

# Enhanced Causal Discovery for Autocorrelated Time Series via Adaptive Momentary Conditional Independence

---

This repository is the official implementation of **Enhanced Causal Discovery for Autocorrelated Time Series via Adaptive Momentary Conditional Independence**.

## Requirements

---

To install requirements:

```
pip install -r requirements.txt
```

Or you can install all required dependencies for Python 3.8 using pip :

- tigramite=5.2.7.0
- igraph=0.11.8
- lingam=1.9.1
- dcor=0.6

## Importing Functions

---

To import the proposed ECD-aMCI algorithm and required simulation functions, run this command:

```
python ECD_aMCI.py
```

Interactive Environment (ensure all `.py` files are in current path)

```
from ECD_aMCI import *
```

# Evaluation

To evaluate the proposed algorithm on the simulated datasets, run:

```
# Simulation experiment on simulated data example
# Define simulation experiment settings
config = {
    # Data generation parameters
    'dim': 10,                    # Number of variables
    in the system
    'tau_true': 5,                # True maximum time lag
    in data generation
    'auto_coeff': 0.9,            # Autoregressive
    coefficient (must be between -1 and 1 for stability)
    'cross_coeff': 0.2,           # Cross-effect
    coefficient between variables
    'instant_ratio': 1,           # Base number of
    instantaneous connections
    'delay_cross_x': 1.5,         # Base number of
    delayed cross connections
    'func': np.copy,              # Function to transform
    causal relationships (np.copy for linear)
    'T': 500,                     # Time series length
    'B': 300,                     # Number of the
    simulated datasets
    #algorithm parameters
    'Method': 1,                  # Method selection (0:
    PCMCI+, 1: ECD-aMCI, 2: NTS-NOTEARS, 3: Bagged-PCMCI+)
    'pc_alpha': 0.01,             # Significance level for
    PC algorithm
    'tau_max': 5,                 # Maximum time lag to
    consider
    'tau_min': 0,                 # Minimum time lag to
    consider
    'cond_ind_test': ParCorr(),    # Conditional
    independence test (ParCorr for linear)
}

# Run parallel experiments using available CPU cores
```

```

num_cores = multiprocessing.cpu_count() - 2 # Leave 2
cores free
results_and_times_and_graphs = Parallel(n_jobs=num_cores)(
    delayed(run_single_experiment)(
        dim=config['dim'],
        auto_coeff=config['auto_coeff'],
        cross_coeff=config['cross_coeff'],
        seed=b, # Using bootstrap index as random seed
        instant_ratio=config['instant_ratio'],
        delay_cross_x=config['delay_cross_x'],
        func=config['func'],
        T=config['T'],
        tau_min=config['tau_min'],
        tau_true=config['tau_true'],
        tau_max=config['tau_max'],
        Method=config['Method'],
        cond_ind_test=config['cond_ind_test'],
        pc_alpha=config['pc_alpha'],
    )
    for b in range(config['B']) # Run B experiments
)
print_metrics(results_and_times_and_graphs, config)

```

## Configuration Parameters

### Data generation parameters

- `func`: Function to transform causal relationships (default: `np.copy` for linear, `np.tanh` for nonlinear)
- `T`: Time series length (default: 500)
- `tau_true`: True maximum time lag in data generation (default: 5)
- `std`: Standard deviation of noise (default: 1)
- `max_tries_stable`: Maximum attempts to find stable structure (default: 200)
- `stability_threshold`: Threshold for considering data stable (default: 1e5)

- `Method`: Causal discovery method (0: PCMCI+, 1: ECD-aMCI, 2: NTS-NOTEARS, 3: Bagged-PCMCI+)
- `cond_ind_test`: Conditional independence test (default: `ParCorr()` for linear)
- `seed`: random seed
- `B`: Number of bootstrap datasets (default: 300)

### **ECD-aMCI ,PCMCI+,Bagged-PCMCI+ Parameters**

- `pc_alpha`: Significance level for conditional independence test (default: 0.01)
- `tau_max`: Maximum time lag to consider (default: 5)
- `tau_min`: Minimum time lag to consider (default: 0)

### **NTS-NOTEARS Parameters (Method 2)**

- `w_thresholds`: Edge detection threshold (default: 0.1)
- `lambda1`: L1 regularization parameter for the kernel weight parameters (default: 0.001)
- `lambda2`: L2 regularization parameter for the parameters of the CNN (default: 0.05)

### **Additional Bagged-PCMCI+ Parameters**

`boot_samples`: Number of bootstrap samples for Bagged-PCMCI+ (default: 50)