The Changing Economics of Knowledge Production

Simona Abis and Laura Veldkamp
Columbia University, NBER and CEPR*

March 15, 2022

Abstract

Big data technologies change the way in which data and human labor combine to create knowledge. Is this a modest technological advance or a data revolution? Using hiring and wage data, we show how to estimate firms' data stocks and the shape of their knowledge production functions. Knowing how much production functions have changed informs us about the likely long-run changes in output, in factor shares, and in the distribution of income, due to the new, big data technologies. Using data from the investment management industry, our structural estimates predict that the labor share of income in knowledge work may fall by 5%. The change associated with big data technologies is similar in magnitude to estimates of the change brought on by the industrial revolution.

Machine learning, artificial intelligence (AI), or big data all refer to new technologies that reduce the role of human judgment in producing usable knowledge. Is this an incremental improvement in existing statistical techniques or a transformative innovation? The nature of this technological shift is similar to industrialization: In the 19th and 20th centuries, industrialization changed the capital-labor ratio, allowing humans to use more machines, factories and sophisticated tools to be more efficient producers of goods and services. Today, machine learning is changing the data-labor ratio, allowing each knowledge worker to leverage

*Email: sa3518@columbia.edu or lv2405@columbia.edu. Columbia Business School, 3022 Broadway, New York, NY 10025. We are grateful to PayScale for use of their wage data. Thanks to Anastassia Fedyk, Ellen McGrattan and the participants of the NBER economic growth and Economics of AI groups, GSU-RFS FinTech Conference, SED, Women in Finance conference, UNC Junior Roundtable and Columbia macro and finance lunches and seminars at the IMF, Princeton, HEC, UVA, CUHK for comments. Thanks also to many graduate students, including Sahil Arora, Sagar Agarwal, Samrat Halder, Anshuman Ramachandran, Ran Liu and Saad Naqvi, Liushiya Chen, Jiashu Sun who substantially advanced the project with their research assistance. We acknowledge the generous financial support and helpful feedback from IBM and the Columbia University Data Science Institute. Keywords: FinTech, production function estimation, machine learning, artificial intelligence, big data, labor share.

more data, to be a more efficient producer of knowledge. Given the myriad of differences between the industrialization era and today's knowledge economy, and the early stage of data technology adoption, how might one compare the magnitude of today's change with its historical counterpart? Economists model industrialization as a change in production technology: a move from a technology with starkly diminishing returns to capital, to one with less diminishing returns. The size of the industrial revolution can therefore be summarized by the magnitude of the change in the production parameter that governs diminishing returns. That same statistic can be estimated for knowledge production, using old and new data technologies. Measuring how much big data technology adoption changes the diminishing returns to data and comparing this to the change that took place during the industrial revolution informs us about whether this is a useful, but common innovation, or the next economic revolution.

The finance industry is a particularly useful laboratory for studying trends in knowledge production because it is an early adopter of new, big-data technologies. Using labor market data from the investment management sector, we estimate two production functions – one for classical data analysis and one for machine learning. The decline in diminishing returns to data shows up as an exponent on data in the production function that is closer to one: We estimate that the data exponent rose by 0.05. This increase in the parameter governing diminishing returns implies that knowledge-producing firms should optimally have more data per worker, or equivalently, fewer workers for a given size data set. This change in production function exponents also affects the distribution of income. It predicts an 5% decline in the share of firm revenue paid to labor. Such a change in the profit share could matter for income inequality. The flip side of the declining labor share is an 5% increase in the share of knowledge revenue paid to data owners. In other words, new data technologies structurally increase the value of data as an asset and enrich those who own the data. Finally, the magnitude of these shifts represents a change in production that is similar to the lower end of historians' estimates of the change experienced during the industrial revolution.

Estimating old and new knowledge production functions is challenging, because for most firms, we do not know how much data they have, nor how much knowledge they create, nor do they announce which technology or what mix of technologies they employ. What we can observe is hiring, skill requirements and wages. A structural model of a two-layer production economy allows us to infer the rest. The two layers of production are as follows: Raw data is turned into usable, processed data (sometimes called information) by data managers; processed data and data analyst labor combine to produce knowledge. Thus, we use hiring of data managers to estimate the size of the firm's data stock, the skills mix of analysts to estimate the mix of data technologies at work, and we bypass the need to measure knowledge by using wage data to construct income shares, which inform us about the returns, and the rate of diminishing returns, to each factor.

To estimate production functions, it is imperative that we precisely categorize job postings and match postings by employer. Unlike other work that measures machine-learning-related employment (e.g., Acemoglu and Restrepo (2018)), our work demands a finer partition of jobs. We need to distinguish between workers that prepare data to be machine-analyzed, workers that primarily use machine learning, and workers that use statistical skills that are of a previous vintage. We also need to know whether data managers are being hired by the same firm that is also hiring machine-learning analysts.

Because different industries have different job vocabularies, we can categorize jobs more accurately by focusing on one industry: finance, more specifically we focus on investment management. Since investment management is primarily a knowledge industry, with no physical output, it is a useful setting in which to tease apart these various types of knowledge jobs. According to Webb (2019) and Brynjolfsson, Mitchell, and Rock (2018a), finance is also the industry with the greatest potential for artificial intelligence labor substitution. We use Burning Glass hiring data, including the textual descriptions of each job, to isolate financial analysis jobs that do and do not predominantly use machine learning, as well as data management jobs, for each company that hires financial analysts. We adjust the number of job postings by a probability of job filling, to measure a company's desired addition to their labor force. This series of worker additions, along with job separations by job category, enables us to build up a measure of each firm's labor stock.

The next challenge is to estimate the amount of data each firm has. Data management work is a form of costly investment in building and maintaining data sets. Therefore, we use the job postings for data managers, the job filling and separation rates for such jobs, and an estimate of the initial data stock to construct data inflows (investments), per firm, each year. To estimate the 2015 initial stock of data of each investment management firm, we estimate which stock best rationalizes the firm's subsequent hiring choices. Combining this initial stock, with a data depreciation rate and a data inflows series allows us to cumulate up the data stock that every investment management firm has in its data warehouse.

Armed with data stocks, labor forces in each category, and wages from PayScale, we estimate the data and labor income shares. These income shares correspond to the exponents in a Cobb-Douglas production function. We estimate a constant-returns Cobb-Douglas specification because we are exploring the analogy that AI is like industrialization and this is the type of production function most often used to describe industrial output. Therefore, we model knowledge production in a parallel way to industrialization, to facilitate comparison, while recognizing the non-rival nature of data. By comparing the estimated exponent for classical data analysis and machine-learning data analysis, we can assess the magnitude of the technological change.

This approach bypasses two forces: The role of capital and the potential for increasing returns. Typically, knowledge is combined with capital, real or financial, to generate profits, in a production function that often exhibits increasing returns. The start of Section 1 shows how we could incorporate capital or increasing returns in our model of firm profits, without changing how we estimate the production of knowledge. As long as there exists some amount of knowledge that produces $1 in profit, at each point in time, we can estimate how data and labor combine to create that amount of knowledge, without taking a stand on how that knowledge will be used to produce profit.

Our data reveals a steady shift underway in the employment of knowledge workers in the investment management sector. We see a steady increase in the fraction of the workforce skilled in new big data technologies. However, while the declining labor share might lead one to expect fewer knowledge workers, we find an increase in the size of the sector large enough so that even though the share shrinks, the number of workers and their pay rises. Even for workers with the old skills, jobs are still abundant. The number of old technology jobs in the sector has not fallen; it simply represents a smaller share of employment. While AI job postings were a tiny fraction of all analysis jobs through 2015, by the end of 2018, about $1/7^{th}$ of all financial analysts in investment management firms had big data or AI-related skills. This shifts we measure are just the first few years of adoption of this new technology. But they indicate the direction of a transformation that we expect to continue for years to come.

Related Literature Our work complements the voluminous literature on information in financial markets, which equates data or signals and knowledge. It is as if, upon obtaining a data set, all the insights from that data were immediately and costlessly revealed. That simplification is appropriate for the questions these papers address. Measuring the changing relationship between data and knowledge does not prove, disprove or compete with any ideas about what effects such knowledge has on investors or markets.[1] Our results add to our understanding of where financial knowledge comes from, help us understand the extent of the technological change we are experiencing, and matter for other questions such as the labor share of income in the financial sector. In our discussion of assumptions after the model setup, we describe how to incorporate some of the equilibrium effects from this literature in our single-firm choice problem.

[1] Papers that explore the consequences of financial knowledge include Goldstein, Ozdenoren, and Yuan (2013), Edmans, Goldstein, and Jiang (2015), Goldstein and Yang (2019), Dugast and Foucault (2020) and Davila and Parlatore (2020).

Our approach relates closely to work that structurally estimates firm production models.[2] The most similar papers in this line of work use structural estimates to infer the value of intangible assets (Eisfeldt and Papanikolaou, 2013; Peters and Taylor, 2017; Belo, Gala, Salomao, and Vitorino, 2021). Another approach uses Q theory to infer intangibles from asset prices and book values (Crouzet and Eberly, 2020; McGrattan, 2020). Our exercise is a complement to work that estimates the value of intangible assets, which is a broad set of assets that includes data. However, the difference in our main question necessitates a different approach. The objective of previous work was decomposing the sources of value in a firm. We are interested in how much two technologies, often both used within the same firm, differ. A hiring-based approach is more suitable for our question because we can identify workers using one technology or another. We cannot tell what firm value is attached to each mode of production, within the same firm. It is the skills required in posted jobs that reveal what technologies the firm is using.

Models of the role of data in the process of economic growth [3] share our model-based approach but also equate data and knowledge. The shifts in production we measure are also related to the long-run changes in the labor share of income documented by (Karabarbounis and Neiman, 2014).

On the topic of big data technologies, many recent working papers use labor market data to investigate how machine learning and artificial intelligence are affecting labor demand. They primarily use a difference-in-difference approach. [4]

In our approach, the number of jobs gained or lost due to machine learning is an important piece of evidence; it informs our work. But labor demand is not our main question. It is just one input into our analysis. Because our focus is on how AI technology affects knowledge production, we need to use a structural, production-function approach. Kogan, Papanikolaou, Schmidt, and Song (2021) shares this focus on technology and employment, but uses a measurement approach more similar to ours.

In what follows, Section 1 sets up a three-equation model that is the basis of our estimation and derives optimality conditions that we estimate to infer its parameters. Section 2 describes the data and how we construct variables that correspond to objects in the model. Section 3

[2]See Belo, Lin, and Bazdresch (2014); Belo, Li, Lin, and Zhao (2017); Kung and Schmid (2015).

[3]Jones and Tonetti (2020); Agrawal, McHale, and Oettl (2018); Aghion, Jones, and Jones (2017); Farboodi and Veldkamp (2019)

[4]Babina, Fedyk, He, and Hodson (2020) examine how firms use AI skills in the production process. Acemoglu and Restrepo (2018), and Deming and Noray (2018) identify industries, firms or regions that are more exposed to machine learning-related technology. Cockburn, Henderson, and Stern (2018); Alekseeva, Azar, Gine, Samila, and Taska (2020) report numbers of AI jobs postings or patents by industry and occupation. Others examine the productivity gains or potential discrimination costs that follow the adoption of AI techniques in providing credit (Fuster, Goldsmith-Pinkham, Ramadorai, and Walther, 2018), in equity analysis (Grennan and Michaely, 2018), or in deep learning more generally (Brynjolfsson, Rock, and Syverson, 2017).

presents the estimation results and describes the features of our data that inform our estimated parameters. We also estimate the value of firms' data stocks. Section 4 concludes.

1 A Model for Measurement

The objective in writing down this model is not to provide insight into new economic mechanisms, nor it is to provide the most realistic, detailed description of financial knowledge production. Rather, the goal is to write down a simple framework that maps objects we observe into those that we want to measure. It needs to relate hiring to labor as well as quantities and prices of labor to data stocks and knowledge production. There are three types of workers: AI (artificial intelligence) analysts, old technology (OT) analysts, and data managers. We use AI as a shorthand to denote a diverse array of big data technologies. The data managers create structured data sets, which, along with labor, are the inputs into knowledge production. Among data managers we also include workers who select, purchase and integrate externally produced data sets into the firm's databases. We define as data (D) only information that is readily available for analysis. This production process is illustrated in Figure 1.

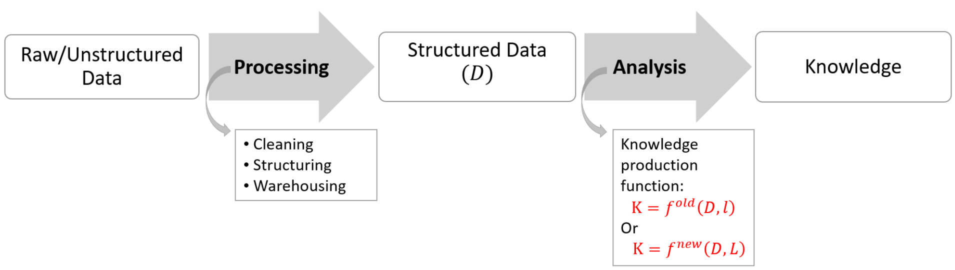

Figure 1: Production process for knowledge

The new technology knowledge production function is:

$$K_{it}^{AI} = A_t^{AI} a_i^{AI} D_{it}^{\alpha} L_{it}^{1-\alpha}, \tag{1}$$

where D_{it} is structured data, L_{it} is labor input for data analysts with machine-learning skills, and K_{it}^{AI} is the knowledge generated using the new technology. The old technology knowledge production function is:

$$K_{it}^{OT} = A_t^{OT} a_i^{OT} D_{it}^{\gamma} l_{it}^{1-\gamma}, \tag{2}$$

where l_{it} is labor input for data analysts with traditional analysis skills, K_{it}^{OT} is the knowledge generated using the old technology. A_t^{AI} and A_t^{OT} are time-varying productivity parameters;

while a_i^{AI} and a_i^{OT} are firm-specific productivity parameters.

We use a Cobb-Douglas production function for knowledge because it offers a clear mapping between incomes shares and the production function parameters and it facilitates our comparison between new data technologies and the changes induced by industrialization. A Cobb-Douglas approach is also supported by Jones (2005). Our specification does embody the non-rival nature of data: both technologies make use of the same data set, at the same time.

Data management and Data Stocks. Data inputs for analysis are not raw data. They need to be structured, cleaned and machine-readable. This requires labor. Suppose that structured data, sometimes referred to as "information," is produced according to $\lambda_{it}^{1-\phi}$, where λ_{it} is labor input for data managers.[5] Labor with diminishing marginal returns can turn raw or purchased data into an integrated, searchable data source that the firm can use. New processed data is added to the existing stock of processed data. But data also depreciates at rate δ. Overall, processed data follows the dynamics below:

$$D_{i(t+1)} = (1-\delta)D_{it} + \lambda_{it}^{1-\phi} = D_{i0}(1-\delta)^{t+1} + \sum_{s=0}^{t}(1-\delta)^{t-s}\lambda_{is}^{1-\phi}. \tag{3}$$

If we estimate the rate of diminishing returns to data management labor λ_{it}, initial data D_{i0} and the depreciation rate δ, we can recover D_{it} from data management labor λ_{it}.

Equilibrium We are interested in a competitive market equilibrium where all firms choose the three types of labor to maximize firm value. We can express this problem recursively, with the firm's data stock as the state variable. In this equilibrium, each firm i solves the following optimization problem:

$$v(D_{it}) = \max_{\lambda_{it}, L_{it}, l_{it}} A_t^{AI} a_i^{AI} D_{it}^{\alpha} L_{it}^{1-\alpha} + A_t^{OT} a_i^{OT} D_{it}^{\gamma} l_{it}^{1-\gamma} - w_{L,t} L_{it} - w_{l,t} l_{it} - w_{\lambda,t}\lambda_{it} + \frac{1}{r} v(D_{i(t+1)}) \tag{4}$$

$$\text{where} \quad D_{i(t+1)} = (1-\delta)D_{it} + \lambda_{it}^{1-\phi}, \tag{5}$$

$v(D_{it})$ is the present discounted value of firm i's data stock at time t and $r > 1$ is the rate of time preference. Note that we have implicitly normalized the price of knowledge to 1. This is not restrictive because knowledge does not have any natural units. In a way, we are saying that one unit of knowledge is however much knowledge is worth \$1. Seen differently, our A

[5] One might be tempted to add a productivity term A^{DM} to the data production function. However, such a term would not be identified. The reason is that data does not have natural units. Multiplying production and initial data by a constant is just a change of units of data. Multiplying D_{it} by a constant simply creates a constant that can be included in A^{AI} and A^{OT}. So if we re-interpret those parameters as productivity, relative to the productivity of data production, the rest of the estimates are unchanged.

parameters measure a combination of productivity and price. We cannot disentangle the two and do not need to for our purposes.

Discussion of Model Assumptions *Increasing returns to knowledge.* Of course, one might object to our assumption of constant returns to scale, within each type of knowledge production. However, keep in mind that this is not different from what the growth literature does with idea production. Idea or technology production is typically produced using constant, or even diminishing returns. Then the ideas or technologies themselves enter into goods production in a way that creates increasing returns. In our setting, the analog to the increasing returns in growth models would be a final goods sector that produced with increasing returns to scale in knowledge, capital and labor: $(\text{final output}_{it}) = (K_{it}^{OT} + K_{it}^{ML})\,\text{capital}^{\zeta}\ \text{labor}^{1-\zeta}$. For our measurement exercise, we do not need to take a stand on this form of final goods production. But our exercise does not rule out increasing returns to knowledge.

Mapping knowledge into profits. The knowledge produced through the process we describe maps neatly into the informative signals that noisy rational expectations models give to their investors. One could write a larger model where the knowledge K_i represents the signal precision of investor i who will form an optimal portfolio, based on that information about future asset payoffs. In such a model, expected utility (risk-adjusted expected profit) is linear in K_i (Van Nieuwerburgh and Veldkamp, 2009). The objective in our problem is to maximize K_i. If instead that objective were maximize a linear coefficient times K_i, nothing would change.

Decreasing returns to information in financial markets. The value of information declines as others acquire more of it (Glode, Green, and Lowery, 2012). This implies that an advance in data technology would cause the mapping from knowledge K_i to risk-adjust profits to decline over time. Recall that units of knowledge are however much knowledge is worth one dollar to the firm. Therefore, a decline in the value of knowledge will be absorbed by the linear productivity multipliers A_t^{AI}, A_t^{OT}, which will re-scale K to represent however much knowledge is worth one dollar in the new period. Similarly, if knowledge produced by AI algorithms is more likely to be short-term, less precise or less valuable due to competition Dugast and Foucault (2018); Dessaint, Foucault, and Fresard (2021); Dugast and Foucault (2020), this will show up as a lower productivity A_t^{AI}. Although we do not identify these kinds of effects, our estimates incorporate them.

If information has decreasing social returns or externalities (Edmans, Goldstein, and Jiang, 2015; Goldstein and Yang, 2019), this is important for policy and welfare, but does not affect our estimation or our conclusion. The technological process for producing knowledge may be changing, and still that knowledge could be socially beneficial or socially costly.

Same type of data used for both technologies. Finally, this structure also implies that the nature of the data inputs is the same for both types of analysis. This simplifies measurement, but the obvious counterfactual would be: AI can make use of a broader array of data types than traditional analysis. One way to interpret this is that it is the source of greater decreasing returns to data from the old technology. Suppose that data is ordered, from easily usable to difficult to use. Once the easiest data is incorporated, the next additional piece of data for traditional analysis has very low marginal value. For AI, that next piece of data has higher marginal value. Thus, the difference in the usability of data could be the primary reason for the difference in returns to data.

No labor matching frictions. When we measure jobs, we will account for the fact that not all job postings result in a worker being hired. We adjust for the job filling rate. What is important is that labor matching frictions not dissociate wages from firms' marginal values. Obviously, firms would never pay more than the marginal product of a worker for a wage because then firing them is a dominant strategy. Matching frictions might result in less pay. But the market for financial analysts is an unusually competitive and liquid labor market. The finance industry pays its data workers particularly well so that they can fill jobs quickly. Addressing this concern is part of why the focus on finance is useful.

Omitted inputs. Of course, physical capital, human capital or management skill, land or other realistic inputs are omitted from the production function. One of the defining features of the knowledge economy is that it is less capital-intensive and more data-intensive in production. The two factor structure we use maintains the parallel with the early two-factor, capital and labor production.

Our approach to estimating production exponents relies on using the fact that the exponents are also the share of revenue paid to each factor of production. If we omit a factor of production, the revenue that would be paid to that factor is now paid to the owners of the remaining factors. In other words, these omissions will bias up the value of the exponents on both data and labor, for both technologies. This does not alter our conclusions unless those omitted inputs vary systematically across firms with different data-labor ratios and across technologies.

Optimal firm hiring and wages. The first order condition with respect to new technology (AI) analyst labor L_{it} is

$$(1-\alpha)K_{it}^{AI} - w_{L,t}L_{it} = 0, \tag{6}$$

which says that total payments to new technology analysis labor $w_{L,t}L_{it}$ are a fraction $(1-\alpha)$ of the value of knowledge output from AI analysis, K_{it}^{AI}. The first order condition with respect to old tech analyst labor l_{it} is

$$(1-\gamma)K_{it}^{OT} - w_{l,t}l_{it} = 0. \tag{7}$$

This says that the total payments to old technology analysis labor $w_{l,t}l_{it}$ are a fraction $(1-\gamma)$ of the value of total output K^{OT}_{it}. Taking the ratio of the two first order conditions implies that

$$\frac{(1-\alpha)K^{AI}_{it}}{(1-\gamma)K^{OT}_{it}} = \frac{w_{L,t}L_{i,t}}{w_{l,t}l_{i,t}} \tag{8}$$

This ratio varies by time t and it measures how much knowledge production technology has changed, for each firm i. The first order condition with respect to data management labor λ_{it} is

$$\frac{1}{r}v'(D_{i(t+1)})(1-\phi)\lambda_{it}^{-\phi} = w_{\lambda,t}. \tag{9}$$

If the marginal value of data today and tomorrow are similar, we can solve for $v'(D)$ and replace $\lambda^{1-\phi}$ by the change in the data stock, to get[6]

$$\frac{(\alpha K^{AI}_{it} + \gamma K^{OT}_{it})(1-\phi)}{r-(1-\delta)} \frac{D_{i(t+1)} - (1-\delta)D_{it}}{D_{it}} - w_{\lambda,t}\lambda_{it} = 0. \tag{10}$$

Intuitively, total payments to data management, $w_{\lambda,t}\lambda_{it}$, depend on the percentage increase in the data stock. This is multiplied by the labor share of income $(1-\phi)$ times the value of the total knowledge produced, $(\alpha K^{AI}_{it} + \gamma K^{OT}_{it})$. Dividing that amount of knowledge term by $r-(1-\delta)$ makes this present discounted value, where the depreciation term $(1-\delta)$ acts like the growth rate in a Gordon growth valuation.

Using these first-order conditions, we derive an expression for the optimal stock of data for a firm, that we will use to impute the initial data stock of each firm. Replace $D_{i(t+1)} - (1-\delta)D_{it}$ in (10) with $\lambda^{1-\phi}_{it}$. Then substitute out the unobserved quantities of knowledge using the first order conditions for AI and OT labor, (6) and (7): $K^{AI}_{it} = w_{L,t}L_{i,t}/(1-\alpha)$ and $K^{OT}_{it} = w_{l,t}l_{i,t}/(1-\gamma)$. This substitution yields:

$$D_{it} - \frac{\left(\frac{\alpha}{1-\alpha}w_{L,t}L_{i,t} + \frac{\gamma}{(1-\gamma)}w_{l,t}l_{i,t}\right)(1-\phi)}{r-(1-\delta)} \frac{\lambda^{-\phi}_{it}}{w_{\lambda,t}} = 0. \tag{11}$$

This equality offers a way to impute firms' data from their observable hiring choices and wage payments. This imputed data stock is the amount of data that best rationalizes the labor decisions of each firm.

[6] See appendix for step-by-step derivation.

2 Data and Estimation

Why look at the investment management industry? Our model is about knowledge production generally, in any industry. But as we turn to estimating this model, we use asset management industry labor and data estimates. One reason we do this is that the investment management industry is primarily a knowledge industry, where information is processed to form forecasts about asset returns and profitable portfolios. But the main reason is that finance is an early adopter of AI and big data technology. If we want to study the nascent adoption of this new technology, it is helpful to look in corners of the economy where adoption is most substantial. In independent studies with different methodologies, Felten, Raj, and Seamans (2018) and Brynjolfsson, Mitchell, and Rock (2018b) both came to the conclusion that the finance/insurance industry was the one with the greatest potential for labor substitution with AI. Acemoglu, Autor, and Hazell (2019) document that finance has the third most number of AI job postings, behind information and business services.

Finally, the financial industry is a useful laboratory because finance jobs are typically filled. JOLTS data tell us that finance is an industry with one of the highest vacancy conversion rates into new employment, presumably because the finance sector pays more than others. Thus, when they want a worker with a specific set of skills, they can buy them. Since our work relies on job postings, it is helpful if many of these postings are, in fact, filled.

Of course, one could include other industries, to broaden our sample and sharpen our estimates. The problem is that distinguishing which workers combine data and labor to produce knowledge relies on industry-specific vocabulary. The type of work that the investment management industry calls an analyst, the retail industry might call an online marketing expert. Because the language used to describe jobs differs, one needs a separate dictionary for each context, which introduces more error. Restricting our analysis to the asset management sector allows us to obtain a cleaner sample of job postings and improve the accuracy of our estimates.

Labor demand Our data is the job postings data set collected by Burning Glass, from January 2010 through December 2018. These postings are scraped from more than $40,000$ sources (e.g. job boards, employer sites, newspapers, public agencies, etc.), with a careful focus on avoiding job duplication. Acemoglu, Autor, and Hazell (2019) show that Burning Glass data covers 60-80% of all U.S. job vacancies. The finance and technology industries have especially good coverage. It includes jobs posted in non-digital forms as well. Importantly, for a large portion of job postings, the data reports employer names, as well as the sector, job title, skill requirements, and sometimes the offered salary range. In addition to the structured data fields, we also make use of the full text of the job posting, as written by employers. We cannot observe some senior positions, directly advertised through word-of-mouth or headhunters. That, though, is not particularly concerning for our application as those roles are likely to be more managerial in nature, while we need to identify positions directly requiring data management or analysis.

Before delving into details, we provide an overview of how we construct this data set, in three broad steps. (1) We subset the data to candidate jobs of interest in the financial industry. (2) We identify finance industry jobs belonging to one of the following categories: data managers, AI analysis or old tech analysis. (3) We match job postings to employers. This procedure leads to the identification of $308,600$ employer-matched job postings categorized as AI, old tech or data management. The unique number of employers goes from 442 in January 2015 to 739 in December 2018. The total number of unique employers is 812. Next, we provide more detail on each of these three steps.

Step 1: The objective of this step is to subset the overall dataset to jobs that likely belong to the finance industry. Our approach is to eliminate jobs that clearly should not belong to our sample, while in following steps we adopt a more direct selection approach. For that purpose we use jobs' NAICS, O*NET and proprietary Burning Glass codes. These are all variables which categorize jobs in progressively finer categories. Not all variables are present for all jobs, for that reason we proceed with the following method. We first drop all job postings which possess a NAICS code that does not belong to one of the following 2-digit NAICS codes: "Professional, Scientific, and Technical Services", "Finance and Insurance", "Information" and "Management of Companies and Enterprises". Since NAICS codes are missing for a non-negligible portion of the data set, we also keep all jobs for which the NAICS code is not available and proceed to use the other job categorization variables to further refine our candidate list of jobs. Next, we compile lists of O*NET codes and Burning Glass proprietary codes (BGT Occupation Group, BGT Career Area) of job categories that should clearly not be in our sample[7]. After eliminating all jobs in irrelevant categories, we are left with a sample of candidate finance jobs. The benefit of proceeding with a selected sample is both computational as well as reducing noise in the following selection steps.

Step 2: For all jobs in the candidate finance sample, we then use the full text of the selected job postings in order to identify analysis jobs and data management jobs. We define "data management" jobs as those requiring skills related to the cleaning, purchasing, structuring, storage and retrieval of data. What define as "analysis jobs" those jobs that combine structured data with skilled labor. We call these analysts because they analyze data in different ways. They are not necessarily what the financial industry calls analysts. Within the analysis jobs we further distinguish between those that mostly require old (Old Technology - OT) or new

[7]Examples of excluded 6-digit O*NET codes that were still present in the sample are: "Bookkeeping", "Accounting, and Auditing Clerks", "Customer Service Representatives", "Cashiers", "Retail Salespersons", ...

(Artificial Intelligence - AI) skills.

This classification is obtained by developing a dictionary of words and short phrases that indicate "data management" or "data analysis", and then counting the relative frequency of these words or expressions in each pre-processed job text.[8] Among the "data analysis" keywords we further identify those clearly indicative of the old and new technologies and we assign jobs to "Old Tech - OT" or "Artificial Intelligence - AI" depending on the relative frequency of words or short phrases of the two types present in the posting. The full dictionaries used are available in Appendix A.1.

While this step is similar in nature to other approaches, working with one type of job in a single industry and using the frequency of skill mentions in a job posting allow us to sort jobs more precisely. Acemoglu and Restrepo (2018) use the standardized skills list provided by Burning Glass. Babina, Fedyk, He, and Hodson (2020) extend this word list by measuring how frequently all structured skills appear in the same job posting as basic AI skills. These standard-skill-based approaches do not work for our exercise: Burning Glass' skills list is not detailed enough to distinguish between different types of data analysis in investment management. Misclassification that might wash out in a job counting exercise is more serious for us. We need to match data and labor stocks firm-by-firm. Instead of using the Burning Glass skills list, we analyze the full text of the job posting. The key advantage of this approach is that is allows us to make use of the frequency of mentions of each type of skill.[9] We illustrate below why word frequency is essential for our classifications.

Step 3: In this step we take an active selection approach by identifying the AI, old tech and data management jobs short-listed in *step 2* which belong to predominantly investment management employers. To match job postings to employers, we use a master list of investment management firms. We identify job postings that belong to each employer, using fuzzy matching of employer names. Appendix A.2 provides a more detailed description of that matching process. We further restrict the sample to employers that posted at least 5 "Old Technology" or "Artificial Intelligence" jobs throughout the entire period of interest (2010-2018).

There is new firm entry. 77% of our firms are in the data set in 2015. The remaining 23% appear for the first time in 2016-2018. Not all 23% are new firms. Many are existing firms that enter our data set when they hire data workers for the first time.

Figure 2 illustrates the frequency of *all* keywords in the samples of job postings categorized as

[8] We pre-process the text of each job posting by first removing symbols, numbers and stop-words (e.g. is, the, and, etc.) and then stemming each word to its root using the Porter stemmer algorithm (thus, e.g. "mathematic", "mathematics", ... = "mathemat").

[9] For instance a job that mentions 'Machine Learning' 10 times within the job text and then also states "Masters in Statistics also accepted", in our approach would be clearly classified in the "AI" category. Looking at the skills lists, instead, the categorization of the job would be ambiguous as it would appear to require both old and new technology skills in the same proportion: "Statistics" and "Machine Learning".

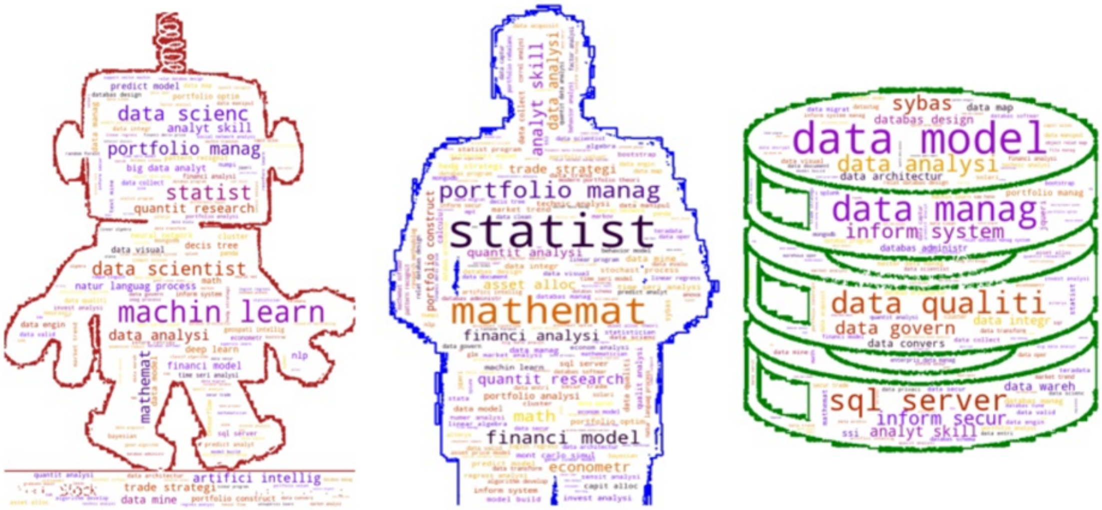

Figure 2: Keywords in the full text of the categorized machine learning, old technology and data management jobs. Larger fonts indicate a higher word frequency. Burning glass job postings, 2010-2018.

AI, old tech or data management (left to right sub-figures); where a greater size indicates a higher frequency. Note that, even if all data analysis and data management keywords are included in all three word clouds, the keywords specific to the assigned category have a significantly higher frequency. That indicates that indeed there exists a clear separation among these three job types, as otherwise the most frequently mentioned skills in each category would be less distinct. This also highlights the importance of utilizing word frequencies in assigning jobs. In fact, accounting for the mere presence of certain keyword in a job posting might lead us to assign it to the wrong category, while considering the relative frequency of mentions allows for a clearer separation of the three groups.

Sample job postings To provide a clear idea of how this methodology classifies jobs, we list three sample job postings here, one each of old technology, AI or big data-related skills, and data management. In this example, all three jobs are posted by the firm Two Sigma. The text of the first job reads:

> "We are looking for world-class quantitative modelers to join our highly motivated team. Quant candidates will have exceptional quantitative skills as well as programming skills, and will write production quality, high reliability, highly-tuned numerical code. Candidates should have: a bachelor's degree in mathematics and/or computer science from a top university; an advanced degree in hard science, computer science, or the equivalent (a field where strong math and statistics skills are necessary); 2 or more years of professional programming experience in Java and

> C, preferably in the financial sector; strong numerical programming skills; strong knowledge of computational numerical algorithms, linear algebra and statistical methods; and experience working with large data sets. (...) ”

This job is classified as old tech because it uses words such as “mathemat” (x1), “math” (x1), “statist” (x2), “algebra” (x1), and does not contain words related to AI or data management skills.

This first posting contrasts with the text of the second job, which reads:

> “Two Sigma combines massive amounts of data, world-class computing power, and statistical expertise to develop sophisticated trading models. We believe that the scientific method is the best way to approach investing. We are looking for talented researchers who can *apply and develop machine learning algorithms for financial datasets*. Researchers work on: Developing trading strategies using statistical and machine learning algorithms. Advancing existing initiatives and opening opportunities to pursue new research topics. Designing solutions for challenges in analyzing real world, large datasets. Minimum qualifications: PhD in quantitative disciplines. Expertise in statistics and machine learning. Intermediate programming skills in Java, C++, or Python. (...) ”

This job is harder to classify. It contains the word “statist” (x3), indicative of old tech. But what ultimately gets this job classified as AI is the higher frequency of AI-related words: “machin learn” (x4). An algorithm that just looked for the presence of skills or words, without measuring their frequency, would likely misclassify this job, and many others like it.

Finally, the text of the third job reads:

> “ (...) Technology drives our business it’s our main competitive advantage and as a result, software engineers play a pivotal role. They tackle the hardest problems through analysis, experimentation, design, and elegant implementation. Software engineers at Two Sigma build what the organization needs to explore data’s possibilities and act on our findings to mine the past and attempt to predict the future. We create the tools at scale to enable vast data analysis; the technology we build enables us to engage in conversation with the data, and search for knowledge and insight. (...) You will be responsible for the following: *Capturing and processing massive amounts of data for thousands of different tradable instruments*, including stocks, bonds, futures, contracts, commodities, and more; (...) ”

This job is classified as data management because of the words, “explor data possibl,” “enabl vast data analysis,” “data specialist,” and “data team.”

Wages Some, but not all jobs in Burning Glass list a salary range. Because listed salaries are not representative, we obtained salary data from a different source. PayScale[10] uses crowdsourcing to collect real-time salary and compensation information. On their website, individuals complete a survey about their current position. In return, they receive a report detailing how their salary compares to that of other individuals with similar characteristics (education, skills, work experience, ...). People may use this service when renegotiating their salary, when looking for other positions or when choosing additional training. So far, PayScale has more than 65 million salary profiles in more than 365 industries.

Importantly for our estimation, individuals are asked to disclose their job title, their total compensation (including bonus) and a list of the top three skills required in their job. The timeliness of this crowd-sourced information, together with its granularity, allowed us to identify 11,041 salary profiles for our job categories of interest in the investment management industry. More specifically, we were able to identify 5,639 Data Management salary profiles, 2,817 profiles by Old Tech analysts and 2,585 by AI analysts. Appendix B details the procedure utilized in identifying and categorizing these profiles.

Every month we compute the mean salary, across all salary profiles, in each of our three job categories of interest.

Figure 3: Average wages. AI analysts earn on average more $26,333 per year than data managers and old technology analysts. Job postings from Burning Glass matched to wage data from PayScale. Wage refers to annual, total compensation.

Figure 3 illustrates the resulting salary time-series.[11] Notice that AI jobs consistently pay

[10] www.payscale.com. Data last updated December 2020.

[11] In unreported results we also utilized median salaries and smoothed salaries utilizing a 12-months rolling

more – \$26,333 more per year on average – than traditional analyst jobs. This difference in wages provides crucial evidence about worker productivity.

Cumulating hiring to get labor. In order to estimate production, we need to know the size of the labor force working in each type of job. The job postings we observe are not the stock of labor. The number of observed job postings for the three categories of interest is displayed in Figure 4, together with the number of employers hiring in each category. We use the following procedure to estimate each firm's labor from their job postings.

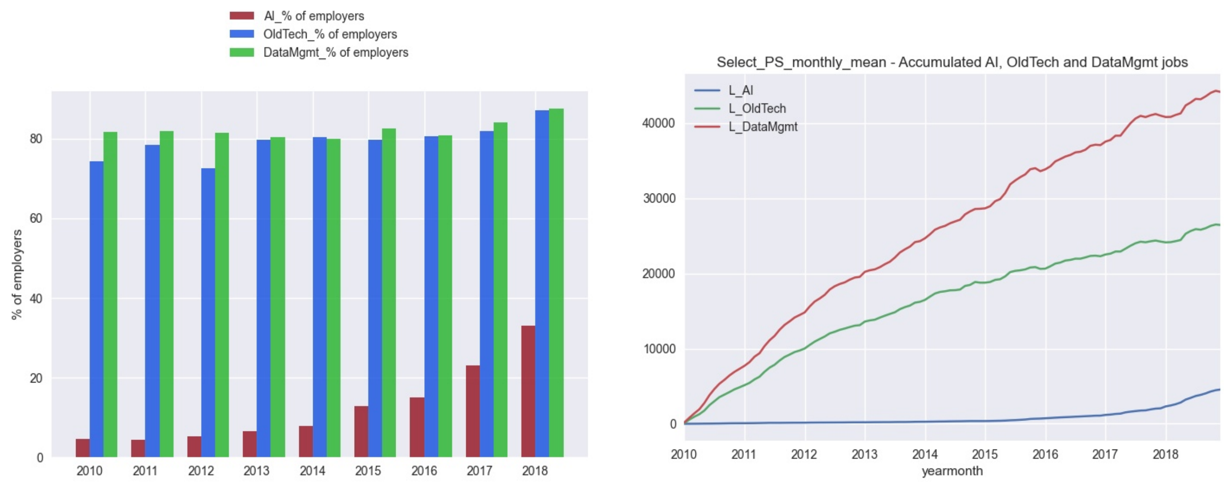

Figure 4: Job postings and Labor Stocks: Panel 1 shows the fraction of employers hiring in each category. Panel 2 shows the stock of labor in each category, measured as a cumulated number of job postings, adjusting for filling and separation rates as in (12).

Job postings are not the same as net hiring. One might be concerned that AI workers, in particular, are so scarce that many postings go unfilled and/or that workers jump from job-to-job. There are two key differences between postings and net hiring: the probability that a vacancy is filled and the probability that an employed worker separates from their job. We adjust for both of these using data on vacancy fill rates and job separation rates from the Bureau of Labor Statistics (BLS).

Each month, the BLS reports the job posting, job filling and separation rate for each occupation. The three occupation brackets present in the final sample are: "Finance and Insurance", "Professional, Scientific and Technical Services" and "Information". Since we want to map our job postings into expected hires, we multiply each job posting number by the fraction of job postings that results in a new hire (h).

window. Results are similar across all specifications.

Of course, AI jobs are not contained in a unique occupation tracked by the BLS. We need a way to map our technology-based job classification into the BLS occupation classification. Fortunately, most Burning Glass job posting have a listed occupation. Of course, different postings have different classifications, even within AI, old technology or data management jobs. Thus, we measure the proportion of jobs in each of our samples that belongs to each occupation. Each month we compute a vector of occupation weights for AI jobs, one for old tech jobs and one for data management jobs that is the fraction of jobs in each category that belongs to each occupation bracket in our sample. We multiply this weight vector by each of the fill and separation rates that month, to get the imputed fill and separation rates for AI financial analysis jobs (h_t^{AI} and s_t^{AI}), the imputed fill and separation rates for old technology financial analysis jobs (h_t^{OT} and s_t^{OT}) and those for data management jobs (h_t^{DM} and s_t^{DM}). See Appendix A.3 for more detail on how BLS data is mapped into our job categories and how h and s are derived from BLS reported rates.

For $type = [AI, OT, DM]$, if s_t^{type} are separation rates by type-month, and h_t^{type} are the fraction of posted vacancies filled by type-month and j_{it}^{type} are Burning Glass job postings by firm-month-type, we cumulate labor flows into stocks as follows:

$$L_{it} = (1 - s_t^{AI})L_{i(t-1)} + j_{it}^{AI} h_t^{AI}, \tag{12}$$

$$l_{it} = (1 - s_t^{OT})l_{i(t-1)} + j_{it}^{OT} h_t^{OT}, \tag{13}$$

$$\lambda_{it} = (1 - s_t^{DM})\lambda_{i(t-1)} + j_{it}^{DM} h_t^{DM}. \tag{14}$$

To use this cumulative approach, we need the initial number of workers of each type (L_{i0}, l_{i0} and λ_{i0}). Unfortunately, that information is not available, but we know that the initial number of workers becomes less relevant the further we are from initialization. For this reason we start the initialization from zero for all job types and we use the first 5 years of data $[2010 - 2014]$ as a burn-in period. We then use the last 4 years $[2015 - 2018]$ for the structural estimation of the model's parameters.[12]

The right panel of Figure 4 shows the imputed labor stocks for each job type. AI workers in investment management are still a small fraction of the overall labor supply, suggesting that the transition to a new model of knowledge production is just in its beginnings. However, what looks like a small uptick on this axis looks like an explosion when we zoom in. Prior to 2015, hiring in AI is mostly flat. From 2015 to 2018, the stock of AI labor increases about 13-times

[12] Five years of burn-in time appears sufficient because when we instead assume that initial labor stock in each category is in steady state (i.e. all hiring is to replace expected separations), we observe much higher labor for the first 5 years, and then convergence with the zero initial labor series. Appendix D replicates all results utilizing steady-state initialization instead.

	Data Management λ_{it}	AI analysts L_{it}^{AI}	Traditional analysts L_{it}^{OT}
mean	53.73	2.33	32.37
stdev	441.17	29.78	204.84
minimum	0	0	0
median	5.72	0	3.63
maximum	11409.26	1765.88	4420.53
Observations	33,392	33,392	33,392

Table 1: **Labor Stock Summary Statistics**. Distribution of the number of AI, old tech or data management workers (stock of labor) per firm in a pooled sample of all firms and time periods of interest.

from 350 to 4537 AI analysts.

Table 1 reports the summary statistics for the stock of each type of labor. What is salient in all three categories is the large dispersion in the size of labor forces. This is helpful because the cross-firm heterogeneity is what identifies the technology parameters. Our final data set contains $33,392$ employer-month observations. These will be used in the structural estimation of the model's parameters.

Cumulating data management to get structured data stocks We measure each firm's stock of data in each period by adding the data management inputs to the depreciated stock of yesterday's data:

$$D_{it} = (1-\delta)^{t+1} D_{i0} + \sum_{s=0}^{t} (1-\delta)^{t-s} \lambda_{is}^{1-\phi}. \tag{15}$$

We fix the depreciation rate of data at $\delta = 0.03$, which is a 3% depreciation rate per month. We also report results for 1% and 10% deprecation. This represents some high-frequency data, whose value lasts for fractions of a second, as well as longer term data used to value companies.

This approach requires firms' initial data stocks. We assume that each firm's initial data is proportional to their first period inflows of data. The 2015 inflows of data for firm i are firm i's 2015 data management labor force, raised to the power $(1-\phi)$. Thus, $D_{i,2015} = \iota \lambda_{i,2015}^{1-\phi}$. The constant of proportionality ι comes from equating our two measures of data.

For each firm, we estimate an initial data level $\tilde{D}_{i,0}$ that makes all subsequent data levels closest to the firm's optimal data level, i.e. the D_{i0} that allows the sequence of D_{it}'s from (15) to best fit to the 2016-2018 sequence of data optimality conditions (11).

The constant ι equates the average initial data stock to the average imputed stock: $(1/N)\sum_i \iota \lambda_{2015,i}^{1-\phi} = \bar{D}_0$.

Then we can express equation 15. Firm i's stock of data in each period t is

$$D_{it} = (1-\delta)^{t+1}\iota\lambda_{2015,i}^{1-\phi} + \sum_{s=0}^{t}(1-\delta)^{t-s}\lambda_{is}^{1-\phi}. \tag{16}$$

where $\iota = N\bar{D}_0/(\sum_{i=1}^{N}\lambda_{i,2015}^{1-\phi})$. After 2015, each firm's data is their discounted initial data, plus the flows of new data added by data management workers.[13] We use the initial data stocks to estimate the production function parameters. Then, given the parameters, we can recover the best-fit initial data and cumulate up a data stock for each firm easily.

Data depreciation and riskless rate. We consider two approaches to estimating data depreciation: depreciating information and depreciating the computer code that collects data. Ultimately, we estimate results based on both estimates. In the first approach, the rate of data depreciation depends on the time-series properties of the variable being forecasted, as well as on the nature of data management. If data is being used to forecast firm earnings, for example, and firm earnings are quite stable over time, with high persistence and small innovations, then data from a few months ago is still quite useful for predicting today's earnings. However, if data is being used to forecast order flow, which has a persistence of only a few days, then order flow data from a month ago is nearly worthless. Data on countries' interest rates or fixed customer characteristics, might depreciate over decades. Firms' data sets are a mixture of these different types of data. Because earnings lies in between the extremes of highly transitory and highly persistent data, we base our first depreciation estimate on the properties of earnings data.

The precision of a signal about an unknown AR(1) state process depreciates at the rate $1-\rho^2(\rho^2+\sigma_\theta^2 D_{it})^{-2}$, where ρ is the persistence of the AR(1) process for earnings and σ_θ^2 is the variance of its innovations (Farboodi and Veldkamp, 2019). Farboodi, Matray, Veldkamp, and Venkateswaran (2020) report these coefficients for the earnings processes of a variety of firms. They find depreciation rates that range from 58% to 91% annually.[14] That is the equivalent to 5-7.5% per month.

Our second approach to measuring data depreciation recognizes that what matters for our exercise is not really how much data depreciates, but how much the output of data management labor depreciates. If what data workers do is to collect each data point, one at a time, and add

[13] Another approach would be to estimate this recursive system of data stocks, production parameters and data inputs for every firm in our sample, the problem quickly becomes unmanageable. However cross-firm data heterogeneity is essential. Estimating only the average D_{i0} and then using a rule to map the average into a firm's initial data greatly reduces the dimensionality of the estimation.

[14] This rate is for amounts of data that increase earnings forecast precision between 0 and 10 times their initial precision.

them to the data set, then depreciation of a piece of information is the relevant depreciation rate. But data workers would never hand-collect a stream of data like this. They automate the collection of a particular type of data. Each month, each day or each microsecond, their system automatically pulls the next piece of data. That process matters because a unit of data management labor produces a stream of data that doesn't depreciate; it automatically updates. In this view of data management, depreciation is hardware breaking, data links changing, or software needing updates. That type of depreciation sounds very much like the standard capital depreciation of macro theory. Typical estimates of 12% per year (1% per month) might then seem to be more appropriate. Standard accounting practice is to amortize data warehouses like software, over 36 months. That translates to a depreciation rate of 3% per month.

Given this range of estimates, we explore depreciation rates of 1%, 3% and 10% monthly, with the understanding that rates around 1-3% more accurately reflect the automated way in which data is collected.

The riskless rate, r, captures the rate of time preference in the model. We use the one-month treasury bill rate quoted at the start of each month as the rate r_t in month t.

Estimating production functions The broad picture here is that the variables of interest are the two production function exponents, α and γ from (1) and (2). There are four variables we jointly estimate: the production exponents α, γ, and ϕ and the initial average data stock $\bar{D}_0$. Using non-linear least squares, we minimize errors from the three first order conditions (6), (7) and (10), as well as the optimal data stock condition (11). We use these estimated parameters to compute time and firm-specific productivity fixed effects. Given those productivity terms, we re-optimize the parameters and repeat until convergence.

More specifically, we initialize the estimation utilizing a guessed set of parameters (α, γ, ϕ and $\bar{D}_0$). Conditional on this set of parameters, we solve for the time-varying productivity parameters A_t^{AI} and A_t^{OT}, utilizing the N-firm averages of the first order conditions 6 and 7 for each time period:

$$A_t^{AI} = \frac{\omega_{Lt} \sum_{i=1}^{N} L_{it}}{(1-\alpha) \sum_{i=1}^{N} a_i^{AI} D_{it}^{\alpha} L_{it}^{1-\alpha}}; \quad A_t^{OT} = \frac{\omega_{lt} \sum_{i=1}^{N} l_{it}}{(1-\gamma) \sum_{i=1}^{N} a_i^{OT} D_{it}^{\gamma} l_{it}^{1-\gamma}} \tag{17}$$

We use time-series averages of the same conditions 6 and 7 to solve for the firm-specific productivity parameters a_i^{AI} and a_i^{OT}:

$$a_i^{AI} = \frac{\sum_{t=1}^{T} \omega_{Lt} L_{it}}{(1-\alpha) \sum_{t=1}^{T} A_t^{AI} D_{it}^{\alpha} L_{it}^{1-\alpha}}; \quad a_i^{OT} = \frac{\sum_{t=1}^{T} \omega_{lt} l_{it}}{(1-\gamma) \sum_{t=1}^{T} A_t^{OT} D_{it}^{\gamma} l_{it}^{1-\gamma}}. \tag{18}$$

As evident from equations 17 and 18, though, the obtained A_t^{OT} is a function of all a_i^{OT}

and A_t^{AI} is a function of all a_i^{AI} for $i = 1, \ldots, N$, and vice-versa. To obtain unique numerical solutions for all productivity parameters of interest we adopt an iterative process. We:

1. Initialize the iteration by assigning to each a_i^{AI} and to each a_i^{OT} a value of $\frac{1}{N}$.

2. Substitute those values into equation 17 to obtain an initial estimate for the time-varying productivity parameters.

3. Iterate between equations 18 and 17 until the sum of squared differences in the estimated parameter values across two consecutive iterations is smaller than a tolerance level (10^{-10}).

Note that this procedure is *not* equivalent to estimating productivity parameters that vary by both time and firm. That would absorb all variation across the AI and old tech first order conditions and would not leave any degrees of freedom to identify the Cobb-Douglas parameters α and γ. To draw a parallel with regression analysis, the above method is akin to utilizing de-meaning to remove time and firm fixed-effects. After doing this, residual means are zero, but there is still lots of variation is our observations. This remaining variation is what we use to pin down the parameters of interest. In particular, the co-variance between data stock and AI/OT hiring is what identifies the Cobb-Douglas parameters α and γ. Section 3.1 elaborates further on identification.

We then substitute the computed productivity parameters into the AI and old tech first order conditions 6 and 7. This allows us to construct a vector of errors for each first order condition, with observations for all relevant firm-months. We then compute a vector of errors for each firm-month for the data management labor first order condition (10) and for the optimal data stock condition (11). We append all these error vectors to obtain a final errors vector containing $(33,392 \times 4)$ equations. Finally, based on this combined vector, we compute the sum of squared errors specific to the chosen set of parameters (α, γ, ϕ and $\bar{D}_0$).

We then utilize non-linear least squares to repeat the above procedure for different combinations of α, γ, ϕ and $\bar{D}_0$. Convergence is obtained when the algorithm identifies the set of parameters which minimizes the sum of squared errors.

As a check on convergence, we also re-estimate the parameters using a grid search method. This is viable because many of our parameters, like the production exponents are bounded between zero and one. While it takes longer to run, our grid search does identify the same solution.

Appendix D re-runs the estimation in three different ways: i) including only time-specific productivity parameters; ii) including only firm-specific productivity parameters, and iii) without constant productivity across firms or over time. The estimates across all specifications are similar.

3 Results

The results have four parts. The first part is our main result, with our estimates of the production function parameters. They reveal that the size of the change in knowledge production is at the lower end of the estimates of the industrial revolution in goods production. The second part explores why we come to this conclusion. It shows why the sensitivities of a firm's data stock to its AI and OT workforce are key statistics for identifying production function exponents. Third, we relate our findings to a literature on labor-replacing technological change. In financial analysis, we find no evidence of AI crowding workers out. Finally, we use our structural model to value data.

3.1 Main Result: Comparing Changes in Knowledge Production to Industrialization

Our main question is: How much does AI change knowledge production? In Table 2, the exponents α and γ represent one minus the diminishing returns to data in the new and old technologies. The fact that $\alpha > \gamma$ means that the rate of diminishing returns to data is less with the new AI technology. In other words, new data technology has significantly raised the productivity of analyzing larger data sets. That is not surprising. The fact that the exponent rose by 0.05, suggests a 5% shift in the share of labor income to data owners, which is economically substantial. The change is statistically significant, at any reasonable threshold.

		$\delta = 1\%$	$\delta = 3\%$	$\delta = 10\%$
AI Analysis	α	0.916	0.867	0.690
		(0.0010)	(0.0015)	(0.0027)
Old Technology Analysis	γ	0.690	0.816	0.640
		(0.0033)	(0.0020)	(0.0032)
Data Management	ϕ	0.280	0.453	0.010
		(0.0063)	(0.0058)	(0.0085)
Change in Labor Share	$\gamma - \alpha$	-23%	-5.1%	-4.9%

Table 2: **Main Result: AI Reduced the Share of Knowledge Income Paid to Labor ($\alpha > \gamma$).** The estimates are for the exponents on data in the knowledge production functions in (1) and (2) and the production of structured data in (3). Data covers 2015-19 from PayScale and Burning Glass. Standard errors in parentheses.

The data depreciation rate matters for this conclusion, but does not diminish the size of our estimate. If data management is mostly maintenance of physical infrastructure and thus depreciates like physical capital, at a rate around 12% per year or 1% per month, then the effect of AI is twice as large. When assuming a very high depreciation rate of data management (10% monthly) our estimate of the technological change is similar to the baseline estimate, still 5%.

The average initial data stock is $(792, 221, 501)$ for $\delta = (0.01, 0.03, 0.1)$. From here on, we present results for the medium depreciation case of $\delta = 3\%$ and report results for the other two cases in the appendix.

The labor first order conditions (6) and (7) tell us that the exponents α, γ also govern the labor share of incomes. Our results imply that owners of data have gained from this technological change. While they used to be paid 82% of the value of the knowledge output, they can now extract 87% of that value. In addition, since more knowledge is being produced, this is 87% of a larger revenue number. While these numbers are much higher than what macroeconomists typically use, they are for a particular industry, finance, with a particularly unequal income distribution. This finding is consistent with the overall economic trend of a decrease in the labor share of income (Karabarbounis and Neiman, 2017).

Of course, owners of data had to pay data managers to build these data sets, just like owners of capital had to pay for the investment in their capital stocks. But once they own these data stocks, they get the income associated with their asset.

Industrial Revolution Comparison. How can we gauge the size of this change in knowledge production? To understand whether this change in factor shares is large, we compare it with estimates from another period where new technologies were adopted, the industrial revolution.

Clark (2005) and Clark (2010) study the historical evolution of income shares in England and find that from 1760-1870, the capital share rose from 21% to 26%. Allen (2019) finds a substantially larger 20% increase, from a 19% capital share of income to 39% of income, between 1759 and 1867. Most of that additional income comes from the land share, with only a 2% decline in the labor share. During a slightly later period, between 1900 and 1920, Klein and Kosobud (1961) find that the labor share of income fell from 0.909 to 0.787. In a production function with only capital and labor inputs, this suggests that the capital share of income would have risen by 12%. Taken together, this suggests changes in the capital factor share in the vicinity of 5-20%.

When historians examine changes to the changes in industrial labor income share, the answers differ. Since much of the additional income paid to capital came from income that was previously paid to land owners, changes in the labor income share are more modest. Using the labor income shares, would make the change in AI production look larger in comparison and make our conclusion look dramatic. But that comparison between labor income shares would be obscuring the large change in industrial production that made capital a more important factor. To be conservative, we compare the change in industrial era capital shares to the modern-day change in the data share of income.

Since the labor share of capital corresponds to the exponent on capital in a Cobb-Douglass production function, like ours, this estimate suggests that the capital exponent in the production function rose by $0.05 - 0.20$ during the industrial revolution. Our rise of $\alpha - \gamma = 0.05$ is in this ballpark. Thus, historical estimates suggest that the magnitude of the technological change in the big data revolution corresponds to some of the lower estimates of the size of the industrial revolution.

What data features identify production parameters? Our exponents are estimated, not calibrated to particular features of the data. So all data features matter. However, some are particularly informative. One feature of the data that is particularly informative about the rate of data diminishing returns α and γ is the sensitivity of data stocks to AI analysis labor, versus the sensitivity to OT analysis labor. The idea is that, if AI knowledge production is a more data-intensive technology, then firms should have more data per AI worker than per old technology worker. However, this sensitivity of data to analysts also depends on the productivity of each type of worker. That productivity is reflected in wages. This is why Cobb-Douglas production estimates are typically measured using ratios of payments, not ratios of inputs. We use our structural model to map production exponents to the sensitivity of data to the share of income paid to data analysts versus data management workers. Then, we plot firms' data stocks against the income ratios, for the two types of workers, and show, visually and econometrically, that when AI workers earn more, it predicts a greater rise in firm data, than when OT workers earn more. The difference in sensitivity of data to AI and OT analysts explains the difference in production function exponents $\alpha - \gamma$.

To derive this data-labor relationship from our model, start from the data manager first-order condition (11). We can re-write this first order condition as

$$D_{it} = \beta_1 \frac{w_{l,t} l_{i,t}}{\lambda_{it}^{\phi} w_{\lambda,t}} + \beta_2 \frac{w_{L,t} L_{i,t}}{\lambda_{it}^{\phi} w_{\lambda,t}}, \tag{19}$$

where

$$\beta_1 = \frac{(1-\phi)}{r-(1-\delta)} \frac{\gamma}{1-\gamma} \quad and \quad \beta_2 = \frac{(1-\phi)}{r-(1-\delta)} \frac{\alpha}{1-\alpha}. \tag{20}$$

Notice that the difference between β_1 and β_2 is just the exponent α for AI analysis or γ for old tech analysis. Thus, the difference in β's maps to a difference between the exponents α and γ.

The slope of the relationship between data and analyst payments is steeper for AI analysts than for OT analysts, as seen in Figure 5. This difference does not speak to causality. Our model predicts that more AI workers prompt a firm to accumulate more data and vice-versa. Both of these forces are embodied in our optimality conditions and both inform our estimation.

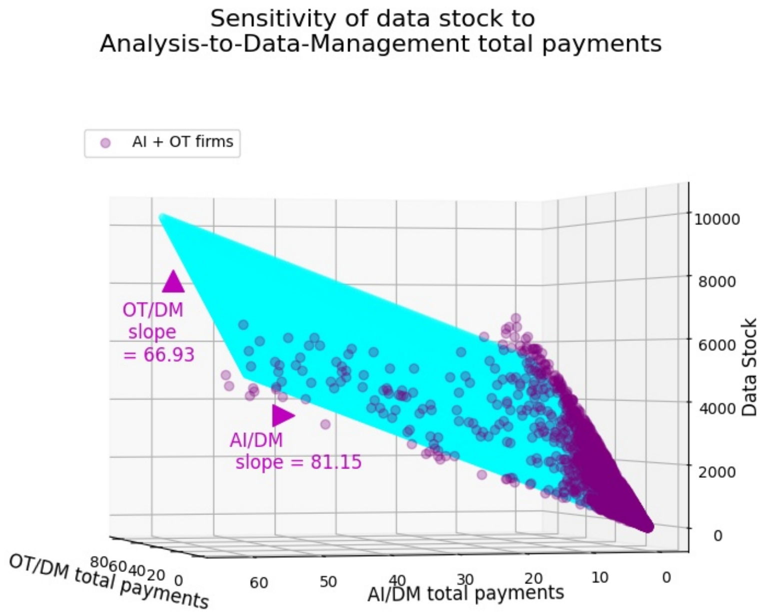

Figure 5: Firms with larger AI worker payments have larger data sets. The set of firms is those that hire some AI workers and some OT workers, over all months of observation. Data stocks are compiled using (15), assuming a data depreciation rate of 3%. Source: Burning Glass, PayScale, 2015-2018.

This bi-directional causality is why a structural estimation is essential for our purposes. Of course, data stocks are something we impute from firms' hiring choices. The hallmark of a firm accumulating more data is its hiring of data management workers. Evidence of the relationship between AI, OT and data accumulation shows up in the cumulated hiring decisions of firms. The firms we estimate to have large data sets are firms that hire more data management workers. Figure 5 also illustrates the enormous heterogeneity in firms' data stocks. In particular, there is a right tail of firms with troves of data. This heterogeneity is helpful for identification.

When we estimate (19) on the set of all firms that hire at least one OT worker, one AI worker and one data manager, at some point in the sample, we find[15]

$$D_{it} = \underset{(0.150)}{66.93} \frac{w_{l,t} l_{i,t}}{\lambda_{it}^{\phi} w_{\lambda,t}} + \underset{(0.458)}{81.15} \frac{w_{L,t} L_{i,t}}{\lambda_{it}^{\phi} w_{\lambda,t}}, \tag{21}$$

Our structural estimates tell us that $1-\phi = 0.547$ and for a depreciation of 3% and the average

[15]These estimates are slightly conservative. When we estimate this relationship on the set of firms with only OT workers, the coefficient on the first, OT payments term is 67.77 (0.076). When we estimate this relationship on firms with only AI workers, we get a coefficient of 94.54 (1.301) on the second AI payments term. Thus the coefficients are similar, with a larger implied difference between α and γ.

interest rate in our sample of $r = 1.007$, we get $(1-\phi)/(r-(1-\delta)) = 14.784$. Dividing each of the coefficients above by this number implies that $\gamma/(1-\gamma) = 4.53$ and $\alpha/(1-\alpha) = 5.489$. This implies that the production coefficients should be $\gamma = 0.819$ for old tech and $\alpha = 0.846$ for AI analysis. This is not identical to, but similar to our structural estimates. The structural estimation differs for a few reasons. First, it uses more data, not just firms hiring all types of workers. Second, it uses a time-varying interest rate r. Third, it optimize coefficients to match all the first-order conditions simultaneously. Finally, structural estimates ensure consistency between α, γ and the other parameters. But the fact that this simple OLS estimation gets quite close to our structural results helps to explain what features of the data are mostly responsible for the answers we find.

3.2 Is Data Replacing Labor?

One of the main concerns people have with new data technologies like AI is that they might be labor replacing. Our results show how even such labor-replacing technologies may be expanding labor demand.

Figure 4 illustrates the aggregate stock of analysis labor. Despite our finding that knowledge production is becoming less labor-intensive, we see that there are more and more workers doing analysis. Production can be less labor-intensive and still have more labor demand because production of knowledge is rising. The expansion is made possible by the improvement in analysis productivity. So even though AI requires less labor per unit of data, it is also labor-demand-enhancing because sector growth more than compensates.

The growing labor force is not an artifact of our parameter estimates. It is also not dependent on most model assumptions. The growing labor result comes from simply counting up the new hires and adjusting for BLS-reported departures. Much of this increase comes from there being more firms in our sample. But the growth of firms working with financial data is hardly a sign of low labor demand.

Both old tech and AI-skilled analysts become more abundant. AI jobs grew at a faster rate (from 350 to 4537). However, they account for only about 35% of the increase. The rest comes from more hiring of old technology analysts. While old technology productivity may not have improved much, these workers are made more productive by the abundance of structured data.

3.3 Estimating the Value of Data

One of the big questions in economics and finance today is how to value firms' data stocks. Four of the five largest firms in the U.S. economy, by market capitalization, have valuations that are well beyond the value that their physical assets might plausibly justify. These firms have future expected revenues based on their accumulated stocks of data. Our structural estimation offers a straightforward way to compute this value.

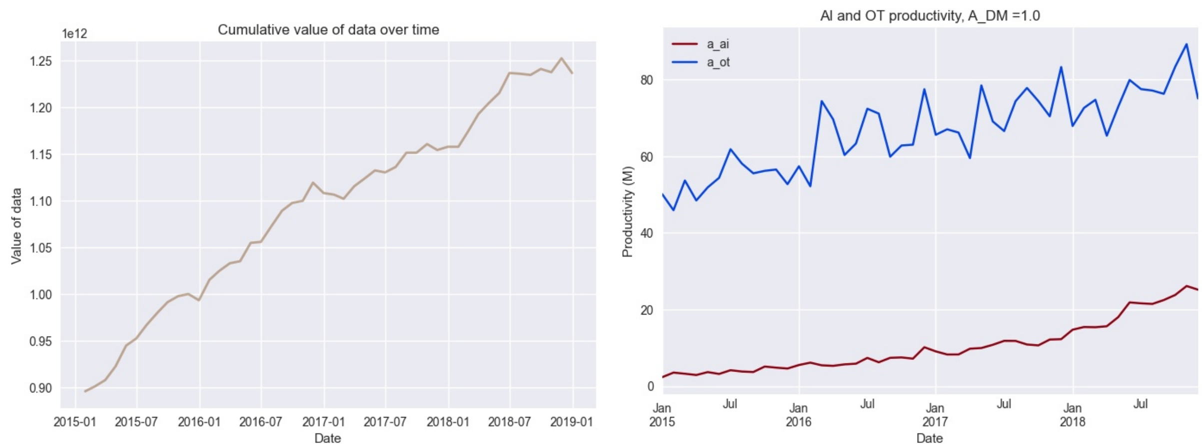

Figure 6: Estimated Value of the Aggregate Stock of Data, in trillions of current U.S. dollars, and the Productivity of Financial Analysts, 2015-2018. Productivity is the estimated values of A^{AI} and A^{OT} for AI and old technology analysts, as defined in equations (1) and (2).

Once we have estimated production parameters and data stocks, we can put them back into our value function. The value function delivers the value of each firm's stock of data in each month. This value is in nominal dollar units, since those are the units of the wages we use. Figure 6 plots this aggregate value. This is our estimate of the value function in (4) for the aggregate stock of data.

Over the time period, 2015-2018, we see a rise in the value of this data stock from about \$ 900 billion to about \$ 1.25 trillion, a 39% increase in value. This means that the combined value of the data, for all the financial firms in our sample is a little less than the market value of Amazon.

Where does this increase in value come from? The first source is simply the accumulation of data. But for most firms, this increase is modest. A second boost to data values comes from the increase in the number of financial analysts that work with data (Figure 4). The more workers there are, the higher is the marginal value of data and the more valuable the stock of data is.

Finally, firms are becoming more productive at using data. More productivity also contributes to the rise in the value of data. The right panel of Figure 6 reports our estimates of the analysis productivity parameters, A^{AI} and A^{OT}, for each month. While productivity with the old technology shows a moderate upwards trend over time, the productivity of working with the new (AI) data technologies has a steep upwards trend, with a jump in 2018. This productivity surge is additional evidence of the transformative power of new, big data technologies.

4 Conclusion

Modern discourse describes new big data technologies as the industrialization of knowledge production. What does that mean? Industrialization was the adoption of new production technologies that involved less human input and less diminishing returns to capital. In other words, the key feature of industrialization is that factor shares changed. Thus if big data technologies are the industrialization of knowledge production, they should offer less diminishing returns to data.

We explored this hypothesis by modeling the production of knowledge, in the same way economists model industrial production. Instead of mixing capital and labor with a Cobb-Douglas production function to produce goods, we described how labor and data can be combined to produce knowledge. Then, just as 20th-century economists estimated the exponents of the industrial production function using labor income shares, we similarly measure the exponents of the knowledge production function using wages and labor flows in a particular type of knowledge production, financial analysis. We find a substantial change in the production function. The magnitude is comparable to the smaller estimates of the change due to industrialization. Thus, describing this change as a new industrialization is a reasonable comparison.

Adoption of AI and big data technologies, as well as the accumulation of stocks of data vary widely by firm. The firms with more data are more prone to hire more big-data or AI workers. This supports the idea that this is a technology that is changing the factor mix of production. This finding matters for the income distribution: It changes the future labor share of income. In a model that did not have constant returns to scale, such a change would alter the optimal size of a firm: Firms with less diminishing returns to data may well take on a larger optimal size. It also suggests that financial knowledge will be significantly more abundant going forward.

Of course, this estimation was for workers doing one type of work in one sector. In other sectors, big data might be more or less of a change to output. It may also be too early to tell, since machine learning is not widely adopted in most other sectors. Much work in this area remains to be done to understand the magnitude and consequences of technological change in data processing. Hopefully, this paper provides an blueprint for how such future measurement might proceed.

References

Acemoglu, D., D. Autor, and J. Hazell (2019): "AI and Jobs: Evidence from Online Vacancies," Working paper, Massachusetts Institute of Technology.

Acemoglu, D., and P. Restrepo (2018): "Artificial Intelligence, Automation and Work," Working Paper 24196, National Bureau of Economic Research.

Aghion, P., B. F. Jones, and C. I. Jones (2017): "Artificial Intelligence and Economic Growth," Stanford GSB Working Paper.

Agrawal, A., J. McHale, and A. Oettl (2018): "Finding Needles in Haystacks: Artificial Intelligence and Recombinant Growth," in *The Economics of Artificial Intelligence: An Agenda*. National Bureau of Economic Research, Inc.

Alekseeva, L., J. Azar, M. Gine, S. Samila, and B. Taska (2020): "The Demand for AI Skills in the Labor Market," .

Allen, R. C. (2019): "Class structure and inequality during the industrial revolution: lessons from England's social tables, 1688–1867," *The Economic History Review*, 72(1), 88–125.

Babina, T., A. Fedyk, A. He, and J. Hodson (2020): "How Does Artificial Intelligence Affect Jobs? Evidence from US Firms and Labor Markets," Discussion paper, Working Paper.

Belo, F., V. Gala, J. Salomao, and M. Vitorino (2021): "Decomposing Firm Value," *Journal of Financial Economics*.

Belo, F., J. Li, X. Lin, and X. Zhao (2017): "Labor-force heterogeneity and asset prices: The importance of skilled labor," *The Review of Financial Studies*, 30(10), 3669–3709.

Belo, F., X. Lin, and S. Bazdresch (2014): "Labor hiring, investment, and stock return predictability in the cross section," *Journal of Political Economy*, 122(1), 129–177.

Brynjolfsson, E., T. Mitchell, and D. Rock (2018a): "What can machines learn, and what does it mean for occupations and the economy?," in *AEA Papers and Proceedings*, vol. 108, pp. 43–47.

——— (2018b): "What can machines learn, and what does it mean for occupations and the economy?," in *AEA Papers and Proceedings*, vol. 108, pp. 43–47.

Brynjolfsson, E., D. Rock, and C. Syverson (2017): "Artificial intelligence and the modern productivity paradox: A clash of expectations and statistics," Discussion paper, National Bureau of Economic Research.

Clark, G. (2005): "The condition of the working class in England, 1209–2004," *Journal of Political Economy*, 113(6), 1307–1340.

——— (2010): "The macroeconomic aggregates for England, 1209–2008," in *Research in economic history*. Emerald Group Publishing Limited.

Cockburn, I. M., R. Henderson, and S. Stern (2018): "The impact of artificial intelligence on innovation," Discussion paper, National bureau of economic research.

Crouzet, N., and J. Eberly (2020): "Rents and Intangible Capital: A Q+ Framework," Northwestern University Working Paper.

Davila, E., and C. Parlatore (2020): "Volatility and Price Informativeness," Working paper.

Deming, D. J., and K. L. Noray (2018): "Stem careers and the changing skill requirements of work," Discussion paper, National Bureau of Economic Research.

Dessaint, O., T. Foucault, and L. Fresard (2021): "Does Alternative Data Improve Financial Forecasting? The Horizon Effect," CEPR Discussion Papers 15786, C.E.P.R. Discussion Papers.

Dugast, J., and T. Foucault (2018): "Data abundance and asset price informativeness," *Journal of Financial Economics*, 130(2), 367–391.

——— (2020): "Equilibrium Data Mining and Data Abundance," Working Papers hal-03053967, HAL.

Edmans, A., I. Goldstein, and W. Jiang (2015): "Feedback Effects, Asymmetric Trading, and the Limits to Arbitrage," *American Economic Review*, 105(12), 3766–3797.

Eisfeldt, A. L., and D. Papanikolaou (2013): "Organization capital and the cross-section of expected returns," *The Journal of Finance*, 68(4), 1365–1406.

Farboodi, M., A. Matray, L. Veldkamp, and V. Venkateswaran (2020): "Where Has All the Data Gone?," Working Paper.

Farboodi, M., and L. Veldkamp (2019): "A Growth Model of the Data Economy," Working Paper, MIT.

Felten, E. W., M. Raj, and R. Seamans (2018): "Linking Advances in Artificial Intelligence to Skills, Occupations, and Industries," in *AEA Papers and Proceedings*.

Fuster, A., P. Goldsmith-Pinkham, T. Ramadorai, and A. Walther (2018): "Predictably unequal? the effects of machine learning on credit markets," *The Effects of Machine Learning on Credit Markets (November 6, 2018)*.

Glode, V., R. Green, and R. Lowery (2012): "Financial Expertise as an Arms Race," *Journal of Finance*, 67(5), 1723–1759.

Goldstein, I., E. Ozdenoren, and K. Yuan (2013): "Trading Frenzies and Their Impact on Real Investment," *Journal of Financial Economics*, 109 (2), 566–582.

Goldstein, I., and L. Yang (2019): "Good Disclosure, Bad Disclosure," *Journal of Financial Economics*, 131(1), 118–138.

Grennan, J., and R. Michaely (2018): "Fintechs and the market for financial analysis," *Michael J. Brennan Irish Finance Working Paper Series Research Paper*, (18-11), 19–10.

Jones, C., and C. Tonetti (2020): "Nonrivalry and the Economics of Data," *American Economic Review*, 110(9), 2819–2858.

Jones, C. I. (2005): "The Shape of Production Functions and the Direction of Technical Change," *Quarterly Journal of Economics*, 120(2), 517–549.

Karabarbounis, L., and B. Neiman (2014): "The Global Decline of the Labor Share," *Quarterly Journal of Economics*, 129(1), 61–103.

——— (2017): "Trends in factor shares: Facts and implications," *NBER Reporter*, (4), 19–22.

Klein, L. R., and R. F. Kosobud (1961): "Some econometrics of growth: Great ratios of economics," *The Quarterly Journal of Economics*, 75(2), 173–198.

Kogan, L., D. Papanikolaou, L. Schmidt, and J. Song (2021): "Technological Innovation and Labor Income Risk," Northwestern Working Paper.

Kung, H., and L. Schmid (2015): "Innovation, growth, and asset prices," *The Journal of Finance*, 70(3), 1001–1037.

Levenshtein, V. I. (1966): "Binary codes capable of correcting deletions, insertions, and reversals.," *Soviet physics doklady*, 10(8).

McGrattan, E. R. (2020): "Intangible capital and measured productivity," *Review of Economic Dynamics*.

Peters, R. H., and L. A. Taylor (2017): "Intangible capital and the investment-q relation," *Journal of Financial Economics*, 123(2), 251–272.

Porter, M. F. (1980): "An algorithm for suffix stripping.," *Program*.

Van Nieuwerburgh, S., and L. Veldkamp (2009): "Information immobility and the home bias puzzle," *Journal of Finance*, 64 (3), 1187–1215.

Webb, M. (2019): "The Impact of Artificial Intelligence on the Labor Market," *Available at SSRN 3482150*.

Not-for-Publication Appendix: Measurement Details, Model Derivations and Robustness Results

A Measurement

A.1 Categorizing Jobs

Jobs are first categorized into "data management" (DM) and "data analysis" by looking at the relative frequency of the "data management" vs. "data analysis" keywords listed below in the full text of the underlying job postings. Jobs identified as "data analysis" are further categorized (where possible) as AI or old technology (OT), by looking at the relative frequency of the AI and OT keywords listed below - these are subsets of the "data analysis" keywords.

All keywords lists are obtained by first tabulating all Burning Glass skills present in the selected sample and identifying skills that best map to the types of jobs described by the model. We then also inspected the text of selected job postings requiring most of the selected skills in order to refine the keywords and phrases to best reflect the format in which they are most frequently present in the text.

Before computing relative frequencies both the keywords lists and the underlying text are pre-processed and stemmed to their root using the Porter Stemming Algorithm.[16]

Data Management keywords : Apache Hive, Information Retrieval, Data Management Platform (DMP), Data Collection, Data Warehousing, SQL Server, Data Visualization, Database Management, Data Governance, Data Transformation, Extensible Markup Language (XML), Data Validation, Data Architecture, Data Mapping, Oracle PL/SQL, Database Design, Data Integration, Teradata, Database Administration, BigTable, Data Security, Database Software, Data Integrity, File Management, Splunk, Relational DataBase Management System, Teradata DBA, Data Migration, Information Assurance, Enterprise Data Management, SSIS, Sybase, jQuery, Data Conversion, Data Acquisition, Master Data Management, Data Capture, Data Verification, MongoDB, Data Warehouse Processing, SAP HANA, Data Loss Prevention, Data Engineering, Database Schemas, Database Architecture, Data Documentation, Data Operations, Oracle Big Data, Domo, Data Manipulation, Data Management Platform, DMP, HyperText Markup Language, Data Access Object (DAO), Structured Query Reporter, SQR, Data Dictionary System, Data Entry, Data Quality, Data Collection, Information Systems, Information Security, Change data capture, Data Management, Data Governance, Data Encryption, Data Cleaning, Semi-Structured Data, Data Evaluation, Data Privacy, Dimensional and Relational Modeling, Data Loss Prevention, Data Operations, Relational Database Design, Database Programming, Information Systems Management, Database Tuning, Object Relational Mapping, Columnar Databases, Datastage, Data Taxonomy, Informatica Data Quality, Data Munging, Data Archiving, Warehouse Operations, Solaris, Data Modeling, data feed management, data discovery, exporting large datasets, exporting datasets, database performance, disigning relational databases, implementing relational databases, designing and implementing relational databases, database development, data production process, normalize large datasets, normalize datasets, create database, Develop database, data onboarding, Data Sourcing, data purchase, data inventory, cloud Security, negotiating data, data attorney, data and technology

[16] This is a standard linguistic tool to remove common morphological and inflectional endings from words in English (Porter (1980)).

attorney, reliability engineering, reliability engineer, data specialist, enable vast data analysis, enable data analysis, Data team, capturing data, processing data, Supporting data, error free data sets, error free datasets, live streams of data, data accumulation, Kernel level development, large scale systems, Hadoop, distributed computing, multi database web applications, connect software packages to internal and external data, explore data possibilities, architect complex systems, build scalable infrastructure for data analysis, build infrastructure for data analysis, solutions for at scale data exploration, solutions for data exploration, information technology security, security engineer, security architect, architect solutions to allow modelers to process query and visualize higher dimensional data

Analysis keywords

- General Analysis: Regression Algorithms, Regression Analysis, Quantitative Analysis, Clustering, Time Series Analysis, Economic Analysis, Model Building, Quantitative Research, pandas, numpy, Hedging Strategy, Quantitative Data Analysis, Investment Analysis, Economic Models, Predictive Analytics, Market Trend, Portfolio Optimization, Portfolio Rebalancing, Financial Derivatives Pricing, Active Alpha Generation, Financial Data Interpretation, Alteryx, Predictive Models, Exploratory Analysis, Sensitivity Analysis, News Analysis, Asset Allocation, Research Methodology, Mathematical Software, Portfolio Construction, Portfolio Analysis, Portfolio Analyst, Market Analysis, Data Techniques, Capital Allocation, Financial Modeling, Algorithm Development, Securities Trading, Trading Strategy, Statistical Programming, Data Mining, Social Network Analysis, Dimensionality Reduction, Principal Components Analysis (PCA), Statistical Software, Portfolio Management, Numerical Analysis, Time Series Models, Asset Allocation Theory, Analytical Skills, Financial Analysis, Financial Modeling, Modern Portfolio Theory, MPT, Portfolio Valuation, strategic portfolio decisions
- Old Technology: Linear Regression, Logistic Regression, Statistic, STATA, Emacs, Technical Analysis, Qualitative Analysis, Qualitative Portfolio Management, Data Trending, Stochastic Optimization, Multivariate Testing, Bootstrapping, Time Series Models, Factor Analysis, Durations analysis, Markov, HMM, Econometrics, Stochastic Processes, Calculus, Statsmodels, Linear Algebra, Mathematics, Maths, Monte Carlo Simulation, Generalized Linear Model, GLM, Linear Programming, Bayesian, Analysis Of Variance, ANOVA, Behavioral Modeling, Black-Scholes, Behavior Analysis, Discounted Cashflow, Numerical Analysis, Correlation Analysis, E-Views, Differential Equations, Algebra, Value at Risk, Asset Pricing Models, Statistician, Mathematician, Econometrician
- AI: Artificial Intelligence, Machine Learning, Natural Language Processing, NLP, Speech Recognition, Gradient boosting, DBSCAN, Nearest Neighbor, Supervised Learning, Unsupervised Learning, Deep Learning, Automatic Speech Recognition, Torch, scikit-learn, Conditional Random Field, TensorFlow, Tensor Flow, Platfora, Neural Network, CNN, RNN, Neural nets, Decision Trees, Random Forest, Support Vector Machine, SVM, Reinforcement Learning, Torch, Lasso, Stochastic Gradient Descent, SGD, Ridge Regression, Elastic-Net, Text Mining, Classification Algorithms, Image Processing, Natural Language Toolkit, NLTK, Pattern Recognition, Computer Vision, Long Short-Term Memory, LSTM, K-Means, Geospatial Intelligence, Big Data Analytics, Latent Dirichlet Allocation, LDA, Backpropagation, Machine Translation, Caffe Deep Learning Framework, Word2Vec, Genetic Algorithm, Evolutionary Algorithm, Data Science, Sentiment Analysis / Opinion Mining, Maximum Entropy Classifier, Neuroscience, Computational Linguistics, Semi-Supervised Learning, Data Scientist

A.2 Identifying jobs for employers of interest:

To match the categorized job postings to the right employers, we use the following procedure:

1. *Create a master list of employers of interest:* We compile a comprehensive list of investment management companies using firms included in two data sources, Preqin and SEC. From Preqin, we select alternative asset managers. From the SEC database, we compile filers of Form 13F, a quarterly report of top ten equity holdings, filed by institutional investment managers with at least $100 million in assets under management. From the final list of firms we exclude commercial banks, insurance companies and private equity firms. To avoid repetitions, we manually cluster entities that refer to the same underlying company (e.g. "citigroup", "citigroup north america"). We then standardize these employers names to create a canonical form for each cluster that uniquely identifies it.[17]

2. *Extract candidate employers:* We use three techniques to identify strings that could potentially be the correct employer:

   (a) *Employer from BGT* - for a significant number of job postings, BGT identifies the employer using both manual and automated techniques. While it is not always available and can be incorrect, this employer will be added to our set of candidates.

   (b) *Keyword Search* - for each employer in the master list, we can generate keywords that identify this employer. We remove keywords that are too general and look for exact matches of these keywords in the job description. These matching words or phrases are added to our set candidates.[18]

   (c) *Named Entity Recognition – NER* - using part-of-speech tagging and word capitalization, we can identify words or phrases that are likely to be named entities (e.g. organization names, countries, people's names, etc.) from the job descriptions. These named entities, which potentially overlap with candidates from the previous methods, are added to our set of candidates.

3. *Standardize candidates and master list:* The first step to matching the candidates to the master list is to standardize employer names on both sides. Our standardization algorithm goes a step further than the basic cleaning applied to the job descriptions. The algorithm removes non-identifying suffixes and prefixes, such as leading the's and corporate designations such as "inc" and "llc". It also intelligently removes generic words (for instance "management" or "capital") only when they are not useful. For example, "The Blackstone Group" will become "blackstone" because "blackstone" is highly identifying. On the other hand, "Capital Group" will remain as "capital group" because stripping out "group" will reduce the employer name to an exceedingly common word "capital".

4. *Map raw candidates to master list:* After collecting a list of raw candidates for each job posting, we first de-duplicate the candidate set, then we compute a similarity score for every possible candidate-master employer pair. The computation of the similarity metric requires as input the frequency of all words appearing in any of the canonical names in the master list of employers.

[17] Job descriptions are scraped and therefore dirty. We remove excessive spaces and line breaks, unrecognizable symbols, HTML codes, and other irrelevant artifacts.

[18] These are also pre-processed, as previously outlined.

word	**frequency (F)**
capital	2,799
asset	745
advisors	684
...	...
sachs	2
vanguard	1

Said differently, for each candidate employer name we identify its optimal match from the master list ($master^*$) as the employer name in the master list with the highest similarity to the candidate of interest, such that:

$$master^* = \arg\max_{master} sim(candidate, master)$$

The computation of the similarity metric will be demonstrated using as an example a single candidate-master employer pair:

$$\text{candidate} = \text{“Royal Banks of Canada."}$$
$$\text{master} = \text{“royal bank canada"}$$

We begin by standardizing the candidate string, removing the period and uppercase in this case:

$$\text{“Royal Banks of Canada."} \rightarrow \text{“royal banks of canada"}$$

Next, we perform a word by word matching between the candidate and master list entry of interest. We compute the Levenshtein similarity (modified to give a slight bonus to exact matches) between any pairs of words and select the pairing which yields the highest similarity.[19] In our example, the words “royal” and “canada” are exact matches, the words “banks” and “bank” obtain a 70% similarity, while the word “of" remains unmatched.

$$\begin{array}{rcccc} sim: & 1.0 & 0.7 & 1.0 & 0 \\ \text{matches:} & \text{(royal, royal),} & \text{(banks, bank),} & \text{(canada, canada),} & \text{(of,)} \end{array}$$

Lastly, we take the weighted sum of all the matched words, we use as weighs the inverse frequency of each word in the whole corpus. We set a minimum weight of 0.1 for matched words, to avoid shrinking the weights of common words to near 0.

$$\begin{array}{rcccc} sim: & 1.0 & 0.7 & 1.0 & 0 \\ \text{matches:} & \text{(royal, royal),} & \text{(banks, bank),} & \text{(canada, canada),} & \text{(of,)} \\ weight: & F(royal)^{-1} & F(bank)^{-1} & F(canada)^{-1} & F(of)^{-1} \end{array}$$

$$sim(\text{“Royal Canada Bank.", “royal bank of canada"}) = 0.867$$

$$sim_{Levenshtein}(\text{“Royal Canada Bank.", “royal bank of canada"})\text{=}0.55$$

We only consider matches above a threshold of 0.75. In this example, the final similarity score of our algorithm is 0.867. That is high, given that half the words have non-exact matches. That is because the frequency in the master list of the words “bank” and “of” relative to “royal” and “canada” is high;

[19] Levenshtein similarity is a standard algorithm utilized in linguistics to compute the similarity between two words. It is based on assessing the number of insertions, deletions or substitutions required to change one word into the other (Levenshtein (1966))

hence they are down-weighted in the similarity score computation. This match, instead, would have been discarded using for instance Levenshtein similarity alone (0.55 similarity).

Our matching algorithm is inspired by how a human would approach the problem. If we encounter "canandian bank" and "bank of canada" for example, it is natural to associate highly similar words and compare locally. This gives us an edge over direct applications of traditional metrics such as the Levenshtein distance.

When a valid match cannot be identified with the above method we perform a final attempt to identify a valid candidate. Given that our algorithm compares words one by one, with a bonus for exact matches, minor typos or spacing issues can cause an otherwise obvious (to the human eye) match to achieve a score below our minimum threshold and be left out. To salvage these edge cases as much as possible, we use the following heuristic:

<u>Example</u>

candidate = "Royale Bank ofCanada inc."
master = "royal bank of canada"

(a) standardize candidate employer

candidate: "Royale Bank ofCanada inc." → "royale bank ofcanada"

(b) remove spaces from both candidate and master to form a single word

candidate: "royale bank ofcanada" → "royalebankofcanada"
master: "royal bank of canada" → "royalbankofcanada"

(c) compute the Levenshtein distance between these two single words and accept matching if similarity score is greater than 0.9.

$sim_{Levenshtein}$("royalebankofcanada", "royalbankofcanada")=0.945

5. *Select the best candidate* At this point, we have a list of candidates for each job posting. All the candidates correspond exactly to a single employer in our master list. The final step is to select the most likely candidate. To evaluate the quality of each candidate, we took into consideration three types of features:

- Consistency - is the same candidate identified using multiple methods (Burning Glass Employer, text search, Named Entity Recognition)?
- Similarity to employer in master - how similar is the raw candidate to the employer in the master list?
- Frequency - for the methods that use the full job posting (text search and Named Entity Recognition), how frequently was the employer mentioned in the text?

More specifically, when the Burning Glass employer is corroborated by at least one of the other two methods, we assign that job posting to the matched employer. When a Burning Glass employer is not corroborated by any of the other two methods, we accept the match only if it is exact (excluding all fuzzy matches). Finally, when the Burning glass employer is not present but both text-based methods agree, we assign that job to the matched employer only if it is an exact match and the employer name is repeated at least 3 times in the text. This reduces matching noise. We do not consider matches that only appear in one of the text-based methods, as too noisy.

If after following this procedure the candidate set is empty, we decide that the true employer is not found in our master list. If the candidate set contains more than one element, we pick the candidate with the highest similarity score.

A.3 Constructing the Labor Inflows Data

Job openings, filling and separation data Our data comes from

`https://www.bls.gov/news.release/jolts.tn.htm`

Job Openings Rate: Job openings information is collected for the last business day of the reference month. A job opening requires that: 1) a specific position exists and there is work available for that position, 2) work could start within 30 days whether or not the employer found a suitable candidate, and 3) the employer is actively recruiting from outside the establishment to fill the position. The job openings rate is computed by dividing the number of job openings by the sum of employment and job openings and multiplying that quotient by 100.

Hiring Rate: The hires level is the total number of additions to the payroll occurring at any time during the reference month, including both new and rehired employees, full-time and part-time, permanent, short-term and seasonal employees, employees recalled to the location after a layoff lasting more than 7 days, on-call or intermittent employees who returned to work after having been formally separated, and transfers from other locations. The hires rate is computed by dividing the number of hires by employment and multiplying that quotient by 100.

Separations Rate: The separations level is the total number of employment terminations S occurring at any time during the reference month, and is reported by type of separation - quits, layoffs and discharges, and other separations. The separations rate is computed by dividing the number of separations by employment and multiplying that quotient by 100: $s = S/E \cdot 100$.

Deriving the probability of filling an opening. If n_O is the total number of posted job openings, n_E is total employment and n_H is the number of new hires in this sub-occupation and month, then the BLS hiring rate is defined to be $r_h = n_H/n_E$, while the job opening rate is $r_o = n_O/(n_E + n_O)$. What we need to adjust the openings data from our model, is the fraction of openings that result in hires, $h = n_H/n_O$.

To solve for h, note that rearranging the definition of the opening rate yields $r_o = (1 - r_o)n_O/n_E$. Dividing r_h by this expression yields $r_h/r_o = (n_H/n_E)/((1-r_o)n_O/n_E) = (n_H/n_O)\cdot 1/(1-r_o)$. Therefore, we can express the n_H/n_O rate we want as $h = r_h(1 - r_o)/r_o$.

Time to Fill a Job Vacancy In our calculations, we have implicitly equated a job posting with a one-month job vacancy. We do that because most of our job postings remain up and unfilled for approximately one month. Below, we report the distribution of the average time that job postings remain open in our data set. This data is for jobs that have the same occupations and regions as our sample for the years 2015, 2016 and 2017. The average time to fill is available for 86% of all the occupation (SOC) - region (MSA) combinations in our sample. Below is the distribution of the average time a Burning Glass job posting stayed online for all the SOC-MSA combinations in our sample for 2015-2017.

If we weight each of these fill times by the number of jobs present in our sample for each the SOC-MSA combinations, we get an average fill times of 38.12 days.

mean	35.6857
std	7.1003
min	14.0000
1%	21.0000
5%	24.0000
10%	27.0000
15%	28.0000
20%	30.0000
25%	31.0000
30%	32.0000
35%	33.0000
40%	34.0000
45%	35.0000
50%	35.0000
55%	36.0000
60%	37.0000
65%	38.0000
70%	39.0000
75%	40.0000
80%	41.0000
85%	43.0000
90%	44.4000
95%	48.0000
99%	54.0000
max	75.0000

Table 3: **Time to Fill Posted Vacancies**.

A.4 Estimation procedure details

Two additional minor details of how we estimated the model.

When we estimate the AI labor first order condition, we use only firms that employ some AI workers and some data management workers. Requiring that the firm currently employs a type of worker does not imply they hired someone that month. Rather, it means that some worker was hired at some time in the past. If we do not exclude these firms, our production exponent estimate would be heavily influenced by the many observations with zero labor and abundant data, or vice-versa. Similarly, when we estimate the traditional labor first order condition, we use only observations from firms that have, at some point, hired a data manager and a traditional analyst.

We normalize the estimated productivity terms by dividing all AI firm-specific productivity parameters by $\sum_{i=1}^{N} a_i^{AI}$ and multiplying the AI time-varying productivity parameters (A_t^{AI}) by the same quantity. Similarly, we divide all OT firm-specific productivity parameters by $\sum_{i=1}^{N} a_i^{OT}$ and multiply the OT time-varying productivity parameters (A_t^{OT}) by the same quantity. This normalization has no effect on the parameter estimates (α, γ, ϕ and $\bar{D_0}$) but it ensures that $\sum_{i=1}^{N} a_i^{AI} = 1$ and $\sum_{i=1}^{N} a_i^{OT} = 1$.

B Wages

In order to identify and categorize PayScale salary profiles we proceed as follows.

First, we obtain all of PayScale salary profiles from 2010 to 2020 in any of the four industries present in our BurningGlass dataset: 'Professional, Scientific, and Technical Services', 'Finance and Insurance', 'Information' and 'Management of Companies and Enterprises'. The data contains a unique identifier for each survey, user and employer, this helps us to clean it and subset it to our sample of interest.

We start by cleaning the data by tracking multiple responses by the same user. When multiple profiles exist by the same user in the same day, we check them for consistency and then keep the one with the most complete information.

Next, we sub-set the data to candidate salary profiles of interest. We only keep jobs for which the list of top three skills is present. Then, we keep all jobs in the 'Finance and Insurance' industry. Additionally we identify all employers who hire predominantly in the 'Finance and Insurance' industry and keep all jobs for those employers in any of the four industries mentioned above. The idea of this second filter is to mimic the structure of our BurningGlass dataset. In fact, in BurningGlass, the majority of jobs postings for our firms of interest are categorized as belonging to the Finance industry. Even though, many of the jobs in the three categories we are interested in are classified as 'Information' or 'Professional, Scientific, and Technical Services'. Similarly to what we had done with BurningGlass, we further exclude all jobs with O*NET codes in the Insurance industry.

Note that, the employer field is not mandatory, so it is present for roughly 60% of observations. Hence, with this filtering we might be excluding some relevant AI and Old Tech analysts jobs categorized in the 'Information' or 'Professional, Scientific, and Technical Services' industries for which the employer had not been disclosed. At the same time, keeping all jobs in those industries would introduce substantial noise in our selection. Considering that our objective is to obtain an equilibrium salary for each job type each month, we opted for a more conservative approach and only included jobs that had a high likelihood of belonging to employers of interest.

Then, we map the identified jobs according to a number of additionally provided fields (employer, O*NET code, job title, employer type and education level) and remove non-investment management jobs based on those criteria. The main categories of eliminated jobs include customer service, sales, administrative, human resources and actuarial jobs. We further exclude jobs that do not require a degree or that only require a high school diploma a non-degree certificate program or a medical degree (common in insurance jobs). We additionally exclude jobs by non-firm employers (e.g. government, universities, military). We finally identify all remaining insurance companies and eliminate all associated salary profiles.

Our last step is to identify salary profiles belonging to our job categories of interest, within this sub-set. We do so by utilizing the provided list of top three skills required by each job. We utilize a similar method as that used to categorize job postings. Indeed, we look at the relative frequency of skills mentioned belonging to the skills dictionaries previously compiled. We use the same dictionaries as utilized in the job postings categorization, with some minor adjustments to account for the different nature of the text (skills list vs. paragraphs). With this procedure we are able to identify 5,639 Data Management profiles, 2,817 Old Tech jobs and 2,585 AI ones.

C Model derivations

Firm i faces the following optimizing problem:

$$v(D_{it}) = \max_{\lambda_{it}, L_{it}, l_{it}} A_t^{AI} a_i^{AI} D_{it}^{\alpha} L_{it}^{1-\alpha} + A_t^{OT} a_i^{OT} D_{it}^{\gamma} l_{it}^{1-\gamma} - w_{L,t} L_{it} - w_{l,t} l_{it} - w_{\lambda,t} \lambda_{it} + \frac{1}{r} v(D_{i(t+1)}) \tag{22}$$

$$\text{where } D_{i(t+1)} = (1-\delta) D_{it} + \lambda_{it}^{1-\phi}. \tag{23}$$

Here the state variable is structured data D_{it}, and the control variables are data management labor λ_{it}, the machine learning analyst labor L_{it} and the old technology analysis labor l_{it}. Plugging (23) into (22), we have

$$v(D_{it}) = \max_{\lambda_{it}, L_{it}, l_{it}} A_t^{AI} a_i^{AI} D_{it}^{\alpha} L_{it}^{1-\alpha} + A_t^{OT} a_i^{OT} D_{it}^{\gamma} l_{it}^{1-\gamma} - w_{L,t} L_{it} - w_{l,t} l_{it} - w_{\lambda,t} \lambda_{it} + \frac{1}{r} v\left((1-\delta) D_{it} + \lambda_{it}^{1-\phi}\right) \tag{24}$$

Taking partial derivative with respect to L_{it}, we have

$$(1-\alpha) A_t^{AI} a_i^{AI} D_{it}^{\alpha} L_{it}^{-\alpha} - w_{L,t} = 0 \implies \frac{(1-\alpha) K_{it}^{AI}}{L_{it}} = w_{L,t}. \tag{25}$$

Taking partial derivative with respect to l_{it}, we have

$$(1-\gamma) A_t^{OT} a_i^{OT} D_{it}^{\gamma} l_{it}^{-\gamma} - w_{l,t} = 0 \implies \frac{(1-\alpha) K_{it}^{OT}}{l_{it}} = w_{l,t}. \tag{26}$$

Taking the partial derivative with respect to λ_{it} and rearranging, we have

$$\frac{1}{r} v'(D_{i(t+1)})(1-\phi) \lambda_{it}^{-\phi} = w_{\lambda,t}. \tag{27}$$

We then totally differentiate (24) to get

$$v'(D_{it}) = \frac{\alpha K_{it}^{AI}}{D_{it}} + \frac{\gamma K_{it}^{OT}}{D_{it}} + \frac{1}{r} v'(D_{i(t+1)})(1-\delta). \tag{28}$$

If we further assume that the marginal value of data today and tomorrow are similar, then

$$v'(D_{it}) \approx \frac{(\alpha K_{it}^{AI} + \gamma K_{it}^{OT})}{D_{it}} \frac{r}{r-(1-\delta)}. \tag{29}$$

Plugging it back to the first order condition (27) and combining it with the structured data dynamics (23), we arrive at

$$\frac{(\alpha K_{it}^{AI} + \gamma K_{it}^{OT})(1-\phi)}{r-(1-\delta)} \frac{D_{i(t+1)} - (1-\delta) D_{it}}{D_{it}} = w_{\lambda} \lambda_{it}. \tag{30}$$

D Robustness

We repeated the structural estimation by varying the construction method of labor stock variables, the data depreciation rate, and the productivity parameters structure. The following sections illustrate the different variables and methods utilized for the estimation, while Table 4 summarizes all estimates. We finally provide a graphical representation of our baseline model for different depreciation rates.

D.1 Labor stock cumulation

In our baseline results, AI, old tech and data management labor demand were cumulated into labor stock starting from a stock of 0 workers in 2010. In this section we provide an alternative method for cumulating old tech and data management labor: we start cumulating from steady-state values in 2010. That implies that firms have already reached the optimal amount of workers of each type and will hire only for replacement. These two cumulation methods should be considered as two opposite extremes. AI labor is always cumulated starting from zero in 2010, as very few AI hires are present in this industry at that time.

In the steady-state cumulation, firms are added at steady-state level as of the first month they appear in 2010. Steady-state is computed as the number of Old tech (data management) job postings in 2010 times the ratio between the filling rate and the separations rate for that specific job type (obtained as described in Section A.3). Firms with no job postings in 2010 are considered as new firms when they first appear in following years and are cumulated from zero.

As can be seen from Figure 7, the two cumulation methods tend to converge after 2015.

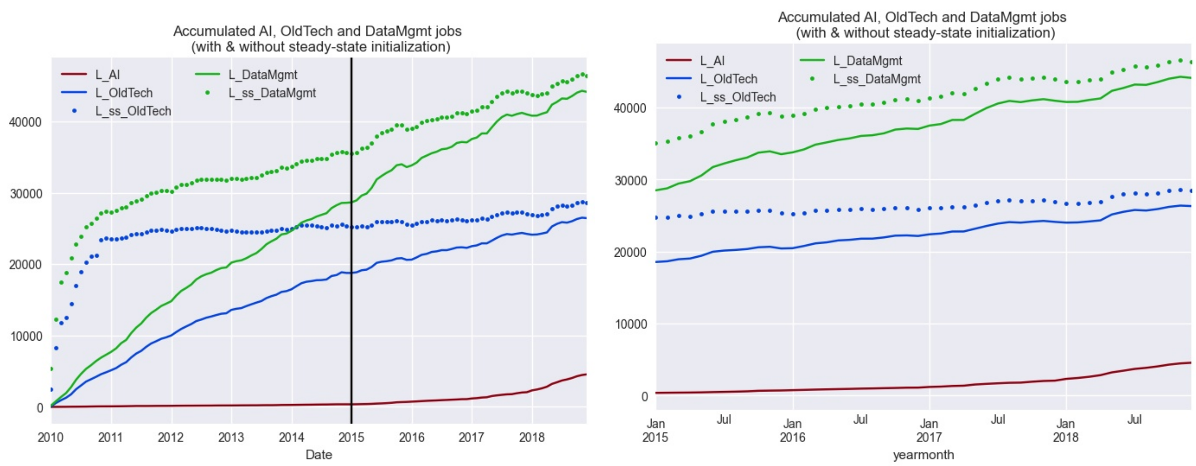

Figure 7: Labor variables cumulated across all firms each month since Jan 2010 (Panel 1) and since Jan 2015 (Panel 2).

D.2 Depreciation Rate

Estimations are repeated for different levels of data depreciation δ between 1% and 10%. Table 4 only report estimates for $\delta = [1\%, 3\%, 10\%]$. Estimates are available for other values of δ.

D.3 Estimation methods

Four different estimation methods were utilized depending on the productivity parameters settings:

1. **None:** No productivity parameters were added to the Cobbb-Douglas production functions.

$$K_{it}^{AI} = D_{it}^{\alpha} L_{it}^{1-\alpha}; \quad K_{it}^{OT} = D_{it}^{\gamma} L_{it}^{1-\gamma} \tag{31}$$

2. **Time:** Time-varying productivity parameters were added to the Cobbb-Douglas production functions. They are computed as cross-sectional averages of the AI and old tech FOCs for all relevant firms each month (t).

$$K_{it}^{AI} = A_t^{AI} D_{it}^{\alpha} L_{it}^{1-\alpha}; \quad K_{it}^{OT} = A_t^{OT} D_{it}^{\gamma} L_{it}^{1-\gamma} \tag{32}$$

3. **Firm:** Firm-specific productivity parameters were added to the Cobbb-Douglas production functions. They are computed as time-series averages of the AI and old tech FOCs for each firm (i).

$$K_{it}^{AI} = a_i^{AI} D_{it}^{\alpha} L_{it}^{1-\alpha}; \quad K_{it}^{OT} = a_i^{OT} D_{it}^{\gamma} L_{it}^{1-\gamma} \tag{33}$$

4. **FirmTime:** Both time-varying and firm-specific productivity parameters were added to the Cobbb-Douglas production functions. They are computed as cross-sectional and time-series averages of the AI and old tech FOCs respectively. Since they are a function one of the other an iterative process is utilized to reach convergence. The solution is scale-invariant, hence all productivity parameters are normalized such that $\sum_i a_i^{AI} = 1$ and $\sum_i a_i^{OT} = 1$.

$$K_{it}^{AI} = A_t^{AI} a_i^{AI} D_{it}^{\alpha} L_{it}^{1-\alpha}; \quad K_{it}^{OT} = A_t^{OT} a_i^{OT} D_{it}^{\gamma} L_{it}^{1-\gamma} \tag{34}$$

E Structural Estimates

Table 4 summarizes the estimated (α, γ, ϕ and $\bar{D_0}$) utilizing the different labor stock variables described in Section D.1, depreciation rates described in Section D.2 and productivity parameters combinations described in Section D.3. The baseline model considered throughout the paper (and highlighted in blue) utilizes *FirmTime* productivity, $\delta = 3\%$ and zero labor variables initialization.

Productivity	δ	α	γ	ϕ	$\bar{d}_0$	$(\alpha - \gamma)$
			Zero Initialization of Labor Stock			
None	0.01	0.9188	0.6992	0.2979	737	0.2196
		(0.0019)	(0.0059)	(0.0115)	(36)	
None	0.03	0.8777	0.8321	0.4992	191	0.0457
		(0.0034)	(0.0046)	(0.0132)	(8)	
None	0.1	0.6909	0.6403	0.01	500	0.0506
		(0.007)	(0.0081)	(0.032)	(103)	
Time	0.01	0.895	0.631	0.1525	1417	0.264
		(0.0005)	(0.0018)	(0.0012)	(15)	
Time	0.03	0.7993	0.7073	0.1469	798	0.0919
		(0.0008)	(0.0008)	(0.0009)	(5)	
Time	0.1	0.7165	0.675	0.1422	223	0.0415
		(0.0012)	(0.0004)	(0.0009)	(1)	
Firm	0.01	0.908	0.6656	0.2368	929	0.2424
		(0.0011)	(0.0032)	(0.0064)	(28)	
Firm	0.03	0.818	0.7199	0.2212	496	0.0981
		(0.0015)	(0.002)	(0.0061)	(15)	
Firm	0.1	0.6886	0.6392	0.01	482	0.0494
		(0.0025)	(0.0027)	(0.001)	(34)	
FirmTime	0.01	0.9159	0.6904	0.2798	792	0.2255
		(0.0011)	(0.0033)	(0.0063)	(22)	
FirmTime	0.03	0.8667	0.8157	0.4531	220	0.051
		(0.0015)	(0.002)	(0.0058)	(4)	
FirmTime	0.1	0.6898	0.6405	0.01	500	0.0493
		(0.0027)	(0.0032)	(0.0085)	(40)	
			Steady-State Initialization of Old Tech and Data Management Labor Stock			
FirmTime	0.01	0.8943	0.6159	0.1197	2145	0.2784
		(0.0013)	(0.0035)	(0.0072)	(80)	
FirmTime	0.03	0.8299	0.7492	0.2579	626	0.0807
		(0.0016)	(0.0023)	(0.0066)	(19)	
FirmTime	0.1	0.7009	0.6389	0.01	625	0.062
		(0.0028)	(0.0032)	(0.006)	(48)	

Table 4: **Model Estimates:** Structural estimates of model parameters for different productivity parameters settings utilizing average Payscale salaries and initializing labor variables from zero in 2010 or from steady-state in 2010. All models are estimated for the depreciation rate of data $\delta = [1\%, 3\%, 10\%]$. The baseline model is highlighted in blue.