Empirical Investigation of an Open Conjecture: AdaBoost Always Cycles? (Global Dynamics Conjecture)

Agentic NL→Lean 4 Pipeline
Job #29

April 26, 2026

Abstract

This report documents the empirical investigation of an open mathematical conjecture that could not be formally proved or disproved in Lean 4 with Mathlib. Numerical experiments were conducted to gather evidence for or against the conjecture. The empirical verdict is: **Inconclusive**. The conjecture remains formally open.

1 Conjecture Statement

Conjecture 1.

AdaBoost Always Cycles? (Global Dynamics Conjecture)

Let

=
{
(

,

)
}

=
1

S={(x
i

,y
i

)}
i=1
m

be a fixed binary-labeled dataset with

{-

1
,
+
1
}
y
i -

{1,+1}, and let

H be a weak-hypothesis class. Consider the discrete AdaBoost update of Freund & Schapire (1997), in the exhaustive weak-learner regime used in Rudin et al. (2012), with weight vectors Δ -

1
w
t Δ-

m1
(the probability simplex). At iteration

t, choose

arg

max

=
1

,

(

)
,
h
t

arg
hH
max

i=1

m

w
t,i

y
i

h(x
i

),

define weighted error

=

=
1

```
,
1
{
(
)
}
,
t
=
i=1
m
w
t,i
1{h
t
(x
i
)
=y
i
},
and update with
=
1
2
log
```

```
1-




,


+
1
,

=


,

exp

(-







(


)
)


,

t


=
2
1


log

t-


1
t
```

```
,w
t+1,i

=
Z
t

w
t,i

exp-(
t

y
i

h
t

(x
i

))

,

where

Z
t

 normalizes to

+
1
,

=
```

```
1

i



w
t+1,i



=1.

Equivalently, with finite hypothesis set

=
{

~
1
,…

,

~

}
H={
h
~
1…


,,
h
~
N



} and matrix



{-

1
,
+
1
}

×

M-{1,+1}
m×N
 defined by
```

```
=



~

(


)
M
ij


=y
i


h
~
j


(x
i


), step

t selects




arg

max



[

]
(




)

,
```

```
=

~


.
j
t


arg
j[N]
max


(w
t



M)
j


,h
t


=
h
~
j
t




.

As specified in Rudin et al. (2012), if this argmax is not unique, ties are
    broken in a fixed deterministic way (for concreteness: pick the smallest
    index

j). The generic no-tie condition means the argmax is unique at every iterate
    , i.e.

(




)
```

```
(

)

for all

 and all

,
(w
t

M)
j

=(w
t

M)
j

for all j

=j

 and all t,

equivalently,

w
t

 never lands on a tie boundary between weak-hypothesis regions of the
    simplex.

This induces a discrete dynamical system
```

```


:Δ -



1→Δ -
```

2 Status

Formal Status: OPEN — no Lean 4 proof or disproof was found.

Empirical Verdict: Inconclusive

The pipeline attempted formal verification in Lean 4 with Mathlib but was unable to produce a compiling proof or disproof. Empirical testing was then conducted to gather numerical evidence.

3 Basic Empirical Testing

The following output was produced by the basic numerical experiment:

```
=== EXPERIMENT PLAN ===

Conjecture (AdaBoost Always Cycles).  For the discrete AdaBoost update of
Freund-Schapire with exhaustive weak learner over a finite H, deterministic
tie-breaking (smallest index), and any finite labeled dataset S, the induced
map T : Delta^{m-1} -> Delta^{m-1} has only periodic orbits: every
    trajectory
{w_t} eventually enters a cycle.

Equivalent encoding: with M in {-1,+1}^{m x N}, M_{ij} = y_i h_j(x_i), pick
j_t = argmin{j : (w_t^T M)_j = max_j' (w_t^T M)_{j'}}; set eps_t = (1 - max)
    /2,
alpha_t = (1/2) log((1-eps_t)/eps_t); w_{t+1,i}   w_{t,i} exp(-alpha_t M_{i,
    j_t}).

Why this is hard: T is a piecewise-real-analytic nonlinear map with N
    polytopal
regions, and only a handful of very small instances (3x3, 4x5 in
Rudin-Schapire-Daubechies) are rigorously known to cycle.  A disproof would
require finding an M where T has an aperiodic orbit.

Experimental probes:
  (1) RANDOM ENSEMBLE: thousands of random ±1 matrices M at many (m,N) sizes
      ,
      run to T_max steps, detect cycling by matching last weight to history
      with tolerance tol=1e-8, extract min-period by re-scanning.
  (2) KNOWN EXAMPLES: Rudin-style 3x3 and 4x5-type cases; confirm cycling.
  (3) STRUCTURED/EDGE CASES: sparse M, redundant duplicated columns,
      rank-deficient M, and near-degenerate near-tie matrices.
  (4) SCALING in (m,N): does cycle length stay bounded or explode?
```

```
  (5) NON-UNIFORM INITIAL WEIGHTS: random Dirichlet inits probe GLOBAL
      dynamics
       (the conjecture asserts cycling from every w_0, not just uniform).
  (6) LONG-RUN STRESS TEST: for instances that did NOT cycle within 1000
      steps,
       re-run with T_max=20000 (counterexample search).
  (7) STATISTICAL TEST: binomial lower bound on cycling probability.

A counterexample would be an M where even at T_max=2x10^4 no weight vector
comes within 1e-8 of a previous one - suggesting aperiodic/quasi-periodic
dynamics.  We explicitly hunt for such cases.


--- Experiment 1: Random ±1 ensemble ---
  trials=3000   wall=8.8s
  cycled detected: 589   terminated early: 1251   not-yet-cycling: 1214
  cycling fraction (excluding early-terminators): 32.668%
  period  : min=1  med=3  mean=5.5  max=92
  transient: min=7  med=38  mean=99.5  max=490
  top-10 period lengths: [(3, 310), (10, 94), (1, 55), (7, 34), (5, 20), (6,
       18), (9, 13), (4, 12), (8, 9), (14, 3)]

--- Experiment 2: Small classical/structured examples ---
  RSD-style 8x8: stop=('zero_edge', 0)  transient=-1  period=-1  j-sequence
      head=[]

--- Experiment 3: Edge-case matrices ---
  edge trials=540   cycled=90   early-term=296   no-cycle=154
  edge-case period distribution: [(1, 46), (3, 36), (10, 5), (6, 2), (4, 1)]

--- Experiment 4: Scaling in (m,N) ---
  (m,N)=( 3, 3): cycled=  0/ 28  cycling%=  0.00%  med(period)=  -1  med(
      transient)=  -1  max(period)=-1
  (m,N)=( 5, 5): cycled= 19/ 89  cycling%= 21.35%  med(period)=   3  med(
      transient)=  36  max(period)=3
  (m,N)=( 7, 7): cycled= 43/118  cycling%= 36.44%  med(period)=   3  med(
      transient)=  36  max(period)=14
  (m,N)=( 9, 9): cycled= 46/132  cycling%= 34.85%  med(period)=   1  med(
      transient)=  48  max(period)=226
  (m,N)=(12,12): cycled= 64/147  cycling%= 43.54%  med(period)=   1  med(
      transient)= 136  max(period)=34
  (m,N)=(15,15): cycled= 60/149  cycling%= 40.27%  med(period)=   1  med(
      transient)= 211  max(period)=126
  (m,N)=(20,20): cycled= 62/150  cycling%= 41.33%  med(period)=   1  med(
      transient)= 302  max(period)=18
  (m,N)=(25,25): cycled= 33/150  cycling%= 22.00%  med(period)=   1  med(
      transient)= 378  max(period)=255

--- Experiment 5: Random Dirichlet initial weights (global dynamics) ---
  Dirichlet-init trials=1500   cycled=342   early-term=573   no-cycle=585
  random-init period median=3  max=14

--- Experiment 6: Stress test (T_max=20000) on hardest cases ---
  candidate matrices from Experiment 1 that had no cycle in 500 steps: 1214
```

```
  stubborn re-tested: 20  -> newly cycled: 20, still-no-cycle after 20000
     steps: 0
    m= 3 N= 3: transient=19996
... [truncated]
```

4 Experiment Code (Basic)

```
import numpy as np
import matplotlib
matplotlib.use("Agg")
import matplotlib.pyplot as plt
import time
from collections import Counter

print("===␣EXPERIMENT␣PLAN␣===")
print("""
Conjecture (AdaBoost Always Cycles).  For the discrete AdaBoost update of
Freund-Schapire with exhaustive weak learner over a finite H, deterministic
tie-breaking (smallest index), and any finite labeled dataset S, the induced
map T : Delta^{m-1} -> Delta^{m-1} has only periodic orbits: every
    trajectory
{w_t} eventually enters a cycle.

Equivalent encoding: with M in {-1,+1}^{m x N}, M_{ij} = y_i h_j(x_i), pick
j_t = argmin{j : (w_t^T M)_j = max_j' (w_t^T M)_{j'}}; set eps_t = (1 - max)
    /2,
alpha_t = (1/2) log((1-eps_t)/eps_t); w_{t+1,i}  w_{t,i} exp(-alpha_t M_{i,
    j_t}).

Why this is hard: T is a piecewise-real-analytic nonlinear map with N
    polytopal
regions, and only a handful of very small instances (3x3, 4x5 in
Rudin-Schapire-Daubechies) are rigorously known to cycle.  A disproof would
require finding an M where T has an aperiodic orbit.

Experimental probes:
  (1) RANDOM ENSEMBLE: thousands of random ±1 matrices M at many (m,N) sizes
     ,
      run to T_max steps, detect cycling by matching last weight to history
      with tolerance tol=1e-8, extract min-period by re-scanning.
  (2) KNOWN EXAMPLES: Rudin-style 3x3 and 4x5-type cases; confirm cycling.
  (3) STRUCTURED/EDGE CASES: sparse M, redundant duplicated columns,
      rank-deficient M, and near-degenerate near-tie matrices.
  (4) SCALING in (m,N): does cycle length stay bounded or explode?
  (5) NON-UNIFORM INITIAL WEIGHTS: random Dirichlet inits probe GLOBAL
     dynamics
      (the conjecture asserts cycling from every w_0, not just uniform).
  (6) LONG-RUN STRESS TEST: for instances that did NOT cycle within 1000
     steps,
      re-run with T_max=20000 (counterexample search).
  (7) STATISTICAL TEST: binomial lower bound on cycling probability.
```

```
A counterexample would be an M where even at T_max=2x10^4 no weight vector
comes within 1e-8 of a previous one - suggesting aperiodic/quasi-periodic
dynamics.  We explicitly hunt for such cases.
""")

rng = np.random.default_rng(20260421)
TOL = 1e-8

# ---------- core AdaBoost + cycle detection ----------
def adaboost(M, T_max=600, w0=None):
    m, N = M.shape
    w = np.ones(m)/m if w0 is None else (np.asarray(w0, float)/np.sum(w0))
    W = np.empty((T_max+1, m))
    J = np.empty(T_max, dtype=np.int32)
    W[0] = w
    for t in range(T_max):
        edges = w @ M                          # shape (N,)
        j = int(np.argmax(edges))              # ties -> smallest idx (numpy
            default)
        J[t] = j
        e = edges[j]
        if e >= 1 - 1e-14:
            return W[:t+1], J[:t], ("perfect", t)
        if e <= -1 + 1e-14:
            return W[:t+1], J[:t], ("degenerate", t)
        if abs(e) < 1e-15:                     # alpha = 0 fixed point
            return W[:t+1], J[:t], ("zero_edge", t)
        eps = (1 - e) / 2.0
        alpha = 0.5 * np.log((1-eps)/eps)
        w = w * np.exp(-alpha * M[:, j])
        s = w.sum()
        if not np.isfinite(s) or s <= 0:
            return W[:t+1], J[:t], ("num", t)
        w = w / s
        W[t+1] = w
    return W, J, ("maxiter", T_max)

def detect_cycle(W, tol=TOL):
    """Return (transient, period) or (-1,-1). Uses w[-1]-to-all comparison,
       then
    re-scans the cycle window to extract min period."""
    n = len(W)
    if n < 3: return -1, -1
    last = W[-1]
    diffs = W[:-1] - last
    d2 = np.einsum('ij,ij->i', diffs, diffs)
    idx = np.where(d2 < tol*tol)[0]
    if len(idx) == 0:
        return -1, -1
    i = int(idx[0])
    ref = W[i]
    tail = W[i+1:]
    d2b = np.einsum('ij,ij->i', tail - ref, tail - ref)
```

```
    pidx = np.where(d2b < tol*tol)[0]
    if len(pidx) == 0:
        return i, n - 1 - i
    return i, int(pidx[0]) + 1

def rand_pm1(m, N, rng):
    return rng.choice([-1.0, 1.0], size=(m, N))

# ---------- Experiment 1: Random ensemble ----------
print("\n--- Experiment 1: Random ±1 ensemble ---")
sizes = [(3,3),(3,4),(4,4),(4,5),(5,5),(5,8),(6,6),(6,10),(8,10),(10,15)]
per_size = 300
T_max = 500
results = []
t0 = time.time()
for (m, N) in sizes:
    for _ in range(per_size):
        M = rand_pm1(m, N, rng)
        W, J, stop = adaboost(M, T_max=T_max)
        tr, per = detect_cycle(W)
        results.append(dict(m=m,N=N,reason=stop[0],t_stop=stop[1],
                            transient=tr, period=per, M=M if per<0 and stop
                               [0]=='maxiter' else None))
print(f"  trials={len(results)}   wall={time.time()-t0:.1f}s")
cycled     = [r for r in results if r['period'] > 0]
term_early = [r for r in results if r['reason'] != 'maxiter']
no_cycle   = [r for r in results if r['period'] <= 0 and r['reason'] == '
   maxiter']
print(f"  cycled detected: {len(cycled)}   terminated early: {len(term_early
   )}   not-yet-cycling: {len(no_cycle)}")
denom = len(cycled) + len(no_cycle)
if denom:
    print(f"  cycling fraction (excluding early-terminators): {len(cycled)/
       denom*100:.3f}%")
if cycled:
    P = np.array([
# ... [truncated]
```

5 Conclusion

The conjecture remains formally open. Numerical experiments were **inconclusive** — neither strong support nor clear counterexamples were found. Further investigation (both formal and empirical) is warranted.