Synthetic Diff-in-Diff

Professor Song Yao
Olin Business School

Customer Analytics

Limitations of Diff-in-Diff and Synthetic Control

- One key limitation of diff-in-diff
  - The parallel trends assumption does not always hold
- One key limitation of synthetic control method (SCM)
  - A single treated unit
- Solution? Synthetic Diff-in-Diff
  - Combine SCM and Diff-in-Diff
  - Improve causal inference with *panel data (aka longitudinal data)*

A Real-World Application

The tabacco tax

- Setting
  - Multiple units and multiple periods
  - At a given time, some treated and some untreated
  - Treatment may not happen at the same time
    - AKA, staggered treatment
- Let's focus on California first
  - Treatment started in 1989
  - 38 donor states

Diff-in-Diff Result

```python
### Diff-in-diff

# Exclude states with large tax hikes soon after 1988
# This is based on some additional Google search during which
# we found out these states raised their tabacco tax significantly
# soon after California's Proposition 99
# Create list of states to exclude
excluded_states = ['Arizona', 'Michigan', 'Massachusetts']

# Create diff-in-diff dataframe excluding those states
df_did = df[~df['state'].isin(excluded_states)].copy()
# Create after_tax indicator for years >= 1989
df_did['after_tax'] = (df_did['year'] >= 1989).astype(int)
# Create treatment indicator for California
df_did['treated'] = (df_did['state'] == 'California').astype(int)

# Run DiD regression with state and year fixed effects
did_model = smf.ols('cigsale ~ treated * after_tax + C(state) + C(year)', data=df_did).fit()
print(did_model.summary())


# Calculate percentage change use average control sales as baseline (donor units after 1989)
sales_baseline_did = \
    df_did[(df_did['treated'] == 0) & (df_did['after_tax'] == 1)]['cigsale'].mean()
did_percent_change = (did_model.params['treated:after_tax'] / sales_baseline_did) * 100
print(f"Percent Change: {round(did_percent_change, 1)}%")
```

```
treated                 -2.1502      1.725     -1.246      0.213      -5.535       1.234
after_tax              -17.0483      1.819     -9.372      0.000     -20.617     -13.479
treated:after_tax      -27.3491      4.409     -6.202      0.000     -36.001     -18.698

Percent Change: -26.8%
```

Diff-in-Diff Reformulation

California $Y_{it} = \beta_0 + \beta_1 Post_t + \beta_2 Treated_i + \beta_3 Treated_i Post_t + e_{it}$

- Data structure

$$D = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$$

$$Y = \begin{bmatrix} \boldsymbol{Y}_{pre,co} & \boldsymbol{Y}_{pre,tr} \\ \boldsymbol{Y}_{post,co} & \boldsymbol{Y}_{post,tr} \end{bmatrix}$$

- Diff-in-diff reformulation

$$\sum_{i=1}^{N}\sum_{t=1}^{T}\left(Y_{it} - (\mu + \alpha_i + \beta_t + \tau D_{it}\right)^2$$

SCM Revisited

```
# The synthetic California's per capita consumption is the weighted sum of
# the donor states' per capita consumptions
synthetic_california_new = np.dot(wide_data[donor_states].values, calif_weights)

# Calculate treatment effect (difference between actual and synthetic)
diff_cigsale = wide_data['California'] - synthetic_california_new

# Calculate average treatment effect on the treated after implementation (post-1989)
att_post = diff_cigsale[wide_data.index >= 1989].mean()
print("\nATT after 1989:", round(att_post, 2), "packs per capita")

# Calculate percentage change
baseline = synthetic_california_new[wide_data.index >= 1989].mean()
percent_change = (att_post / baseline) * 100
print(f"Percent Change: {round(percent_change, 1)}%")
```

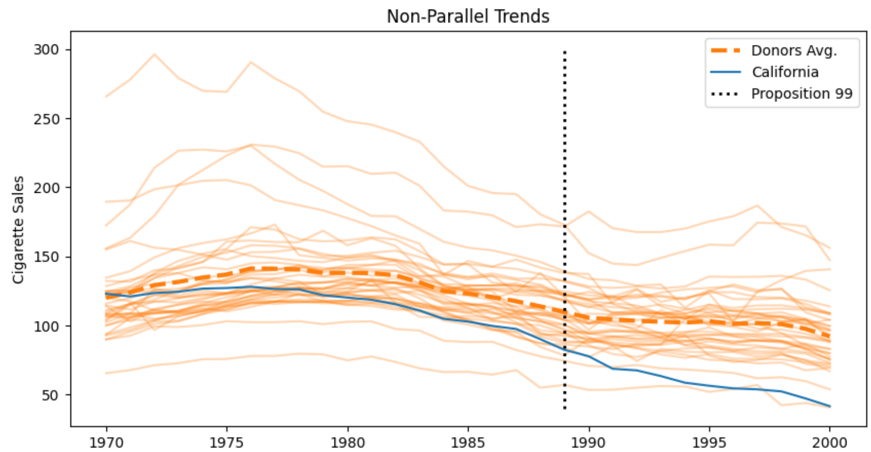
```
ATT after 1989: -19.51 packs per capita
Percent Change: -24.4%
```

$$\sum_{i=1}^{N}\sum_{t=1}^{T}\left(Y_{it} - \beta_t - \tau D_{it}\right)^2 \hat{w}_i^{sc}$$

Reasons behind the different effect size estimates?

Assumptions of Diff-in-Diff

- **<u>Parallel trends (key assumption)</u>**
  - California and other donor states are too different
- No spillovers (SUTVA)



Synthetic Diff-in-Diff

Overview of the intuition

- Construct unit weights w as we did in SCM
- Construct *<u>time weights k</u>*
- Apply *<u>weighted DiD</u>* regression to estimate the treatment effect

Formulation of SDID

Combine SCM and DiD

- SCM

$$\sum_{i=1}^{N}\sum_{t=1}^{T}\left(Y_{it}-\beta_t-\tau D_{it}\right)^2\hat{w}_i^{sc}$$

- DiD

$$\sum_{i=1}^{N}\sum_{t=1}^{T}\left(Y_{it}-(\mu+\alpha_i+\beta_t+\tau D_{it}\right)^2$$

- SDiD

$$\sum_{i=1}^{N}\sum_{t=1}^{T}\left(Y_{it}-(\mu+\alpha_i+\beta_t+\tau D_{it}\right)^2\hat{w}_i^{sdid}\hat{k}_t^{sdid}$$

What does the time weights k do?

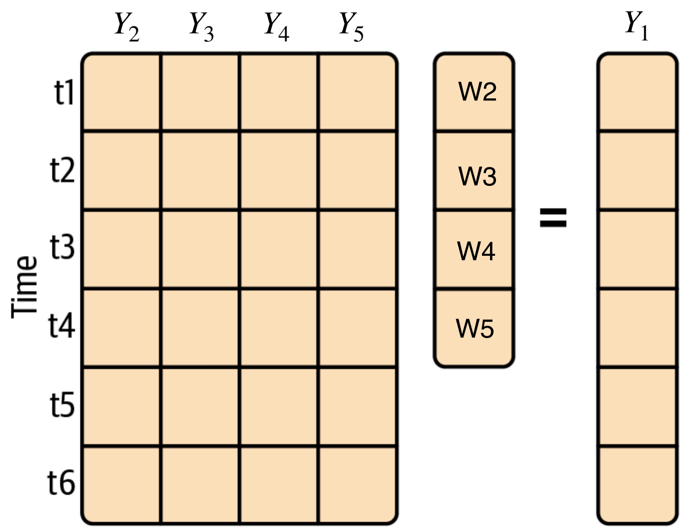
Weights that minimize the effect of noisy pre-treatment periods

- Some pre-treatment periods are noisy
  - E.g., control units' outcomes have much larger deviations from the post-treatment average
- Weight those stable/close pre-treatment periods more
- Weight those volatile/distant pre-treatment periods less

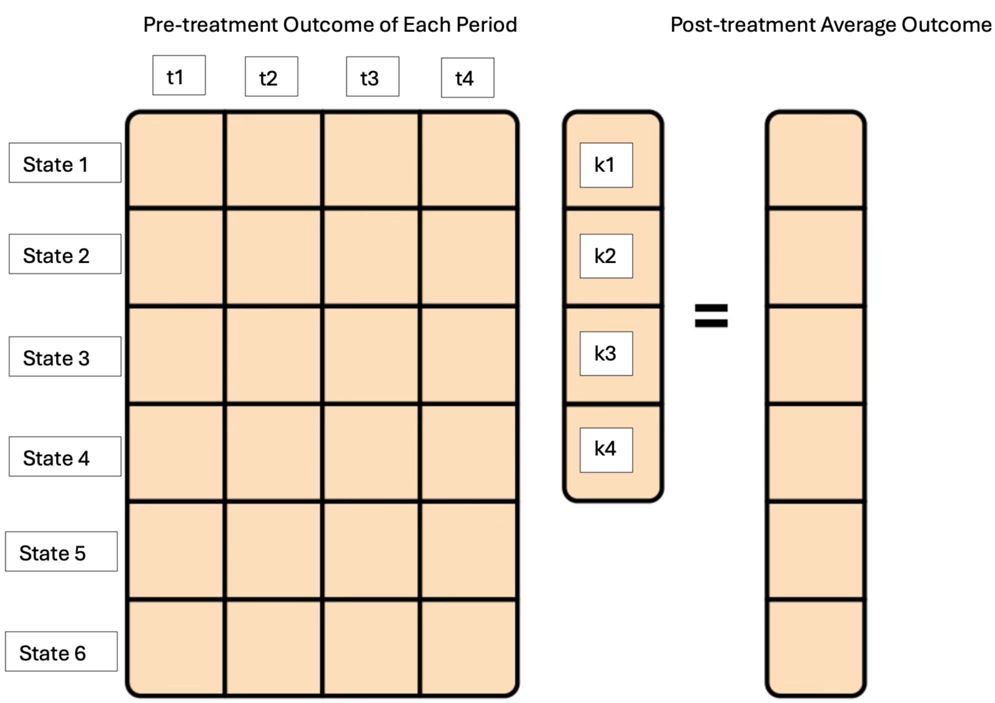
$$\hat{k}^{sdid}=\underset{k}{\text{argmin}}\ ||\bar{\boldsymbol{y}}_{post,co}-(\boldsymbol{k}_{pre}\boldsymbol{Y}_{pre,co}+k_0)||_2^2$$

$$\text{s.t}\quad \sum k_t=1 \text{ and } \ k_t>0\ \forall\ t$$

Very Similar to how we get the W in SCM



Very Similar to how we get the W in SCM



Implementation

Very similar to how we get the W in SCM

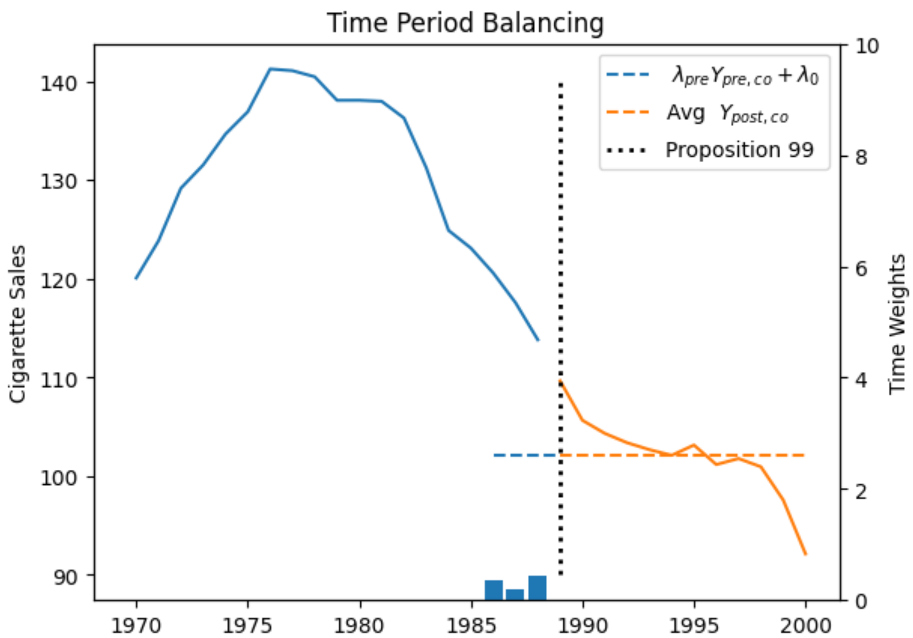
- Focusing on control (donor) states
- Compute the average outcome of post-treatment of each state (i.e., average cigsale across all control states and all post-treatment years)
- Regress
  - Each state's post-treatment average ~ Each state's pre-treatment yearly cigsale * k
  - sum(k) = 1 and k is bounded between 0 and 1

Estimating the Time Weights

More details during demo

```
import cvxpy as cp # convex optimization library
def fit_time_weights(data, outcome_col, year_col, state_col, treat_col, post_col):

        control = data.query(f"~{treat_col}")

        # pivot the data to the (T_pre, N_co) matrix representation
        y_pre = (control
                 .query(f"~{post_col}")
                 .pivot(index=year_col, columns=state_col, values=outcome_col))

        # group post-treatment time period by units to have a (1, N_co) vector.
        y_post_mean = (control
                       .query(f"{post_col}")
                       .groupby(state_col)
                       [outcome_col]
                       .mean()
                       .values)

        # add a (1, N_co) vector of 1 to the top of the matrix, to serve as the intercept.
        X = np.concatenate([np.ones((1, y_pre.shape[1])), y_pre.values], axis=0)

        # estimate time weights
        w = cp.Variable(X.shape[0])
        objective = cp.Minimize(cp.sum_squares(w@X - y_post_mean))
        constraints = [cp.sum(w[1:]) == 1, w[1:] >= 0]
        problem = cp.Problem(objective, constraints)
        problem.solve(verbose=False)

        # print("Intercept: ", w.value[0])
        return pd.Series(w.value[1:], # remove intercept
                         name="time_weights",
                         index=y_pre.index)
```

Noisy Pre-treatment Periods Have Little Weight



Diff-in-Diff with Unit and Time Weights (w,k)

$$\sum_{i=1}^{N} \sum_{t=1}^{T} \left(Y_{it} - (\mu + \alpha_i + \beta_t + \tau D_{it}\right)^2 \hat{w}_i^{sdid} \hat{k}_t^{sdid}$$

```
sdid_data = join_weights(weights_df, unit_weights, time_weights,
                        year_col="year",
                        state_col="state",
                        treat_col="treated",
                        post_col="after_tax")

sdid_data.head()
```

	year	state	cigsale	lnincome	beer	age15to24	retprice	treated	after_tax	time_weights	unit_weights	weights
0	1970	Rhode Island	123.900000	NaN	NaN	0.183158	39.299999	0	0	-4.600031e-14	1.290447e-03	-0.0
1	1970	Tennessee	99.800003	NaN	NaN	0.178044	39.900002	0	0	-4.600031e-14	-1.322115e-16	0.0
2	1970	Indiana	134.600010	NaN	NaN	0.176516	30.600000	0	0	-4.600031e-14	1.031292e-02	-0.0
3	1970	Nevada	189.500000	NaN	NaN	0.161554	38.900002	0	0	-4.600031e-14	1.241939e-01	-0.0
4	1970	Louisiana	115.900000	NaN	NaN	0.185185	34.299999	0	0	-4.600031e-14	-8.281903e-17	0.0

Weighted Least Squared

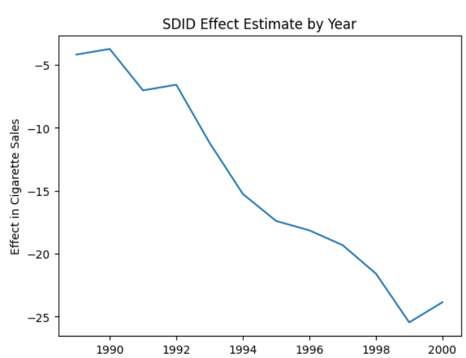
```
sdid_model = smf.wls("cigsale ~ after_tax*treated",
                     data=sdid_data,
                     weights=sdid_data["weights"]+1e-10).fit()

sdid_model.summary().tables[1]
```

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	120.4060	1.272	94.665	0.000	117.911	122.901
after_tax	-19.1905	1.799	-10.669	0.000	-22.720	-15.661
treated	-25.2601	1.799	-14.043	0.000	-28.789	-21.731
after_tax:treated	-15.6054	2.544	-6.135	0.000	-20.596	-10.615

Treatment Effects Overtime—SDiD for Each Period

Note: Only CA treated in our data—But the intuition carries.

```
effects = {year: synthetic_diff_in_diff(weights_df.query(f"~after_tax|(year=={year})"),
                                        outcome_col="cigsale",
                                        year_col="year",
                                        state_col="state",
                                        treat_col="treated",
                                        post_col="after_tax")
           for year in range(1989, 2001)}

effects = pd.Series(effects)

plt.plot(effects);
plt.ylabel("Effect in Cigarette Sales")
plt.title("SDID Effect Estimate by Year");
```



How Do We Deal with Multiple Treated Units?

"Staggered Treatment"

- First, we can run the SDiD for each post-treatment period
  - Run SDiD for 1989, 1990, …, 2000 iteratively
  - The average effect across years gives the ATT
- If there California was treated in 1989, Massachusetts in 1993, Arizona and Michigan in 1994
  - SDiD for 1989-1992, Treated==1 for CA
  - SDiD for 1993, Treated==1 for CA, MA
  - SDiD for 1994-2000, Treated==1 for CA, MA, AZ, and MI
  - The average effect across years gives the ATT