https://a16z.com/2019/05/09/data-network-effects-moats/

The Empty Promise of Data Moats

by Martin Casado and Peter Lauten

Data has long been lauded as a competitive moat for companies, and that narrative's been further hyped with the recent wave of AI startups. Network effects have been similarly promoted as a defensible force in building software businesses. So of course, we constantly hear about the combination of the two: "data network effects" (heck, we've talked about them at length ourselves).

But for enterprise startups — which is where we focus — we now wonder if there's practical evidence of data network effects at all. Moreover, we suspect that even the more straightforward data *scale* effect has limited value as a defensive strategy for many companies. This isn't just an academic question: It has important implications for where founders invest their time and resources. If you're a startup that assumes the data you're collecting equals a durable moat, then you might underinvest in the other areas that actually *do* increase the defensibility of your business long term (verticalization, go-to-market dominance, post-sales account control, the winning brand, etc).

Treating data as a magical moat can misdirect founders from focusing on what's really needed to win

In other words, treating data as a magical moat can misdirect founders from focusing on what is really needed to win. So, do data network effects exist? How might a scale effect behave differently from the traditional network effect? And once we get past the hype of having to have them… how can startups establish more *durable* data moats — or at least figure out where data best plays into their strategy?

Data + network effects ≠ data network effects

Broadly defined, a "network" is at play when a system of users/customers/endpoints/etc. are structurally arranged in a network. In our context, such networks are often built around a technology, product, or service supporting the network structure, whether constructed around engagement features (e.g., social networks) and/or protocols (e.g., Ethernet, email, cryptocurrencies).

Network *effects* occur when the value of participating in a network goes up for the participants as more nodes come on to the network, or as engagement increases between existing nodes. Imagine trying to have a one-way phone conversation or call only five people in the world and no one else; the telephone system became more valuable as more users joined the network. Other common, more modern examples of network effects may include social networks, online marketplaces, and cryptonetworks.

Systems with network effects generally have the property of *direct* interactions between the nodes over a defined interface or protocol. Joining the network requires conforming to some standard, which increases direct interaction for all nodes and makes those interactions increasingly stickier. But when it comes to the popular narrative around *data* network effects, we don't often see the same sticky, direct interaction play out (let alone mechanical interdependencies between nodes due to protocols or interfaces).

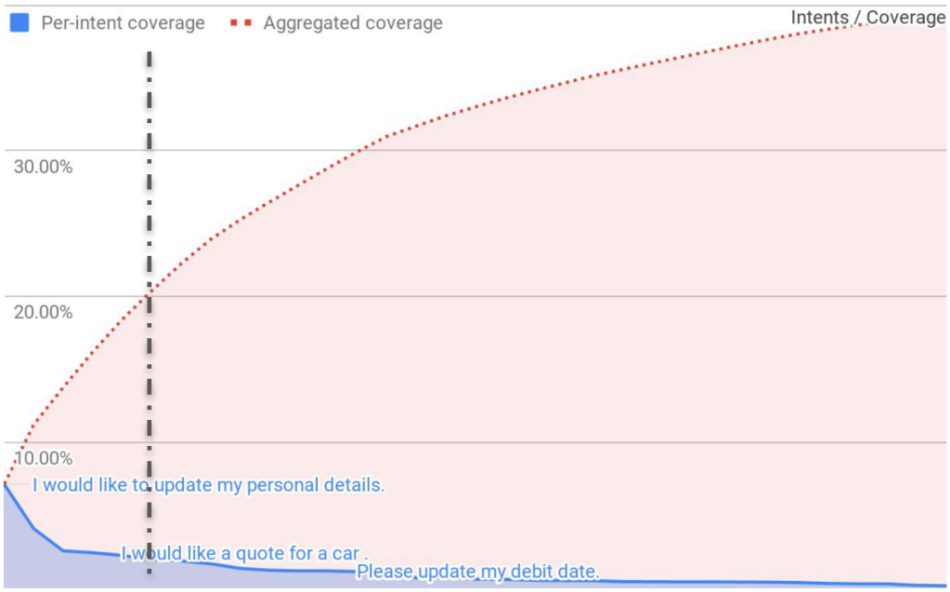
There generally isn't an inherent network effect that comes from merely having more data.

Most data network effects are really scale effects

Most discussions around data defensibility actually boil down to scale effects, a dynamic that fits a looser definition of network effects in which there is no direct interaction between nodes. For instance, the Netflix recommendation engine can predict that you're likely to enjoy show *Y* if most of the viewers of your favorite film *X* also tend to watch show *Y*, even though those users don't directly interact with each other. More data means better recommendations, which means more customers, and even more data… the famous "flywheel".

Yet even with scale effects, our observation is that data is rarely a strong enough moat. Unlike traditional economies of scale, where the economics of fixed, upfront investment can get increasingly favorable with scale over time, the exact *opposite* dynamic often plays out with data scale effects: The cost of adding unique data to your corpus may actually go up, while the value of incremental data goes down!

Take the case of a company using a chat bot to respond to customer support inquiries. As you can see from the graph below, creating an initial corpus from customer support transcripts is likely to provide answers to simple inquiries ("Where is my package?"). But the vast majority of inquiries are far messier, many of which are only ever asked once ("Where is that thing that I've been waiting to arrive on my front door step?"). So in this limiting case, collecting useful inquiries becomes more difficult over time. And, after 40% of the queries have been collected in this case, there is actually no advantage to collecting more data at all!



The above graph came from a study (shared with permission) by Arun Chaganty of Eloquent Labs, for questions submitted to a chat bot in the customer support space. In it, he finds that 20% of the effort into the data distribution tends to only get you around 20% coverage of use cases. Beyond that point, the data curve not only has diminishing marginal value, but is increasingly expensive to capture and clean. Also notice that the distribution approaches an asymptote of 40% intent coverage, demonstrating the extent to which it's difficult to automate all conversations depending on the context.

Of course, the point beyond which the data scale effect diminishes varies by domain. But regardless of exactly when this happens, the ultimate outcome is often the same: the ability to stay ahead of the pack tends to slow down, not speed up, with data scale. Instead of getting stronger, the defensible moat erodes as the data corpus grows and the competition races to catch up.

Instead of getting stronger, the data moat erodes as the corpus grows

The point of this is not to make a categorical statement about the utility of data as a defensive moat — our point is that *defensibility is not inherent to data itself*. And unless you understand the lifecycle of the data journey for your target domain, you're not guaranteed defensibility; the following framework may help.

A practical framework for understanding the data journey

Minimum Viable Corpus

When most people talk about network effects, they focus on overcoming the bootstrapping or cold-start problem (colloquially called the "chicken-egg" problem) of acquiring enough early nodes to make the network useful for all nodes (and make the economics of the business competitive). The bootstrapping problem is hard to solve in most network effects businesses, particularly when you need the network already up and running to attract volume.

But this isn't necessarily true for many enterprise businesses with a data scale effect. Bootstrapping what we think of as the "minimum viable corpus*"* is sufficient to start training against, and is the first inflection point along a startup's data journey. This initial corpus can come from a variety of sources: automating data capture from available sources, such as crawling the web; getting early users to trade their data for something in return; repurposing data from other domains through transfer learning; and even synthetically generating data, where you programmatically create data to train against.

Early in the data journey, getting to the minimum viable corpus requires relatively low investment and will clearly not be a durable moat.

Data Acquisition Cost

In a given corpus, getting the next piece of data tends to become *more* expensive to capture over time. Unique data that brings new signal to your corpus may be harder to find in the noise, is more of a hassle to secure, and takes longer to cleanly label over time. This is true in many domains that rely on so-called "data network effects".

With traditional network effects, on the other hand, user acquisition costs go down over time, because the value of joining the network increases. Further, with traditional network effects there also tends to be an accompanying, more inherent virality where nodes are incented to grow the network themselves and therefore propagate to add more value to the network. Neither of these properties apply to data effects: costs of data go up.

Incremental Data Value

As you gather data, the data also tends to become *less* valuable to add to the corpus. Why? Even if the new arbitrary batch of data has the same cost to collect as the last batch acquired, it yields less value given some of the new data you acquire already overlaps with your existing corpus. And this only gets worse over time: Benefits of new data go down.

In most of the startups we've seen, new data early on applies to the entire customer base. But beyond a certain point — such as the asymptote in the example graph above — new data collected will only apply to the small subsets that lie in the "long tail" of special use cases. As such, any data scale effect moats also become less valuable as the data set gets expanded.

Data Freshness

This point may seem obvious but can't be emphasized enough: In many real-world use cases, data goes stale over time… it is no longer relevant. Streets change, temperatures change, attitudes change, and so on.

Not just that, but any proprietary insight many data startups have initially weakens over time because the value of data decreases as more people collect it: Your prediction edge erodes as competitors chase you in the same domains. And the amount of work required just to keep an existing corpus fresh over time — let alone ahead of the pack — increases with scale.

Data, in this sense, is like a commodity.

When IS data defensible, and what can you do to manage this?

None of this is to suggest data is pointless! But it does need more thoughtful consideration than leaping from "we have lots of data" to "therefore we have long-term defensibility". Because data moats clearly don't last (or automatically happen) through data collection alone, carefully thinking about the strategies that map onto the data journey can help you compete with — and more intentionally and proactively keep up with — a data advantage. It's way better to plan for it than being blindsided when an asymptote or point of diminishing returns suddenly hits your company.

Data effects need more thoughtful consideration than leaping from 'we have lots of data' to 'therefore we have long-term defensibility'

Bootstrap the initial corpus to compete with incumbents

Bootstrapping data is not so difficult in some domains, as described earlier. Yet founders can actually use this to their advantage to go head to head with incumbents that have data, but fail to apply it properly. After bootstrapping into a minimum viable corpus, startups with a head start on building out the right dataset can use that know-how to accelerate ahead of incumbent competition before those incumbents figure out how to make sense of the data.

Generating synthetic data is another approach to catch up with incumbents housing large tracks of data. We know of a startup that produced synthetic data to train their systems in the enterprise automation space; as a result, a team with only a handful of engineers was able to bootstrap their minimum viable corpus. That team ultimately beat two massive incumbents relying on their existing data corpuses collected over decades at global scale, neither of which was well-suited for the problem at hand.

Know the distribution of the data

Having a sharp understanding of the distribution of the data corpus will inform your data strategy, and how much defensibility you can actually create, depending on the application space.

The distribution of data, and its corresponding value, varies significantly by domain. As such, it's critical to intimately understand the shape of the distribution, and to craft the right strategy to capture it. Is there a fat tail of critical data that's hard to acquire? If so, what's the plan to scale the corpus into the long tail? How important is accuracy in your domain? What error rate is acceptable — it's not the end of the world if machine learning predicts the wrong auto-complete in an email to a colleague, but inaccurate object classification in the world of autonomous vehicles can, quite literally, be a matter of life and death. Misunderstanding the data distribution

might even be hard to detect if not looking closely, for instance if weights aren't properly applied over time-series data (e.g., see "catastrophic forgetting").

The challenge we shared earlier — that so much of the learnings in many domains are in the long tail of exceptional use cases — can also be an advantage if you're a first mover. This is particularly true for enterprise companies that embed these learnings into the product *and* the sales process. While some investors aren't drawn to the grit of wading through complex markets because they only see the difficulties on scale and margins, we believe that earning your way into a complex market creates scar tissue that is itself defensible.

Understand the extent to which data improves your product

In some domains, having more data results in a dramatically better product. So much so that it will overcome the increasing overhead and diminishing value of data over time. For instance, if you have a cancer screen that is 85% accurate, it is far more likely to get used than one which is 80% accurate. That use will provide additional data, which could in turn improve accuracy.

While we haven't seen many of these effects play out in practice, there are a few instances where a data advantage could create a winner-take-all style advantage in the product that is clearly the foundation for a strong moat.

Of course, understanding the extent to which data contributes to a product is not always straightforward. Often choice of algorithms or other product-feature tuning has a far greater impact than having more data alone.

Weigh the tradeoff between quality and quantity

One of the trickiest tradeoffs in nurturing a data corpus is how to balance quality versus quantity. Why is there a tradeoff? Solving too much for scale can result in okay estimates across a broad range of use cases, but not great estimates for any one of them. Solving too little for scale can result in a corpus that is well-equipped to solve a narrow problem, but underdelivers on the entire set of use cases your customers expect.

In practice, this may mean focusing more effort on labeling rich data for a narrow use case, or opening the aperture more broadly to data that is useful across far more use cases. Clearly, both depth and breadth are critical properties of any corpus, but getting the balance wrong in either direction can severely impact performance. When it comes to maintaining an edge on the competition, staying on top of the quality / quantity tradeoff for your specific domain will enable you to maximize the value of incremental data added to a data moat.

Secure proprietary data sources

The question we're posing throughout this post — and that we want founders to ask themselves — is *where* does the data scale effect really exist, and how long will it last? That doesn't mean that there isn't practical defensibility a company can get from proprietary data; there is clearly a long list of industries (e.g., pharma) and counterexamples that have dominated their markets for

decades, particularly when they have access to proprietary data sets for industry structure reasons (e.g., Equifax, LexisNexis, Experian, etc.).

Accumulating proprietary data is a defensible strategy that is strongest when the sources are scanty or are reticent to provide data to more than one vendor (such as government buyers). As the bar for security requirements and compliance standards rises to an all-time high, surviving vendor scrutiny to get access to sensitive data can itself be a moat against competitors.

Even taking on all the upfront costs to assemble, clean, and standardize big pools of public datasets can create a scale effect that emerging competitors will have to recreate from the ground up. Especially in cases where specialized know-how is essential to find, understand, and clean the data in the first place. Startups that prove to be responsible data custodians can earn trust from their customers, who will then share increasingly sensitive data only with them, creating a moat.

Wither data moats…

Data is fundamental to many software companies' product strategies, and there are ways it can contribute to defensibility — but don't rely on it as a magic wand. Most of the narrative around data network effects is really around data scale effects, and as we've outlined in this post, those sometimes have the opposite effect if not planned correctly. But don't even assume you have a data network effect (you likely don't), or that the data scale effect will last in perpetuity (it almost certainly won't).

Instead, we encourage startups to think more holistically about defensibility. Greater long-term defensibility is more likely to come from packaging differentiated technology; understanding the domain and reflecting that in your product as you verticalize across industries; dominating the go-to-market race; and winning the talent war to build a world-class team. These efforts will pay off in defending and winning in the markets far more than data alone.