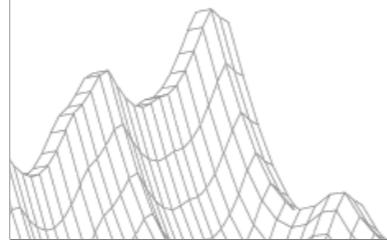


The Fitness of Ideas



Vanessa Ferdinand

cocosci Melbourne lab meeting
June 18, 2020



I've annotated this presentation so that you can read what I talked about on each slide.

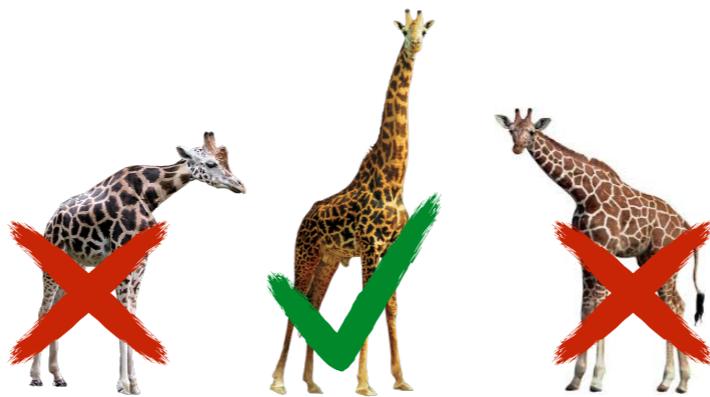
First I asked everyone what they thought the fitness of ideas was – or what they thought causally affected the relative frequencies of different ideas (in some kind of population: a network of humans, a single mind, etc.) over time.

There are tons of different things that could affect the fitness of ideas.

In these slides, I present ONE particular take on that and provide ONE particular definition of what the fitness of ideas is.

Part 1

Fitness



Fitness

Fitness is the number of children an organism has.

one particular
organism



$$f(Biden) = 4$$

one type of
organism *on average*

$$f(Democrats) = 1.47$$



$$f(Trump) = 5$$

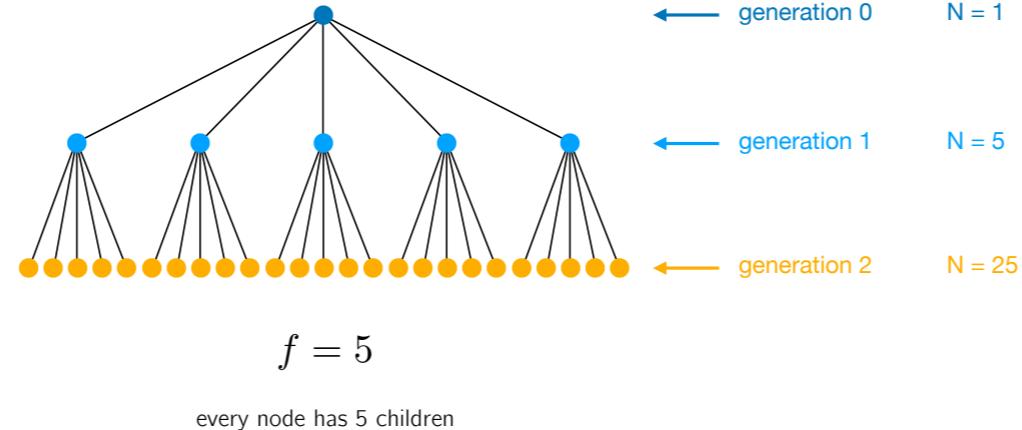
$$f(Republicans) = 2.08$$

This is the most basic, widespread definition of what fitness is in biology. There are many other ways to think of it – see Sober (1984) "The Nature of Selection" for awesome ideas.

Exponential population growth

$$N = f^g$$

b^x exponent = depth of tree
base of exponent = # branches



First we need to understand exponential growth processes.

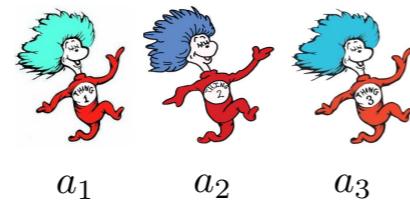
Let's assume a population is growing exponentially, without limit. N is the population size at each generation. This equation shows how N relates to fitness and the number of generations that have elapsed. The base of the exponent is the growth rate.

Main point: think of fitness as a growth rate. This is a more general concept than number of children.

Exponential population growth

$$f(a_1, a_2, a_3) = (3, 2, 1)$$

generation	$N(a_1)$	$N(a_2)$	$N(a_3)$	$q(a_1)$	$q(a_2)$	$q(a_3)$
0	1	1	1	0.33	0.33	0.33
1	3	2	1	0.50	0.33	0.17
2	9	4	1	0.64	0.29	0.07
3	27	8	1	0.75	0.22	0.03
4	81	16	1	0.83	0.16	0.01
5	243	32	1	0.88	0.12	0.00



← population has 3 types of organisms
(ex: 3 genotypes)

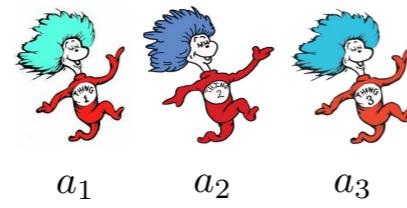
But of course there can be more than one type of organism in a population and each type can have a different growth rate.

Here's an example of a population with three types of organisms – each one has a different fitness.

Exponential population growth

$$f(a_1, a_2, a_3) = (3, 2, 1)$$

generation	$N(a_1)$	$N(a_2)$	$N(a_3)$	$q(a_1)$	$q(a_2)$	$q(a_3)$
0	1	1	1	0.33	0.33	0.33
1	3	2	1	0.50	0.33	0.17
2	9	4	1	0.64	0.29	0.07
3	27	8	1	0.75	0.22	0.03
4	81	16	1	0.83	0.16	0.01
5	243	32	1	0.88	0.12	0.00



number of each type
in the population

frequency of each type
in the population

We can look at the differential growth in this population in terms of N (the census count of each type in the population) or in terms of q (the frequency of each type in the population).

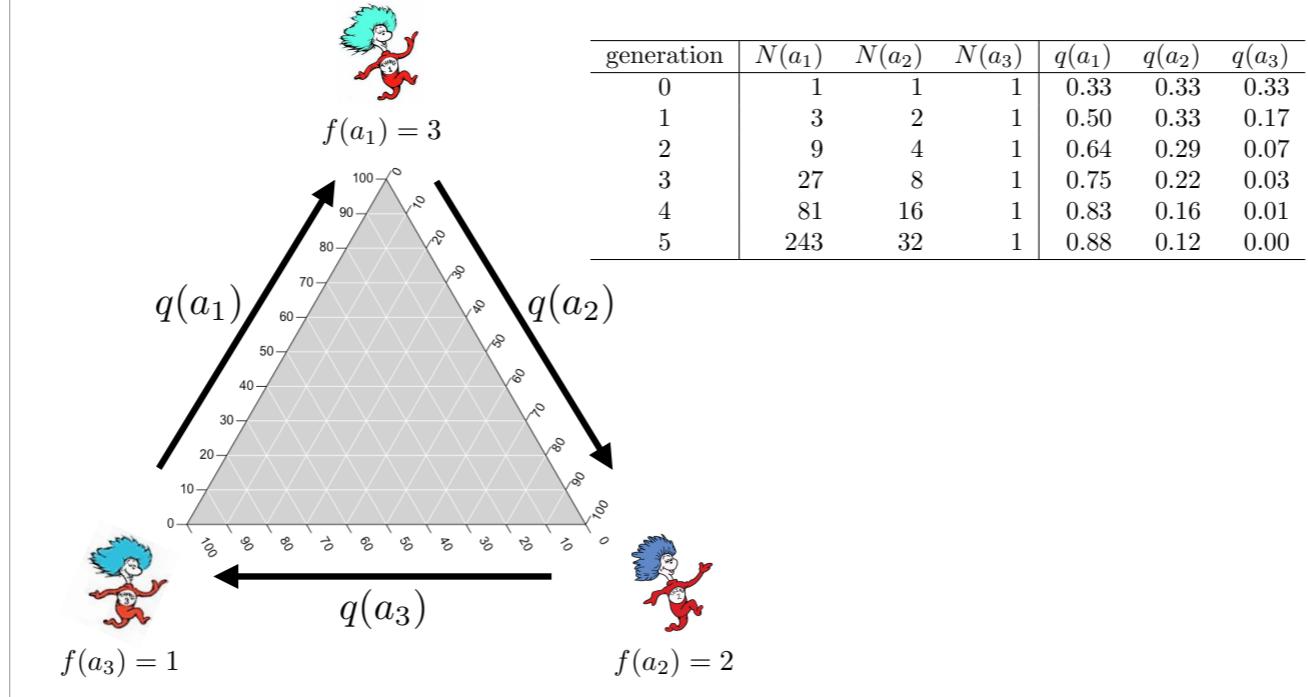
How to get the frequency:

Divide each $N(a)$ by the total fitness of the population. Ex: total fitness in generation 1 is 3, in generation 2 is 6, etc. Frequencies must sum to one: the frequency representation of a population is a stochastic vector.

Important note:

If we scrap the assumption that populations can growth without bound (i.e. we impose some carrying capacity) the frequencies (q) still accurately tell us what the expected proportion of each type is at each generation, no matter if we limit the population to a size of 100, 10k, or infinity.

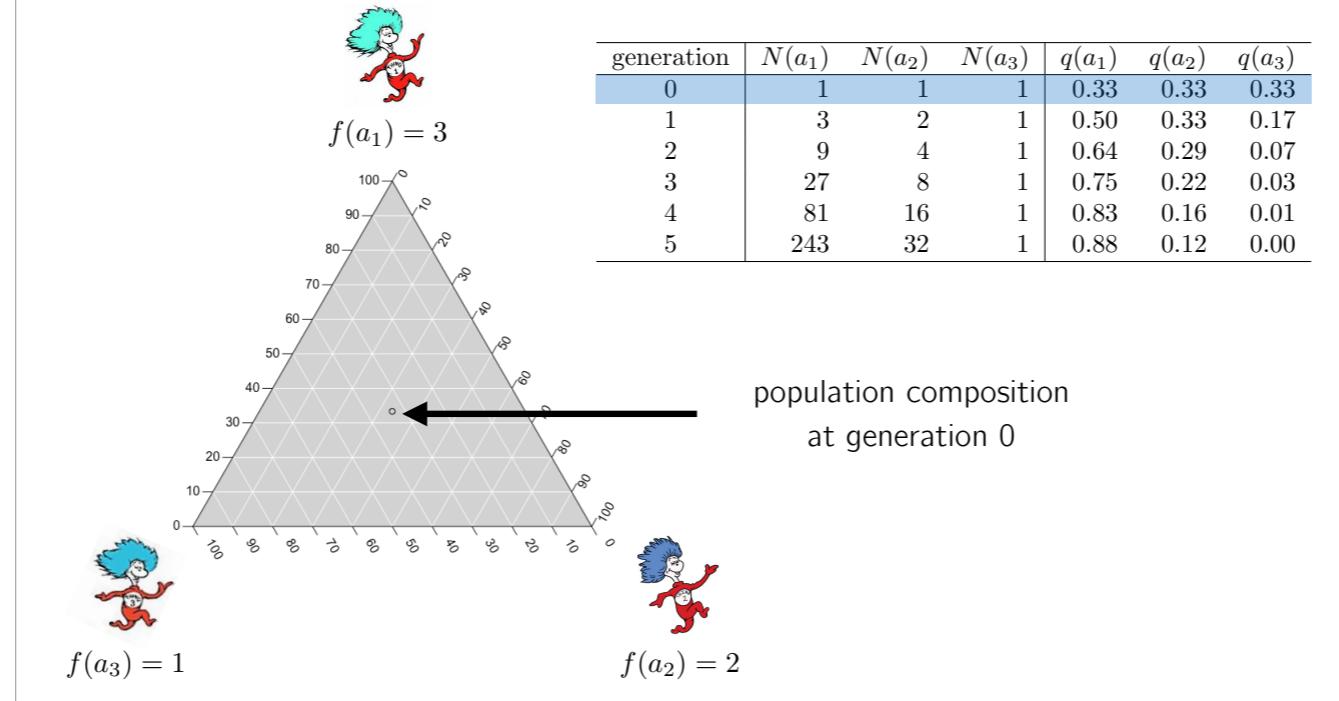
Replicator dynamics



An evolving population can be nicely represented on a simplex. Here's how to read a simplex plot.

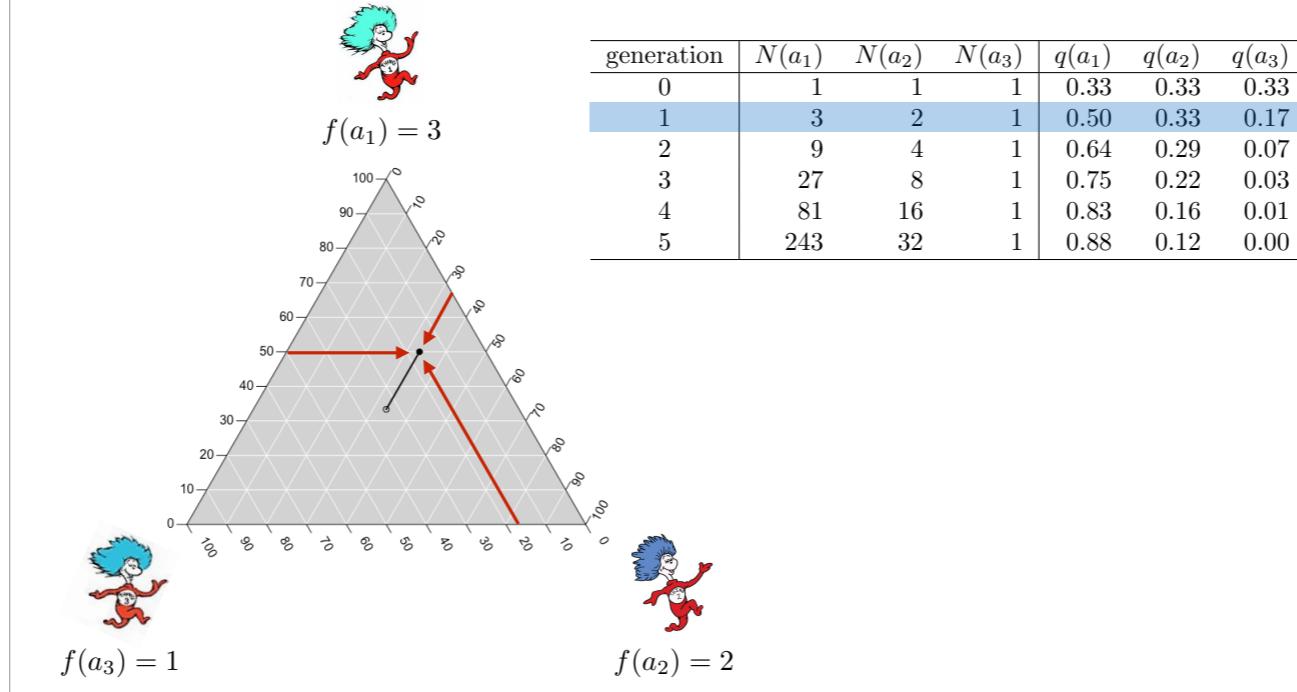
Read each of the 3 axes in the direction of these arrows. Each axis is the frequency one type in the population (these plots are showing frequency out of 100 though...pretend the axes run 0 to 1).

Replicator dynamics



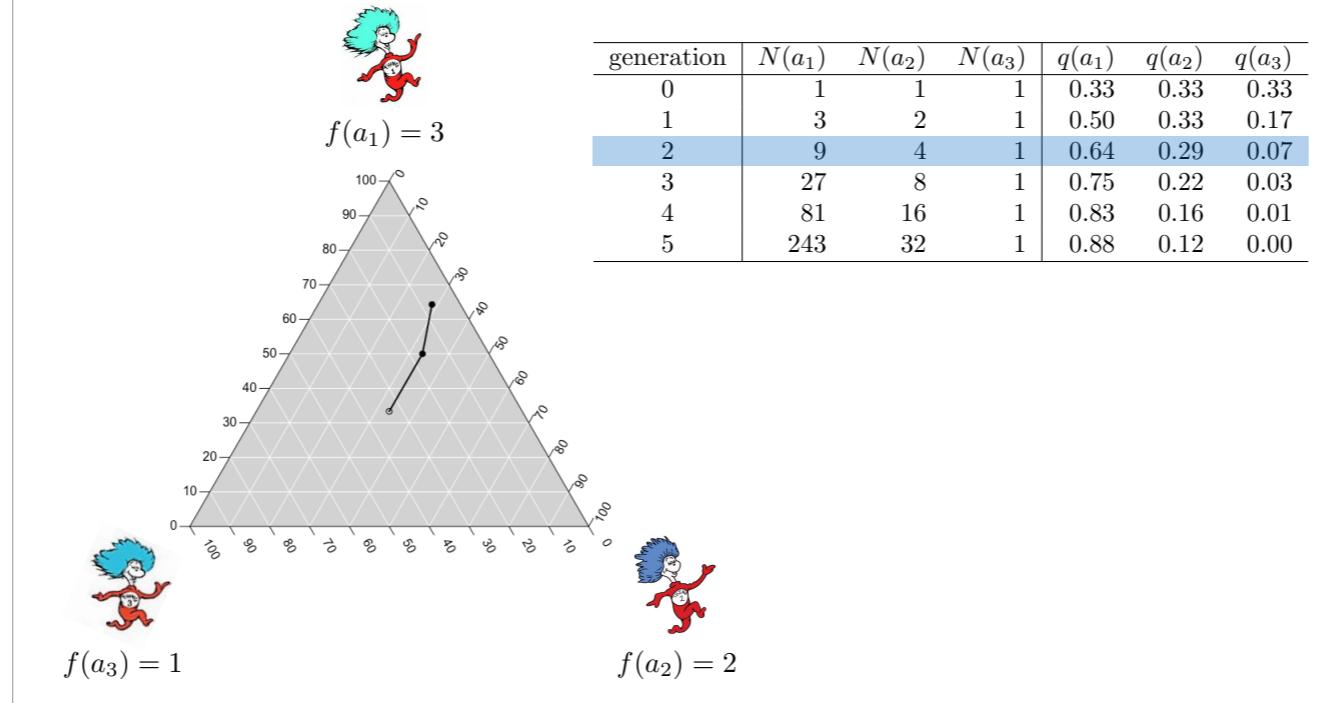
When all organisms have the frequency (i.e. the population is in a uniform distribution) then that's represented by the point at the exact middle of the plot.

Replicator dynamics



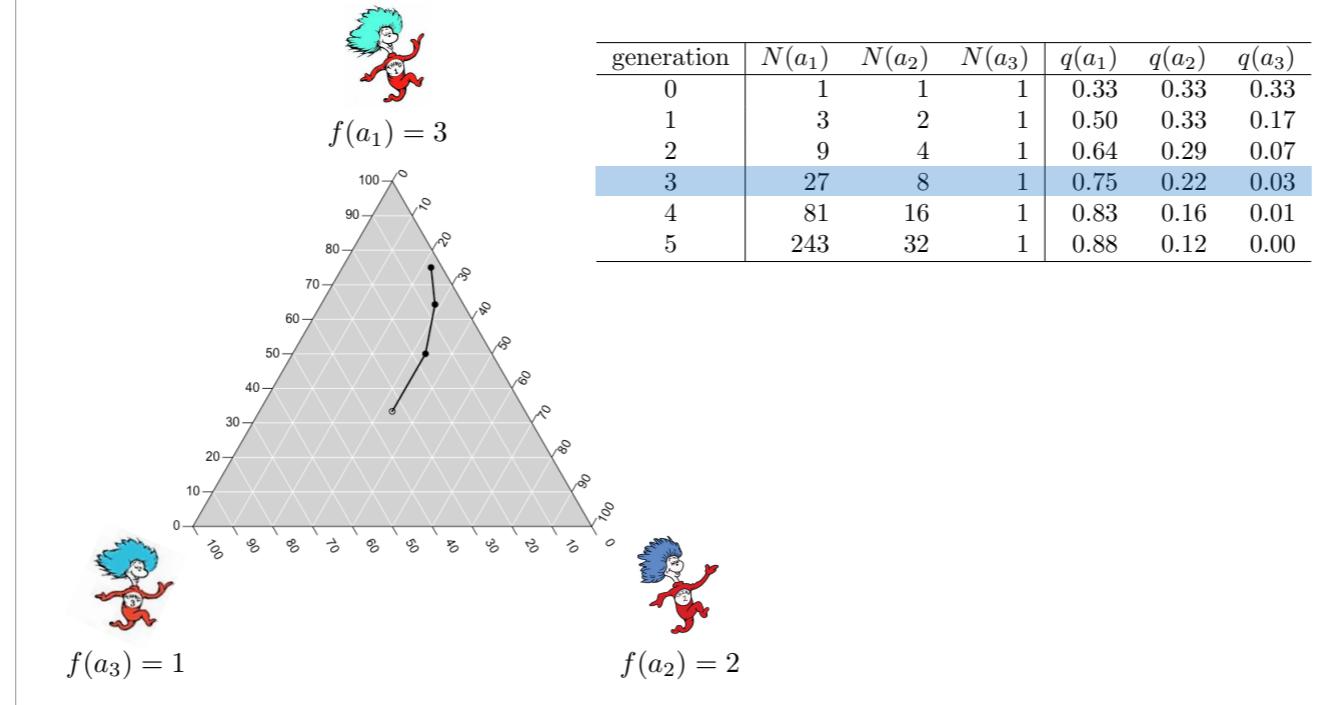
Here's how to read down each of the axes to plot the population at generation 1.

Replicator dynamics



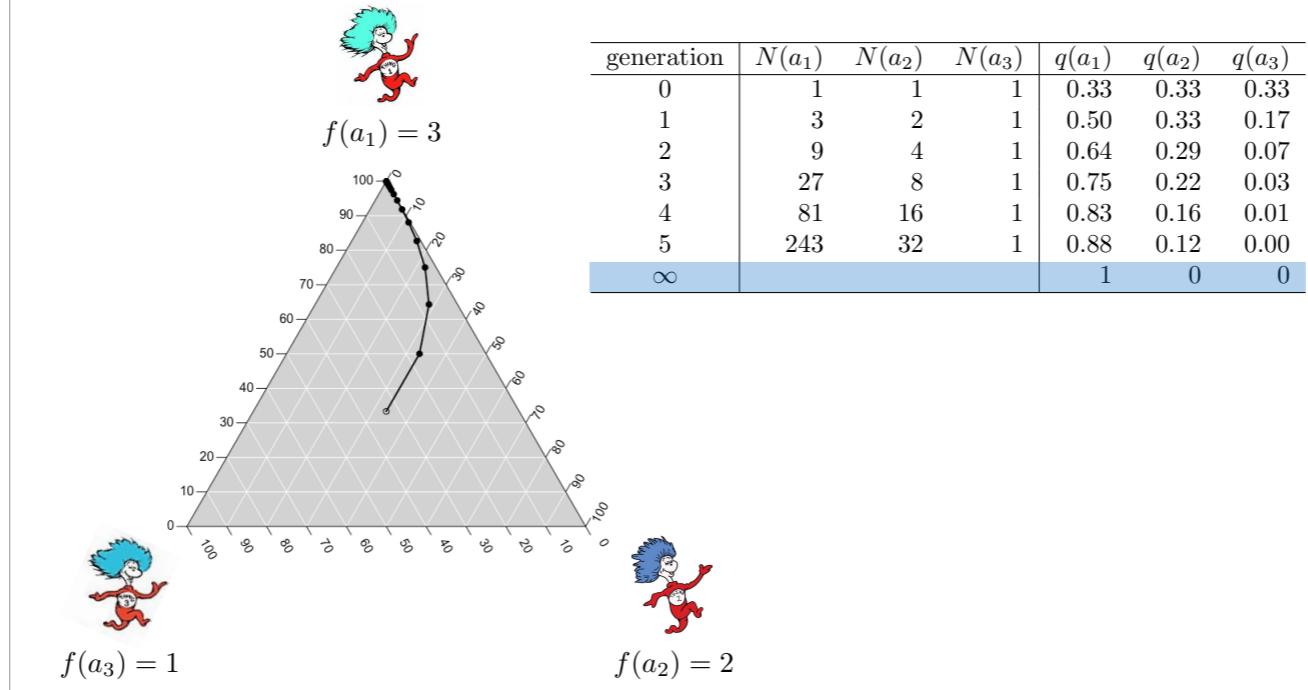
Here's generation 2.

Replicator dynamics



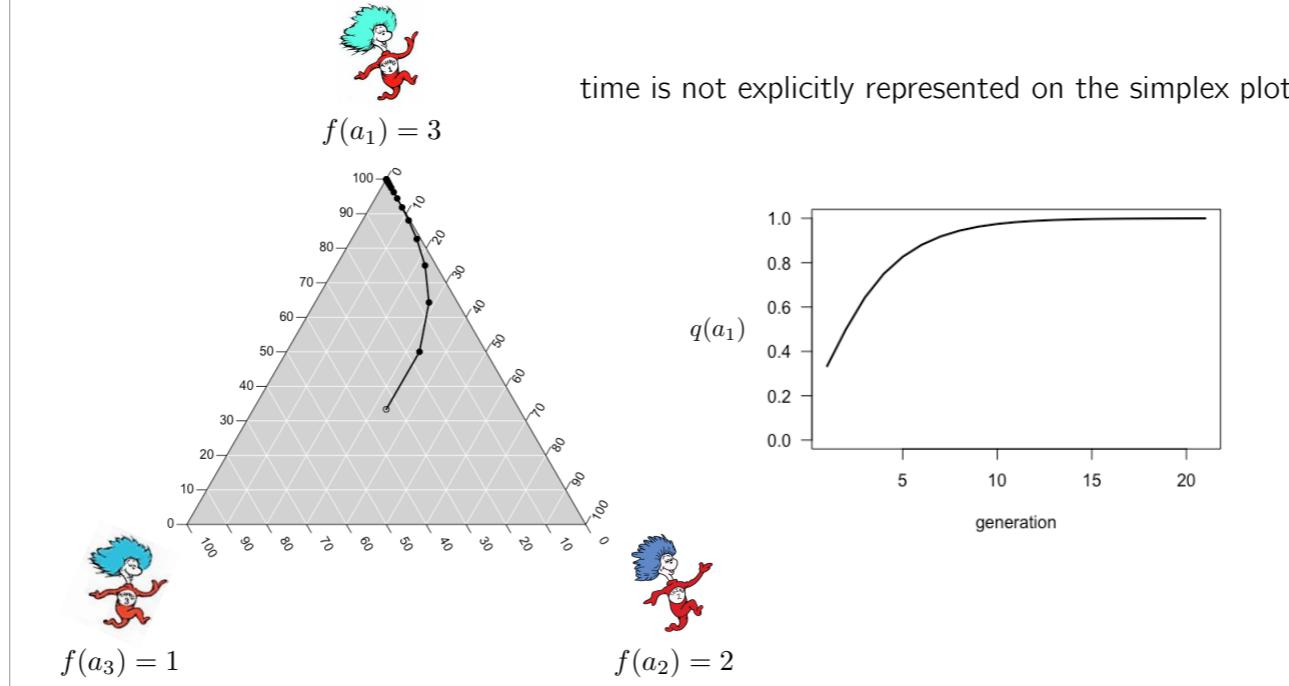
Here's generation 3.

Replicator dynamics



And as you run infinite generations, the frequencies approach a point distribution, where the organism with the highest fitness takes over the population.

Replicator dynamics



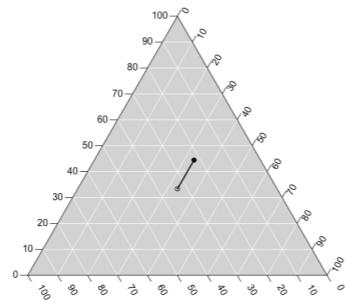
The simplex plot does not have an axis which represents time. How have to infer the direction of the curve.

Here is a plot of the same data with time on the x axis and the frequency of type a_1 on the y axis. We had to drop the information about the other frequencies to plot this line. The simplex plot is a nice way to show the frequencies of all population types together.

Differential fitness

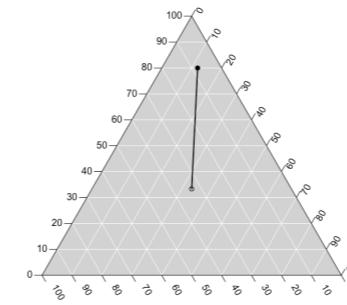
low differential fitness

$$f(a_1, a_2, a_3) = (2, 3, 4)$$



high differential fitness

$$f(a_1, a_2, a_3) = (2, 3, 20)$$



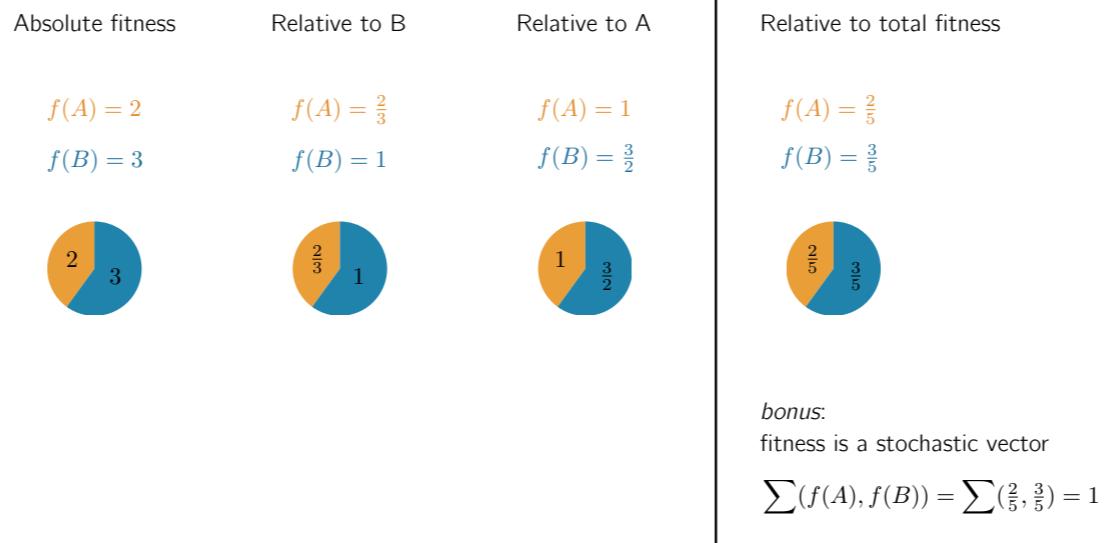
higher differential fitness means larger evolutionary steps are taken each generation

The most interesting thing about fitness is how differential it is – meaning, how similar or different the growth rates of each type are.

Similar growth rates = low differential fitness = smaller rates of change in the population = smaller evolutionary steps each generation.

Different growth rates = high differential fitness = larger rates of change in the population = larger evolutionary steps each generation.

Fitness representations



There are two main ways to represent fitness: absolute fitness and relative fitness.

Notice how all of these representations maintain the same proportion between growth rates. If you divide a bunch of things by the same number (i.e. a constant) then you maintain the proportions. So, you can represent fitness relative to anything you want. Representing it relative to total fitness is very useful and has the added bonus of making fitness into a stochastic vector too – which puts it into the same format as the frequency representation of the population. Having the fitness and the population both be stochastic vectors makes it easy to do a bunch of math things to them. Like calculate the relative flow of information from 1) the population at time t and 2) the fitnesses into 3) the population at time t+1. Stuff like that.

Replicator dynamics

The replicator dynamics equation:

$$q(a_i)' = \frac{q(a_i)f(a_i)}{\sum_{k=1}^n q(a_k)f(a_k)}$$

make everything sum to one

generation	$N(a_1)$	$N(a_2)$	$N(a_3)$	$q(a_1)$	$q(a_2)$	$q(a_3)$	$f(a_1, a_2, a_3) = (3, 2, 1)$
0	1	1	1	0.33	0.33	0.33	
1	3	2	1	0.50	0.33	0.17	
2	9	4	1	0.64	0.29	0.07	
3	27	8	1	0.75	0.22	0.03	
4	81	16	1	0.83	0.16	0.01	
5	243	32	1	0.88	0.12	0.00	

This equation gives us an easy way to calculate the frequencies over time. So now we can scrap working with the raw numbers entirely.

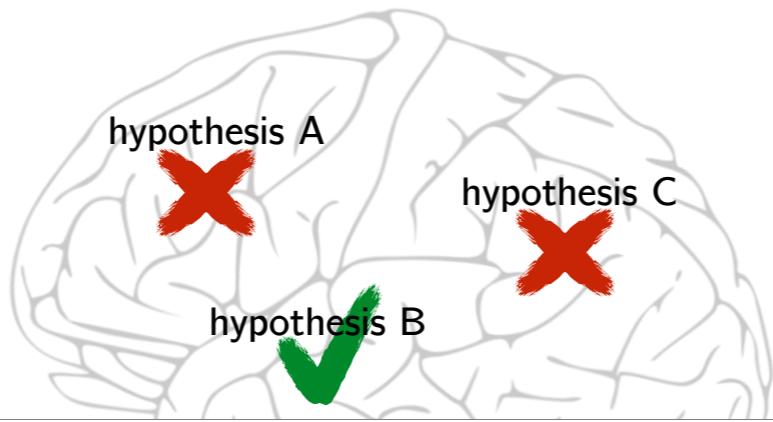
See Nowak (2006) "Evolutionary Dynamics: exploring the equations of life" for a complete introduction to replicator dynamics.

So that's what fitness and replicator dynamics are.

We'll come back to that later.

Part 2

Learning



Bayesian inference

Let's watch a Bayesian learner learn about some coin flips.

Learning goal:

What is the bias on the coin that's being flipped?



Let's say the learner only has **3 hypotheses**:

- 1) it's a H:T = 50:50 (fair) coin
- 2) it's a H:T = 70:30 coin
- 3) it's a H:T = 30:70 coin

And let's say the learner begins with equal belief in the three hypotheses.

This is represented by a **uniform prior**: $p(h_1, h_2, h_3) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$

Switching gears now, let's walk through an example of how learning works. We'll use one particular model of learning called Bayesian updating.

Bayesian inference

Flip a fair coin 100 times. The first flip comes up **tails**.

Use Bayes rule to calculate the learner's new beliefs.



- 1) Get the likelihood of the data under each hypothesis

$$p(\text{tails})|h_1 = 50 : 50 = .5$$

$$p(\text{tails})|h_2 = 70 : 30 = .3 \quad p(d|h_i) = (0.5, 0.3, 0.7)$$

$$p(\text{tails})|h_3 = 30 : 70 = .7$$

- 2) Multiply each hypothesis' prior probability by its likelihood under the data. Normalize.

$$p(h_i) \times p(d|h_i) \propto p(h_i|d) \quad p(h_i) \rightarrow p(h_i|d)$$

$$\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right) \times (0.5, 0.3, 0.7) = (0.17, 0.10, 0.23) \quad \sum \neq 1 \quad \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right) \rightarrow (0.34, 0.20, 0.46)$$
$$(0.34, 0.20, 0.46) \quad \sum = 1$$

$p(d|h)$ is called the likelihood
 $p(h)$ is called the prior
 $p(h|d)$ is called the posterior

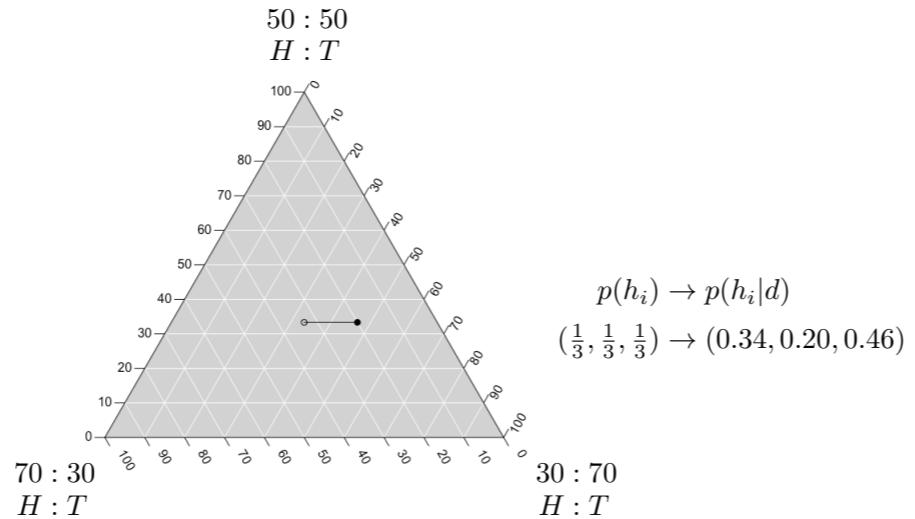
$p(h_i)$ and $p(h_i|d)$ each must sum to one (they are stochastic vectors)
 $p(d|h_i)$ does not have to sum to one, it can be composed of any type of number

In Bayesian updating, the posterior at learning cycle t becomes the new prior using in learning cycle $t+1$.

Bottom right of slide:
That's how the prior updates given the data in this example.

Bayesian inference

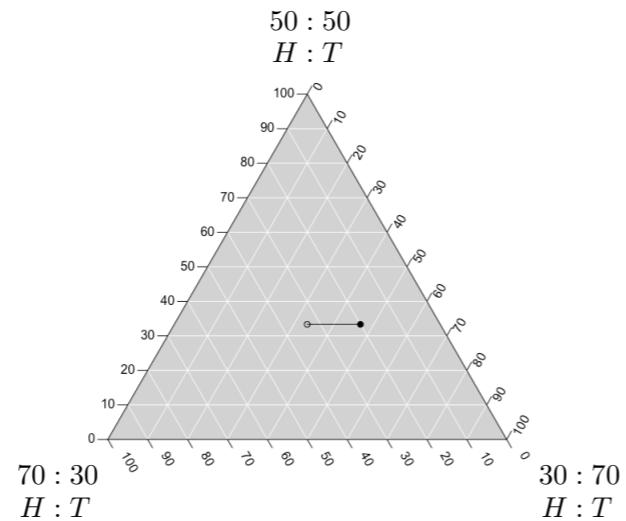
Flip a fair coin 100 times. The first flip comes up **tails**.



Now let's visualize the change in relative probability of the hypotheses over time on a simplex.

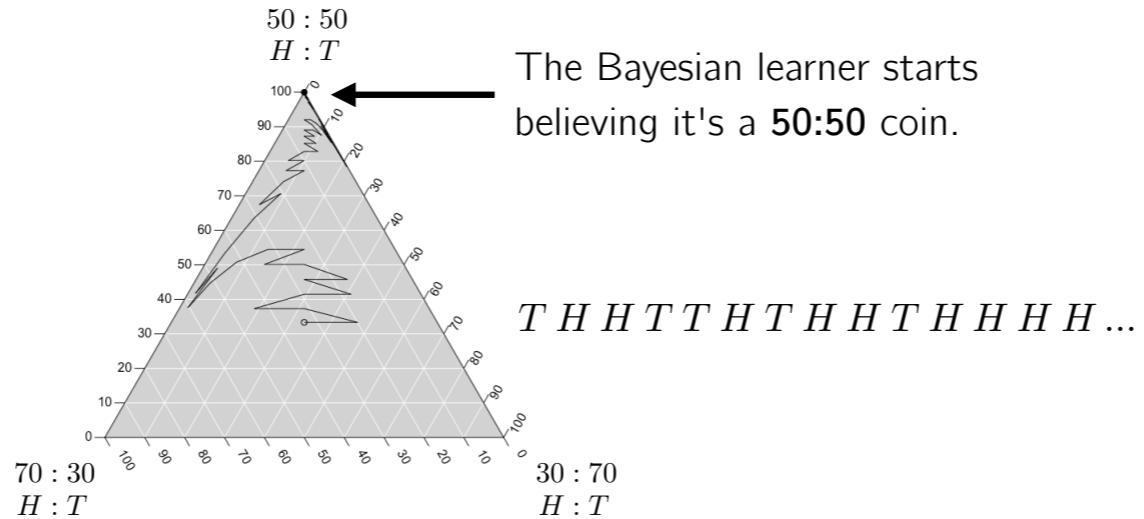
Bayesian inference

Update beliefs for each of the 100 coin flips.



Bayesian inference

Update beliefs for each of the 100 coin flips.



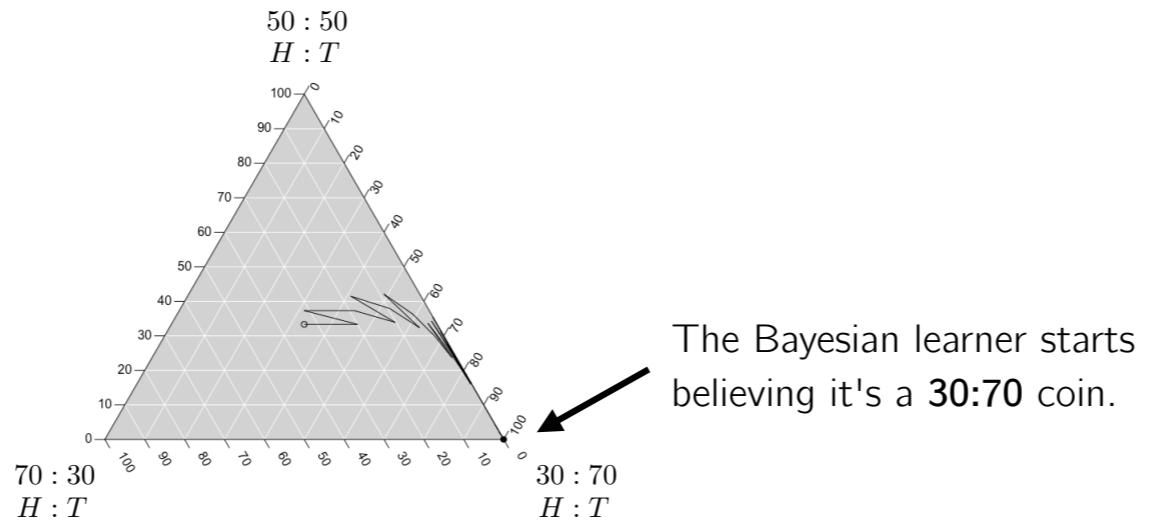
Here's a complete learning trajectory for a Bayesian learning observing 100 coin flips.

I've shown the first 14 flips so you can trace the learner's responses to each one. The first flip is a tails so you see it makes the learner favor hypothesis 30:70. But then the next flip is a heads, so the learner moves back, but closer to hypothesis 50:50, later the run of heads takes the learner toward hypothesis 70:30. And so on.

The sequence of 100 flips used here contained 50 heads and 50 tails.

Bayesian inference

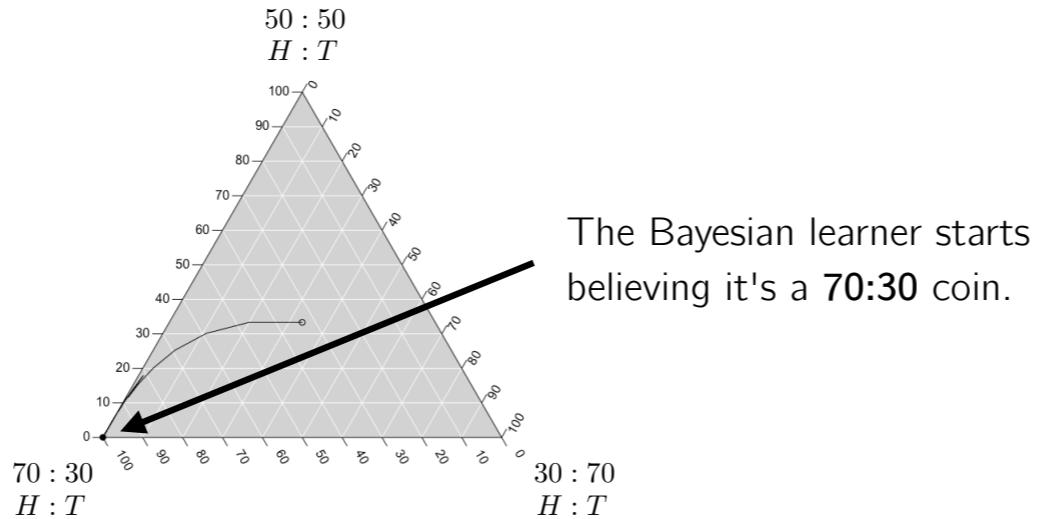
Now flip a **30:70** coin 100 times...



Here the sequence contained 70 tails and 30 heads.

Bayesian inference

Now flip a 70:30 coin 100 times...



Here the sequence contained 30 tails and 70 heads.

Each time the Bayesian updater learns the correct hypothesis. And it is guaranteed to settle on the correct one 1) given enough data and 2) if the correct hypothesis is one of the hypotheses it is entertaining. If the correct hypothesis isn't in the mix, then it will converge to the next best explanation. The "best explanation" is the hypothesis that exhibits the highest likelihood under the data.

Do you notice any qualitative differences between the learning trajectories from the three sets of data shown in these last 3 slides?

Bayesian inference

Bayes rule:

$$p(h_i|d) = \frac{p(h_i)p(d|h_i)}{\sum_{k=1}^n p(h_k)p(d|h_k)}$$

make everything sum to one

prior probability
of each hypothesis

likelihood of the data
given each hypothesis

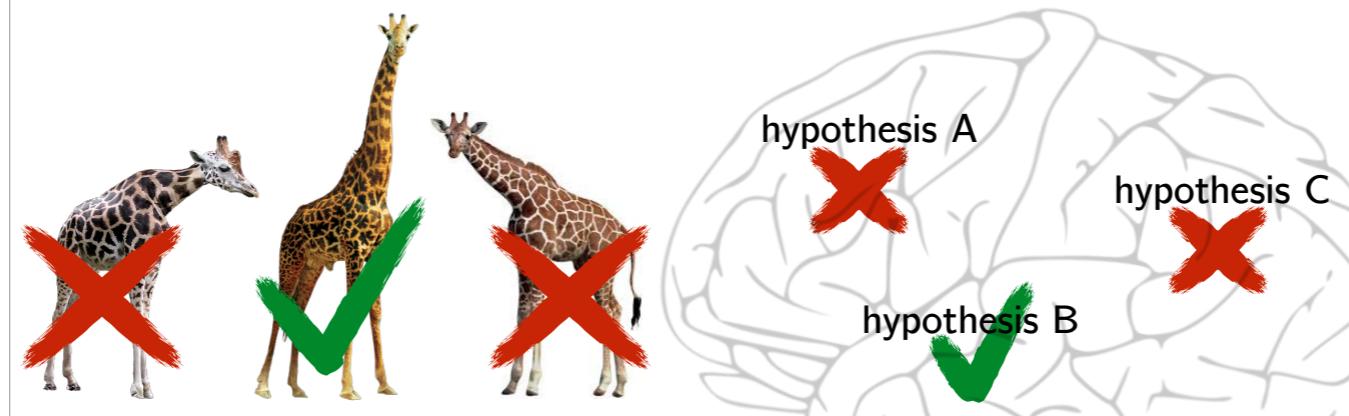
posterior probability
of each hypothesis

If you feel like you've seen this before, well you have...

Here's the full equation for Bayes rule and all of its parts.

Part 3

Equivalence



Equivalence between evolution and learning

Replicator dynamics

$$q(a_i)' = \frac{q(a_i)f(a_i)}{\sum_{k=1}^n q(a_k)f(a_k)}$$

Bayesian inference

$$p(h_i|d) = \frac{p(h_i)p(d|h_i)}{\sum_{k=1}^n p(h_k)p(d|h_k)}$$

Harper (2010), Shalizi (2010), Krakauer (2015), Suchow et al (2017)

The equation for replicator dynamics and Bayes rule are formally equivalent. "Formally equivalent" means each part of each equation maps on to one another.

Equivalence between evolution and learning

Replicator dynamics

$$q(a_i)' = \frac{q(a_i)f(a_i)}{\sum_{k=1}^n q(a_k)f(a_k)}$$

Bayesian inference

$$p(h_i|d) = \frac{p(h_i)p(d|h_i)}{\sum_{k=1}^n p(h_k)p(d|h_k)}$$

Harper (2010), Shalizi (2010), Krakauer (2015), Suchow et al (2017)

Equivalence between evolution and learning

Replicator dynamics

$$q(a_i)' = \frac{q(a_i)f(a_i)}{\sum_{k=1}^n q(a_k)f(a_k)}$$

Bayesian inference

$$p(h_i|d) = \frac{p(h_i)p(d|h_i)}{\sum_{k=1}^n p(h_k)p(d|h_k)}$$

This is the money shot. This is what's so interesting.

What is the fitness of ideas?

No one has actually developed this idea yet.

It is the support they get from the data at hand.

Harper (2010), Shalizi (2010), Krakauer (2015), Suchow et al (2017)

This equivalence has been noted in passing by the four citations shown here, but no one has really "done" anything with this equivalence yet, except point it out.

If an analogy between two domains (X and Y) is useful, it should give us insight into X based on how we think about things in Y, or vice versa. What can our knowledge of Bayesian inference make us realize about replicator dynamics? And what can our knowledge about replicator dynamics make us realize about Bayesian inference?

I'm working on this analogy and the best insight it gave me is centered on the idea of differential fitness. What does the concept of differential fitness from evolution tell us about the dynamics of the evolution of ideas?

Note: evolution just means "change over time". It's perfectly ok to talk about the evolution of ideas. I'm not using a Darwinian framework to initialize my thinking about the evolution of ideas – it would be a total shame to explore something as awesome and varied as learning and cultural evolution with such strict initial conceptual blinders on! Let's use the full power of our imaginations until there's a reason not to. So, ideas evolve if we can demonstrate that they (themselves, or a population of them) change over time.

Cool implications

"More data means less influence of the prior"

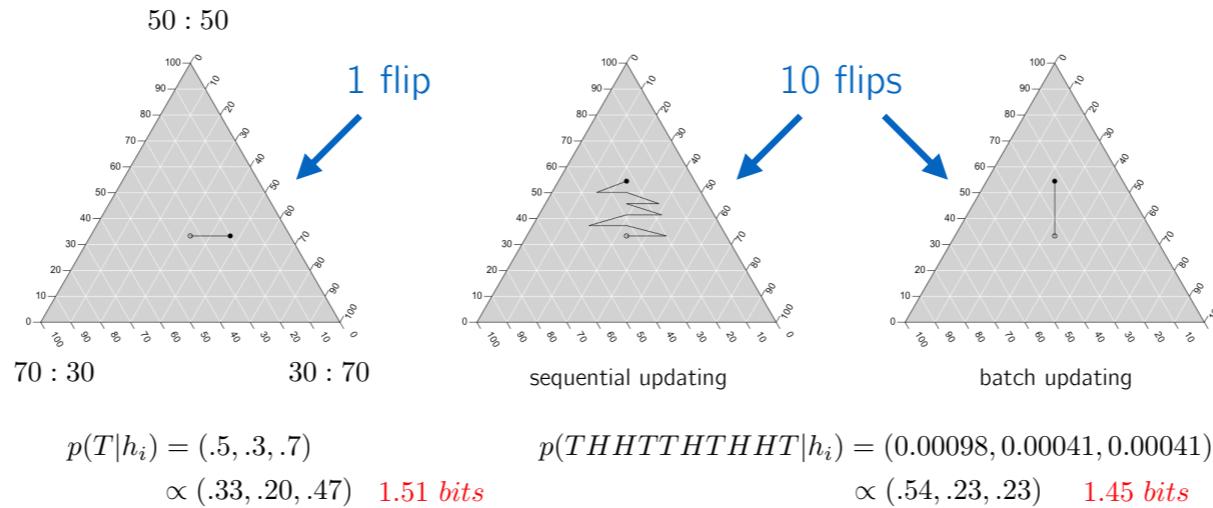
Why???

Here's a common thing you hear about Bayesian learning: "the more data you get, the less the influence of the prior". Or "seeing more and more data overwhelms the prior".

Cool implications

"More data means less influence of the prior"

Why???



Well the answer there is that differential fitness grows with the amount of data being seen. I need to prove this, but I think it's always the case when the "true" hypothesis is in the mix. And probably the case all of the time – unless you're only entertaining diametrically-opposed ideas, like only to 30:70 and 70:30 hypothesis: no matter how much data you see from a 50:50 source, the relative fitness of these two hypotheses won't change.

Main point: more data = higher differential fitness = larger evolutionary steps.

Here I've quantified the strength of differential fitness using Shannon entropy of the fitness values $p(d|hi)$. So higher entropy means lower differential fitness.

Here's a better way I like to quantify differential fitness: If there are N variants, $\log(N, \text{base})$ – Shannon entropy of $p(d|hi, \text{base})$ and if you make the base of the logarithm = to N, then your measure of differential fitness is bounded between 0 and 1, where 0 means no differential fitness and 1 means maximal differential fitness.

Cool implications

Also, the competition matters.

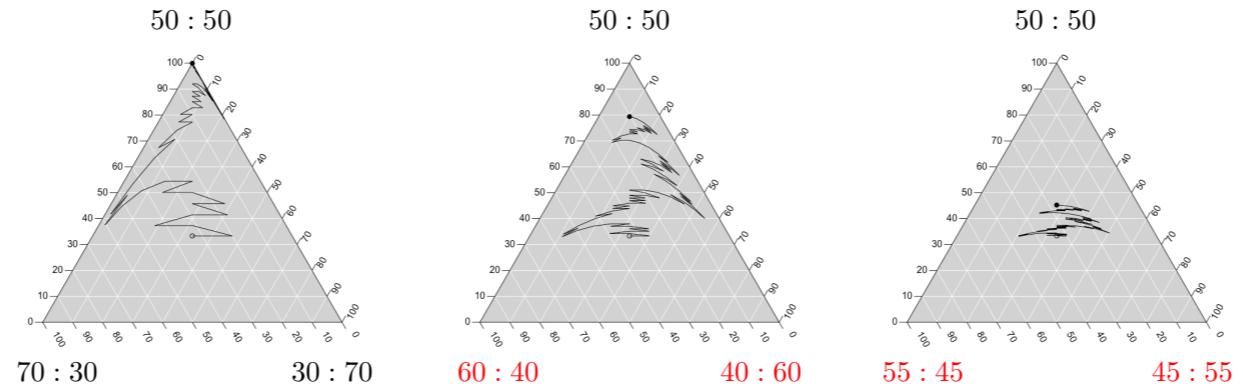
Why???

Differential fitness also helps us understand why the competition matters. Here, competition = the set of hypotheses that the Bayesian learner is entertaining.

Cool implications

Also, the competition matters.

Why???



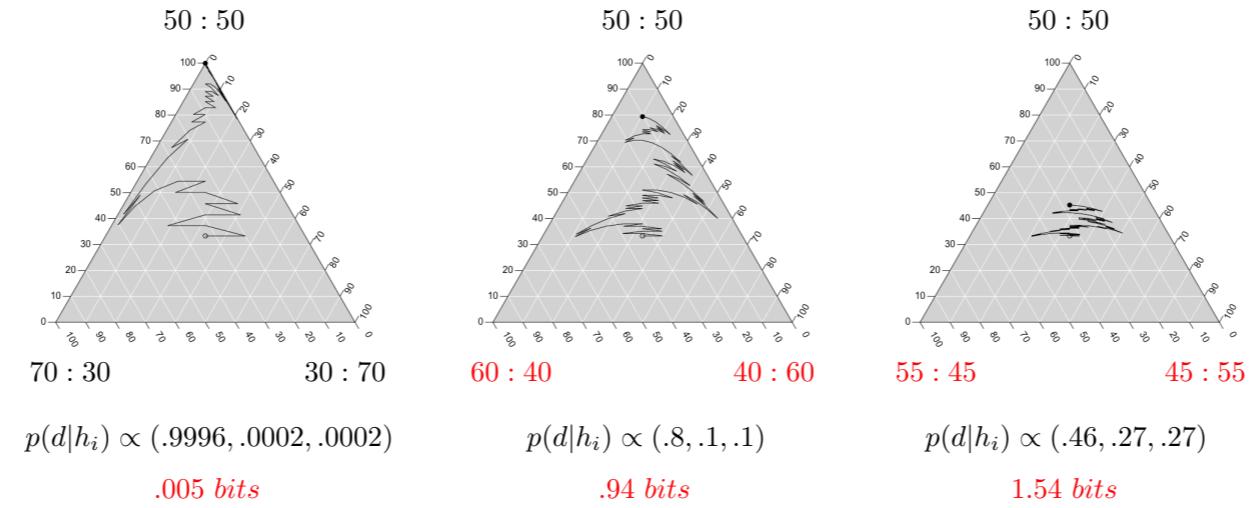
Each plot uses the same sequence of 100 flips, but different hypotheses.

Evolutionary trajectories for the same data sequence, but for three different sets of hypotheses.

Cool implications

Also, the competition matters.

Why???



As the likelihoods of the data under the hypotheses becomes more similar, differential fitness goes down.

Cool implications

Replicator dynamics

$$q(a_i)' = \frac{q(a_i)f(a_i)}{\sum_{k=1}^n q(a_k)f(a_k)}$$

Here fitness is a static, supervenient property

Bayesian inference

$$p(h_i|d) = \frac{p(h_i)p(d|h_i)}{\sum_{k=1}^n p(h_k)p(d|h_k)}$$

Here fitness changes each time new data is sampled

Fitness is usually represented as values that are averaged across all possible things that contribute to fitness (i.e. marginalized over all these things). This is different to the way we treat likelihoods in Bayesian learning.

Bayesian likelihoods are usually calculated for a current sample of data. However, we could calculate the likelihood of hypotheses given the source that generates a particular distribution over data types (i.e. marginalize over all samples that the source could produce).

Cool implications

Replicator dynamics

$$q(a_i)' = \frac{q(a_i)f(a_i)}{\sum_{k=1}^n q(a_k)f(a_k)}$$

Bayesian inference

$$p(h_i|d) = \frac{p(h_i)p(d|h_i)}{\sum_{k=1}^n p(h_k)p(d|h_k)}$$

Different environments can cause the fitness of organisms to change.

$$f(a_i|e_1) \text{ versus } f(a_i|e_2)$$

Ideas can be located in different data environments.

Ferdinand (2018)

$$f(h_i|d_j) = p(d_j|h_i)$$

Let's push the analogy further. We can make the mapping between these two equations even more similar by:

A) working with likelihoods that are calculated for the complete generating source of data

or

B) working with fitness values that are stochastically generated from a source of different fitness values

Biologists do B, but cognitive scientists don't really do A.

In Ferdinand (2018) I used A above to model the coevolution of ideas and data. I found this leap, that $f(h|d) = p(d|h)$, to be a useful conceptual tool, that allowed me to relate populations of ideas to populations of data. Note: this looks like I flipped a conditional probability, but $f(h|d)$ is not a probability. It is the growth rate of ideas given a particular data environment.

Fitness in different environments

drought years vs rainy years



Here's the basic biology concept that an organism's fitness can vary in different environments.

Fitness in different environments

fitness in rainy year: $f(a_1|e_1) = 4$

$f(a_2|e_1) = 6$

fitness in drought year: $f(a_1|e_2) = 3$

$f(a_2|e_2) = 2$



a_1



a_2

Both organisms in this example do better in rainy years. But the fern does better than the cactus in rainy years and does worse than the cactus in drought years.

Average fitness

Average fitness depends on base rate of each environment

	a_1	a_2	a_3
<i>environment 1</i>	4	6	8
<i>environment 2</i>	3	2	1
average fitness	3.5	4	4.5

$$p(e1) = 0.5$$

$$p(e2) = 0.5$$

	a_1	a_2	a_3
<i>environment 1</i>	4	6	8
<i>environment 2</i>	3	2	1
average fitness	3.3	3.2	3.1

$$p(e1) = 0.3$$

$$p(e2) = 0.7$$

Can update the population based on average fitness across environments

$$N_t(A) \times f(A|E) \times p(E) = N_{t+1}(A)$$

Average fitness is, well, the average fitness of the organism across all possible environments, weighted by the probability of each environments. So, marginalize over all environments to get the average fitness.

Left: average fitness when environments occur with equal fitness

Right: average fitness when environment 1 has a 70% of occurring and environment 2 has a 30% chance of occurring

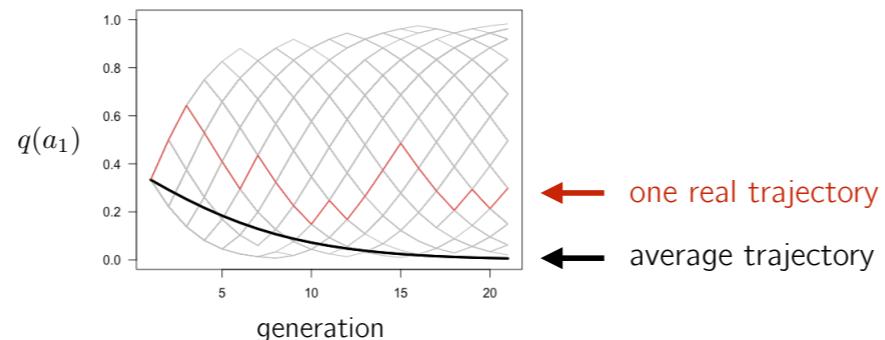
Different environment base rates can lead to different organisms taking over the population.

Average fitness

Average fitness depends on base rate of each environment

	a_1	a_2	a_3
environment 1	4	6	8
environment 2	3	2	1
average fitness	3.5	4	4.5

	a_1	a_2	a_3
environment 1	4	6	8
environment 2	3	2	1
average fitness	3.3	3.2	3.1



Here's the frequency of a_1 over time, solved using the replicator dynamics equation with the average fitness (black line). Variant a_3 eventually takes over the population.

The red line shows a simulated evolutionary trajectory. The simulation: Each generation, one environment was randomly sampled according to the probabilities $p(e_1) = 0.5$ and $p(e_2) = 0.5$. Then the population was updated using the replicator dynamics equation and the fitness values under the environment that was sampled.

The grey lines show a lattice of all possible evolutionary trajectories that could occur – each one is determined by a unique historical sequence of environments. Ex: (e_1, e_1, e_2, e_1) , (e_2, e_1, e_2, e_1) , etc.

The average trajectory is the average of all possible trajectories, weighted by the probability of each trajectory. Instead of running all possible trajectories and averaging them, you can just solve the average trajectory using the replicator dynamics equation with the average fitness.

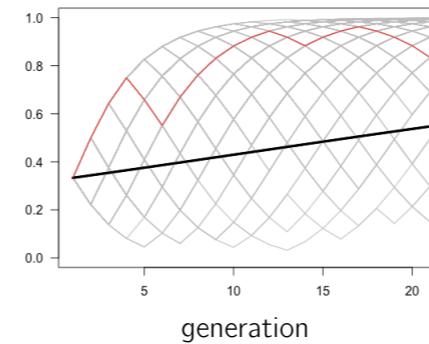
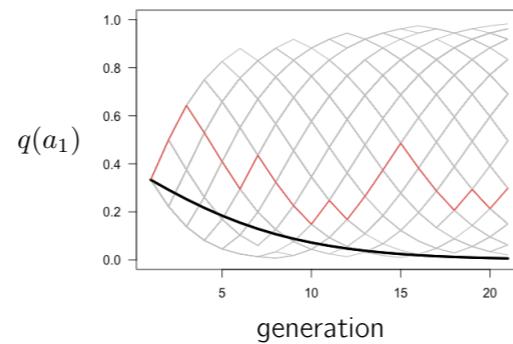
Note: the average trajectory does not correspond to an achievable evolutionary history (i.e. it isn't restricted to steps on the lattice), but it's still the best estimate. If we ran our simulations in an infinite number of parallel universes, the average trajectory would be our best guess for where any particular trajectory is at generation g , but no single trajectory would actually line up with our best guess.

Average fitness

Average fitness depends on base rate of each environment

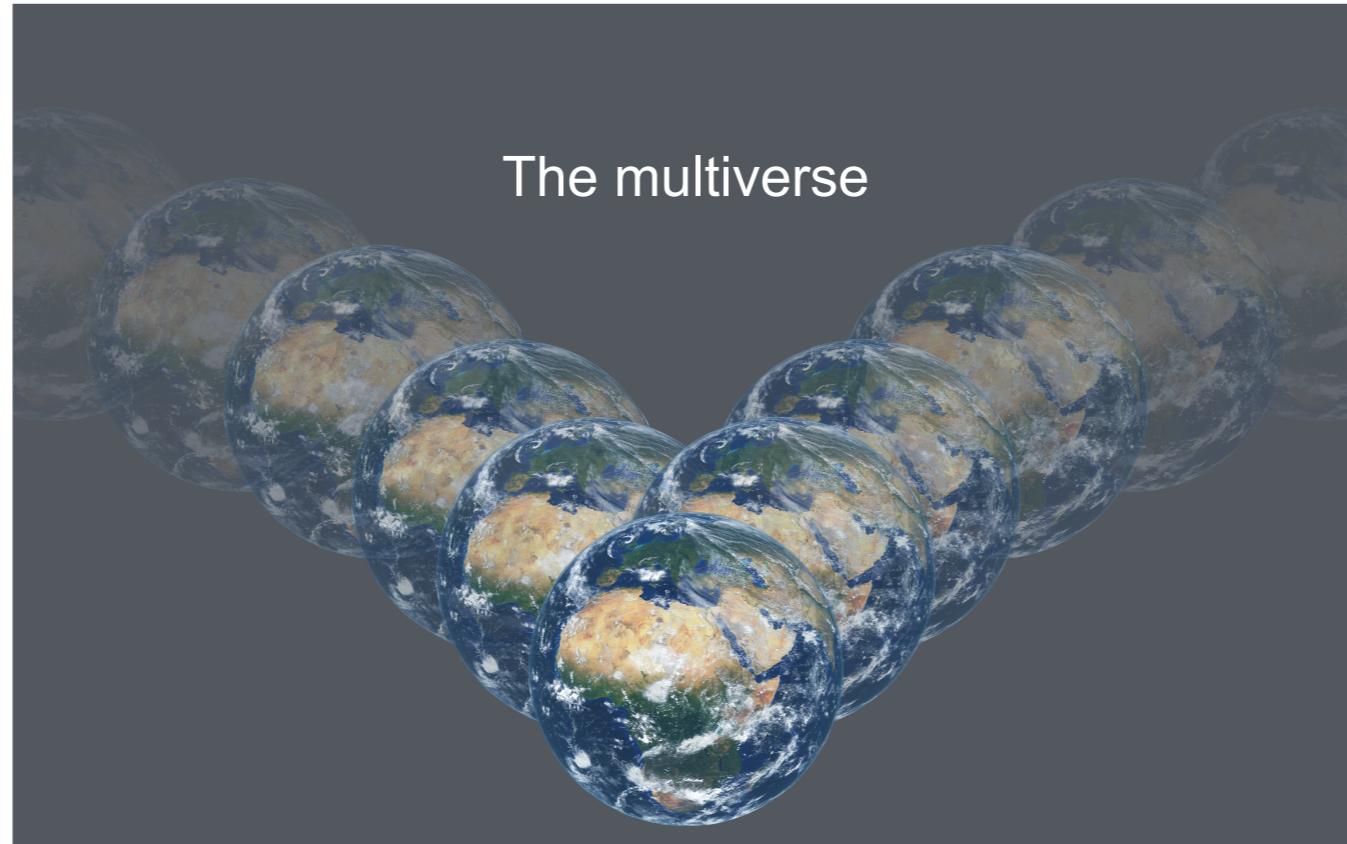
	a_1	a_2	a_3
environment 1	4	6	8
environment 2	3	2	1
average fitness	3.5	4	4.5

	a_1	a_2	a_3
environment 1	4	6	8
environment 2	3	2	1
average fitness	3.3	3.2	3.1



Here's the same plot for $p(e1) = 0.3$ and $p(e2) = 0.7$

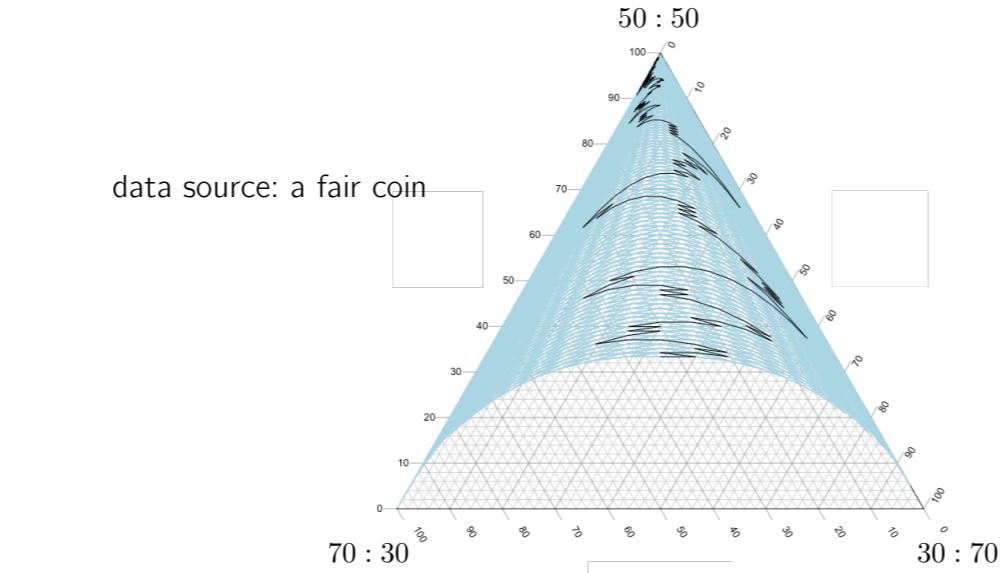
Variant a_1 eventually takes over the population. And it does so at a much slower rate than a_3 did in the scenario on the left, because the differential fitness is lower in the scenario on the right.



Here I'll just reiterate the usefulness of solving the average fitness for all possible permutations of environmental histories: it gives us our best estimate and simplest general characterization of the evolutionary process we're studying.

Average fitness

Multiverse lattice for our original Bayesian updating example



The blue lines are the lattice of all possible learning trajectories a Bayesian updater can take when receiving 100 flips of a fair coin.

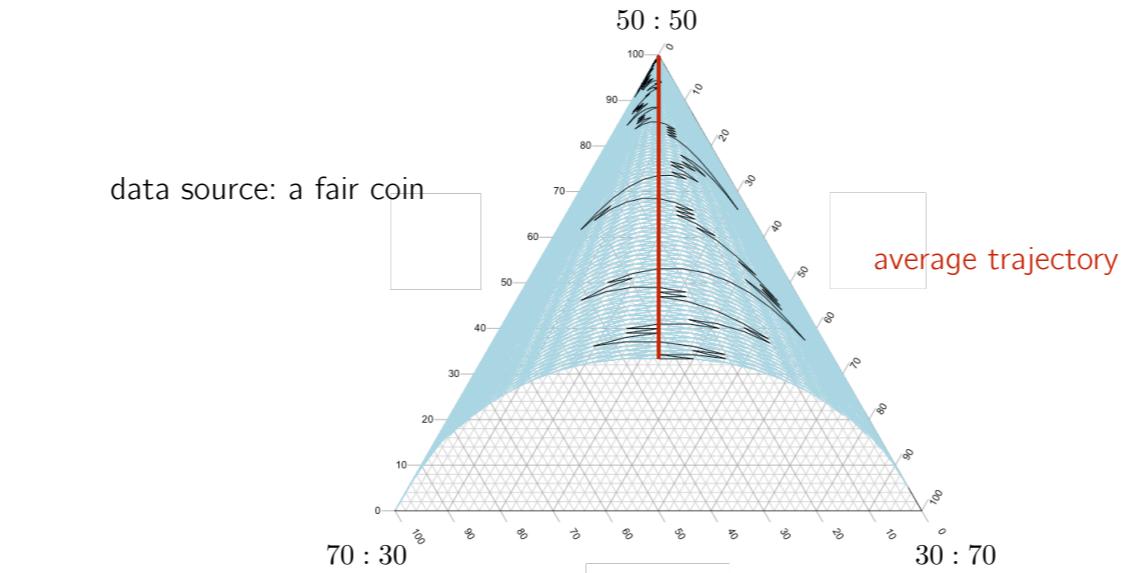
This lattice was solved by iterating Bayes rule on each permutation of heads and tails for 100 flips of a fair coin.
There are $2^{10} = 1024$ permutations.

Note:

The area not covered by the lattice is a "no go" zone of posterior probabilities that a Bayesian (i.e. rational) agent would never infer (on the basis of this data source and the initial prior).

Average fitness

Multiverse lattice for our original Bayesian updating example



The black line is the trajectory from one simulation. The red line is the average of all trajectories.

Main point: the fitness of ideas for an entire source that generates data can be summed up as the average fitness of ideas given the source. The average fitness corresponds to an average trajectory.

Another cool thing:

We can also look at the area of the evolutionary state space that is covered by different ideas (or in their basins of attraction). I'm currently working on the evolutionary dynamics of generalist vs specialist ideas. Generalist ideas correspond to a larger, flatter area of the evolutionary state space and specialist ideas correspond to a smaller, peakier area of the evolutionary state space. Here, the 50:50 hypothesis is more of a generalist idea and the 70:30 hypothesis is more of a specialist idea, because the binomial distribution over all possible data types given $p=0.5$ has higher variance, $Np(1-p) = 2.5$, than for $p=0.7$, with a variance of $Np(1-p) = 2.1$. More on this in another talk.

Voting model

Different types of voting systems:

- 1) Each voter casts as many votes as they want toward each candidate

	a_1	a_2	a_3
voter type 1	4	6	8
voter type 2	3	2	1
average support	3.5	4	4.5

- 2) Every voter gets one vote to divy up among the candidates

	a_1	a_2	a_3
voter type 1	.22	.33	.44
voter type 2	.50	.33	.17
average support	.29	.33	.38

- 3) Voters must divy up their votes so that all votes for each candidate sum to one.

	a_1	a_2	a_3
voter type 1	.5	.7	.3
voter type 2	.5	.3	.7
total support	1	1	1

this is a pathologically weird voting system

So far we've covered average fitness as in #1. But there are many different ways you could represent how environments or data "support" different organisms or ideas.

Let's use a voting analogy, where different environments vote for different organism types (i.e. candidates). And then the organisms reproduce on the basis of the differential support they received.

There are many rules we could put in place to constrain how the votes are distributed. Here are three.

Voting model

and it's how fitness works in Bayesian inference

	h_1	h_2	h_3
<i>data type 1 (heads)</i>	.5	.7	.3
<i>data type 2 (tails)</i>	.5	.3	.7
total support	1	1	1

stay tuned for more on this later...

- 3) Voters must divy up their votes so that all votes for each candidate sum to one.

	a_1	a_2	a_3
<i>voter type 1</i>	.5	.7	.3
<i>voter type 2</i>	.5	.3	.7
total support	1	1	1

this is a pathologically weird voting system

If we commit ourself to this particular constraint on how fitness works in Bayesian inference, then this constitutes a subset of all possible evolutionary dynamics. These particular dynamics will characterize "how ideas evolve" under Bayesian learning.

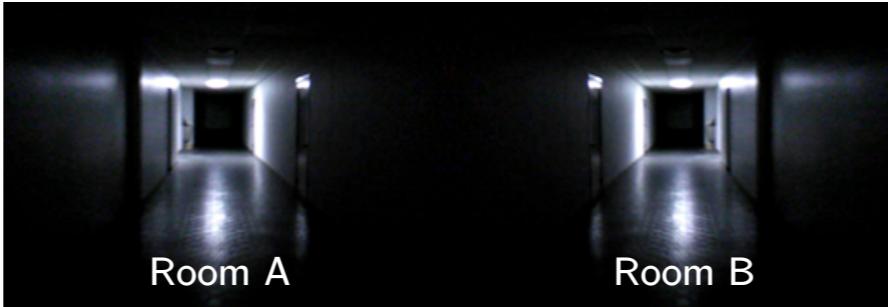
As Amy pointed out: this looks like Wolpert's "no free lunch" theorem: no idea (or model) can do better than any other idea "overall".

But in the evolution biological variants, it's quite common for one variant to do better than others overall. That's because we don't live in a set of environments structured in such a way to serve all variants equally well, in the aggregate. :(

Can ideas really have fitness?

The fitness of ideas

Creepy highschool auditorium thought experiment



T H H T T H T H T... H H H H H H H H H H...

Do you feel comfortable saying the fitness of the idea that the coin is **fair** is higher in the environment of Room A than in Room B?

You friend's kid asks you to come see their high school art program's exit show. You go.

In Room A, there's someone seated flipping a fair coin. A large audience is watching this.

In Room B, there's someone else seated flipping a double-headed coin. A large audience is watching this.

Each audience member's head contains a population of ideas – maybe you feel comfortable saying that. And each room contains a population of ideas – maybe you feel comfortable saying that. Do you feel comfortable saying the fitness of the "fair coin" idea is higher in the environment of Room A than Room B? Aka, can we actually talk about the fitness of ideas?

Limitations

The equivalence between replicator dynamics and Bayesian learning is all about the data.

What are other ways ideas gain support, aside from just via data?

Tons of stuff could contribute to the average, differential growth rates of ideas. We could add more stuff into the fitness of ideas than just their support under data. Get thinkin bout that.

References

- Ferdinand, V. (2018). The coevolution of data and hypotheses in Bayesian cultural evolution. *Proceedings of the 12th International Conference on the Evolution of Language*.
- Harper, M. (2009). The replicator equation as an inference dynamic. *arXiv preprint arXiv:0911.1763*.
- Hofbauer, J., & Sigmund, K. (1998). *Evolutionary games and population dynamics*. Cambridge university press.
- Krakauer, D. C. (2011). Darwinian demons, evolutionary complexity, and information maximization. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 21(3), 037110. [See Section: Isomorphism among dynamics of adaptive plasticity and the demon](#)
- Mustonen, V., & Lässig, M. (2010). Fitness flux and ubiquity of adaptive evolution. *Proceedings of the National Academy of Sciences*, 107(9), 4248-4253.
- Nowak, M. A. (2006). *Evolutionary dynamics: exploring the equations of life*. Harvard university press.
- Shalizi, C. R. (2009). Dynamics of Bayesian updating with dependent data and misspecified models. *Electronic Journal of Statistics*, 3, 1039-1074. [See Appendix A: Bayesian updating as replicator dynamics](#)
- Sober, E. (2014). *The nature of selection: Evolutionary theory in philosophical focus*. University of Chicago Press.
- Suchow, J. W., Bourgin, D. D., & Griffiths, T. L. (2017). Evolution in mind: Evolutionary dynamics, cognitive processes, and bayesian inference. *Trends in cognitive sciences*, 21(7), 522-530.
- Wolpert, D. H. (1996). The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7), 1341-1390.