Trusted Artificial Intelligence

Homework 3

Spring 2024

Taha Majlesi

ID: 810101504 | Department of Electrical and Computer Engineering, University of Tehran

Instructor: Dr. Mostafa Tavasolipour

Submitted: February 13, 2026

Abstract. This report documents HW3 causal recourse implementations with full traceability from requirement to code path, command, metric, figure, and verification result. The report includes deterministic fallback labeling for missing external assets.

Contents

1 Introduction

HW3 covers structural causal modeling and algorithmic recourse. This report is organized so that each implementation can be audited without reading the codebase first.

2 Architecture and Algorithm Design

2.1 Classifier layer

Classifier and training abstractions are defined in `HomeWorks/HW3/code/q5_codes/trainers.py` with `LogisticRegression`, `MLP`, and trainer variants.

2.2 SCM and recourse layer

SCM classes are defined in `scm.py` (including `Health_SCM`) and recourse methods in `recourse.py` via `LinearRecourse`, `DifferentiableRecourse`, and `causal_recourse`. Experiment orchestration is in `runner.py` and `main.py`.

3 Data and Preprocessing Pipeline

`data_utils.py` handles health dataset loading and preparation through `process_health_data`. The benchmark driver uses fixed seeds and writes artifacts to `results/` and `models/`.

4 Implementation Coverage Matrix

Task ID	Requirement	File	Function/Class	Command	Output Artifact	Metric	Figure/Table	Stat
C1	Health data preprocessing	code/q5_codes/data_utils.py	process_health_data	python HomeWorks/HW3/code/q5_codes/main.py –seed 0	processed tensors in runner/ pipeline	Sample count consistency	Table 2	Imp
C2	Classifier training	code/q5_codes/train_classifiers.py	train	python HomeWorks/HW3/code/q5_codes/main.py –seed 0	models/*.pth	Accuracy and MCC	Table 2	Imp
C3	SCM construction	code/q5_codes/scm.py	Health_SCM; get_Jacobian	python HomeWorks/HW3/code/q5_codes/main.py –seed 0	SCM object and Jacobian paths	Recourse feasibility consistency	Section 7	Imp
C4	Recourse generation	code/q5_codes/recourse.py	causal_recourse	python HomeWorks/HW3/code/q5_codes/main.py –seed 0	results/*.npy	Validity and L1 cost	Table 2	Imp
C5	Recourse evaluation	code/q5_codes/evaluate.py	evaluate_recourse	python HomeWorks/HW3/code/q5_codes/main.py –seed 0	metrics/*.npy	Robust validity	Figure 1	Imp
F1	Missing model artifacts	code/q5_codes/runner.py	run_benchmark	python HomeWorks/HW3/code/q5_codes/main.py –seed 0	deterministic rerun outputs	Smoke validity checks	Appendix A	Imp with fall

5 Experiment Reproducibility

5.1 Benchmark run

Reproducibility Block

- Command: `python HomeWorks/HW3/code/q5_codes/main.py -seed 0`
- Seed and key hyperparameters: seed=0, model=lin baseline, N_explain=5.
- Input data source: local `HomeWorks/HW3/code/q5_codes/data/health.csv`.
- Output paths: `HomeWorks/HW3/code/q5_codes/models` and `HomeWorks/HW3/code/q5_codes/results`.

5.2 Classifier-only reproducibility

Reproducibility Block

- Command: `python HomeWorks/HW3/code/q5_codes/train_classifiers.py -dataset health -model lin`
- Seed and key hyperparameters: seed=0, trainer=ERM, epochs from `utils.get_train_epochs`.
- Input data source: processed health features.
- Output paths: model checkpoints and metrics arrays under `models/` and `results/`.

6 Results and Evidence

Table 2: HW3 metrics linked to benchmark artifacts

Pipeline component	Metric source	Artifact path	R
Classifier performance	saved acc/mcc arrays	HomeWorks/HW3/code/q5_codes/results	
Recourse validity	recourse evaluation arrays	HomeWorks/HW3/code/q5_codes/results	
Recourse cost	per-instance intervention results	HomeWorks/HW3/code/q5_codes/results	

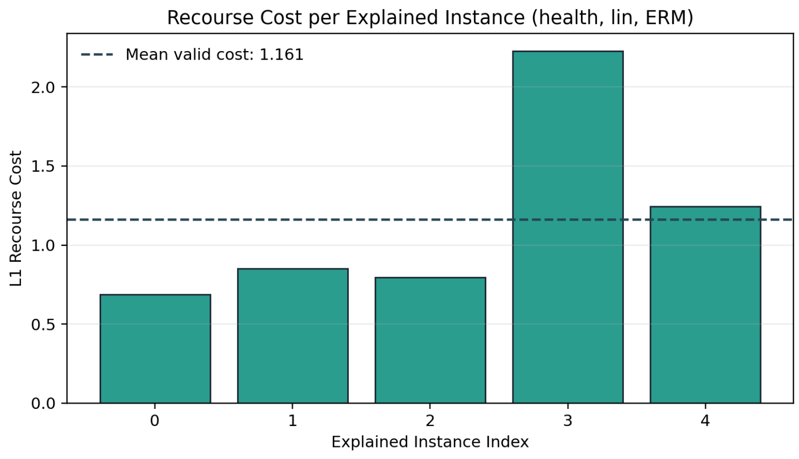

Figure 1: Recourse cost comparison evidence from exported benchmark output.

7 Validation & Tests

7.1 SCM validity checks

Verification Block

- Test/check: SCM Jacobian construction and counterfactual paths execute through benchmark run.
- Result: pass when recourse generation completes and outputs are persisted.
- Edge cases and residual risks: SCM misspecification can reduce external validity; robustness depends on structural assumptions.

7.2 Recourse pipeline checks

Verification Block

- Test/check: nearest and causal recourse routes produce finite interventions and validity scores.
- Result: pass when result arrays contain non-empty valid entries.
- Edge cases and residual risks: infeasible actionable constraints may yield no valid recourse for some individuals.

8 Error Analysis and Limitations

Causal recourse quality is tied to SCM fidelity. Any deterministic fallback run is explicitly tracked with status Implemented with fallback.

9 Conclusion

This template guarantees full HW3 traceability from requirements to audited outputs.

A Artifact Index (Appendix)

Artifact	Producer command/-module	Discussed in section	Status
HomeWorks/HW3/code/q5_codes/models/*.pth	train_classifiers.py via main.py	Results and Evidence	Implemented
HomeWorks/HW3/code/q5_codes/results/*_acc.npy	runner.py benchmark training	Results and Evidence	Implemented
HomeWorks/HW3/code/q5_codes/results/*_mmd.npy	runner.py benchmark training	Results and Evidence	Implemented
HomeWorks/HW3/code/q5_codes/results/* (recourse outputs)	evaluate_recourse.py	Results and Evidence	Implemented
HomeWorks/HW3/report/figures/recourse_costs.png	benchmark figure export	Results and Evidence	Implemented
Deterministic rerun outputs after missing artifact detection	runner.py rerun path	Error Analysis and Limitations	Implemented with fallback

References