Contents

From Classical Econometrics to Causal Machine Learning

TARNet for Housing Price Policy Analysis

Project: Estimating the Causal Impact of LTV (Loan-to-Value) Tightening on Housing Prices using Treatment-Agnostic Representation Networks

Author: Dimaso
Date: 2024
Framework: TensorFlow/Keras

Table of Contents

Executive Summary

This project demonstrates the transition from classical Structural Vector Autoregression (SVAR) econometric methods to modern causal machine learning approaches for analyzing macroeconomic policy impacts. Using a Treatment-Agnostic Representation Network (TARNet), we estimate the causal effect of LTV tightening policies on housing price changes, allowing for:

- **Nonlinear relationships** between macroeconomic variables
- **Heterogeneous treatment effects** across different regions and economic conditions
- **Counterfactual predictions** for policy evaluation

Key Result: The model successfully estimates individual and average treatment effects, revealing heterogeneous policy impacts that vary by credit conditions, GDP growth, and regional characteristics.

Introduction: From SVAR to Causal ML

Motivation

Traditional SVAR models in econometrics assume: - Linear relationships - Aggregate effects only - Fixed structural relationships

Modern causal ML approaches (like TARNet) enable: - **Nonlinear patterns** via neural networks - **Individual-level predictions** (ITE - Individual Treatment Effects) - **Conditional treatment effects** (CATE - Conditional Average Treatment Effects) - **Natural counterfactual generation**

Research Question

How do macroprudential policies (specifically LTV tightening) causally affect housing prices, and how do these effects vary across different economic conditions?

Theoretical Background

Causal Inference Fundamentals

The Fundamental Problem of Causal Inference For each unit i, we observe: - $Y_i(0)$: Potential outcome under control (no treatment) - $Y_i(1)$: Potential outcome under treatment - But we can only observe **one** of these outcomes

Observed outcome: $Y_i = T_i \cdot Y_i(1) + (1 - T_i) \cdot Y_i(0)$

where $T_i \in \{0, 1\}$ is the treatment indicator.

Average Treatment Effect (ATE)

$$ATE = E[Y(1) - Y(0)] = E[Y(1)] - E[Y(0)]$$

Interpretation: The average causal effect of treatment across the entire population.

Individual Treatment Effect (ITE)

$$ITE_i = Y_i(1) - Y_i(0)$$

Interpretation: The causal effect for a specific individual/region/time period.

Conditional Average Treatment Effect (CATE)

$$CATE(x) = E[Y(1) - Y(0)|X = x]$$

Interpretation: The average treatment effect conditional on specific characteristics (e.g., high credit gap, high GDP regions).

TARNet: Treatment-Agnostic Representation Network

TARNet addresses causal inference by:

1. **Learning a shared representation** $\Phi(X)$ of covariates
2. **Predicting both potential outcomes:**
   - $\hat{Y}(0) = f_0(\Phi(X))$ - outcome under control
   - $\hat{Y}(1) = f_1(\Phi(X))$ - outcome under treatment
3. **Computing ITE:** $I\hat{T}E_i = \hat{Y}_i(1) - \hat{Y}_i(0)$

Key Insight: Both heads share the same representation, allowing the model to generalize counterfactual predictions.

Research Design

Treatment Variable

LTV Tightening (Binary): - $T = 1$: LTV policy tightened (restrictive policy) - $T = 0$: No LTV tightening (baseline policy)

Outcome Variable

Housing Price Change (%): - Year-over-year or quarter-over-quarter percentage change - Continuous variable

Control Variables (Covariates)

1. **GDP Growth** - Economic activity
2. **Credit Gap** - Credit cycle position (deviation from trend)
3. **Inflation** - Price stability
4. **Interest Rate** - Monetary policy stance
5. **Exchange Rate** - External conditions
6. **Household Debt-to-Income** - Household leverage
7. **Unemployment** - Labor market conditions
8. **Lagged Housing Price** - Price momentum

Key Assumptions

1. **Unconfoundedness (Ignorability):**

   $$Y(0), Y(1) \perp T | X$$

   Treatment assignment is independent of potential outcomes given observed covariates.

2. **Overlap (Positivity):**

   $$0 < P(T = 1|X) < 1$$

   Every unit has a positive probability of receiving either treatment.

Data and Methodology

Data Generation (Synthetic)

For this demonstration, we generate synthetic data with: - **3,000 observations** across 5 regions - **Clear causal structure** with heterogeneous treatment effects - **Realistic correlations** between macroeconomic variables - **Selection bias** (treatment assignment depends on covariates)

Preprocessing Pipeline

1. **Missing Value Handling:**
    - Forward-fill for time-series variables
    - Median imputation for others
2. **Feature Normalization:**
    - StandardScaler (mean=0, std=1)
    - Critical for neural network training
3. **Train/Validation/Test Split:**
    - Train: 70% (2,100 samples)
    - Validation: 10% (300 samples)
    - Test: 20% (600 samples)
    - Stratified by treatment to maintain balance

Model Architecture: TARNet

Network Structure

```
Input (8 features)
    ↓
Shared Representation Network
      Dense(128) → BatchNorm → Dropout(0.2)
      Dense(64) → BatchNorm → Dropout(0.2)
      Dense(32) → BatchNorm
    ↓
Representation Φ(X)
     → Outcome Head 0 (Control)
         Dense(32) → Dropout(0.2)
         Dense(16) → Dropout(0.2)
         Dense(1) → Ŷ(0)

     → Outcome Head 1 (Treatment)
         Dense(32) → Dropout(0.2)
         Dense(16) → Dropout(0.2)
         Dense(1) → Ŷ(1)
```

Training Strategy

Loss Function: - MSE for each outcome head - Each head trained only on observed outcomes: - Control samples ($T = 0$) train $Y(0)$ head - Treated samples ($T = 1$) train $Y(1)$ head

Regularization: - Dropout (0.2) to prevent overfitting - L2 regularization (0.001) on weights - Batch normalization for stable training

Training Details: - Optimizer: Adam (learning_rate=0.001) - Batch size: 32 - Epochs: 100 - Early stopping based on validation loss

---

Results and Interpretation

1. Average Treatment Effect (ATE)

Formula:

$$ATE = \frac{1}{n}\sum_{i=1}^{n}[\hat{Y}_i(1) - \hat{Y}_i(0)]$$

Interpretation: - **ATE < 0:** On average, LTV tightening **decreases** housing price growth - **ATE > 0:** On average, LTV tightening **increases** housing price growth - **Magnitude:** The size indicates the strength of the policy effect

Example Result:

```
ATE = -0.489
95% CI: [-0.555, -0.424]
```

Interpretation: LTV tightening reduces housing price growth by an average of **0.49 percentage points**. The negative sign suggests the policy is effective in cooling the housing market.

2. Individual Treatment Effects (ITE)

Formula:

$$ITE_i = \hat{Y}_i(1) - \hat{Y}_i(0)$$

Distribution Analysis:

- **Mean ITE:** Typically close to ATE
- **Standard Deviation:** Measures heterogeneity
- **Range:** Shows variation in policy impact

Interpretation: - **Negative ITE:** Policy reduces prices for this unit - **Positive ITE:** Policy increases prices for this unit (rare but possible) - **Large variance:** High heterogeneity suggests policy effectiveness depends on context

Visualization: A histogram of ITE values shows: - Most units have negative effects (policy works as intended) - Some units have stronger/weaker responses - Distribution shape indicates heterogeneity level

3. Conditional Average Treatment Effects (CATE)

CATE by Credit Gap **High Credit Gap Regions:**

```
CATE = -0.65 ± 0.12
```

Low Credit Gap Regions:

```
CATE = -0.32 ± 0.09
```

Interpretation: - Policy is **more effective** (larger negative effect) in regions with high credit gaps - This suggests LTV tightening works better when credit cycles are overheated - **Policy implication:** Target policies to regions with credit imbalances

CATE by GDP Growth **High GDP Growth Regions:**

```
CATE = -0.45 ± 0.11
```

Low GDP Growth Regions:

```
CATE = -0.55 ± 0.10
```

Interpretation: - Policy effects are similar but slightly stronger in slower-growing regions - May reflect different market dynamics during economic expansions vs. contractions

CATE by Treatment Status **Treated Group:**

```
CATE = -0.52 ± 0.15
```

Control Group (Counterfactual):

```
CATE = -0.47 ± 0.14
```

Interpretation: - Both groups would experience similar average effects if treated - Suggests minimal selection bias (good model fit)

4. Counterfactual Analysis

What would have happened without the policy?

For treated units ($T = 1$): - **Observed:** Housing price change with LTV tightening - **Counterfactual:** Predicted price change without tightening - **Difference:** Policy effect

Visualization: Time series plots show: - Red line: Observed trajectory (with treatment) - Blue dashed line: Counterfactual (without treatment) - Gray fill: Treatment effect magnitude

Key Insights: 1. **Treated regions:** Shows how prices evolved with policy vs. without 2. **Control regions:** Shows predicted effect if policy had been applied 3. **Aggregate effect:** Average treatment effect over time

---

Key Findings

1. Policy Effectiveness

☐ **LTV tightening effectively reduces housing price growth** - Average reduction: ~0.49 percentage points - Effect is statistically significant (narrow confidence intervals)

2. Heterogeneous Effects

☐ **Policy effectiveness varies significantly across conditions:** - **2x stronger** in high credit gap regions (-0.65 vs. -0.32) - Suggests targeted policy application could be more effective

3. Nonlinear Patterns

☐ **Neural network captures complex relationships:** - Traditional linear models might miss these patterns - TARNet reveals context-dependent policy effects

4. Practical Utility

☐ **Actionable insights for policymakers:** - Identify regions where policy will be most effective - Predict individual-level policy impacts - Generate counterfactual scenarios for policy evaluation

---

Practical Applications

For Policymakers

1. **Targeted Policy Application:**
   - Apply LTV tightening in regions with high credit gaps
   - Expect stronger effects in overheated markets
2. **Policy Evaluation:**
   - Compare actual outcomes to counterfactual predictions
   - Assess policy effectiveness post-implementation
3. **Scenario Planning:**
   - Predict effects of policy changes before implementation
   - Evaluate different policy intensities

For Researchers

1. **Methodological Advancement:**
   - Demonstrates superiority of ML approaches over linear models
   - Enables discovery of heterogeneous effects
2. **Replication:**
   - Clear pipeline from data to results
   - Reproducible methodology

For Practitioners

1. **Real Estate Analysis:**
   - Understand policy impact on housing markets
   - Identify regions most affected by policy changes
2. **Investment Decisions:**
   - Anticipate policy effects on property values
   - Adjust strategies based on predicted impacts

Code Structure

Main Components

1. Data Generation

```python
def generate_synthetic_housing_data():
    # Creates realistic housing market data
    # With known causal structure
    return df
```

2. Preprocessing

```python
def preprocess_data(df, feature_cols, target_col):
    # Handles missing values
    # Normalizes features
    # Splits into train/val/test
    return data_dict
```

3. Model Building

```python
def build_tarnet(input_dim):
    # Shared representation network
    # Two outcome heads (y0, y1)
    return model
```

4. Causal Effect Estimation

```python
def compute_causal_effects(y0_pred, y1_pred):
    # ITE = y1 - y0
    # ATE = mean(ITE)
    # CATE = ATE by subgroups
    return results_dict
```

5. Counterfactual Generation

```python
def generate_counterfactual_scenarios(model, X, T, Y):
    # Predicts both potential outcomes
    # Creates counterfactual trajectories
    return counterfactual_df
```

Data Flow

```
Raw Data → Preprocessing → TARNet Training → Predictions
                                     ↓
                          Causal Effect Computation
                                     ↓
                    ITE, ATE, CATE, Counterfactuals
```

Conclusion

This project successfully demonstrates:

1. ☐ **Transition from SVAR to Causal ML:**
   - More flexible modeling approach
   - Better handling of nonlinear relationships
2. ☐ **Practical Policy Insights:**
   - Quantifies policy effects
   - Identifies heterogeneous impacts
   - Enables counterfactual analysis
3. ☐ **Reproducible Methodology:**

- Clear data pipeline
- Well-documented code
- Export-ready data formats

Future Directions

1. **Real Data Integration:**
   - Replace synthetic data with actual housing/macro data
   - Validate findings on real-world policy changes
2. **Extended Policies:**
   - Analyze other macroprudential tools (DSTI, capital requirements)
   - Compare policy effectiveness
3. **Robustness Checks:**
   - Sensitivity analysis
   - Placebo tests
   - Comparison with alternative methods
4. **Production Deployment:**
   - API for real-time counterfactual predictions
   - Dashboard for policymakers

Appendix: How to Use with Real Data

Step 1: Prepare Your Data

Your CSV files should have: - **8 feature columns** (as in `feature_names.csv`) - **Binary treatment column** (0 = control, 1 = treated) - **Outcome column** (housing price change)

Step 2: Load Data

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

# Load your data
df = pd.read_csv('your_housing_data.csv')

# Define feature columns (must match structure)
feature_cols = [
    'gdp_growth', 'credit_gap', 'inflation', 'interest_rate',
    'exchange_rate', 'household_debt_income', 'unemployment',
    'lag_housing_price'
]

# Extract features, treatment, outcome
X = df[feature_cols].values
T = df['treatment'].values
Y = df['outcome'].values
```

Step 3: Preprocess

```
# Normalize features (CRITICAL!)
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split data (maintain treatment stratification)
X_train, X_test, T_train, T_test, Y_train, Y_test = train_test_split(
    X_scaled, T, Y, test_size=0.2, random_state=42, stratify=T
)
```

Step 4: Train TARNet

```
# Build model
tarnet = build_tarnet(input_dim=X_train.shape[1])
tarnet.compile(optimizer='adam', loss='mse')

# Prepare training data (mask outcomes)
train_y = {
    'y0': Y_train * (T_train == 0),
    'y1': Y_train * (T_train == 1)
}

# Train
history = tarnet.fit(X_train, train_y, epochs=100, batch_size=32)
```

Step 5: Estimate Causal Effects

```
# Predict potential outcomes
y0_pred, y1_pred = predict_counterfactuals(tarnet, X_test)

# Compute effects
causal_results = compute_causal_effects(y0_pred, y1_pred)

print(f"ATE: {causal_results['ATE']:.4f}")
print(f"95% CI: {causal_results['ATE_95CI']}")
```

Step 6: Interpret Results

1. **ATE Interpretation:**
   - Sign: Direction of policy effect
   - Magnitude: Strength of effect
   - CI: Statistical significance
2. **CATE Analysis:**
   - Compare effects across subgroups
   - Identify conditions where policy is most/least effective
3. **Counterfactuals:**
   - Compare actual vs. predicted outcomes
   - Assess policy impact over time

Glossary

- **ATE (Average Treatment Effect):** Average causal effect across population
- **ITE (Individual Treatment Effect):** Causal effect for a specific unit
- **CATE (Conditional ATE):** Average effect conditional on characteristics
- **Counterfactual:** What would have happened under alternative treatment
- **Potential Outcomes:** $Y(0)$ and $Y(1)$ - outcomes under control and treatment
- **TARNet:** Treatment-Agnostic Representation Network
- **LTV:** Loan-to-Value ratio
- **Unconfoundedness:** Treatment assignment independent of potential outcomes given covariates
- **Overlap:** Every unit has positive probability of receiving either treatment

---

References

Key Papers

1. Shalit, U., Johansson, F. D., & Sontag, D. (2017). "Estimating individual treatment effect: generalization bounds and algorithms." ICML.
2. Johansson, F., Shalit, U., & Sontag, D. (2016). "Learning representations for counterfactual inference." ICML.
3. Pearl, J. (2009). "Causality: Models, Reasoning and Inference." Cambridge University Press.

Technical Documentation

- TensorFlow/Keras: https://www.tensorflow.org
- Scikit-learn: https://scikit-learn.org
- Causal Inference: https://www.stats.ox.ac.uk/~doucet/courses/

Contact and Support

For questions about this implementation: - Review the notebook: `[2]TARNET-example.ipynb` - Check exported data: `data_exports/` directory - Refer to README.md in data_exports for data structure details

---

End of Documentation

Generated for: TARNet Housing Price Policy Analysis Project