictions.

Keywords: Counterfactual Prediction; Recurrent Neural Networks; Randomization Inference; Synthetic Controls; Time-Series Cross-Section Data

PhD Candidate, Department of Political Science, 210 Barrows Hall #1950, Berkeley, CA 94720-1950.
Email: poulos@berkeley.edu. *Telephone:* +1-510-642-6323. I acknowledge support of the National Science Foundation Graduate Research Fellowship (DGE 1106400). This work used the computer resources of Stampede2 at the Texas Advanced Computing Center (TACC) under an Extreme Science and Engineering Discovery Environment (XSEDE) startup allocation (TG-SES180010). The Titan Xp GPU used for this research was donated by the NVIDIA Corporation.

1 Introduction

An important problem in the social sciences is estimating the effect of a binary intervention on an outcome over time. When interventions take place at an aggregate level (e.g., a state), researchers make causal inferences by comparing the post-intervention (“post-period”) outcomes of affected (“treated”) units against the outcomes of unaffected (“control”) units. A common approach to the problem is the synthetic control method (SCM) (Abadie et al., 2010), which predicts the counterfactual outcomes of treated units by finding a convex combination of control units that match the treated units in term of lagged outcomes. Correlations across units are assumed to remain constant stable time.

The SCM has several limitations. First, the convexity restriction of the synthetic control estimator precludes dynamic, nonlinear interactions between multiple control units. Intuitively, one can expect that the treated unit may exhibit nonlinear or negative correlations with the control units. Ferman and Pinto (2016) demonstrate that the convexity restriction implies that the SCM estimator may be biased even if selection into treatment is only correlated with time-invariant unobserved covariates. Second, Ferman and Pinto (2018) demonstrate that the SCM is generally biased if treatment assignment is correlated with unobserved confounders, even when the number of pre-intervention periods grows (Ferman and Pinto, 2018). Moreover, the authors show that while the SCM minimizes imbalance in pre-period outcomes, the likelihood of finding exact balancing weights vanishes as the number of time periods increase, which results in bias.

While the strength of the SCM lies in its simplicity in setup and implementation, several problems arise from the lack of guidance on how to specify the SCM estimator. The specification of the estimator can produce very different results: Ferman et al. (2018) show, for example, how cherry-picking between common SCM specifications can facilitate p -hacking. Kaul et al. (2015) show that the common practice of including lagged versions of the outcome variable as model inputs can render all other covariates irrelevant. Klößner et al. (2017) demonstrates that the common practice of using cross-validation to select importance weights can

yield multiple values and consequently different results.

This paper proposes an alternative to the SCM that is capable of automatically selecting appropriate control units at each time-step, allows for nonconvex combinations of control units, and does not rely on pre-intervention covariates. The method uses recurrent neural networks (RNNs) to predict a counterfactual time-series of treated units using only control unit outcomes as model inputs. RNNs are a class of neural networks that take advantage of the sequential nature of time-series data by sharing model parameters across multiple time-steps (El Hihi and Bengio, 1995). Non-parametric models such as RNNs are useful for prediction problems because we do not have to assume a functional form on the data. In addition, RNNs can learn the most useful nonconvex combination of control unit outcomes at each time-step for generating counterfactual predictions. Relaxing the convexity restriction is useful when the data-generating process underlying the outcome of interest depends nonlinearly on the history of its inputs. RNNs have been shown to outperform various linear models on time-series prediction tasks (Cinar et al., 2017).

The proposed method builds on a new literature that uses machine learning methods for data-driven synthetic controls, such as matrix completion (Athey et al., 2017; Poulos, 2019), or two-stage estimators that reduces data dimensionality via L1-regularized regression (Doudchenko and Imbens, 2016; Carvalho et al., 2018) or matrix factorization (Amjad et al., 2018) prior to regressing the outcomes on the reduced data. These methods are data-driven in the sense that they are capable of finding an appropriate subset of control units for comparison in the absence of domain knowledge or pre-intervention covariates.

RNNs are end-to-end trainable and very flexible to a given sequential prediction problem. For example, they are capable of sharing learned parameters across time-steps and multiple treated units. While the SCM can be generalized to handle multiple treated units (e.g., Dube and Zipperer, 2015; Xu, 2017), the generalized the SCM is not capable of sharing model weights when predicting the outcomes of multiple treated units. Regularization methods such as dropout can easily be incorporated into RNN architectures to prevent overfitting

during the training process, which is problematic when the networks learn an overreliance on a few model inputs. Moreover, an attention mechanism can be included in the model in order to discern the contribution of each model input to the predicted counterfactual.

In the section immediately below, I describe the approach of using RNNs for counterfactual time-series prediction; Section 4 details the procedure for evaluating the models in terms of predictive accuracy and statistical significance; Section 5 presents the results of the placebo tests and discusses when the proposed method is expected to outperform the SCM; Section 7 concludes by discussing the contributions of the paper and offering potential avenues for future research.

2 Counterfactual prediction

The proposed method estimates the causal effect of a discrete intervention in observational panel data; i.e., settings in which treatment is not randomly assigned and there exists both pre- and post-period observations of the outcome of interest. Let \mathbf{Y} denote a $N \times T$ matrix of outcomes for each unit $i = 1, \dots, N$ at time $t = 1, \dots, T$. \mathbf{Y} is incomplete because we observe each element Y_{it} for only the control units and the treated units prior to time of initial treatment exposure, $T_0 < T$. Let \mathcal{O} denote the set of (it) values that are observed and \mathcal{M} the set of (it) missing values. Let the values of the $N \times T$ complete matrix \mathbf{W} be $W_{it} = 1$ if $(it) \in \mathcal{M}$ and $W_{it} = 0$ if $(it) \in \mathcal{O}$. Note that the process that generates W_{it} is referred to the treatment assignment mechanism in the causal inference literature (Imbens and Rubin, 2015) and the missing data mechanism in missing data analysis (?). The pattern of missing data is assumed to follow from the simultaneous treatment adoption setting, where treated units are exposed to treatment at time T_0 and every subsequent period.

This setup is motivated by the Neyman (1923) potential outcomes framework, where for each it value there exists a pair of potential outcomes, $Y_{it}(1)$ and $Y_{it}(0)$, which represents the response to treated and control regimes, respectively. The observed outcomes are

$$Y_{it} = \begin{cases} Y_{it}(0) & \text{if } W_{it} = 0 \text{ or } t < T_0 \\ Y_{it}(1) & \text{if } W_{it} = 1 \text{ and } t \geq T_0. \end{cases} \quad (1)$$

The problem of counterfactual prediction is that we cannot directly observe the missing potential outcomes and instead wish to impute the missing values in $\mathbf{Y}(0)$ for treated units with $W_{it} = 1$. The potential outcomes framework explicitly assumes unconfoundedness. In an observational setting, this assumption requires $(\mathbf{Y}(0), \mathbf{Y}(1)) \perp\!\!\!\perp \mathbf{W} | \mathbf{Y}(\mathcal{O})$, where $\mathbf{Y}(\mathcal{O})$ is the observed data.

The potential outcomes framework also implicitly assumes treatment is well-defined to ensure that each unit has the same number of potential outcomes (Imbens and Rubin, 2015). It also excludes interference between units, which would undermine the framework by creating more than two potential outcomes per unit, depending on the treatment status of other units (Rubin, 1990).

2.1 Relationship to matrix completion and covariate shift

The proposed approach is similar to the method of matrix completion via nuclear norm minimization (MC-NNM) proposed by Athey et al. (2017) to predict counterfactual outcomes. Matrix completion methods attempt to impute missing entries in a low-rank matrix by solving a convex optimization problem via NNM, even when relatively few values are observed in \mathbf{Y} (??). The estimator recovers a $N \times T$ low-rank matrix by minimizing the sum of squared errors via nuclear norm regularized least squares. The estimator reconstructs the matrix by iteratively replacing missing values with those recovered from a singular value decomposition (?).

Athey et al. (2017) note two drawbacks of MC-NNM. First, the errors may be autocorrelated because the estimator does not account for time-series dependencies in the observed

data. The estimator estimate patterns row- and column-wise, but treat the data as perfectly synchronized (?). In contrast, the RNN-based approach described in Section 3 exploits the temporal component of the data and therefore does not have the problem of autocorrelated errors.

Second, the MC-NNM estimator penalizes the errors for each observed value equally without regard to the fact that the probability of missingness (i.e, the propensity score), increases with t . Athey et al. (2017) suggest weighting the loss function by the propensity score, which is similar to the importance weighting scheme proposed by Cortes et al. (2008) to address the problem of covariate shift, which is a special case of domain adaptation (Huang et al., 2007; Bickel et al., 2009; Cortes et al., 2010).¹

The covariate shift problem occurs when training and test data are drawn from different distributions. For notational ease, define the set of predictors $\mathbf{X} = \mathbf{Y}(\mathcal{O}), \forall t < T_0$. In both the matrix completion and RNN-based approach to counterfactual prediction, the models are trained on \mathbf{X} to predict $\mathbf{Y}(\mathcal{O})$ for $t \geq T_0$. The trained model is used to impute the missing potential outcomes, $\mathbf{Y}(\mathcal{M})$. This approach to counterfactual prediction can be connected to recent work in transfer learning (Ben-David et al., 2007; Ganin et al., 2015). The approach assumes similarity between the distributions of pre-period outcomes of treated and control units. An extension of the RNN-based approach would consider weighting the training loss by the propensity score to reduce any discrepancy between these two distributions.

2.2 Nonparametric regression

In its most basic form, counterfactual prediction can be approached by nonparametrically regressing the post-period observed outcomes on the pre-period observed outcomes,

$$\hat{\mathbf{Y}} = \hat{f}_0(\mathbf{X}) + \epsilon^{(t)}, \quad (2)$$

¹Schnabel et al. (2016) first connected the matrix completion problem with causal inference in observational settings in the context of recommender systems under confounding. Johansson et al. (2016) formulates the general problem of counterfactual inference as a covariate shift problem.

where the noise variables $\epsilon^{(t)}$ are assumed to be i.i.d. standard normal and independent of the observed data.

The nonlinear function \hat{f}_0 is estimated by minimizing the mean squared error, $\text{MSE} = \text{E} \left[\left(\mathbf{Y} - \hat{\mathbf{Y}} \right)^2 \right]$ for $t \geq T_0$. The estimated causal effect of the intervention is the difference between the observed time-series of the treated units and the counterfactual time-series that would have been observed in the absence of the intervention:

$$\hat{\phi}^{(t)} = \mathbf{Y}(\mathcal{M})^{(t)} - \hat{\mathbf{Y}}^{(t)} \quad \text{for } t = T_0, \dots, T. \quad (3)$$

This treatment effect is calculated at every post-period time-step and is thus useful for understanding the temporal evolution of the causal effect.

3 RNNs for counteractual prediction

RNNs (Graves, 2012; Goodfellow et al., 2016) consist of an input $\mathbf{X} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n_x)})$, an output $\mathbf{Y} = (\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(n_y)})$, and a shared hidden state $\mathbf{h}^{(t)}$. In the encoder-decoder network architecture described below, n_x and n_y can vary in length; in the plain vanilla RNN it is assumed $n_x = n_y = T$.

At each t , RNNs input $\mathbf{x}^{(t)}$ and pass it to the $\mathbf{h}^{(t)}$, which is updated with a function $g^{(t)}$ using the entire history of the input, which is unfolded backwards in time:

$$\begin{aligned} \mathbf{h}^{(t)} &= g^{(t)} \left(\mathbf{x}^{(t)}, \mathbf{x}^{(t-1)}, \dots, \mathbf{x}^{(1)} \right) \\ &= f_1 \left(\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}; \theta \right), \end{aligned} \quad (4)$$

where the nonlinear function $f_1(\cdot)$ and parameter θ are learned on all t and $\mathbf{x}^{(t)}$. Parameter sharing is particularly useful in the current application because it allows for better generalization when the dimension of the training data is relatively small. The updated hidden state (4) is used to generate a sequence of values $\mathbf{o}^{(t)}$ in the form of log probabilities corresponding

to the output. The loss function computes $\hat{\mathbf{y}}^{(t)} = \text{linear}(\mathbf{o}^{(t)})$ and calculates the loss. The total loss for the input-output pair is the sum of the losses over all t .

The RNNs are trained to estimate the conditional distribution of $\mathbf{y}^{(t)}$ given the past inputs and also the previous output. This is accomplished by offsetting the input-output pairs by one time-step so that the networks receive $\mathbf{y}^{(1)}$ as input at $t + 1$ to be conditioned on for predicting subsequent outputs. This popular training procedure is known as teacher forcing because it forces the networks to stay close to the ground-truth output $\mathbf{y}^{(t)}$ (Lamb et al., 2016). Specifically, the RNNs are trained to maximize the log-likelihood

$$\log \Pr \left(\mathbf{y}^{(t)} | \mathbf{x}^{(1)} \dots \mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \dots, \mathbf{y}^{(t-1)} \right). \quad (5)$$

3.1 Encoder-decoder networks

A special variant of RNNs that are suitable for handling variable-length sequential data are encoder-decoder networks (Cho et al., 2014). Encoder-decoder networks are the standard for neural machine translation (Bahdanau et al., 2014; Vinyals et al., 2014) and are also widely used for predictive tasks, including speech recognition (Chorowski et al., 2015) and time-series forecasting (Zhu and Laptev, 2017).

Encoder-decoder networks are trained to estimate the conditional distribution of the output sequence given the past input sequence, e.g., $p(\mathbf{x}^{(t)}, \dots, \mathbf{x}^{(t)} | \mathbf{x}^{(t)}, \dots, \mathbf{x}^{(t)})$, where the input and output sequence lengths can differ. The encoder RNN reads in $\mathbf{x}^{(t)}$ sequentially and the hidden state of the network updates according to Eq. 4. The hidden state of the encoder is a context vector \mathbf{c} that summarizes the input sequence, which is copied over to the decoder RNN. The decoder generates a variable-length output sequence by predicting $\mathbf{x}^{(t)}$ given the encoder hidden state and the previous element of the output sequence. Thus, the hidden state of the decoder is updated recursively by

$$\mathbf{h}^{(t)} = f \left(\mathbf{h}^{(t-1)}, \mathbf{y}^{(t-1)}, \mathbf{c} \right), \quad (6)$$

and the conditional probability of the next element of the sequence is

$$P(\mathbf{x}^{(t)}|\mathbf{y}^{(t)}, \dots, \mathbf{y}^{(t-1)}, \mathbf{c}) = f\left(\mathbf{h}^{(t)}, \mathbf{y}^{(t-1)}, \mathbf{c}\right). \quad (7)$$

Effectively, the decoder learns to generate outputs $\mathbf{x}^{(t)}$ given the previous outputs, conditioned on the input sequence.

3.2 Variational autoencoder

The encoder-decoder model described above predicts the next element of a sequence conditioned on the previous element and an evolving hidden state. While this architecture is effective for many sequence-based predictive tasks, the model does not learn a vector representation of the full sequence. The variational autoencoder (VAE) (Kingma and Welling, 2013) is a generative model that learns a latent variable model for input sequences such that new sequences can be generated from sampling from the latent space. Similar to encoder-decoder networks, the VAE has an encoder that learns a latent representation of the input sequence and a decoder that maps the representation back to the inputs. The VAE architecture differs from encoder-decoder networks in that the VAE doesn't have a final dense layer that compares the decoder outputs to output sequences $\mathbf{x}^{(t)}$ (i.e., it is a "self-supervised" technique).² The other difference is that the VAE maps the inputs to a distribution over latent variables.

The RNN-based VAE proposed by several researchers (Fabius and van Amersfoort, 2014; Chung et al., 2015; Bowman et al., 2015) for sequence modeling consists of an encoder RNN that maps $\mathbf{x}^{(t)}$ to a distribution over parameters of a latent space q . The model then randomly samples \mathbf{z} from the latent distribution, $q(\mathbf{z}|\mathbf{x}^{(t)}) = q(\mathbf{z}; f(\mathbf{x}^{(t)}; \theta))$, where $f(\cdot)$ is a differentiable function with respect to parameter θ (pp. 699 Goodfellow et al., 2016).³ Finally, a decoder RNN takes the form of a conditional probability model $P(\mathbf{x}^{(t)}|\mathbf{z})$.

²Fig. SM-1 and Fig. SM-2 illustrates the architectures of encoder-decoder networks and the VAE, respectively.

³In the empirical applications, $f(\cdot)$ take the form of a log-normal distribution.

The parameters of the model are learned by maximizing the loss function, which takes the difference between the log-likelihood between the decoder outputs and $\mathbf{x}^{(t)}$ and the relative entropy between $q(\mathbf{z}|\mathbf{x}^{(t)})$ and the model prior $p(\mathbf{z})$. The latter component of the loss function acts as regularizer by forcing the learned latent distribution to be similar to the model prior.

4 Predictive accuracy and statistical significance

In applications in which the counterfactual time-series is known (i.e., placebo tests) models can be evaluated in terms of the MSPE between the predicted and actual post-intervention time-series among control units. Specifically, I calculate:

$$\text{MSPE} = \frac{1}{T - n} \sum_{n+1:T} \left(\hat{\phi}^{(t)} \right)^2, \quad (8)$$

where $\hat{\phi}^{(t)}$ is defined in Eq. 3. Eq. 8 measures the accuracy of the counterfactual predictions, and consequently the accuracy of the estimated treatment effect. However, this metric does not tell us anything about the statistical significance of estimated treatment effects.

Abadie et al. (2010) propose a randomization inference approach for calculating the exact distribution of placebo effects under the sharp null hypothesis of no impact. Cavallo et al. (2013) extends the placebo-based testing approach to the case of multiple (placebo) treated units by constructing a distribution of *average* placebo effects under the null hypothesis. Firpo and Possebom (2018) derive the conditions under which the randomization inference approach is valid from a finite sample perspective.⁴ Randomization p -values are obtained following these steps:

1. Estimate the observed test static μ^* by estimating Eq. 8 for all J , which results in a matrix of dimension $(T - n) \times J$. Taking the row-wise mean results in a $T - n$ -length array of observed average placebo treated effects.

⁴Hahn and Shi (2017) analyze the approach from a repeated sampling perspective.

2. Calculate every possible average placebo effect μ by randomly sampling without replacement which $J - 1$ control units are assumed to be treated. There are $\mathcal{Q} = \sum_{g=1}^{J-1} \binom{J}{g}$ possible average placebo effects. The result is a matrix of dimension $(T - n) \times \mathcal{Q}$. Note that \mathcal{Q} can be computationally burdensome when there are many control units. In the applications described below, I set $\mathcal{Q} = 10,000$ in which $J > 16$.
3. Take a column-wise sum of the number of μ that are greater than or equal to μ^* .

Each element of the $(T - n) \times J$ matrix of counts obtained from the last step is divided by \mathcal{Q} to estimate an array of exact two-sided p values, \hat{p} . I then calculate a single false positive rate by $\text{FPR} = \frac{\text{FP}}{(T - n) \times J}$, where FP is the number of false positives defined as the number of p -values less than or equal to $\alpha = 0.5$ and J is the number of placebo treated units.

4.1 Randomization confidence intervals

Assuming that treatment has a constant additive effect Δ , I construct an interval estimate for Δ by inverting the randomization test. Let δ_Δ be the test statistic calculated by subtracting all possible μ by Δ . I derive a two-sided randomization confidence interval by collecting all values of δ_Δ that yield \hat{p} values greater than or equal to a significance level α . I find the endpoints of the confidence interval by randomly sampling 1,000 values of Δ .

5 SCM placebo tests

I evaluate the proposed RNN-based approach on three datasets common to the SCM literature. In each dataset, I remove the actual treated unit and evaluate the models on their ability to produce low error rates on control units; i.e., estimating treatment effects of zero. A secondary evaluation criteria is the probability of falsely rejecting the null hypothesis of the randomization test as measured the FPR. The synthetic control estimator is implemented using the publicly available R code associated with each of the three referenced studies, and importance weights are chosen by cross-validation.

5.1 RNNs implementation details

I train a baseline RNN in the form of a single unidirectional Long Short-Term Memory (LSTM) network (Schmidhuber and Hochreiter, 1997) with output space dimensionality equivalent to the number of treated units. The encoder takes the form of a two-layer bidirectional LSTMs, each with 128 hidden units, and the decoder is a single-layer Gated Recurrent Unit (GRU) (Chung et al., 2014) also with 128 hidden units. An attention mechanism in the form of a softmax mask is included before the first hidden layer in order to generate a normalized distribution of the importance of each time-step regarding an input. In contrast to attention mechanisms used in neural machine translation to assist networks in learning the correct alignment between image pixels and target characters (Cho et al., 2014; Poulos and Valle, 2017), the attention mechanism used in this paper is not expected to help the model perform better, but to help understand which time-steps and inputs contribute to the prediction.

RNN weights are learned with stochastic gradient descent on the MSE using **Adam** stochastic optimization (Kingma and Ba, 2014). As a regularization strategy, I apply dropout to the inputs and L2 regularization losses to the network weights. The networks are implemented with the **Keras** neural network library (Chollet et al., 2015) in Python on top of a TensorFlow backend. RNNs are trained in batches of size four or eight for 5,000 to 10,000 epochs, which takes about 20 minutes to run on a 12GB NVIDIA Titan Xp GPU.

5.2 Basque Country

Abadie and Gardeazabal (2003) estimate the economic impact of terrorism in the Basque Country during the period of 1968 to 1997 by comparing per-capita gross domestic product (GDP) in the Basque Country against a synthetic control region without terrorism. The synthetic control is constructed using pre-period measures of illiteracy, educational attainment, investment, and means of GDP. The time series begins in 1955, which leaves only $n = 14$ pre-period time-steps.

Figs. SM-7 plots estimated treatment effects on control units for each model. We observe considerable variability in the SCM estimates and not as much variability in the RNNs. This is explained by the fact that the SCM cannot handle multiple (placebo) treated units and thus a separate model has to be run for each (placebo) treated unit. RNNs can handle multiple treated units and thus benefit from parameter sharing across treated units. The standard deviation from the mean MPSE, which is reported in the first column of Panel A of Table 1, indicates that the SCM is comparatively more variable in terms of its predictive accuracy. The baseline LSTM yields the lowest mean MSPE, 0.007 ± 0.002 .

Table 1: Evaluation metrics on SCM placebo tests.

Panel A: MSPE	Basque Country	California	West Germany
Encoder-decoder	0.01 ± 0.002	0.005 ± 0.004	0.03 ± 0.01
LSTM (baseline)	0.007 ± 0.002	0.008 ± 0.004	0.02 ± 0.01
SCM	0.007 ± 0.01	0.11 ± 0.13	0.06 ± 0.11
Panel B: FPR			
Encoder-decoder	0.28	0.38	0.25
LSTM (baseline)	0.32	0.33	0.25
SCM	0.27	0.33	0.25

NOTE: Error bars represent \pm one standard deviation from the MSPE.

Fig. SM-8 plots the per-period randomization p -values corresponding to treatment effects on treated and control units. p -values corresponding to treatment effects on the actual treated unit (i.e., Basque Country) are made by comparing the per-period treatment effects on Basque Country against the null distribution of average placebo effects. The SCM has the comparatively lowest FPR (Panel B of Table 1) among all models: the model falsely rejects the null hypothesis about a quarter of the time. Note that the reported p -values are not adjusted for multiple comparisons.

5.3 California

Abadie et al. (2010) applies the SCM to estimate the effects of a large-scale tobacco control program implemented in California in 1988. The study spans the period of 1970 to 2000,

providing $n = 19$ pre-period time-steps. The synthetic control is constructed using pre-period covariates including income, beer sales, demographics, and means of the dependent variable, which is log per-capita cigarette consumption.

Fig. SM-10 shows that for RNNs, control unit treatment effects are tightly centered around zero — as expected — whereas SCM control treatment effects (Fig. SM-??) are more dispersed. Encoder-decoder networks yield the lowest mean MSPE, 0.005 ± 0.004 , while yielding comparatively higher FPR.

5.4 West Germany

Lastly, Abadie et al. (2015) constructs a synthetic West Germany in order to estimate the impact of the 1990 German reunification on log real per-capita GDP during a post-period that extends to 2003. The time-series begins in 1960, which leaves $n = 30$ pre-period time-steps. The synthetic control is constructed using pre-period means of GDP, trade, industry, schooling, and investment.

The SCM and RNN-based approaches both assume the absence of spillover effects. Abadie et al. (2015) acknowledge that spillover effects is a valid concern in their study because German reunification likely have effects on GDP in the 16 OECD member countries that serve as controls. Indeed, Fig. SM-13 shows that RNNs estimate mostly positive (and increasing) treatment effects on control units, which suggests that reunification might have had a less negative impact on German GDP than the authors' estimates suggest.

Overall, the baseline LSTM yields the lowest error in terms of mean MSPE, 0.02 ± 0.01 , with a FRP comparable to the SCM.

5.5 Discussion

I compare the predictive accuracy of the RNN-based approach against the SCM by running a series of placebo tests using data from three datasets common to the SCM literature. The models are evaluated primarily on their ability to produce low error rates on control units

(i.e., estimating treatment effects of zero); a secondary evaluation criteria is minimizing the probability of falsely rejecting the null hypothesis of the randomization test.

I find that either encoder-decoder networks or LSTM outperform the SCM on each of the three datasets in terms of having the lowest MSPE, with FPRs comparable to the SCM. The baseline LSTM outperforms encoder-decoder networks in two of the three datasets. When applied to datasets with low-dimensional predictor sets, the LSTM performs well but deeper networks such as encoder-decoder networks are susceptible to overfitting. Overfitting in this case means that the networks learn dependencies on a small subset of predictors and cannot generalize well to unseen data. Overfitting occurs when training encoder-decoder networks on the Basque Country dataset (Fig. SM-3a), which has the lowest dimensions of the three SCM datasets. Even in this case of obvious overfitting, model check-pointing is employed so that the model with the lowest validation error is used to produce counterfactual time-series.

The results suggest that RNNs should outperform the SCM in all cases as long as the complexity of the network architecture is proportional to the dimension of the predictor set. Encoder-decoder networks outperform the other models when the predictor set is comparatively large (i.e., $J = 38$ in the California dataset), while the baseline LSTM outperforms all other models on smaller predictor sets ($J = 16$ for Basque Country and West Germany datasets).

6 Application: Homestead acts and state capacity

In this section, I estimate the causal impacts of homestead acts on state government education spending. I create a state-level measure of state government education spending from the records of 48 state governments during the period of 1783 to 1932 (Sylla et al., 1993) and the records of 16 state governments during the period of 1933 to 1937 (Sylla et al., 1995a,b). Comparable measures for 48 states are drawn from U.S. Census special reports for the years 1902, 1913, 1932, 1942, 1962, 1972, and 1982 (Haines, 2010).

The data pre-processing steps are as follows. The measure is inflation-adjusted according to the U.S. Consumer Price Index (Williamson, 2017) and scaled by the total free population in the decennial census (Haines, 2010). Missing values are imputed separately in the pre- and -post-periods by carrying the last observation forward and remaining missing values are imputed by carrying the next observation backward. The raw outcomes data are log-transformed to alleviate exponential effects. Lastly, I remove states with no variance in the pre-period outcomes, resulting in complete $N \times T$ matrices of size 33×159 and 34×158 for the expenditures and revenues outcomes, respectively.

In this application, public land states are the treated units and state land states — i.e., states that were not crafted from the public domain and were therefore not directly affected by homestead policies — serve as control units. This group includes states of the original 13 colonies, Maine, Tennessee, Texas, Vermont, and West Virginia. Aggregating to the state level approximately 1.46 million individual land patent records authorized under the HSA,⁵ I determine that the earliest homestead entries occurred in 1869 in about half of the western frontier states, about seven years following the enactment of the HSA.

I train an encoder-decoder network to predict the counterfactual time-series of public land states, using only their previous history to generate predictions. Similar to the placebo tests on the SCM datasets, I evaluate the models two ways: first, I monitor the loss over 2,000 training epochs and save the model weights with the lowest error on a validation set consisting of the final 10% of the time-series.⁶ Second, I calculate MSPE (Eq. 8) on $J = 18$ state land states that serve as placebo treated units. Encoder-decoder networks outperform the baseline LSTM in terms of minimizing the MSPE in placebo tests (Tables SM-1 and SM-2).

Fig. 17 plots the observed and counterfactual time-series for each outcome and region. Counterfactual predictions of state government finances in the absence of homestead acts

⁵Land patent records provide information on the initial transfer of land titles from the federal government and are made accessible online by the U.S. General Land Office (<https://glorerecords.blm.gov>).

⁶The models use both dropout and L2 regularization to control for overfitting on the training set. Figs. SM-?? and SM-15 record the training history of each model.

generally tracks the observed time-series until the turn of the century, at which the counterfactual flattens and diverges from the increasing observed time-series. This delay can potentially be explained by the facts that homesteaders were required to make improvements on land for five years before filing a grant and homestead entries did not substantially accumulate until after the 1889 cash-entry restriction (Figs. SM-?? and SM-??).

Taking the difference between the observed and predicted time-series (Eq. 3) yields time-specific estimates of treatment effects. Fig. 18 plots the temporal evolution of treatment effect estimates over the entire post-period and 95% randomization confidence intervals that are constructed by inverting the randomization test described in the previous section.⁷ Fifty years after its passage, the estimated impact of the HSA on western state government education spending and revenue is 0.005 log points $[-0.16, 0.19]$, and 0.61 log points $[-0.19, 1.53]$, respectively. The confidence intervals surrounding these time-specific estimates contain zero, which implies that the estimated impacts are not significantly more extreme than the placebo treated effects estimated at the same time-step. The confidence interval on the estimated impact of the HSA on western state government expenditure in 1912, an increase of 0.17 log points $[0.004, 0.3]$, does not contain zero, which implies that the estimated impact is significantly more extreme than the placebo treated effects.

An important characteristic of the estimates plotted in Fig. 18 is the progressive widening of confidence intervals. Intuitively, counterfactual predictions become more uncertain as we move farther into the future. In the present application, confidence intervals may become implausibly wide because the post-period extends well into the twentieth century. In order to compare with the DD estimates described in the section below, I average over the entire post-period and find no evidence that the HSA impacted western state government education spending, 0.07 $[-0.32, 0.46]$, expenditure, 0.16 $[-0.21, 0.57]$, or revenue, 0.05 $[-0.33, 0.46]$. Estimates on the impact of the SHA on state capacity in the South are in the same direction and similar magnitudes.

⁷Fig. SM-19 plots the time-specific estimates of randomization p -values inferred from the exact distribution of average placebo effects under the null hypothesis.

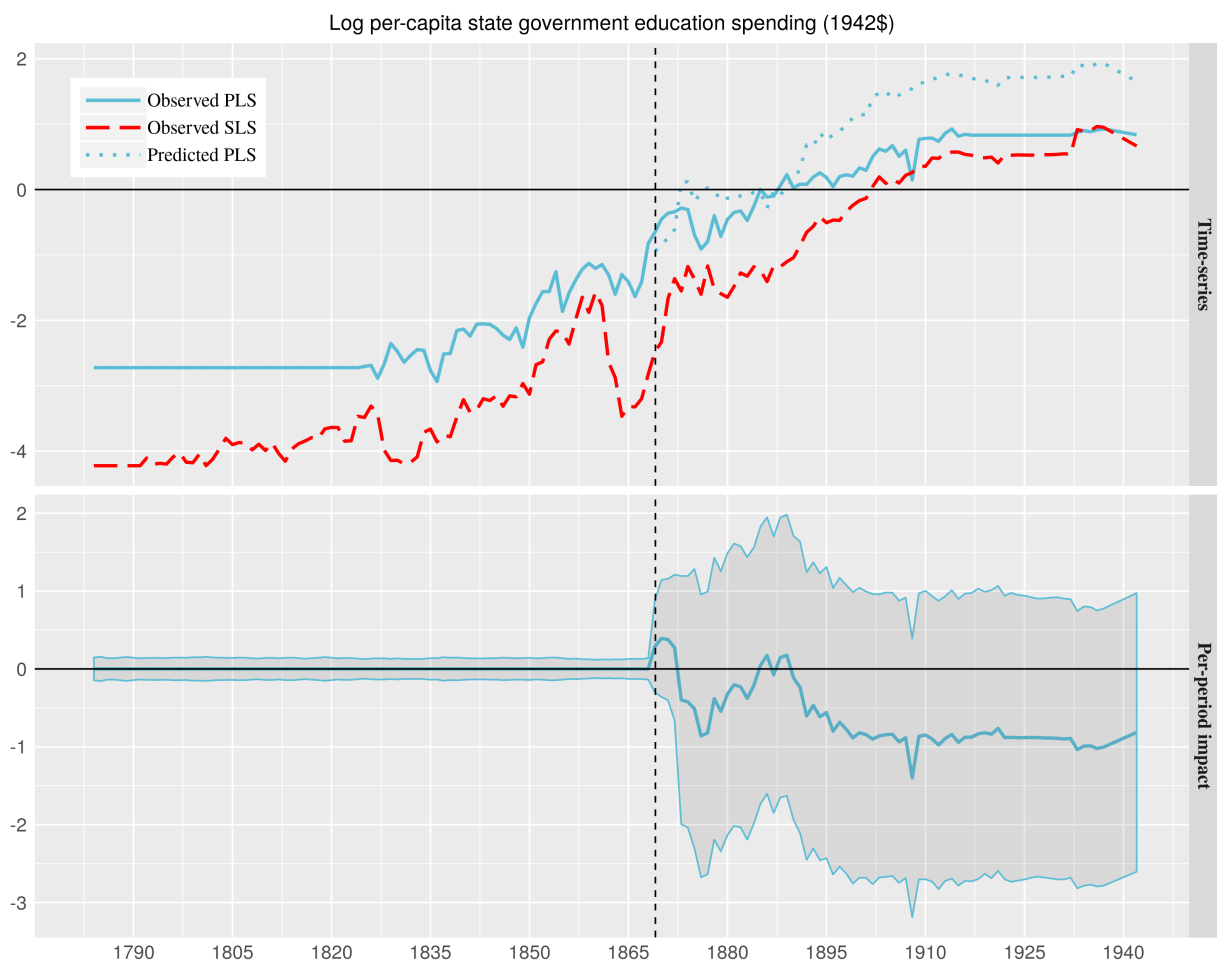


Figure 1: MC-NNM estimates of treatment exposure on state government education spending, 1809 to 1982.

The attention mechanism embedded within the networks provides clues as to which predictors the networks favor at each time-step during the training process. Fig. 3 plots the attention mechanism applied to the test data, where each cell of the heatmap represents the mean attention weight regarding each time-step and predictor across all test samples. The attention distribution suggests that the networks favor heavily winner margins from the 1975 election and also winner margins from the 1988 and 2003 elections when predicting the post-period. Attention is generally distributed evenly across predictors, which suggests that the networks rely on many different non-treated cities to construct the counterfactual.

Figure 2: Attention mechanism as a function of predictors (i.e., winner margins by city) and time-steps for encoder-decoder networks. Attention is the normalized (softmax) distribution of the importance of each time-step regarding a predictor.

6.1 Discussion

7 Conclusion

This paper makes a methodological contribution in proposing a novel alternative to the SCM for estimating the effect of a policy intervention on an outcome over time in settings where appropriate control units are unavailable. The SCM is growing in popularity in the social sciences despite its limitations — the most obvious being that the choice of specification can lead to different results, and thus facilitate *p*-hacking. By inputting only control unit outcomes and not relying on pre-period covariates, the proposed method offers a more principled approach than the SCM.

In placebo tests, RNN-based models outperform the SCM in terms of predictive accuracy while yielding a comparable proportion of false positives. RNNs have advantages over the SCM in that they are structured for sequential data and can learn nonconvex combinations of predictors, which is beneficial when the data-generating process underlying the outcome of interest depends nonlinearly on the history of its inputs. RNNs are also capable of handling

multiple treated units and can learn nonconvex combinations of control units, which is useful because the model can share parameters across treated units, and thus generate more precise predictions in settings in which treated units share similar data-generating processes.

The RNN-based approach joins a new generation of data-driven machine learning techniques for generating counterfactual predictions. Machine learning techniques in general have an advantage over the SCM in that they automatically choose appropriate predictors without relying on pretreatment covariates; this capability limits “researcher degrees of freedom” that arises from choices on how to specify the model. RNNs have an advantage over alternative machine learning algorithms because they are specifically structured to exploit the sequential nature of time-series data by sharing model parameters across time-steps.

Future research might investigate through simulations how the interaction between RNN complexity (as determined by the number of hidden layers or nodes) and data dimensionality impacts predictive accuracy. Simulations will also allow us to assess the exact impact of data dimensionality, the proportion of treated units, convexity versus non-convexity in the modeled relationship, and the length of the pre-period on the choice between RNNs and the SCM.

References

- Abadie, A., Diamond, A. and 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**, 493–505.
- (2015) Comparative politics and the synthetic control method. *American Journal of Political Science*, **59**, 495–510.
- Abadie, A. and Gardeazabal, J. (2003) The economic costs of conflict: A case study of the basque country. *The American Economic Review*, **93**, 113–132.
- Amjad, M., Shah, D. and Shen, D. (2018) Robust synthetic control. *The Journal of Machine Learning Research*, **19**, 802–852.
- Athey, S., Bayati, M., Doudchenko, N., Imbens, G. and Khosravi, K. (2017) Matrix Completion Methods for Causal Panel Data Models. *ArXiv e-prints*.
- Bahdanau, D., Cho, K. and Bengio, Y. (2014) Neural Machine Translation by Jointly Learning to Align and Translate. *ArXiv e-prints*.
- Ben-David, S., Blitzer, J., Crammer, K. and Pereira, F. (2007) Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, 137–144.
- Bickel, S., Brückner, M. and Scheffer, T. (2009) Discriminative learning under covariate shift. *Journal of Machine Learning Research*, **10**, 2137–2155.
- Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R. and Bengio, S. (2015) Generating sentences from a continuous space. *arXiv:1511.06349*.
- Carvalho, C., Masini, R. and Medeiros, M. C. (2018) Arco: an artificial counterfactual approach for high-dimensional panel time-series data. *Journal of Econometrics*, **207**, 352–380.
- Cavallo, E., Galiani, S., Noy, I. and Pantano, J. (2013) Catastrophic natural disasters and economic growth. *Review of Economics and Statistics*, **95**, 1549–1561.
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y. (2014) Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *ArXiv e-prints*.
- Chollet, F. et al. (2015) Keras. <https://keras.io>.
- Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K. and Bengio, Y. (2015) Attention-based models for speech recognition. In *Advances in Neural Information Processing Systems*, 577–585.
- Chung, J., Gulcehre, C., Cho, K. and Bengio, Y. (2014) Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *ArXiv e-prints*.
- Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C. and Bengio, Y. (2015) A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems*, 2980–2988.
- Cinar, Y. G., Mirisae, H., Goswami, P., Gaussier, E., Aït-Bachir, A. and Strijov, V. (2017) Position-based content attention for time series forecasting with sequence-to-sequence rnns. In *International Conference on Neural Information Processing*, 533–544. Springer.
- Cortes, C., Mansour, Y. and Mohri, M. (2010) Learning bounds for importance weighting. In *Advances in Neural Information Processing Systems*, 442–450.
- Cortes, C., Mohri, M., Riley, M. and Rostamizadeh, A. (2008) Sample selection bias correction theory. In *International Conference on Algorithmic Learning Theory*, 38–53. Springer.
- Doudchenko, N. and Imbens, G. W. (2016) Balancing, Regression, Difference-In-Differences and Synthetic Control Methods: A Synthesis. *ArXiv e-prints*.
- Dube, A. and Zipperer, B. (2015) Pooling multiple case studies using synthetic controls: An application to minimum wage policies.
- El Hihi, S. and Bengio, Y. (1995) Hierarchical recurrent neural networks for long-term dependencies. In *Neural Information Processing Systems*, vol. 400, 409.
- Fabius, O. and van Amersfoort, J. R. (2014) Variational recurrent auto-encoders. *arXiv:1412.6581*.
- Ferman, B. and Pinto, C. (2016) Revisiting the synthetic control estimator. Available at <https://mpra.ub.uni-muenchen.de/81941/>.

- (2018) Synthetic controls with imperfect pre-treatment fit. Available at: <https://sites.google.com/site/brunoferman/research>.
- Ferman, B., Pinto, C. and Possebom, V. (2018) Cherry picking with synthetic controls. Available at: https://mpira.ub.uni-muenchen.de/85138/1/MPRA_paper_85138.pdf.
- Firpo, S. and Possebom, V. (2018) Synthetic control method: Inference, sensitivity analysis and confidence sets. *Journal of Causal Inference*, **6**.
- Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M. and Lempitsky, V. (2015) Domain-Adversarial Training of Neural Networks. *arXiv e-prints*, arXiv:1505.07818.
- Goodfellow, I., Bengio, Y. and Courville, A. (2016) *Deep Learning*. Cambridge, MA: MIT press.
- Graves, A. (2012) Neural networks. In *Supervised Sequence Labelling with Recurrent Neural Networks*, 15–35. Springer.
- Hahn, J. and Shi, R. (2017) Synthetic control and inference. *Econometrics*, **5**, 52.
- Haines, M. R. (2010) Historical, Demographic, Economic, and Social Data: The United States, 1790–2002. Ann Arbor, MI: Inter-university Consortium for Political and Social Research [distributor], 2010-05-21. doi.org/10.3886/ICPSR02896.v3.
- Huang, J., Gretton, A., Borgwardt, K. M., Schölkopf, B. and Smola, A. J. (2007) Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems*, 601–608.
- Imbens, G. W. and Rubin, D. B. (2015) *Causal Inference in Statistics, Social, and Biomedical Sciences*. Cambridge: Cambridge University Press.
- Johansson, F., Shalit, U. and Sontag, D. (2016) Learning representations for counterfactual inference. In *International Conference on Machine Learning*, 3020–3029.
- Kaul, A., Klößner, S., Pfeifer, G. and Schieler, M. (2015) Synthetic control methods: Never use all pre-intervention outcomes together with covariates. Available at: http://www.oekonometrie.uni-saarland.de/papers/SCM_Predictors.pdf.
- Kingma, D. P. and Ba, J. (2014) Adam: A Method for Stochastic Optimization. *ArXiv e-prints*.
- Kingma, D. P. and Welling, M. (2013) Auto-encoding variational bayes. *arXiv:1312.6114*.
- Klößner, S., Kaul, A., Pfeifer, G. and Schieler, M. (2017) Comparative politics and the synthetic control method revisited: A note on Abadie et al. (2015). *Swiss Journal of Economics and Statistics*.
- Lamb, A. M., Goyal, A. G. A. P., Zhang, Y., Zhang, S., Courville, A. C. and Bengio, Y. (2016) Professor forcing: A new algorithm for training recurrent networks. In *Advances In Neural Information Processing Systems*, 4601–4609.
- Neyman, J. (1923) On the application of probability theory to agricultural experiments. *Annals of Agricultural Sciences*, **51**. Reprinted in Splawa-Neyman et al. (1990).
- Poulos, J. (2019) State-Building through Public Land Disposal? An Application of Matrix Completion for Counterfactual Prediction. *arXiv e-prints*, arXiv:1903.08028.
- Poulos, J. and Valle, R. (2017) Attention networks for image-to-text. *arXiv e-prints*, arXiv:1712.04046.
- Rubin, D. B. (1990) Comment: Neyman (1923) and causal inference in experiments and observational studies. *Statistical Science*, **5**, 472–480.
- Schmidhuber, J. and Hochreiter, S. (1997) Long short-term memory. *Neural Computation*, **9**, 1735–1780.
- Schnabel, T., Swaminathan, A., Singh, A., Chandak, N. and Joachims, T. (2016) Recommendations as treatments: Debiasing learning and evaluation. *arXiv:1602.05352*.
- Splawa-Neyman, J., Dabrowska, D., Speed, T. et al. (1990) On the application of probability theory to agricultural experiments. essay on principles. section 9. *Statistical Science*, **5**, 465–472.
- Sylla, R. E., Legler, J. B. and Wallis, J. (1993) Sources and Uses of Funds in State and Local Governments, 1790–1915: [United States]. Ann Arbor, MI: Inter-university Consortium for Political and Social Research, 2017-05-21. doi.org/10.3886/ICPSR06304.v1.

- (1995a) State and Local Government [United States]: Sources and Uses of Funds, Census Statistics, Twentieth Century [Through 1982]. Ann Arbor, MI: Inter-university Consortium for Political and Social Research, 2017-05-21. doi.org/10.3886/ICPSR06304.v1.
- (1995b) State and Local Government [United States]: Sources and Uses of Funds, State Financial Statistics, 1933-1937. Ann Arbor, MI: Inter-university Consortium for Political and Social Research, 2017-05-21. <http://doi.org/10.3886/ICPSR06306.v1>.
- Vinyals, O., Kaiser, L., Koo, T., Petrov, S., Sutskever, I. and Hinton, G. (2014) Grammar as a Foreign Language. *ArXiv e-prints*.
- Williamson, S. H. (2017) Seven ways to compute the relative value of a us dollar amount, 1774 to present. *MeasuringWorth.com*. [Online; accessed 01-October-2017].
- Xu, Y. (2017) Generalized synthetic control method: Causal inference with interactive fixed effects models. *Political Analysis*, **25**, 57–76.
- Zhu, L. and Laptev, N. (2017) Deep and Confident Prediction for Time Series at Uber. *ArXiv e-prints*.