Wright-Fisher Process

(as applied to costly signaling)

Today:

1) new model of evolution/learning (Wright-Fisher)
2) evolution/learning ➔costly signaling

(We will come back to evidence for costly signaling next class)

(First, let's remind ourselves of the game)

- Male either
  - farmer (probability p)
  - or teacher (probability 1-p)
- Male chooses length of nail
- Female observes nail, not occupation
- Female chooses whether to “accept” or “reject” male

  (perhaps based, at least party, on how beautiful she finds his nails.)

IF…

1) Longer nails cumbersome for all males, more cumbersome for farmers
(-1/cm, -2/cm)
2) Females benefit from accepting teachers, but not farmers
(+10, -10)
3) All males benefit from being accepted
(+5,+5)

THEN

Exists a Nash equilibrium s.t.:
-farmers don't grow nails
-teachers grow nails to length l
(where l is some number between 2.5 and 5 cm)
-females accept those with nails at least length l

Now let’s discuss…

Learning/Evolution

First:

Why do we need learning/evolution?

We have argued costly signaling is Nash, but…

Why is Nash relevent?

The Khasi villagers NOT “choosing” what to find beautiful!

Why would their notion of beauty coincide with Nash?

(Similar issue for evolutionary applications like peacock tails!)

We have seen that evolution/learning lead to Nash, but…

1) may not converge
2) there are multiple Nash. E.g…

”pooling”: good and bad senders send cheapest signal, and receivers ignore signal

(no incentive to start attending to signal since noone sends, no incentive to start sending expensive signal bc ignored.)

Maybe THIS is what evolves?

There are some UBER-rational arguments against this equilibrium:

e.g. receiver “infers” that if anyone were to send a costly signal it MUST be the high type

“universal divinity”
(i.e. UBER-rational)

What about when agents aren’t divine?

Turns out evolution/learning gets you to costly separating!

Not just separating, but “efficient separating”
(i.e. l=2.5)

(which is what god would have wanted.)

(And empiricists too!)

Not trivial to show, replicator doesn’t do the trick!

➔wright-fisher

(wright-fisher will be REALLY useful! Also easy to code. And some added insights!)

Let’s start with the intuition

(then will become clear why replicator doesn’t suffice)

Suppose we start in a world where no one has long nails, and no one finds them beautiful

Suppose there is some experimentation (or mutation):

Some farmers grow long nails

➔They QUICKLY change back (or die off)

Some teachers grow long nails

➔They TOO change back (b/c costly), but SLOWLY (b/c less costly)

Some females start to find long nails beautiful and "match" with men who are beautiful

➔They find themselves more likely to mate with teachers and MAINTAIN this sense of beauty (or are imitated or have more offspring)

Over time …

-teachers with long nails start to perform well because enough females like them, counterbalancing the nail cost

-farmers with long nails NEVER do well

Eventually…

-All teachers have long fingernails

-All females like males with long fingernails

-No farmers have long fingernails

And once there, REALLY hard to leave!

Problem with replicator:

CAN leave separating
(just takes complicated “path”)

CAN leave pooling too
(just takes simpler path)

(likewise for ostentatious separating)

Replicator can just tell us if NO paths leave.

Can’t tell us if “more” paths leave.

Doesn’t distinguish between “more stable” and “less stable”

THIS is why noone had solved this model before

(Grafen 1990 is seminal paper; claimed to solve, but really just showed was Nash!)

Needs “stochastic” model!

➔Wright-Fisher!

An ad from our sponsor:

Program For evolutionary Dynamics

Martin Nowak

Drew Fudenberg

Let’s learn Wright-Fisher

And in so doing, let’s see that leads to costly signaling

Simulations require numbers

(although important to show robust! We will!)

And easier with small number of strategies

(take fewest needed to get insight, show robust later)

So, let’s assume…

-1/3 good, 2/3rd bad

-available signals: 0,1,2,3
Costs: 0,1,2,3 vs 0,3,6,9

-for each possible signal, 0,1,2,3, receivers either accept or reject that signal
Senders get 5 if accepted
receivers get 5 if accept good and -5 if accept bad

The Nash equilibrium are:

1) ”pooling”: good and bad senders send 1, and receivers never accept any signal
2) “efficient separating”: good sends signal 3, bad sends 1, and receiver accepts 3 (and 4?)
3) “ostentatious separating”: good sends signal 4, bad sends 1, and receiver accepts only signal 4

(prove this?)

Why four signals?

1) Pooling
2) Efficient separating
3) Ostentatious separating
4) Non-equilibrium separating
    (bad sends 0, good sends 1)

Will simulate…

(Proof?

I don’t know how!

But simulations VERY compelling! And robust! And simple to code! And give additional insight, e.g. into “why”)

Basics of Wright-Fisher:

Start each of N players with randomly chosen strategy

In each generation:

-Payoffs determined

(e.g. all senders play against all receivers, so depends on frequency of each strategy)

-Fitness determined

(e.g. f=1-w+w*payoffs, or f=e^w*payoffs where w measures “selection strength”; In replicator this doesn’t matter)

-Each individual has offspring proportional to fitness; N offspring born in total

-Offspring take random strategy with probability mu

(“mutation” or “experimentation)

-Otherwise, offspring take strategy of “mom”

(this can be “imitation”; ignores sexual reproduction)

-Mom’s generation dies

Repeat for M generations

Display time trend

Perhaps repeat many such simulations, and display averages across all simulations

(notice as population gets large, this approaches replicator dynamic with mutations)

Let’s apply this to our costly signaling model…

```
Start each of N players with randomly  chosen strategy

In each generation:
        -Payoffs determined
        (e.g. all senders play against all receivers, so depends on frequency of each strategy)
        -Fitness determined
        (e.g. f=1-w+w*payoffs, or f=e^w*payoffs where w measures “selection strength”;
        In replicator this doesn’t matter)
        -Each individual has offspring proportional to fitness; N offspring born in total
                -Offspring take random strategy with probability mu
                (“mutation” or “experimentation)
                -Otherwise, offspring take strategy of “mom”
                (this can be “imitation”; ignores sexual reproduction)
        -Mom’s generation dies

Repeat for M generations

Display time trend

Perhaps repeat many such simulations, and display averages across all simulations
```

Start with 50 low quality senders, 25 high quality senders, 75 receivers with randomly chosen strategy. E.g.:

-Low quality senders: 40 send 0 and 10 send 2

-High quality senders: 20 send 0 and 5 send 2

-Receivers: 70 accept ≥0, 5 only accept ≥2

Start each of N players with randomly chosen strategy

In each generation:

- -Payoffs determined
  (e.g. all senders play against all receivers, so depends on frequency of each strategy)
- -Fitness determined
  (e.g. f=1-w+w*payoffs, or f=e^w*payoffs where w measures “selection strength”;
  In replicator this doesn’t matter)
- -Each individual has offspring proportional to fitness; N offspring born in total
  - -Offspring take random strategy with probability mu
    (“mutation” or “experimentation)
  - -Otherwise, offspring take strategy of “mom”
    (this can be “imitation”; ignores sexual reproduction)
- -Mom’s generation dies

Repeat for M generations

Display time trend

Perhaps repeat many such simulations, and display averages across all simulations

Payoffs for low quality senders (present):

- If send 0, 70/75 chance accepted and pay no cost

payoff= .93*5 – 0 = 4.67

- If send 2, 75/75 chance accepted and pay 6 cost

payoff= 1*5 – 6 = -1

Payoffs for high quality senders (present):

- If send 0, 70/75 chance accepted and pay no cost

payoff= .93*5 – 0 = 4.67

- If send 2, 75/75 chance and pay 2 cost

payoff= 1*5 – 2 = 3

Payoffs for receivers (present):

- If accept ≥0, 50/75 chance accept bad type

$$\text{payoff} = 2/3 * -5 + 1/3 * 5 = -1.67$$

- If only accept ≥2, 5/75 chance match with good type, 10/75 chance match with bad type, and 60/75 don’t match

$$\text{payoff} = .07 * 5 + .13 * (-5) + .8 * 0 = -0.3$$

```
Start each of N players with randomly  chosen strategy

In each generation:
        -Payoffs determined
        (e.g. all senders play against all receivers, so depends on frequency of each strategy)
        -Fitness determined
        (e.g. f=1-w+w*payoffs, or f=e^w*payoffs where w measures “selection strength”;
        In replicator this doesn’t matter)
        -Each individual has offspring proportional to fitness; N offspring born in total
                -Offspring take random strategy with probability mu
                (“mutation” or “experimentation)
                -Otherwise, offspring take strategy of “mom”
                (this can be “imitation”; ignores sexual reproduction)
        -Mom’s generation dies

Repeat for M generations

Display time trend

Perhaps repeat many such simulations, and display averages across all simulations
```

For each, we let f=e^(.1*payoff)

E.g. for low quality senders who send 0

payoff = 4.67

f = e^(.1*payoff)=1.60

```
Start each of N players with randomly  chosen strategy

In each generation:
        -Payoffs determined
        (e.g. all senders play against all receivers, so depends on frequency of each strategy)
        -Fitness determined
        (e.g. f=1-w+w*payoffs, or f=e^w*payoffs where w measures “selection strength”;
        In replicator this doesn’t matter)
        -Each individual has offspring proportional to fitness; N offspring born in total
                -Offspring take random strategy with probability mu
                (“mutation” or “experimentation)
                -Otherwise, offspring take strategy of “mom”
                (this can be “imitation”; ignores sexual reproduction)
        -Mom’s generation dies

Repeat for M generations

Display time trend

Perhaps repeat many such simulations, and display averages across all simulations
```

How do we allocate offspring:

Fitness for low quality senders:

- If send 0➔ payoff=4.67, f=1.60
- If send 2➔ payoff=-1, f=.90

➔ For any given offspring, chance offspring 10*.9/(40*1.6 + 10*.9) has a signal 2 mother. O.w. must have signal 0 mother.

➔ For any given offspring, chance that she is signal 2 is chance that mother is signal 2 and not a *(1m-u)+mu/2

➔ Probability of having exactly X offspring who send signal 2 and 0 who send signal 0, is the binomial with probability of "success" of p=10*.9/(40*1.6 + 10*.9)*(1-mu)+mu/2 and 50" trials."

(50 choose X) *[p^X+ (1-p)^(50-X)

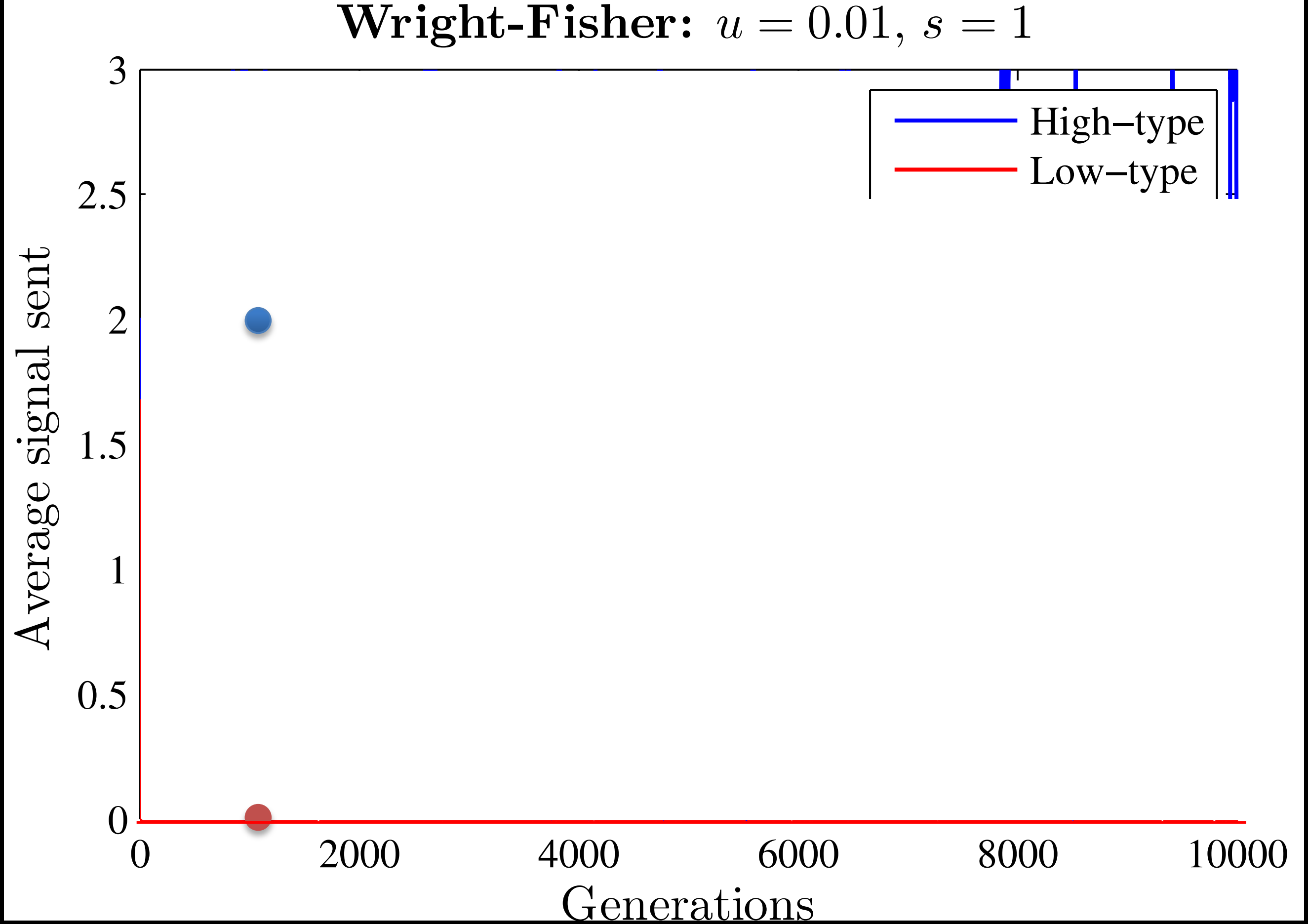
(With more than 2 strategies, we must use the multinomial distribution)

```
Start each of N players with randomly  chosen strategy

In each generation:
        -Payoffs determined
        (e.g. all senders play against all receivers, so depends on frequency of each strategy)
        -Fitness determined
        (e.g. f=1-w+w*payoffs, or f=e^w*payoffs where w measures “selection strength”;
        In replicator this doesn’t matter)
        -Each individual has offspring proportional to fitness; N offspring born in total
                -Offspring take random strategy with probability mu
                (“mutation” or “experimentation)
                -Otherwise, offspring take strategy of “mom”
                (this can be “imitation”; ignores sexual reproduction)
        -Mom’s generation dies

Repeat for M generations

Display time trend

Perhaps repeat many such simulations, and display averages across all simulations
```

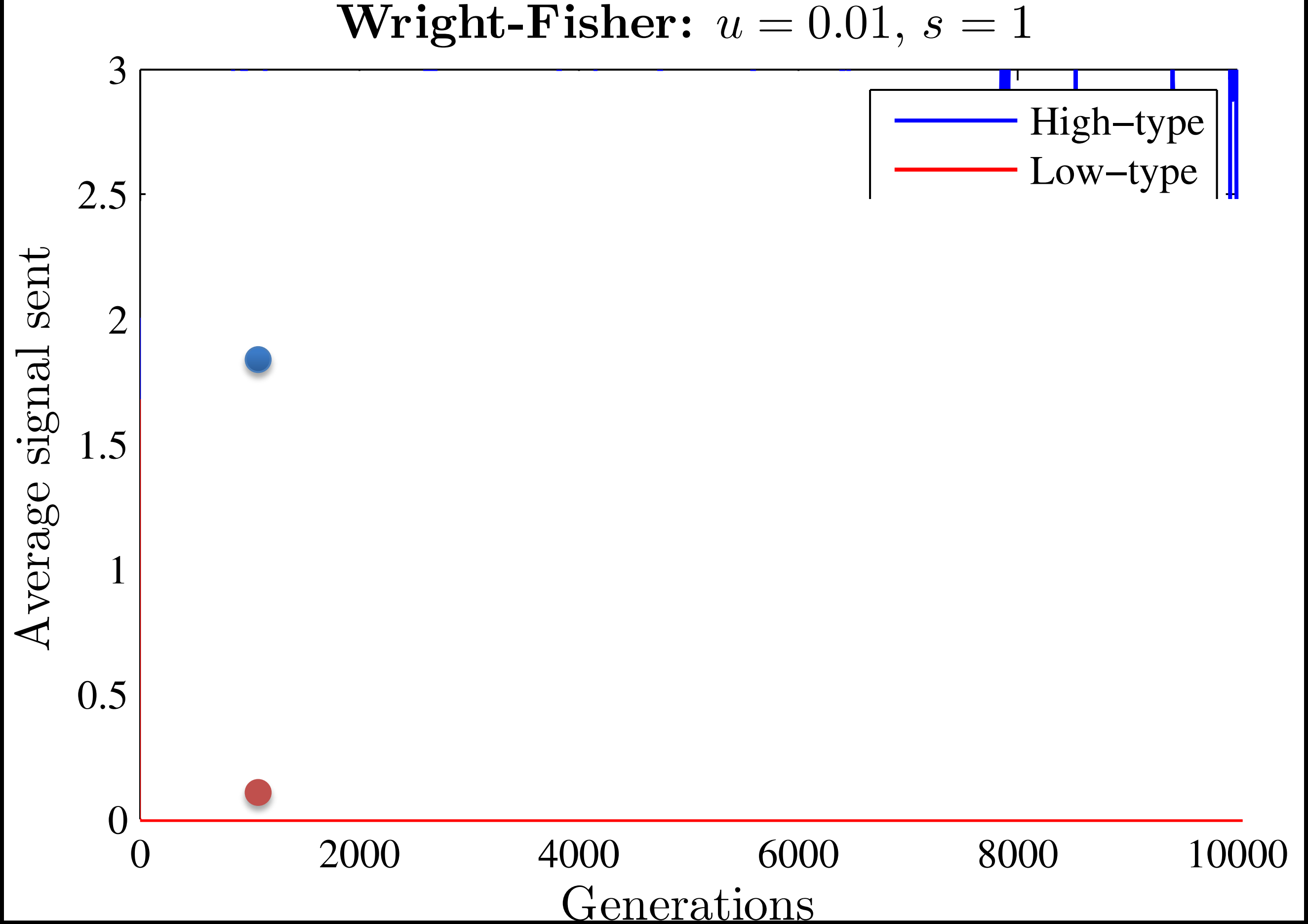
(Need to figure out good way to represent info visually!)

Average the signal values for each sender type and report for each generation in a graph

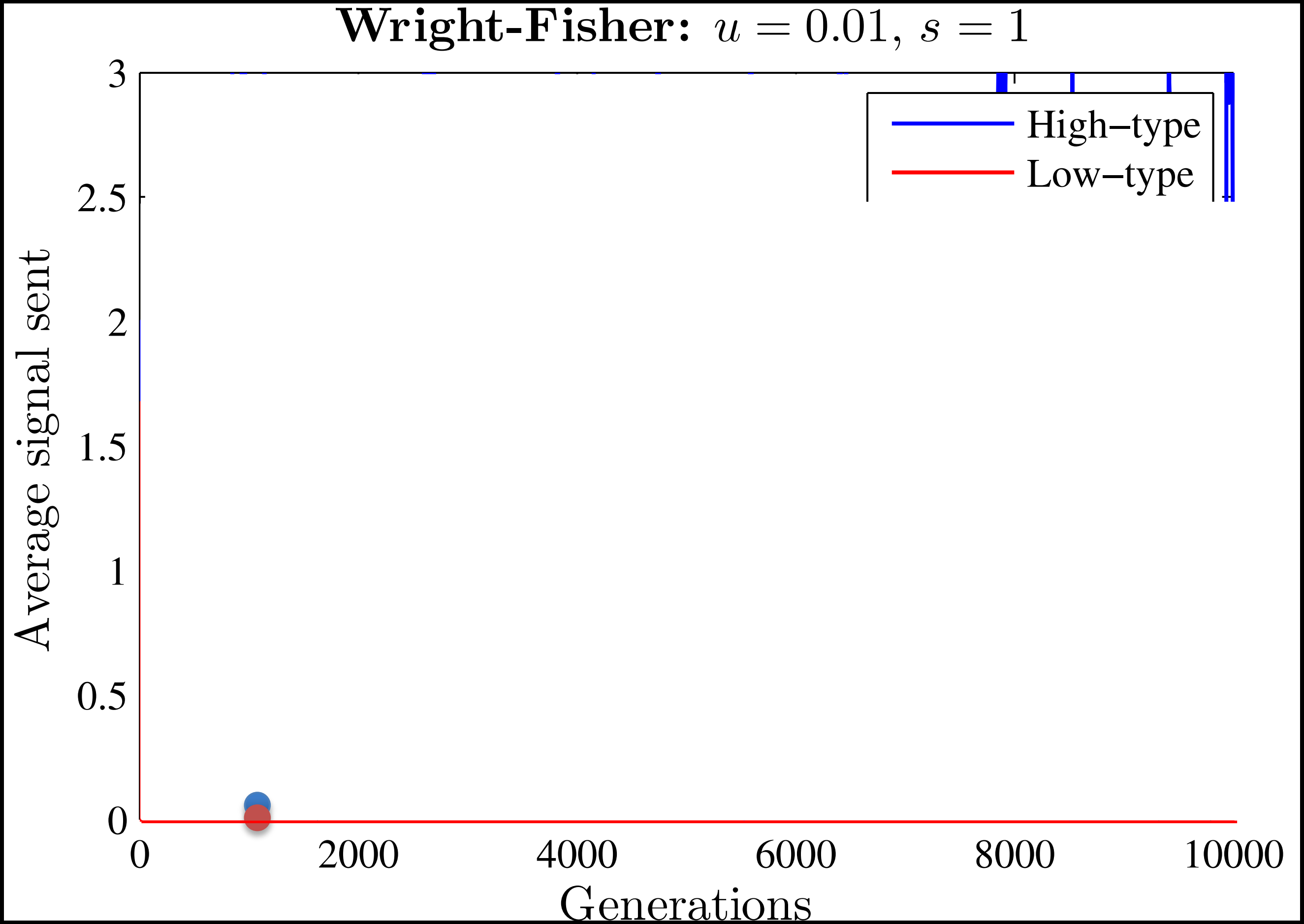
Efficient Separating equilibrium looks like this:



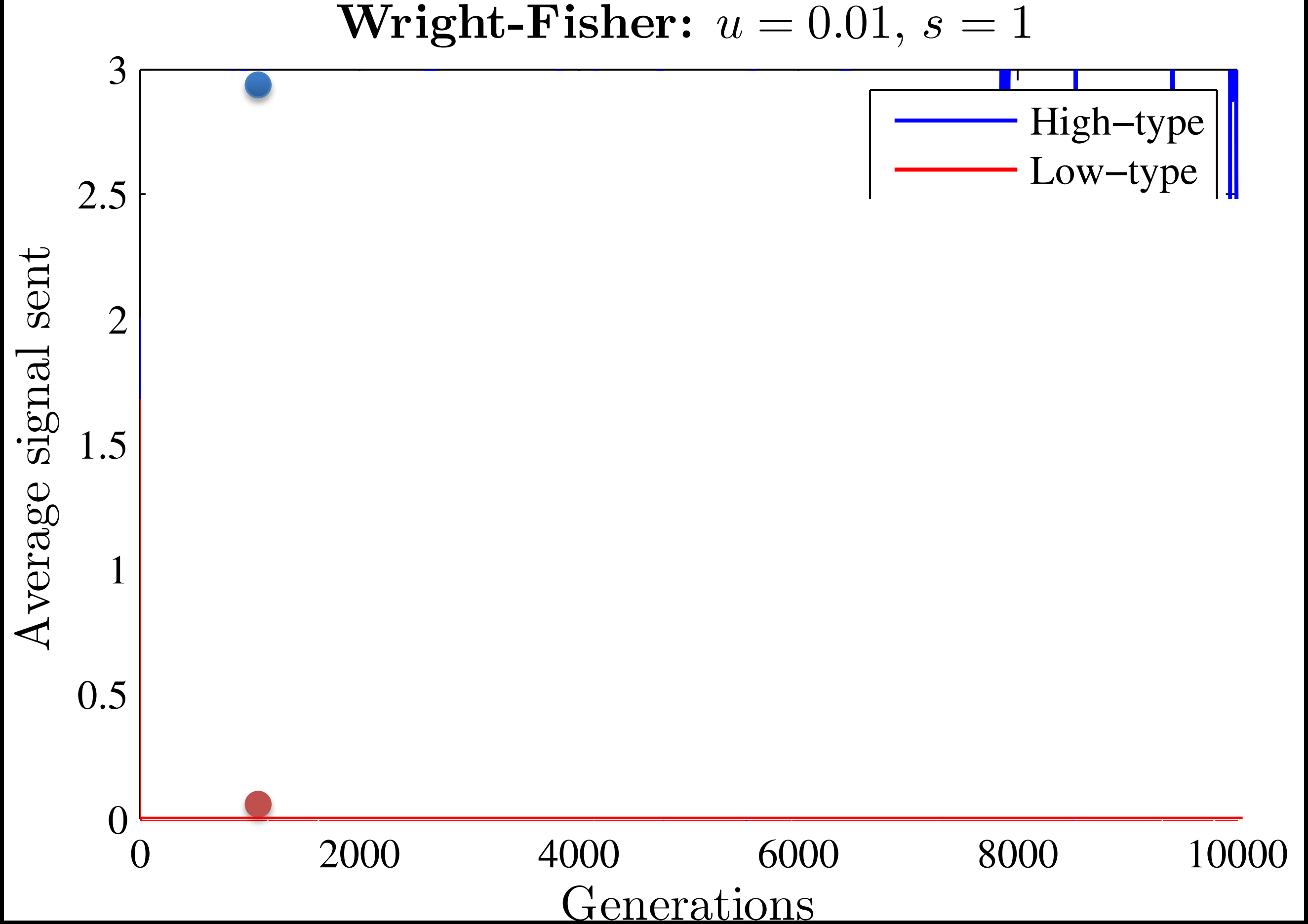
or this (b/c mutants):



Pooling equilibrium looks like this:

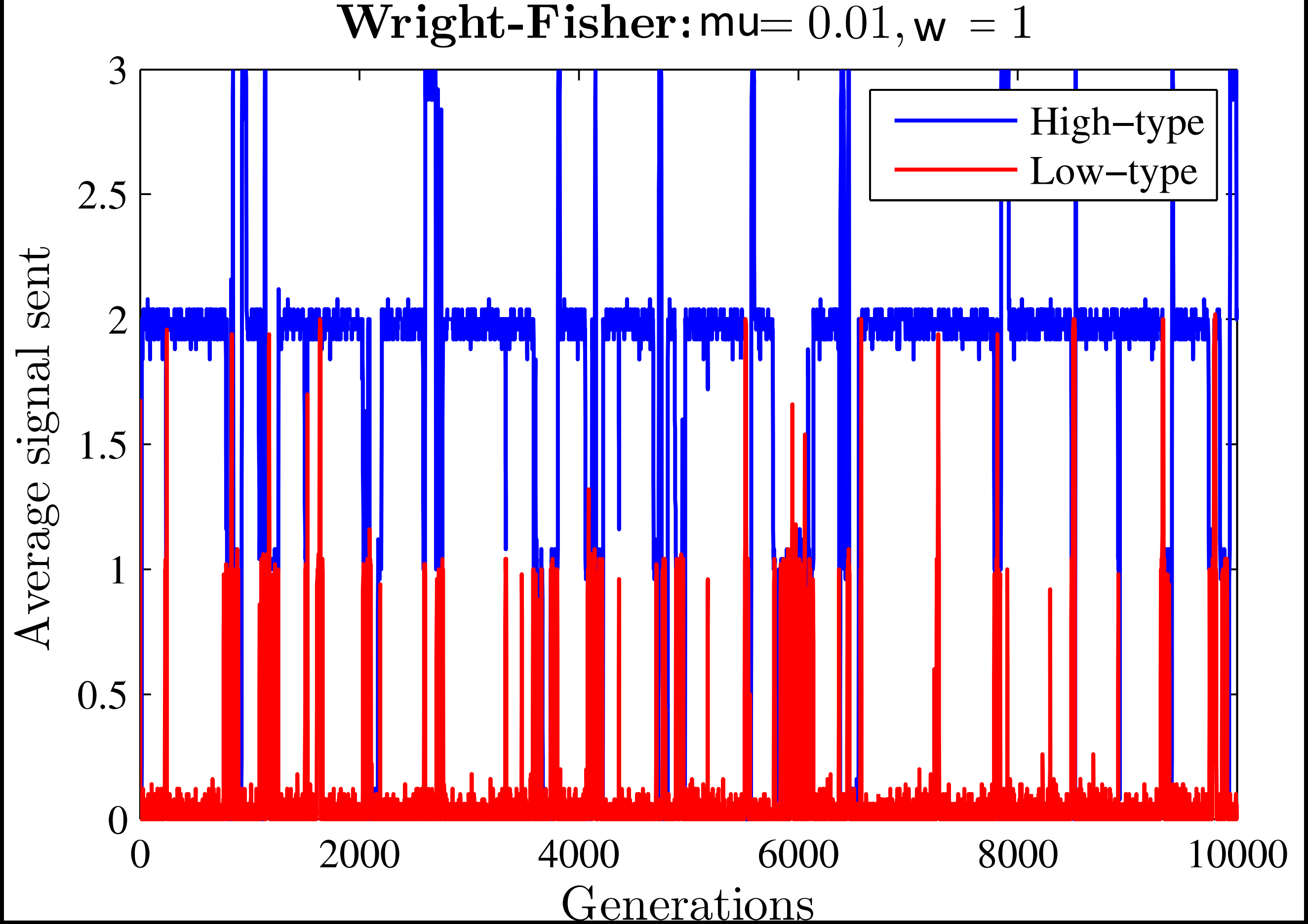


Ostentatious separating equilibrium looks like this:

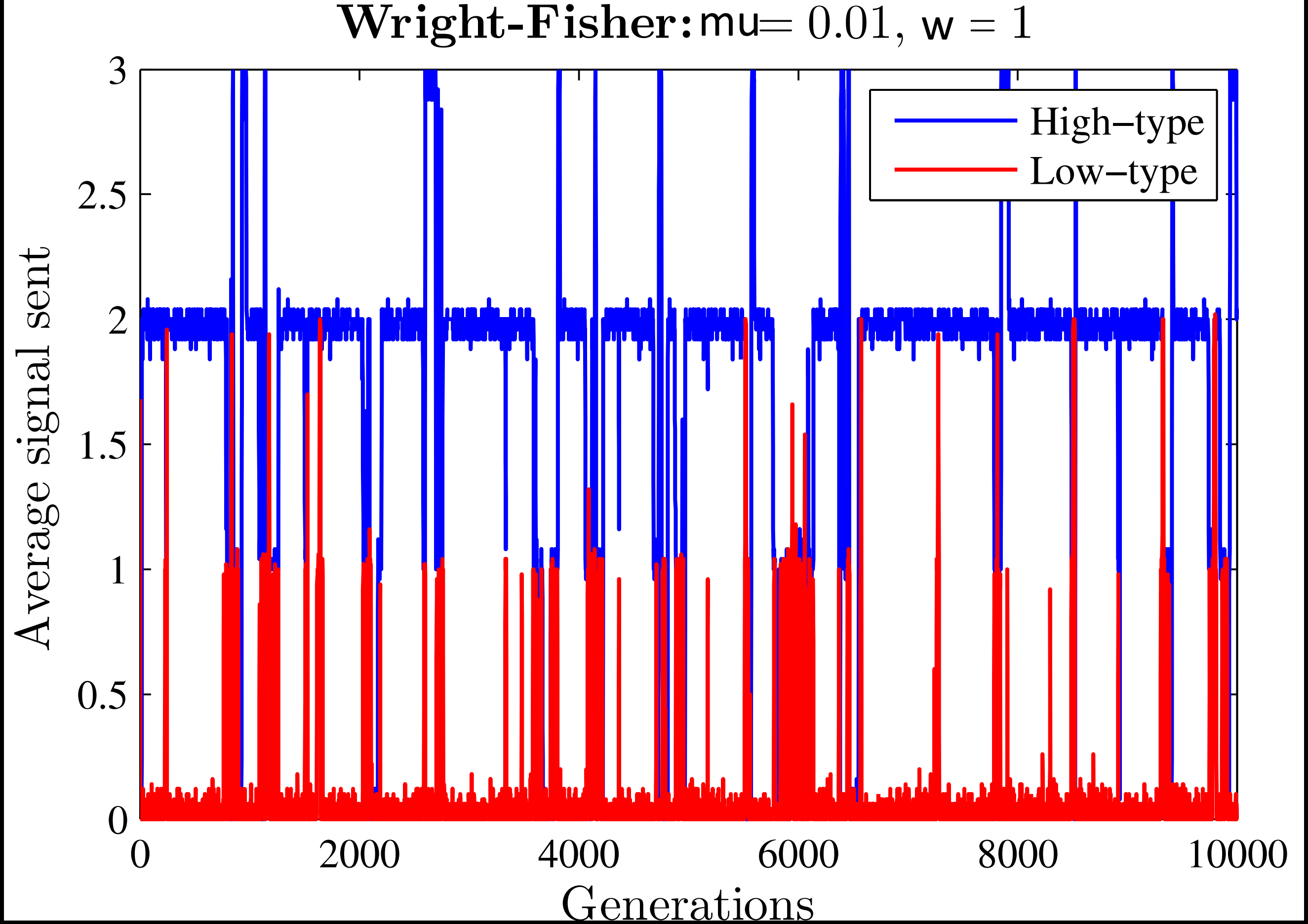


Simulation Results?

Here is an example time trend



Notice almost always at efficient separating (although does leave sometimes)



Freak occurrence?

Or almost always at separating?

For any given generation:

We can “categorize” the population according to:

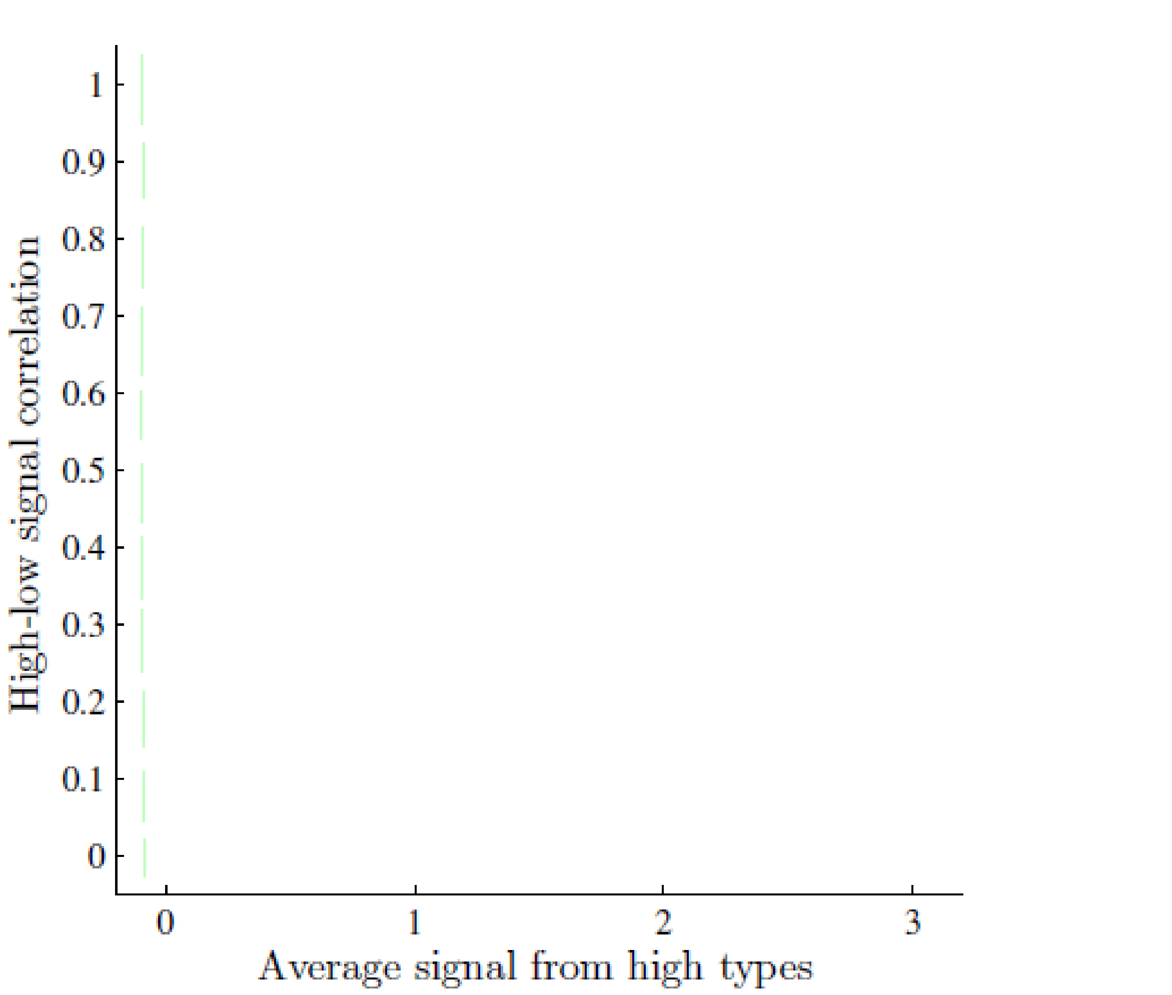
1) The average signal of high (averaged over all 25 high players, in that generation)

   E.g., If 24 high types send signal 2 and 1 sends signal 3, then the average signal is 2.04

2) “Correlation” between high and low signals

   E.g., (1/25,0,24/25,0)* (50/50,0,0,0)=4%

Results – Evolution/Imitation



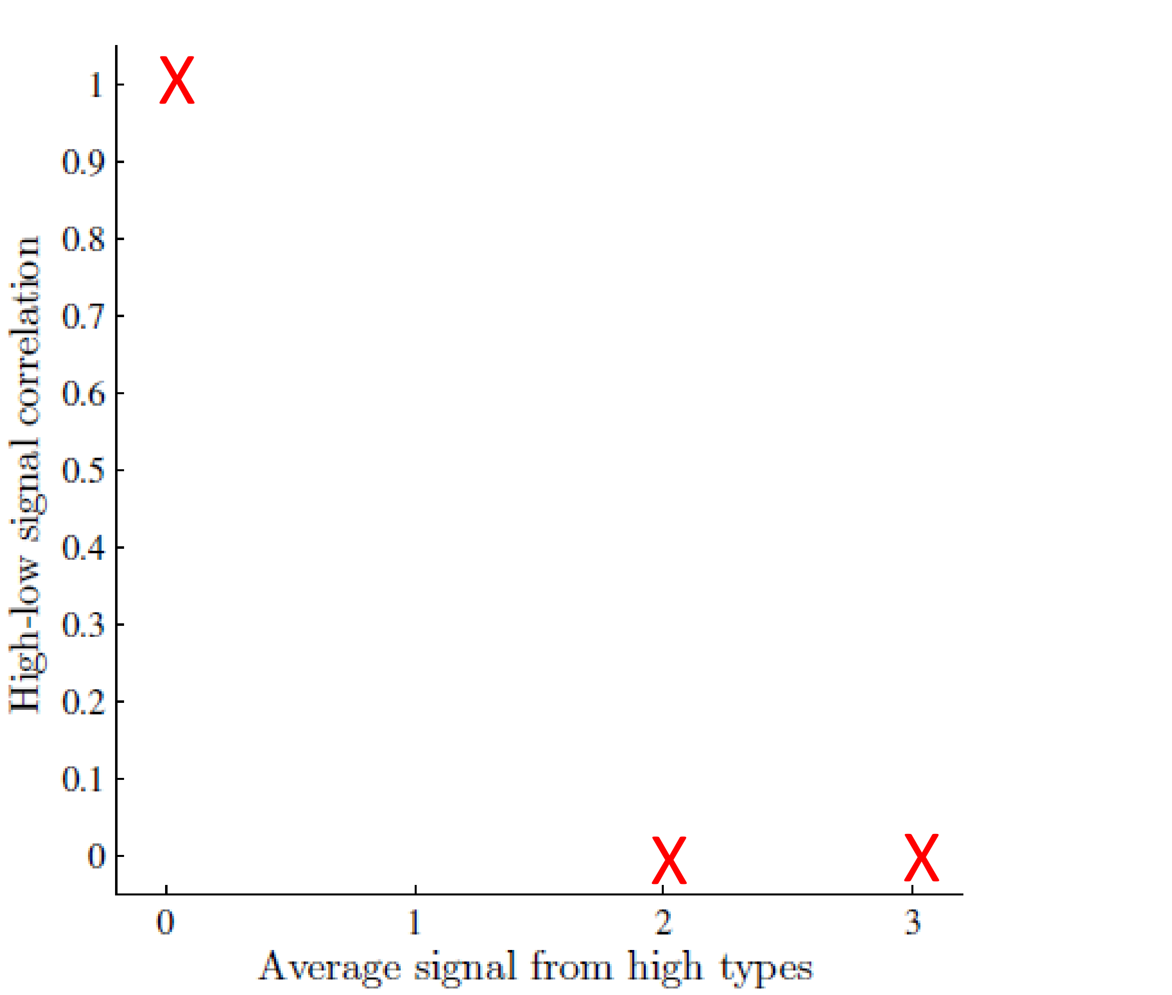
Notice that the 3 equilibrium can be plotted on this graph as follows:

1) Pooling:
high sends signal 0, low sends same signal
➔ (0,1)

2) Efficient Separating:
high sends signal 2, low sends signal 0
➔(2,0)

3) Ostentatious Separating:
high sends signal 3, low sends signal 0
➔(3,0)

Results – Evolution/Imitation



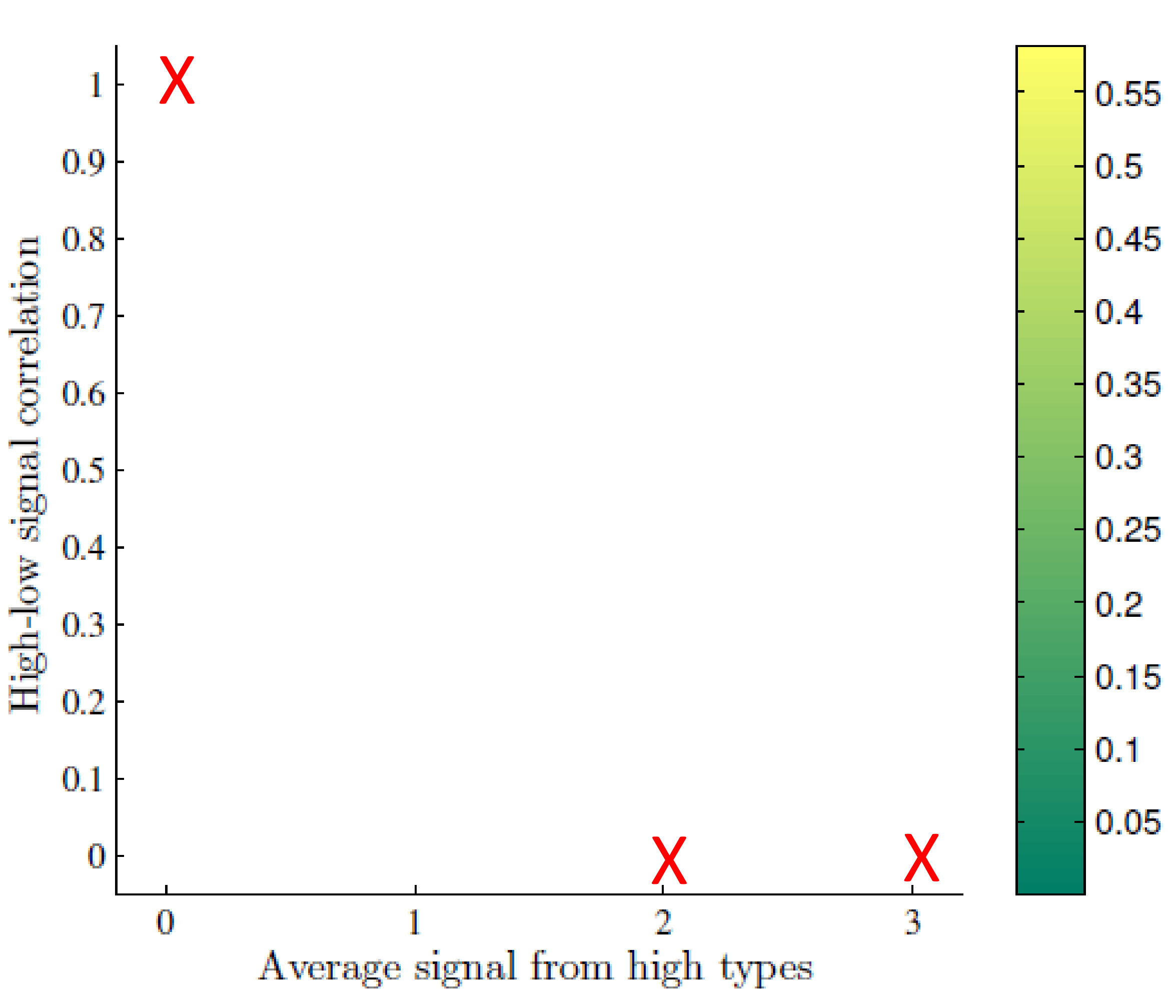
Let’s run this simulation 20 times for a million generations each.

Let’s count how frequently (in terms of total number of generations) the population is at each “point” in this graph

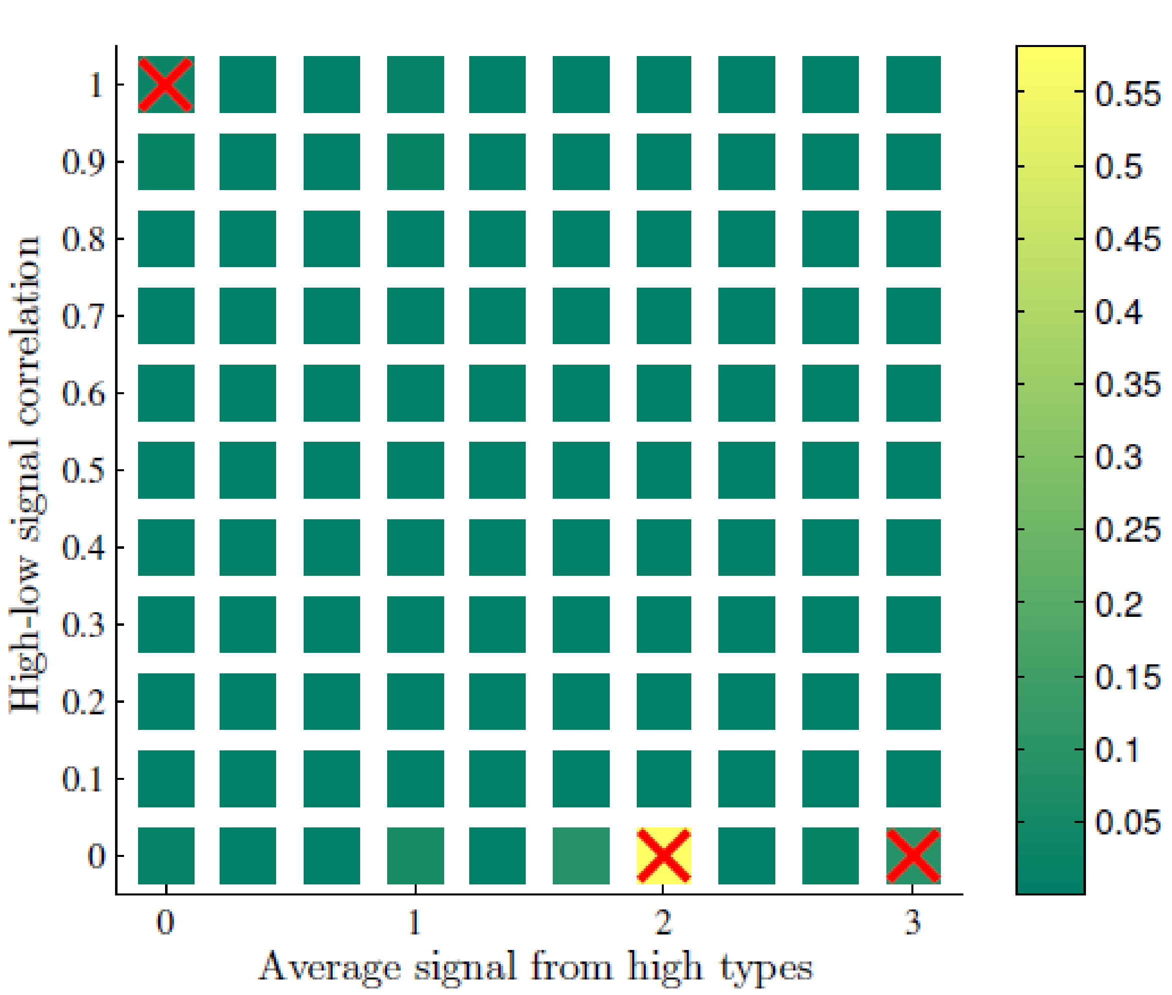
We can display frequency using color code (yellow=frequent, green=infrequnt)

(Since always some “experimentation,” points = “boxes,”)

Results – Evolution/Imitation

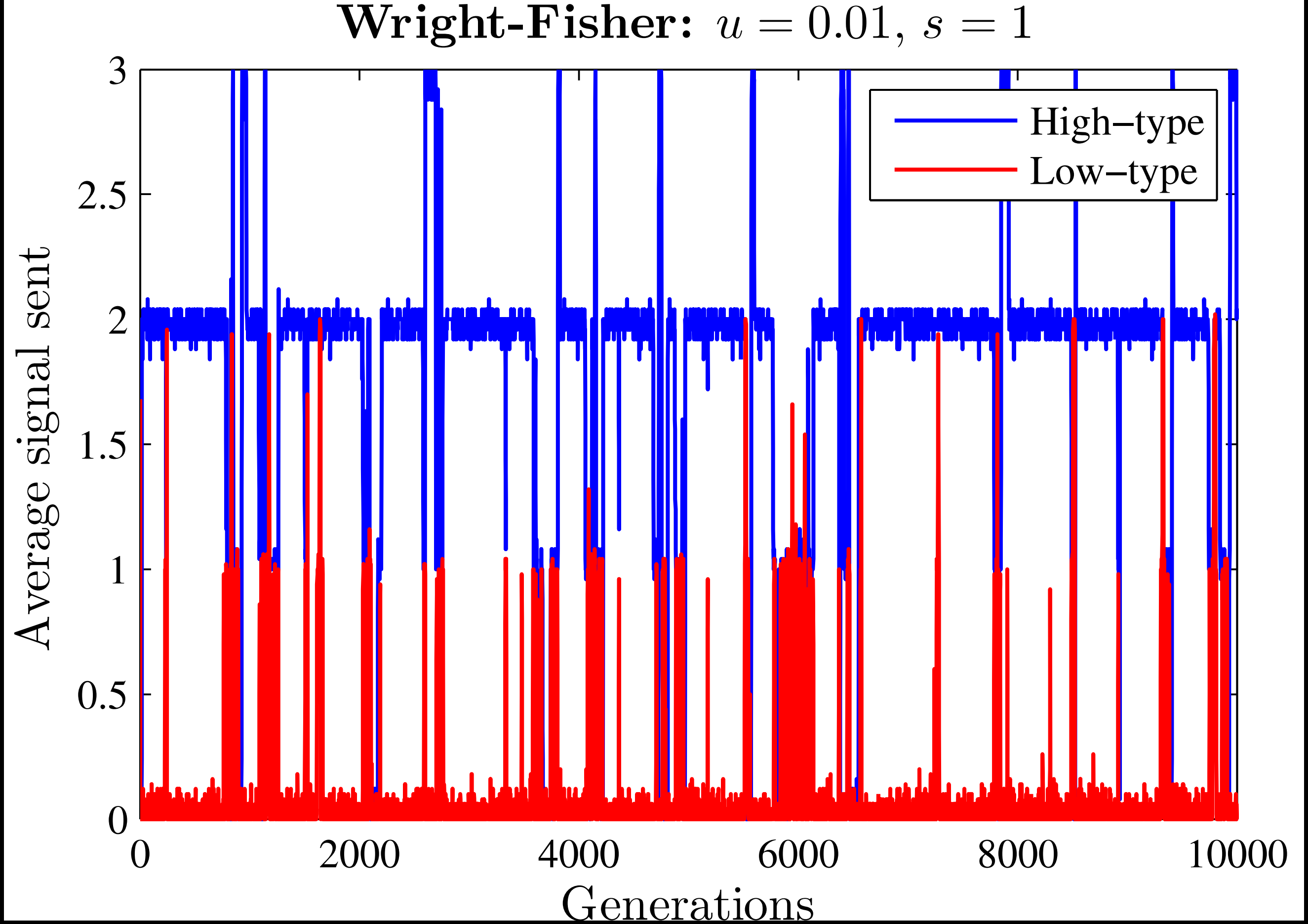


Results – Evolution/Imitation



Why?

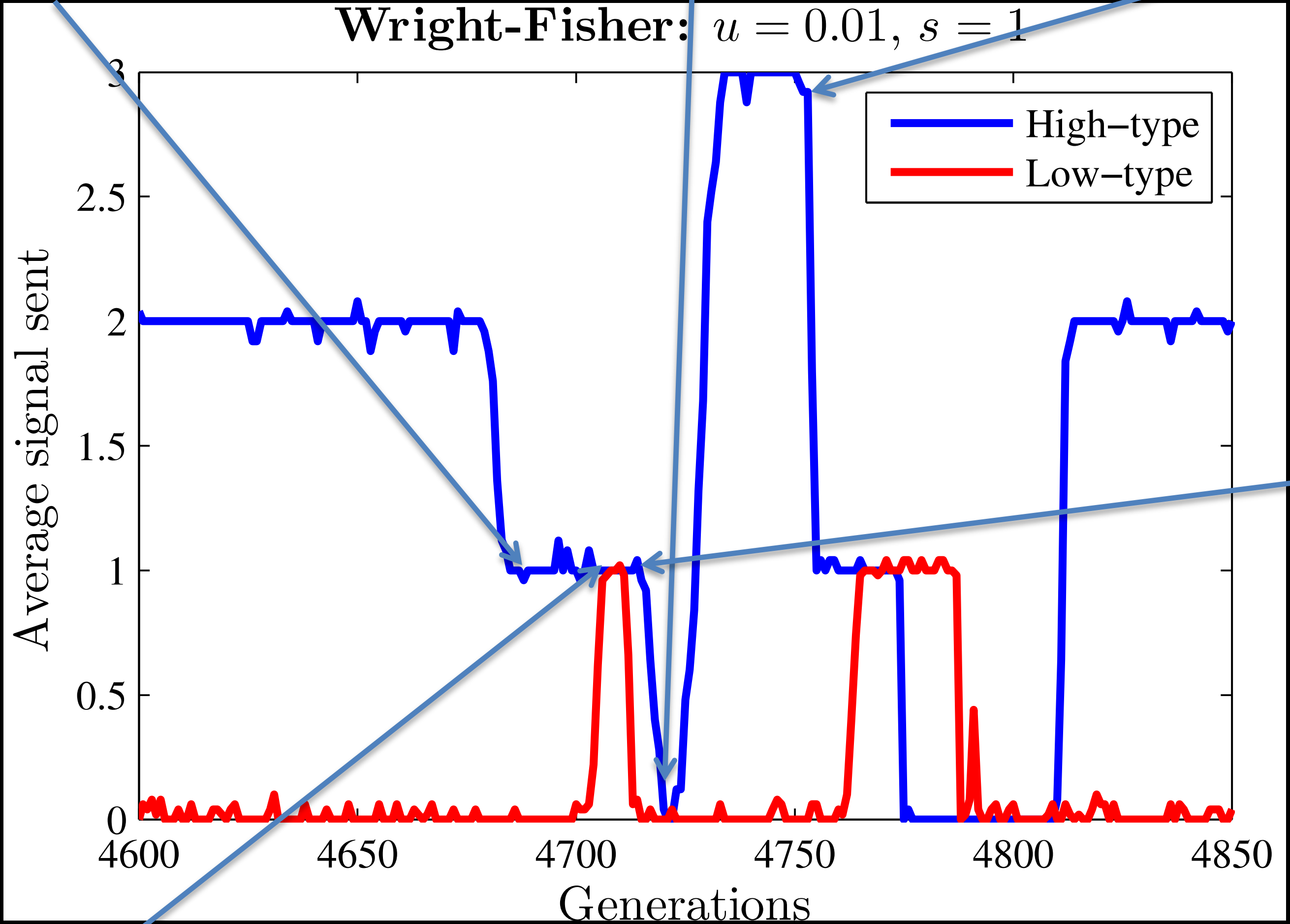
Here is an example time trend



As soon as receiver drifts to accepting 2 or 3

Enough receivers must have "neutrally drifted" to accept 1 so worth for good but not bad types

As soon as receiver drifts to accepting 1 or 2



Very quickly
After bad start
Sending 1,
receivers stop
Accepting 1

If in meantime
Receivers stop
Accepting 2
(by drift), then
Both good and
Bad better
Sending 0

Since good but not bad sending 1, receivers start accepting 1, to point where bad start sendin

Must leave efficient separating via …

1) receiver drift to accepting 1
2) good send 1

3) Bad send 1, but beforehand receivers drift away from accepting 2

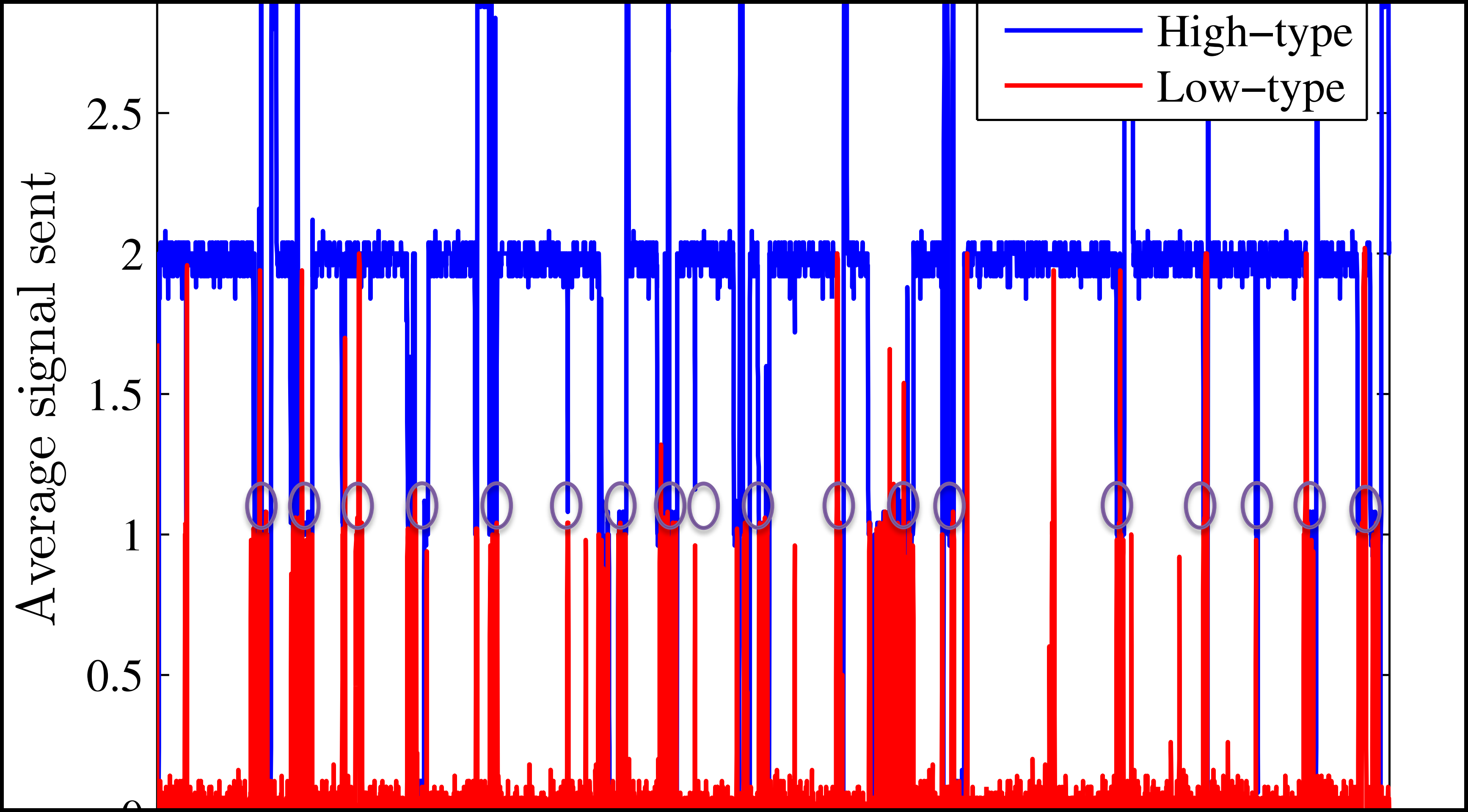
To leave pooling, just need…

1) Receiver drift to accept 2 or 3

To leave ostentatious separating, just need…

1) Receiver drift to accept 2 or 1

Here is an example time trend



Robust:

Robust? You will show in HW:

1) Doesn’t depend on parameters chosen for payoffs
2) Doesn’t depend on details of learning rule or evolutionary rule (e.g. if fitness is linear)
3) Still works even if REALLY small or FAIRLY large “experimentation”

Does it work for a continuum of signals (not just 0, 1, 2, 3)

And/or continuous actions (not just accept/reject) for the receiver?

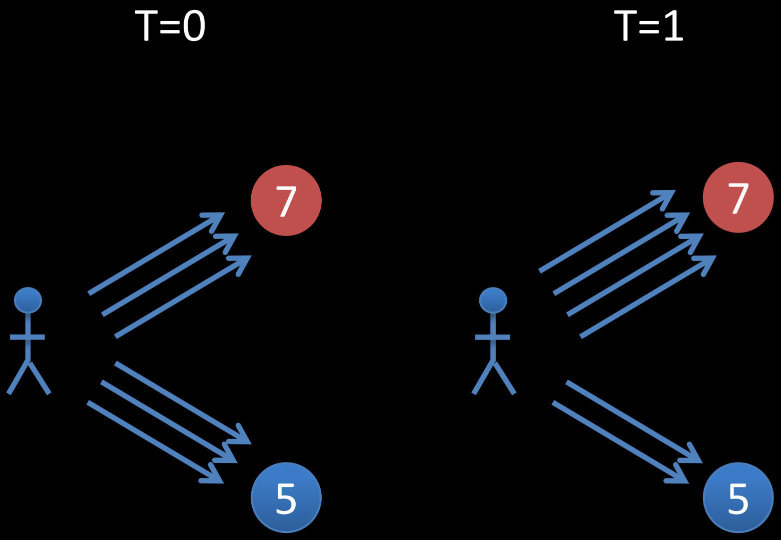
This would make a great final project

What about other models of communication?

(e.g. if not all senders want receiver to take “highest” action, but instead higher senders want receivers to take higher action, and receivers have similar preferences accept always want slightly less high.)

Reinforcement Learning Model

Reinforcement Learning



More successful behaviors
held more tenaciously

Basics of Reinforcement Learning:

Each of N players is assigned initial “values” for each strategy.

In each period

- Players adjust their values based on their payoffs
-values determine propensities
-choose strategy proportional to propensity
- Payoffs determined

Repeat for T periods

Display time trend

Perhaps repeat many such simulations, and display averages across all simulations

Let’s take a closer look at how the “values” adjust:

v_{t+1} (x)= v_t + a*(realized payoff –v_t (x)

Small a means adjust slowly
(a must be between 0 and 1)
(can also limit “memory”)

Value increases if payoffs higher than value.
(sometimes only for strategy played, sometimes for all)

Let’s take a close look at how propensities determined by values:

Propensity(x) = e^(g*v(x)) / [e^(g*v(x)) + e^(g*v(y))]

- y is another strategy
(assume only 2 for now)

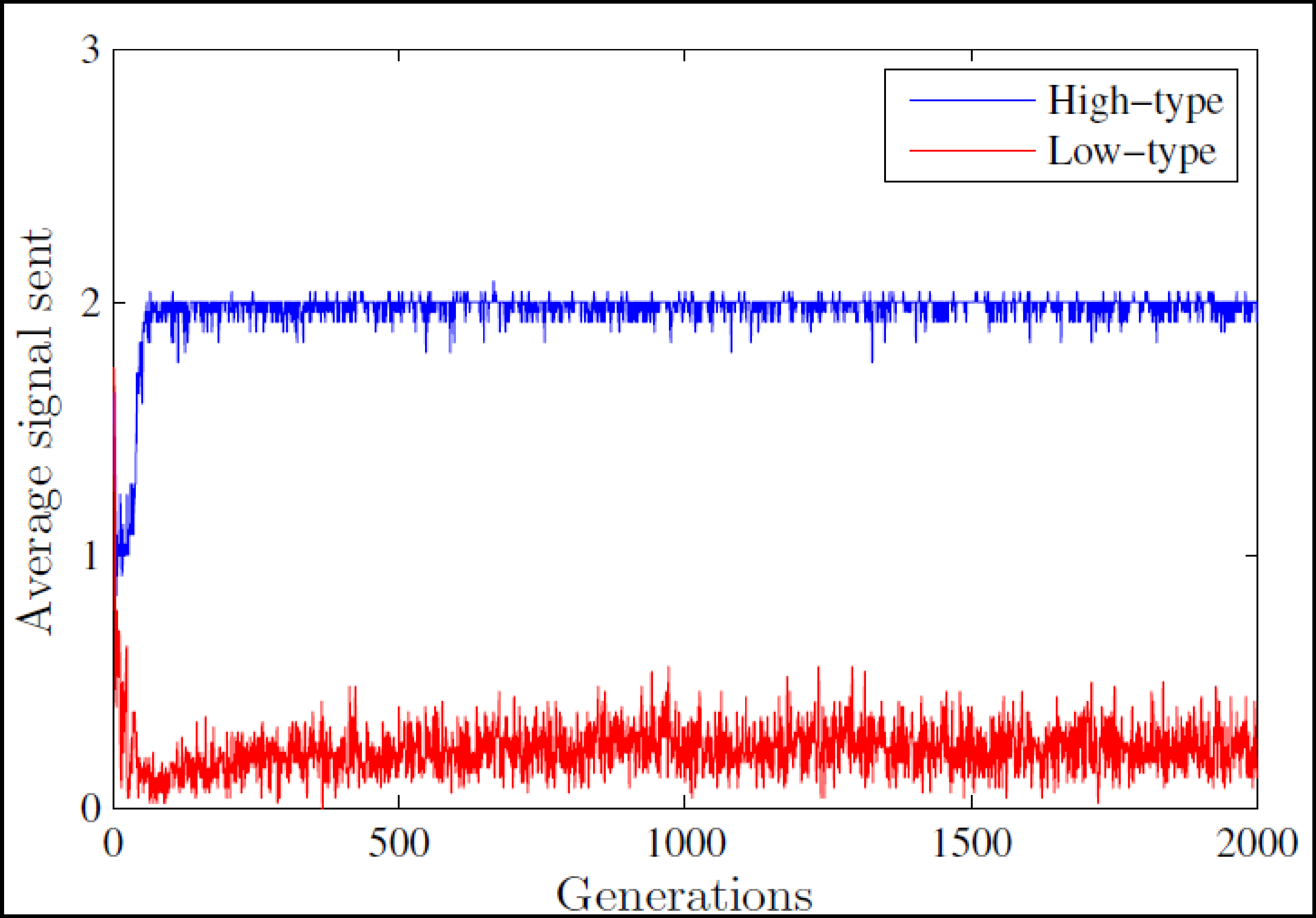
- g determines “selection strength”

-need not be exponential

Applying this to our costly signaling case…

Results…

Results – Reinforcement Learning



Even if start at pooling…

Always get to efficient separating, and stay there.

14.11 Insights from Game Theory into Social Behavior
Fall 2013