Inter-Entity Bookkeeping Networks: Representations and Applications*

Jing Li
School of Business
University of Hong Kong
acjli@hku.hk

Pierre Jinghong Liang
Tepper School of Business
Carnegie Mellon University
liangj@andrew.cmu.edu

Hyun Hwang
Tepper School of Business
Carnegie Mellon University
hyunh@andrew.cmu.edu

(Preliminary)
April 2019

Abstract

In this paper, we explore the representation of accounts and transactions of multiple entities when (1) each entity practices double-entry bookkeeping and (2) inter-entity transactions take place. As such, entity-to-entity links created by "real" inter-entity transactions and account-to-account links within an entity created by "artificial" double-entry bookkeeping are integrated into an account network through the well-known quadruple-entry bookkeeping of the inter-entity transactions: each inter-entity transaction is recorded in four accounts (two within each entity). We focus on loops in such an account network and distinguish loops created from different accounting events (such as cash transactions, non-cash inter-entity transactions and non-cash intra-entity transactions) and how these loops are affected when the account-network expands. Applications of the representation and some results from simulations are also presented.

Keywords: double entry bookkeeping

*This document is prepared specifically for 2019 UCSD Workshop. Please do not distribute beyond this use. The General Research Fund of Hong Kong financially supports a larger research project on which this paper is based. We are grateful for encouraging comments from R. Ravi. All errors are our own. The authors gratefully acknowledge financial support from the General Research Fund of Hong Kong and the University of Hong Kong. Pierre Jinghong Liang also has research affiliations with Shanghai Advanced Institute of Finance and China Academy of Financial Research at Shanghai Jiaotong University.

1 Introduction

Double-entry bookkeeping remains a foundation of the financial infrastructure in any modern organization. Not surprisingly, it is one of favorite research topics of many scholars, including Professor Yuji Ijiri of Carnegie Mellon. Among many of his research interests, double-entry bookkeeping occupies a special place within Ijiri's work, spanning its underlying algebraic foundation (see Ijiri [1965]) to its poetic beauty (see Ijiri [2014]). High praises of the bookkeeping invention have been attributed to Johann Wolfgang von Goethe and Arthur Cayley. Its important role in the rise of capitalism has been raised by Sombart, Weber, and Schumpeter.

As recognized long ago, a deep connection between linear algebra and double-entry bookkeeping exists. In recording each transaction in two accounts, the double-entry system links all accounts of an entity together and in the process creates laws that governs the relation between the transactions and account balances. Such laws can be represented as properties of a matrix or, equivalently, as properties of a graph, as succinctly summarized by a famous theorem from Leonhard Euler, according to Professor John Fellingham in Fellingham [2018].[1]. Beyond its elegance, the structure proves useful in a varieties of scenarios. For example, one such use is its economic function in providing information. That is, the structure can be thought of as part of an information source for an economic decision-making purpose, as envisioned by Butterworth [1972]. In Arya et al. [2000a], a specific inference problem was formulated to assess the role of double-entry bookkeeping structure. See related work in Arya et al. [2000b] and Arya et al. [2004].

In this paper, we explore the representation of accounts and transactions of multiple entities when (1) each entity practices double-entry bookkeeping and (2) inter-entity transactions take place. As such, two types of links arise. Within each entity, every transaction links two accounts together. This link is "artificial" in the sense that they are the result of entity simply recording the transaction on its own (using such a double-entry bookkeeping system). Between entities, inter-entity transactions links two entities together. This link is "real" in the sense two entities decide to transact with each other for some "real" economic reason independently of their recording. As such, each inter-entity transaction is recorded four times (two within each entity's books). Combining these two types of links, accounts from both entities are integrated into an expanded account network through the quadruple-entry bookkeeping of the inter-entity transactions.

This expanded bookkeeping network is the subject of our study. We first explore the properties of the bookkeeping system such as the four fundamental spaces. We then focus on loops in the bookkeeping network and distinguish loops created from different accounting events (such as cash transactions, non-cash inter-entity transactions and non-cash intra-entity transactions) and how these loops are affected when the account-network expands. Applications of the representation and some results from simulations are also presented.

By studying the quadruple entry bookkeeping through its matrix representation, we show that introducing inter-entity transactions may fundamentally alter the properties of the bookkeeping process of transforming transactions into accounts. In particular, the dimension of the left-nullspace of the quadruple entry matrix may be greater than two (2) and the dimension of the null-space of the matrix may be less than the sum of the dimensions of the two individual double-entry matrices. Correspondingly, by studying its graph representation, we show visually that the loops in the double-entry graph can be broken while new loops can be

[1] On page 1, Professor Fellingham states that "One way to describe a general result is a famous theorem from Leonhard Euler: The number of nodes minus one plus the number of enclosed regions equals the number of arcs (see, for example, Trudeau, 1993). Another way is to use accounting words: The number of T-accounts minus one plus the number of loops equals the number of journal entries. There is also a linear algebraic expression about the matrix underlying the system: The dimension of the row space plus the dimension of the null space equals the number of columns in the matrix."

formed in the quadruple entry graph. We apply the quadruple entry bookkeeping to an existing application, namely inferring transaction from financial statements as introduced in Arya et al. [2000a]. In particular, we add account information from another nearby entity who also practice double-entry bookkeeping and engage in inter-entity transactions with the entity of intersts. We investigate how such additional improve the inference activity. Not surprisingly, the improvement depends on the nature of account structure of the nearby entity. Through numerical exampes and associated simulation, a few observations emerge. First, a spanning-tree structure of the nearby entity helps the inference program greatly, especially the inter-entity transaction is involved in one or multiple pre-existing loops in the double-entry graph of the original entity. Further, user's priors may also play a significant role. For example, bringing in a nearby entity whose inter-entity transactions have a higher prior uncertainty improve inference more effectively.

The paper process with Section 2 reviewing the properties of the double-entry bookkeeping in both matrix and graph form as well the idea of the transaction loops corresponding to the nullspace of the double-entry matrix. Section 3 develops the Quadruple entry bookkeeping in its matrix and graph representations and explores its effect on the transaction loops and other properties introduced in section 2. Section 4 presents an inference-type application and present examples and simulation results illustrating our main findings.

2 Double Entry Bookkeeping

2.1 Matrix Representation

We adopt the notation used in Arya et al. [2000a].

- Transaction vector y is an $n \times 1$ column vector
- Change in accounting balance vector x is an $m \times 1$ column vector with $m \leq n$
- Transformation matrix A is an $m \times n$ matrix representing Double-entry Bookkeeping

$$x = Ay$$

Here is an example of the matrix representation of double-entry bookkeeping.

$$\begin{bmatrix} \text{Cash} & 10 \\ \text{Inventory} & 0 \\ PPE & 0 \\ \text{Sales} & 0 \\ COGS & 0 \\ G\&A & 0 \end{bmatrix} + A \cdot \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 6 \\ -10 \\ 5 \\ 3 \end{bmatrix}$$

x_0 : opening balances
x_1 : ending balances
y : Transactions
$x = x_1 - x_0$: changes in account balances
A : Transformation matrix double entry

$$\Rightarrow \boxed{\underset{m\times 1}{x} = \underset{m\times n}{A} \cdot \underset{n\times 1}{y}}$$

$$A = \begin{array}{c} \begin{array}{cc} (y_1 - y_4) & (y_5 - y_7) \\ \text{Cash Transactions} & \text{Deferrals \& Accruals} \\ \longleftarrow - - - - - - \longrightarrow & \longleftarrow - - - \longrightarrow \end{array} \\ \begin{vmatrix} -1 & -1 & -1 & +1 & 0 & 0 & 0 \\ +1 & 0 & 0 & 0 & -1 & +1 & 0 \\ 0 & +1 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 & 0 & 0 \\ 0 & 0 & +1 & 0 & 0 & 0 & +1 \end{vmatrix} \end{array}$$

2.2 Graph Representation

It is well-known that the incidence matrix has a directed graph representation (see the example in Figure 1). We can call this an double-entry bookkeeping network.

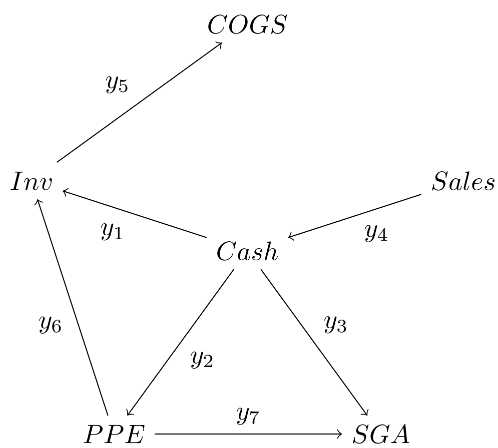


Figure 1: Example: A Double-entry Bookkeeping Graph (from Arya et al. [2000a])

2.3 Equivalences

- **Journal entries, A's columns and edges in graph** Each journal entry is either presented by an column of the A matrix or equivalently, a directed edge in the graph.
- **Null spaces and loops** Let $Ay' = 0$. Then if y is a solution to $Ay = x$, then $y + \alpha y'$ is also a solution. In the language of liner algebra, y' lies in the null space of A with dimension $n - (m - 1)$

  Using the example above, these are two independent combinations of transactions that produce zero change in Account balances,

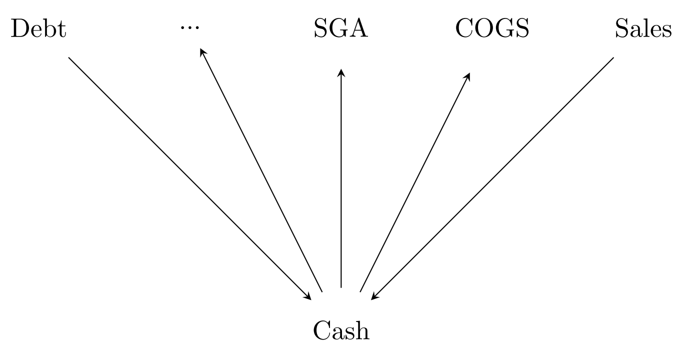
  - $\langle -y_1, y_2, y_6 \rangle$
  - $\langle y_2, -y_3, y_7 \rangle$

  any combination of the above two combinations generates zero change in balance. The existence of loops in the bookkeeping network (or equivalently the existence of null space) is the source of information loss in conveying transaction information using account balances.

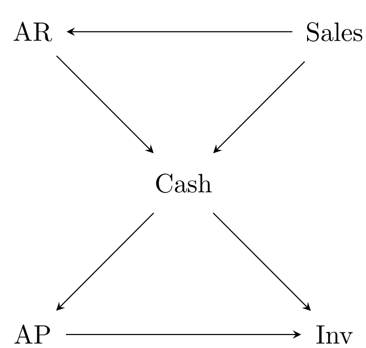
2.3.1 Sources of Loops in Bookkeeping Network

The main representation idea is that bookkeeping account balances are treated as x-vectors in the matrix-representation and, equivalently, nodes in the graph-representation while bookkeeping transactions are treated as y-vectors in the matrix representation and, equivalently, the directed edges in the graph-representation. Now we construct the connections between some bookkeeping features and null spaces of the A-matrix or the loops in the graph. Consider the following simple and intuitive observations.

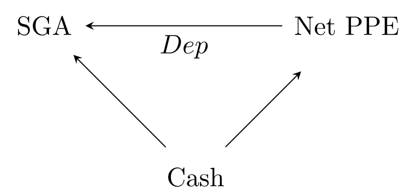
- No Loops in **Cash-based Bookkeeping**. Under cash-based double-entry bookkeeping, only cash transactions are recognized and recorded. Accordingly, in the corresponding bookkeeping network, all non-cash accounts (nodes) are only connected to the cash account (node); so no loops are created in the bookkeeping network. This makes sense. Intuitively, this bookkeeping system allows readers to perfectly infer the total (or net) transactions recorded under cash-based bookkeeping. For example, when sales are recorded on cash-basis, the sales (trial-) balance at the end of the period communicate perfectly the total cash sales during the period. No information is lost due to bookkeeping.



- Loops created by **external transactions** in Accrual-based Bookkeeping. Under accrual-based system, a non-cash transaction involving an external party, combined with two existing cash transactions, generates a loop. For example, a loop among the cash, accounts receivable, and sales revenue emerges if the reporting entity aggregates both credit and cash sales into a single reported Sales Revenue financial statement line-item (FSLI). Intuitively, an outside reader is unable to pin down transaction volume from only knowing the three account balances. Some information is lost due to bookkeeping.

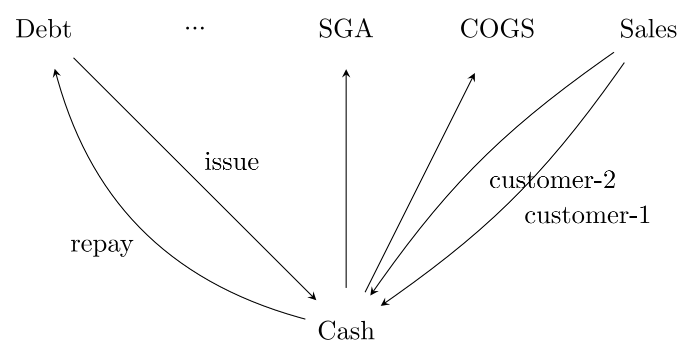


- Loops created by **internal transactions** in Accrual-based Bookkeeping. Under accrual-based system, a non-cash transaction not involving an external party, combined with two existing cash transactions, generates a loop. For example, a loop among the cash, net PPE, and SGA emerges if the recording entity aggregates both cash SGA and non-cash SGA into a single reported SGA FSLI. Intuitively, an outside reader is unable to pin down transaction volume from only knowing the three account balances. Some information is lost due to bookkeeping.

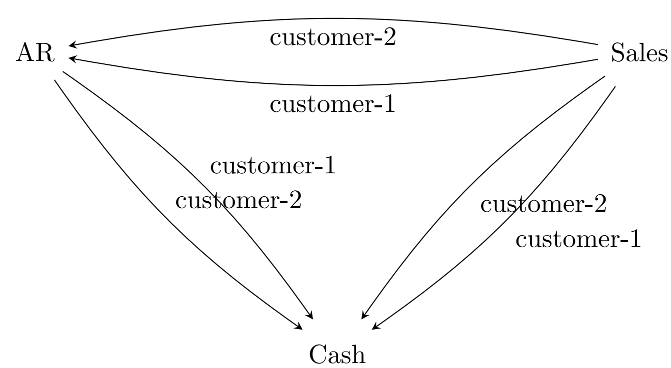


Things become more complicated when the users' interest lies in learning transactions at a more disaggregated level.

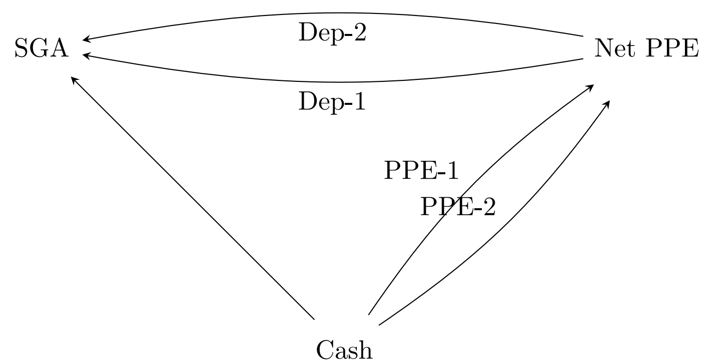
- Loops in **Cash-based Bookkeeping**. Suppose the users are interested in learning cash sales to customer 1 separately from customer 2, or interested in learning the debt issues separately from debt repayments. Then the aggregated nature of Sales Revenue or Debt FSLI, even under cash-based bookkeeping, would generate loops.



- More Loops created by **external transactions** in Accrual-based Bookkeeping. Suppose the users are interested in learning cash and credit sales to customer 1 and 2. Then the aggregated nature of Sales Revenue FSLI, Accounts Receivable would generate more loops.



- More Loops created by **internal transactions** in Accrual-based Bookkeeping. Suppose the users are interested in learning depreciation amount from one types of PPE and another types of PPE to the SGA. Then the aggregated nature of SGA FSLI and net PPE would generate more loops.



3 Quadruple Entry Bookkeeping

3.1 Expansion of double entry to Quadruple entry

Now we expand the setting to include more than one accounting entity. Naturally, each entity's bookkeeping system is governed separately by a self-contained double-entry system, i.e., its own $\{x, y, A\}$ triplet. To illustrate with two entities, a and b, we expand notation accordingly

- Transaction vector y^i is an $n_i \times 1$ column vector for entity $i \in \{a, b\}$
- Change in accounting balance vector x^i is an $m_i \times 1$ column vector entity $i \in \{a, b\}$
- Transformation matrix A is an $m_a \times n_a$ matrix for entity a and similarly for the $m_b \times n_b$ matrix B.

$$\begin{aligned} x^a &= Ay^a \\ x^b &= By^b \end{aligned}$$

or equivalently,

$$\begin{bmatrix} x^a \\ x^b \end{bmatrix}_{(m_a+m_b)\times 1} = \begin{bmatrix} A & \varnothing \\ \varnothing & B \end{bmatrix}_{(m_a+m_b)\times(n_a+n_b)} \begin{bmatrix} y^a \\ y^b \end{bmatrix}_{(n_a+n_b)\times 1}$$

Now we consider transactions between two entities. In the simple case, suppose a particular transaction, say entity a's Cash Sales, is conducted entirely with entity b. Further, suppose b's purchases is conducted entirely with entity a. Such a unique relation between the two entities has a strong but powerful implication to their bookkeeping networks. In this case, let y_{a1} denotes a's Cash Sales transaction and let y_{b2} denotes b's purchase of inventory, this strong link between the two entities can now be represented by the constraint that $y_{a1} = y_{b2}$. Generally, we will deploy the following notation for multi-entity bookkeeping setting:

- Let transaction vector y^{ab} is a $k \times 1$ column vector representing the common transactions between entities a and b. Assume $k < ((n_a + n_b)) - (m_a + m_b)$.

- Let transaction vector $y^{b\backslash a}$ is an $(n_b - k) \times 1$ column vector representing firm b's transactions not common with entity a.
- Let Transformation matrix $[B\backslash A]$ is an $m_b \times (n_b - k)$ matrix transforming the $(n_b - k)$ transactions into B's m_b balances. Similarly define $[A\backslash B]$.
- Let Transformation matrix $[A_{AB}]$ is an $m_a \times k$ matrix transforming the k transactions common to entities a and b into a's m_a balances. Similarly define $[B_{AB}]$.

With this notation, we can create a shortened transaction column vector $\left[y^a, y^{b\backslash a}\right]^T$ eliminating the redundant those transactions in y^b having an exact counter-part in y^a. With a suitable rearrangement of column-positions in matrices A and B, the two-entity level bookkeeping can be represented as

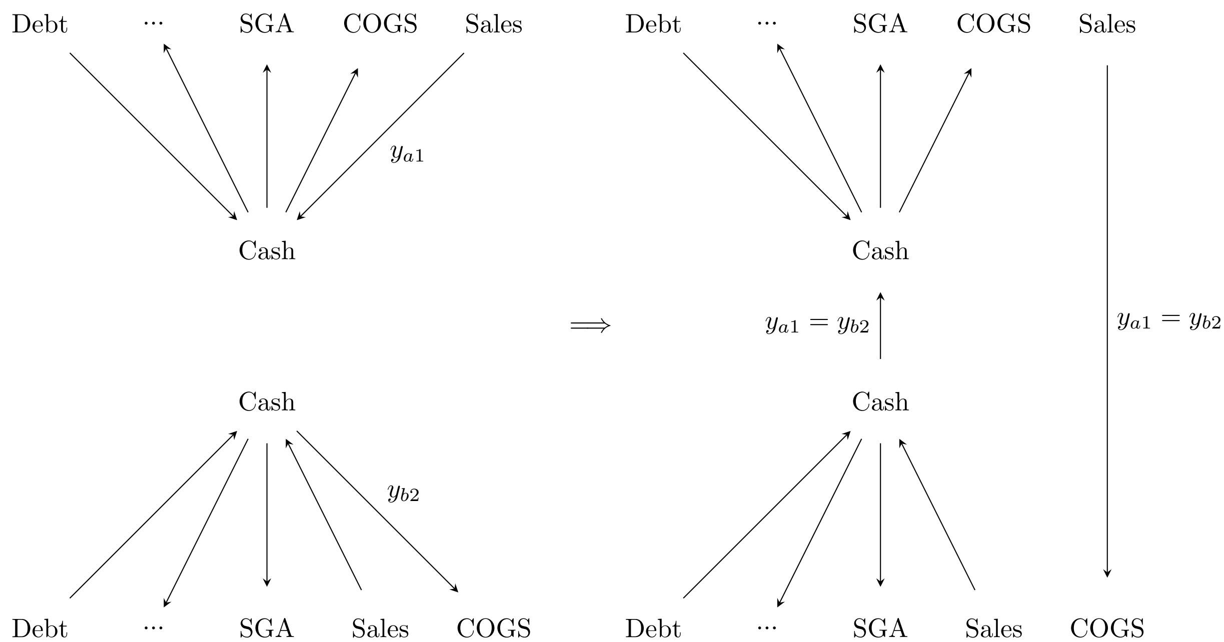
$$\begin{bmatrix} x^a \\ x^b \end{bmatrix}_{(m_a+m_b)\times 1} = \begin{bmatrix} [A\backslash B]_{m_a\times(n_a-k)} & [A_{AB}]_{m_a\times k} & \varnothing \\ \varnothing & [B_{AB}]_{m_b\times k} & [B\backslash A]_{m_b\times(n_b-k)} \end{bmatrix} \begin{bmatrix} y^{a\backslash b} \\ y^{ab} y^{b\backslash a} \end{bmatrix}_{(n_a+n_b-k)\times 1}$$

Denote matrix M_{AB} as the transformation matrix in the equation above. Notice that each inter-entity transactions y^{ab} are recorded four times through the center columns of the transformation matrix in the equation above. Specifically, each transaction in y^{ab} is recorded twice in entity a's book, represented by the corresponding column in matrix A_{AB}, and two more times in entity b's book, represented by the corresponding column in matrix B_{AB}. The transactions in either $y^{a\backslash b}$ or in $y^{b\backslash a}$ are recorded only twice. As we add more and more entity into the transaction vector, more and more transactions will be recorded four times. In the limit, all but internal transactions are recored four times. We call this new transformation matrix in the equation above the Quadruple entry matrix.

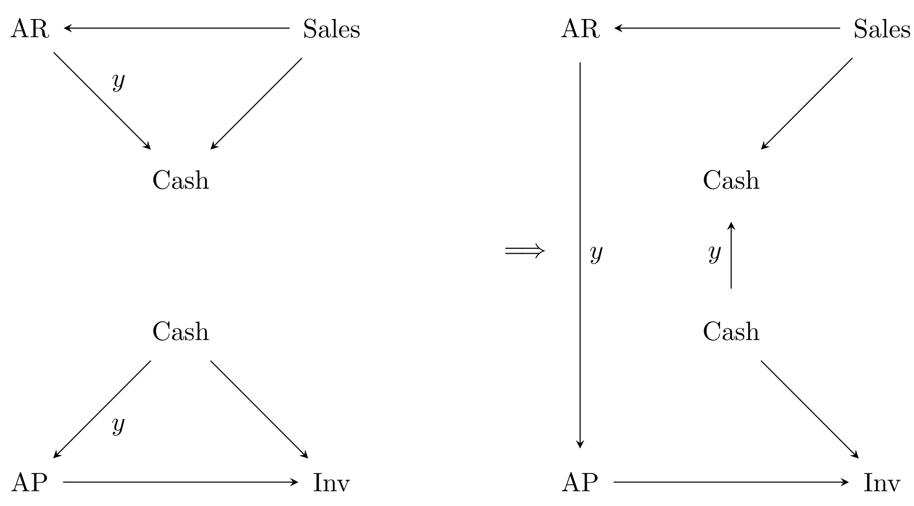
3.2 Graph Representation of Inter-entity Transactions

In the presence of inter-entity transactions, the expansion from double- to quadruple-entry bookkeeping opens two alternative representations of the network created by the bookkeeping practice. One alternative is to represent the network as multiple disjoint, but self-contained individual bookkeeping networks created by each individual entity's double-entry bookkeeping system (as described by Arya et al. [2000a]). An alternative, the one we emphasize here, is to exploit the inter-entity transactions to connect the individual account networks by the common transaction shared by a pair of entities.

Simple Inter-entity Transaction Here is a simple example of the alternative graph representation. Suppose we use entity a's cash-sale and entity b's cash-purchase example above. The common transaction volume in y^{ab} (for example, $y_1^a = y_2^b$) is recorded four times in the books of entities a and b, or equivalently drawn on two edges in the account network. In particular, under the original graph representation, the edges are from entity a's Sales to entity a's Cash and from entity B's Cash to entity B's Cost of Goods Sold (COGS). Under this representation, both edges are intra-entity. An alternative representation is that the two edges are from entity a's Sales to entity B's COGS and from entity a's Cash to entity a's Cash. Under this representation, both edges are inter-entity instead.



Inter-entity Transactions and Loops We now show an example of re-drawing the edges representing an inter-entity transaction changes the loops of the resulting bookkeeping graph. In this example, a collection of receivable of entity a is matched with the payoff of a payable by entity b. As a result, the alternative representation replaces two three-node **intra-entity** loops with a multi-note **inter-entity** loop.



3.3 Properties

Before we turn to the application of the quadruple entry bookkeeping matrix, we briefly discuss some properties of the matrix, especially as it relates to that of the double-entry matrix well-studied in prior work. Specifically we look into the four fundamental spaces of the Quadruple-entry matrix:

left-null space The left nullspace of any matrix A consists of all vectors w orthogonal to each column in A. Furthermore, if the graph is connected, there is one and only one independent vector in the left nullspace. Accordingly, for the quadruple-entry matrix, the dimension of its left nullspace is at least two (2) because each entity's accounts (or nodes in the graph) are connected. However, due the existence of inter-entity transactions, a subset of accounts (or nodes) of one entity may become connected with a subset of accounts of another entity, thus forming an additional row vector orthogonal to each column, independent of the existing two row vectors.

row space The row space of a matrix consists of all possible linear combination of all the rows of the matrix. Accordingly the dimension of the row space is the matrix's rank. For the Quadruple-entry matrix, the rank r is at most $m_a + m_b - 2$ since double-entry matrix for each entity has a rank of $m_i - 1$. The inter-entity transaction can further reduce the rank of the Quadruple-entry matrix.

null space The nullspace of any matrix A consists of vectors, y, that solve the equation $Ay = 0$. The loops of the directed graph form a basis for the nullspace. In bookkeeping terms, a transaction vector that produces zero changes in the account balances forms such a loop. Consider one such loop-forming transaction vector within a single entity's account network, such a vector will continue to generate zero balances for accounts of both entities if no transactions involved in the loop is an inter-entity transaction. If not, then accounts of another entity will be affected by the transaction vector. As a result, the dimension of the null space of the Quadruple-entry matrix may be less the sum of the dimensions of the nullspace of two associated single-entity double-entry matrices.

column space The column space of Quadruple-entry matrix consist of all journal entries with some four entries and others two entries.

We end this section with a basic property of the nullspace of the quadruple entry matrix. The property links its nullspace to the nullspace of the double-entry matrix.

Proposition 1 *If a vector y lies in the null space of the quadruple matrix, then the sub-vector of y representing entity a's transaction lies in the null space of the double-entry matrix A while the sub-vector of y representing entity b's transaction lies in the null space of the double-entry matrix B.*

Proof. Given vector y lies in the null space of the quadruple matrix, then,

$$\varnothing_{(m_a+m_b)\times 1} = My \Rightarrow \varnothing_{m_a} = \left[[A \backslash B]_{m_a \times (n_a - k)}, [A_{AB}]_{m_a \times k}, \varnothing \right] y$$

$$\Rightarrow \varnothing_{m_a} = \left[[A \backslash B]_{m_a \times (n_a - k)}, [A_{AB}]_{m_a \times k} \right] y_a \Rightarrow [A]_{m_a \times (n_a)}\, y_a$$

So y_a lies in the nullspace of A, the sub-vector of y representing entity a's transaction lies in the null space of the double-entry matrix A. The proof of the statement on entity b is similar, noting $y_b = [y_{ab}, y_{b\backslash a}]$ ■

4 Applications and Simulation Exercises

4.1 Inference Transactions from Account Balances

The basic application we wish to explore is borrowed from Arya et al. [2000a]. That is, the user wishes to use financial statement information (i.e., account balances x's) to infer about the underlying transactions (the y's) Here is the main inference theorem stated on page 375 of their paper:

- Let A^+ be the $n \times m$ matrix denotes the pseudoinverse of A
- Let N be the $(n-m+1) \times n$ matrix denotes the null space of A
- Let $\bar{y}$ be the prior on the unknown transaction vector y

Proposition 2 ***Due to Arya et al. [2000a]*** *Given financial statements x, the reader's best guess of unknown transactions y is*

$$A^+x + N^+N\ \bar{y}$$

In this section we provide simulation examples that illustrate the application of quadruple-entry bookkeeping accounting network structure. We are interested in two broad types of questions: 1) from the perspective of a single firm, to what extent including the financial statements information from other firms' may help with the inference of transactions of the target firm? 2) from the perspective of a multi-firm transaction network, how can the quadruple-entry accounting structure help with the inference about the network flows when we vary the network structure among the entities?

4.2 Simulation with two firms

We first follow the single-firm example in Arya et al. [2000a], and use exactly the same types of transactions and numerical numbers illustrated in the paper for our target firm. We are interested in knowing how the other firm's information may help with estimating the transactions of this target firm. We label this single firm as Firm A. We use the Eculidean distance (the square root of the sum of square errors) between the estimated transactions vector and the true transactions vector, to compare the estimation result from different scenarios. For the best guess of transactions $\hat{y}_a$ using a quadruple entry accounting network, the Eculidean distance is $v = ||y_a - \hat{y}_a|| = \sqrt{\sum_1^n (y_{ai} - \hat{y}_{ai})}$.

The basic set of transactions considered by Arya et al. [2000a] is presented in the previous Figure 1. The vector of transaction amounts is assumed to be $y_a = (8, 9, 1, 10, 5, 1, 2)$, which results the financial statement accounts of $Ay_a = x_a = (-8, 4, 6, -10, 5, 3)$. The financial statements reader's prior expectation of transactions is $\bar{y}_a = (7, 9, 1, 10, 5, 1, 2)$. Notice that in this example, the reader's prior for most transactions coincide with the true amounts, except for the transaction y_{a1}–the purchase of inventory. Given the financial statement x_a and the prior $\bar{y}_a$, the reader's best guess of true transactions is given by $\hat{y}_a = A^+x + N^+N\bar{y}_a = (7.625, 9.25, 1.125, 10, 5, 1.375, 1.875)$. Note that when a transaction does not reside in any basis of the null space, it is perfectly inferred, e.g., transactions y_{a4} and y_{a5}. For other transactions, the non-empty nullspace prevents the perfect (and unique) inference of the transactions from the statements. In this example, the estimation error measured by Eculidean distance is 0.6124.

We now consider the inference problem when including the financial statement information from a second firm that has a direct transaction with A. We label this firm as B. We explore several different setting with respect to the structure of B's accounting network structure, and the nature of the transaction between these two firms.

Case 1: B's double-entry accounts network has empty nullspace. That is, Firm B's accounts network is a spanning tree. For example, we consider the following transactions of B,

$$\begin{aligned} y_{b1}: &\quad \text{purchase inventory for cash} \\ y_{b3}: &\quad \text{cash expenses} \\ y_{b4}: &\quad \text{cash sales} \\ y_{b5}: &\quad \text{cost of good sold} \end{aligned}$$

Clearly there is no loop in B's accounts network, and all nodes are connected. Thus B's accounts network is a spanning tree. Figure 2 places the two Firm A and B's account graphs side-by-side. We consider the following different types of transactions that link A and B.

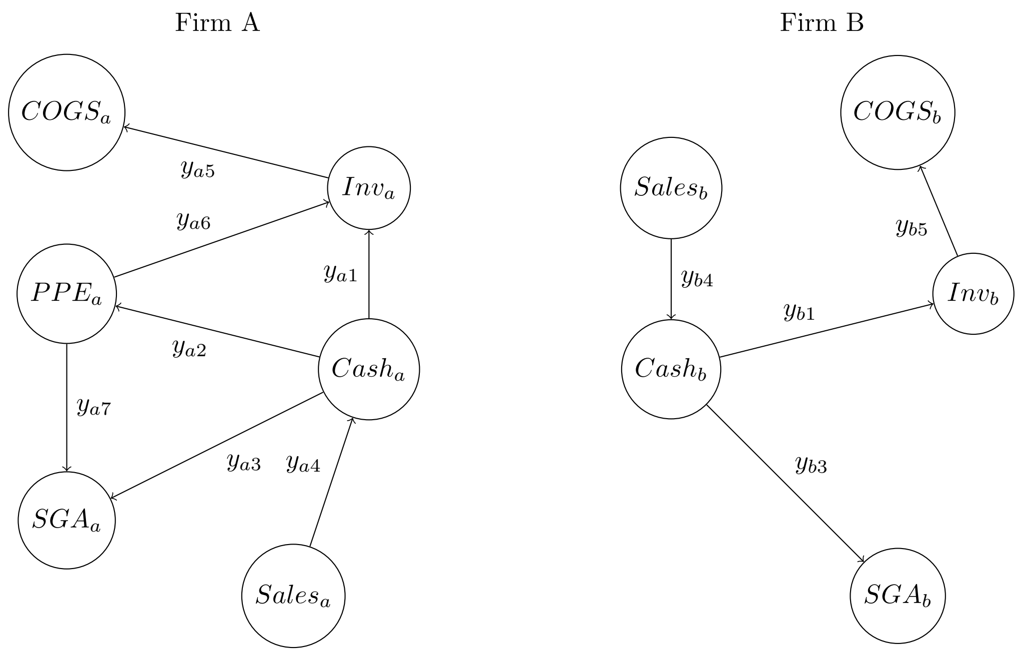


Figure 2

B sells inventory to A. The transaction y_{a1} in A's bookkeeping network is linked to a transaction (sales) recognized by firm B.

B's transaction y_{b4} and A's transaction y_{a1} are linked to each other, which allows us to connect the cash account in B to the cash account in A, and the sales account in B to the inventory account in A. In terms of the digraph presentation, the connected digraph becomes as in Figure 3. From the graph, one can see that these two firms' subgraphs become connected because of the inter-company sales transaction.

By connecting to firm B's accounts through the quadruple entry bookkeeping network, one edge ($Cash_a- > Inv_a$) in the loop that connects the nodes ($Cash_a, Inv_a, PPE_a$) is removed. Since firm B's transactions space do not form any loop, these two accounts that connect to the counter-accounts in firm B are not in any loop of the connected graph. The connected graph AB has only one loop from the original graph of firm A.

To see how firm B may help improve the inference about the transactions in firm A, assume firm B's transactions amounts are given by $y_b = (7, 1, 8, 6)$, with a prior of $\bar{y}_b = (6, 2, 7, 5)$. B's sales transaction is the same amount as A's inventory purchase transaction. Thus B's accounts balances are $x_b = B * y_b = (0, 1, -8, 6, 1)$.

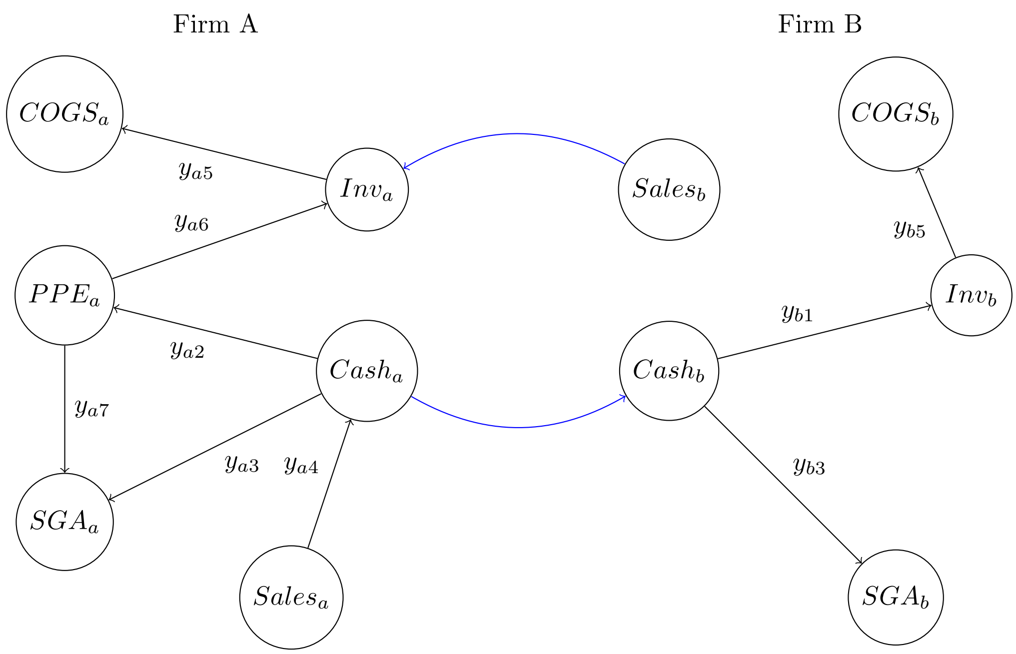


Figure 3

The quadruple entry matrix M is

$$
\begin{array}{cc}
 & \begin{array}{ccccccc|ccc} \mathrm{y}_{ab} & \mathrm{y}_{a2} & \mathrm{y}_{a3} & \mathrm{y}_{a4} & \mathrm{y}_{a5} & \mathrm{y}_{a6} & \mathrm{y}_{a7} & \mathrm{y}_{b1} & \mathrm{y}_{b3} & \mathrm{y}_{b5} \end{array} \\
\begin{array}{c} \mathrm{Cash}_a \\ \mathrm{Inv}_a \\ \mathrm{PPE}_a \\ \mathrm{Sales}_a \\ \mathrm{COGS}_a \\ \mathrm{SGA}_a \\ \hline \mathrm{Cash}_b \\ \mathrm{Inv}_b \\ \mathrm{Sales}_b \\ \mathrm{COGS}_b \\ \mathrm{SGA}_b \end{array} &
\left[\begin{array}{ccccccc|ccc}
-1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\hline
1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{array}\right]
\end{array}
$$

We relabel the A's transaction of purchasing inventory as y_{ab}, representing the inter-firm nature of this transaction. Given x_a and x_b, the best guess of firm A and B's transactions is $\hat{y}{=}M^{+}x + N_m^{+}N_m\bar{y}$, where $x = (x_a, x_b) = (-8, 4, 6, -1, 5, 3, 0, 1, -8, 6, 1)$ and $\bar{y} = (\bar{y}_a, \bar{y}_b/\bar{y}_{b4}) = (7, 9, 1, 10, 5, 1, 2, 6, 2, 5)$. We obtain that $\hat{y} = (8, 9, 1, 10, 5, 1, 2, 7, 1, 6)$. The best guess of A's transaction $\hat{y}_a$ is the first seven elements in the vector. In this case A's transactions are perfectly inferred, $\hat{y}_a = y_a$. Clearly the estimation error is reduced when B's accounts balances are used in the estimation using a quadruple entry matrix. Though the quadruple entry matrix has non-zero nullspace, all transactions are still perfectly backed out. The reason is because the connected network has only one loop, the best estimate of these transactions in the nullspace is simply the prior, which happens to be perfect. We will see later with more dimensions of the nullspace, the transactions cannot be perfectly inferred even if the prior is precise. Note that B's transactions are perfectly inferred though the prior is not precise, because B's accounts network is a spanning tree without any cycle.

To see the effect of prior, let's instead assume that $\bar{y}_a = (7, 8, 1.5, 9, 6, 1, 2)$, then the inferred transactions using the transformation matrix A alone is $\hat{y}_a = (7.81255, 8.625, 1.5625, 10, 5, 1.1875, 1.4375)$, with an Eculidean distance of 0.9186. If we use the quadruple entry matrix M to infer the transactions, we obtain that $\hat{y}_a = (8, 8.5, 1.5, 10, 5, 1, 1.5)$, with an Eculidean distance of 0.866, which is smaller than the Eculidean distance from using A matrix alone. With uncertain prior, the transactions that remain residing in a loop cannot be perfectly inferred, but we still achieve a reduction in estimation error. A's direct transaction with B can be perfectly inferred when using the quadruple entry matrix, which is due to the fact that B's accounts network is a spanning tree. Meanwhile, other transactions that reside in the same loop but not belong to any other loop, can also be perfectly inferred regardless of the prior.

Observation 3 *When the counter-party's accounts network is a spanning tree, the inter-firm transaction can be perfectly inferred.*

We take a detail look of the fundamental sub-spaces of the quadruple entry matrix. The number of columns in M_{ab} is $n_{ab} = 10 = 7 + 4 - 1$, which is the total number of transactions in firm A and B respectively, minus the number of inter-entity transactions between these two firms. The total number of columns is reduced by one because of the inter-entity transaction. The number of rows in M_{ab} is simply $m_{ab} = m_a + m_b = 11$, where m_a and m_b represent the number of accounts in each firm. Now we look at the left nullspace and nullspace. The nullspace of the matrix has only one dimension because there is only one loop now. Thus the column space has a dimension of $n_{ab} - 1 = 9$. The new graph is still *connected* when we remove the original edges in each subgraph of A and B, and draw the edges from one firm's accounts to its counter-party's accounts. The left-nullspace of the quadruple entry matrix consists of two basis: one basis for adding all accounts in the graph to get zero, and the second basis for adding the accounts of one single firm. Thus the left-nullspace is 2, even though the new graph appears to be fully connected. The row space is $m_{ab} - 2 = 9$, the same as the column space.

Observation 4 *If the edge from the inter-entity transaction resides in a loop of the single firm's digraph, the quadruple entry graph remains connected, but the left nullspace of the connected graph is larger than 1.*

B sells PPE to A We next consider B sells PPE to A, in which case B's sales y_{b4} is linked to A's PPE purchase y_{a2}. Figure 4 shows the connection through quadruple entry bookkeeping among the accounts in two firms. The quadruple entry matrix M is represented in a similar fashion, with y_{ab} being the inter-entity transaction that is recorded in A as debiting PPE_a and crediting $Cash_a$, and in B as crediting $Sales_b$ and debiting $Cash_b$ in the same column.

There are at least two differences between the inter-entity transaction of PPE purchase and the inter-entity transaction of inventory purchase:

1. Removing the edge from $Cash_a$ to PPE_a in A's accounts network breaks one existing loop, but the remaining loop in A has four edges among $(Inv_a, PPE_a, Cash_a, SGA_a)$. In the inventory example, after removing the edge from $Cash_a$ to Inv_a, the remaining loop in A has three edges $(Inv_a, Cash_a, SGA_a)$. This is due to the fact that PPE account is involved in more internal allocation and adjustment, which result in more loops related to the PPE purchase transaction.

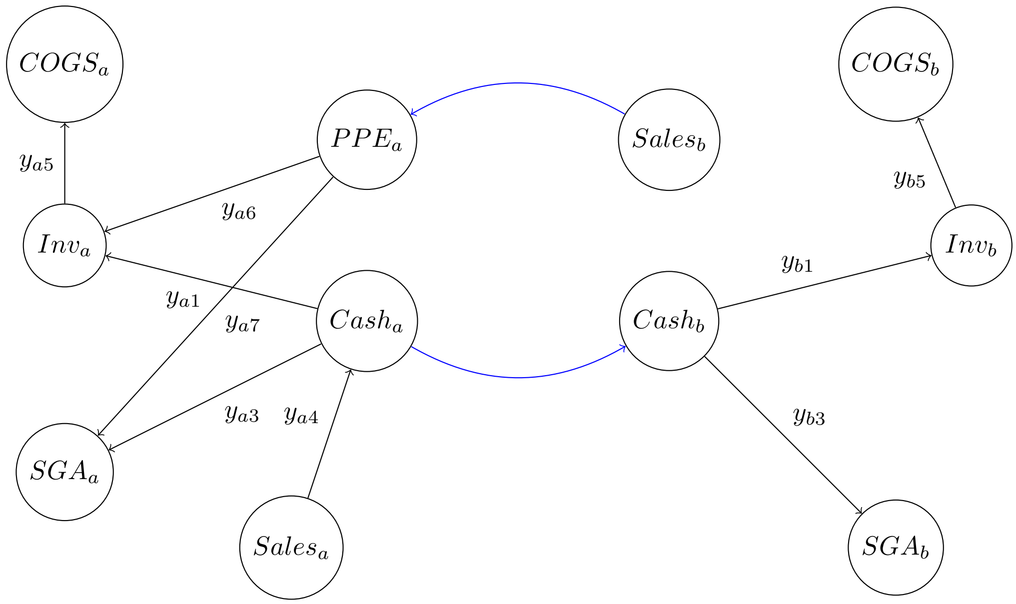


Figure 4

2. The PPE purchase in A has a perfect prior ex-ante, while the inventory purchase's prior is different from the true transaction. How does the prior uncertainty affect the inference problem when we include the counter-party's financials in the problem?

Comparing these two types of transactions may help us understand the effects of prior uncertainty and the structure of loops. We skip writing out this new transformation matrix and show directly the estimation results. Using the expanded transformation matrix and the financial accounts balances (x_a, x_b), we obtain the reader's best guess of transactions is $\hat{y}_a = (7.75, 9, 1.25, 10, 5, 1.25, 1.75)$. We see that the estimated purchase of PPE transaction is 9, which is perfectly inferred. Though the other transaction edges that reside in the loop cannot be perfectly inferred, the Eculidean distance is 0.5, smaller than the single-firm based estimation error 0.6124.

It seems that the estimation result is worse than the one from inventory purchase. However, as we observed, the transaction differs for two different reasons, and it is not clear which effect is driving the result. Therefore, instead of following the original example's transaction amounts, we perform simulations using randomly generated numbers with variation in the prior uncertainty. Specifically, we perform the following set of simulation. We randomly generated a set of transactions with the same prior, and follow the inference procedure to estimate the true transactions in these two types of links: i) Inventory purchase, ii) PPE transaction. We calculate the Eculidean distance, d_1 and d_2, for each set of transactions, and repeat the random process of the same prior to obtain 1,000 data points. We then test the statistical difference between these two sets of Eculidean distances.

We generate random 10 transactions that follow the log-normal distribution with parameters of (μ_y, σ_y), where $\mu_y = \begin{pmatrix} \mu_{y_{a1}} \\ \mu_{y_{a2}} \\ \mu_{y_{a3}} \\ \mu_{y_{a4}} \\ \mu_{y_{a5}} \\ \mu_{y_{a6}} \\ \mu_{y_{a7}} \\ \mu_{y_{b1}} \\ \mu_{y_{b3}} \\ \mu_{y_{b5}} \end{pmatrix} = \begin{pmatrix} Ln(8) \\ Ln(8) \\ Ln(1) \\ Ln(10) \\ Ln(5) \\ Ln(1) \\ Ln(2) \\ Ln(7) \\ Ln(1) \\ Ln(2) \end{pmatrix}$, and $\sigma_y = \begin{pmatrix} var_1 \\ var_2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$.

Notice that B's sale transaction y_{b4} is the same as either y_{a1} or y_{a2}, depending on which transaction we are examining. We try to generate numbers that are close to the example we have used, but we intentionally assume the same mean of the transaction amount of the purchase of inventory and PPE. The variances of these two transactions vary, and the other transactions all have the perfect prior. In other words, the randomly generated amounts for other transactions are the same for all the data points we obtain. We first allow these two types of transactions have the same variance (fixing the prior uncertainty) in the distribution, and use the same group of random numbers to infer the transactions by matching B's sales with either inventory purchase or PPE purchase. We label the Eculidean distance from the estimation using the inventory purchase quadruple entry as e_1, and the distance from using the PPE purchase quadruple entry as e_2. Then we vary the variance by making one type of transaction more precise than the other when generating the random numbers, and then use the same group of random numbers to infer the transactions in each case. Below is the estimation results:

Variance		e_1	e_2	T-stat ($e_1 - e2$)
INV PPE	$\frac{1}{2}Ln(8)$ $\frac{1}{2}Ln(8)$	5.359	4.440	2.588 ($p = 0.0098$)
INV PPE	$\frac{1}{4}Ln(8)$ $\frac{1}{4}Ln(8)$	2.109	1.886	2.4937 ($p = 0.0128$)
INV PPE	$\frac{1}{2}Ln(8)$ $\frac{1}{4}Ln(8)$	5.247	2.018	9.471 ($p < 0.000$)
INV PPE	$\frac{1}{4}Ln(8)$ $\frac{1}{2}Ln(8)$	1.765	5.759	-12.001 ($p < 0.000$)

Table 1: Simulation results

From Table 1, we see that when we fix the prior uncertainty (either $1/2Ln(8)$ or $1/4Ln(8)$, matching the inter-entity PPE purchase transaction provides more precise estimation (with Eculidean distance of 4.44 and 1.886 respectively) than the inter-entity inventory purchase transaction (with Eculidean distance of 1.886 and 2.109 respectively). We also perform a t-test with respect to the difference between these two, and show that t-statistic is 2.588 and 2.4937, suggesting that the PPE transaction has significantly lower distance at 1% level.

When we vary the prior uncertainty in the random number generating process, we find that prior uncertainty plays an important role. It is always the case that the transaction with a larger uncertainty is more crucial in the inference problem. The Euclidean distance from the first pair of variances ($1/2Ln(8)$

for inventory purchase and $1/8Ln(8)$ for PPE purchase) is 5.247 and 2.018 for the inter-company inventory purchase and inter-company PPE purchase, respectively. The t-statistics of the difference in the mean of the two distances is 9.471, statistically significant with $p < 0.000$. When we switch the variance on each transaction, we find the opposite result. Notice that in all cases, because of the spanning tree structure of firm B, the inter-entity transaction (either inventory or PPE purchase) is always perfectly inferred. The improvement on the estimation errors comes from the remaining transactions that reside in the loop after joining the two firms' graphs together.

We can make two observations from the simulation results in Table 1.

Observation 5 *Removing an edge that resides in multi-loops (a shared edge) through the quadruple entry bookkeeping link reduces the estimation errors more than removing an edge that resides in a single unique loop.*

Observation 6 *Removing an edge with more prior uncertainty through the quadruple entry bookkeeping link reduces the estimation errors more than removing one with less uncertainty.*

Case 2: when B's double entry accounts network has loops (cycles). We use the example of the first type transaction (B sells inventory from A) in this part and consider more complex accounts network of B other than a spanning tree.

we start with the example in Figure 2, and add one more transaction to firm B so that B's accounts form a loop that involves the inter-entity transaction. It is generally not common to see the sales transaction reside in a loop. Nonetheless, we make up a hypothetical transaction which may not fit into current accounting standard's requirement, but allows us to examine the effect of a loop. Suppose some sales return and allowances are recognized as SGA, which is debiting SGA and crediting Sales, as shown in Figure 5. Now B has one loop which contains the edge ($Sales_b \to Cash_b$) from the inter-entity transaction. Assume the amount of this transaction is 2 and the prior is also 2. All the other transactions follow the example in Figure 2.

We redraw the edges following the quadruple entry link between these two firms. The quadruple entry digraph is presented in Figure 6. Because the inter-entity transaction in each firm's stand-alone network resides in one loop, reconnecting the accounts across firms to form new edges actually connects all the accounts in two original loops together: a new loop now forms among $(Cash_a, PPE_a, Inv_a, Sales_b, SGA_b, Cash_b)$. Though the loop that contains the inter-entity transaction does not break as in the example with spanning tree structure, the total dimension of nullspace still decreases by one because of the quadruple entry link. Using the quadruple entry matrix, the best guess of A's transactions is $\hat{y_a} = (7.7857, 9.1429, 1.0714, 10, 5, 1.2143, 1.9286)$ with an Eculidean distance of 0.3499, smaller than the estimation error (0.6124) from stand-alone matrix A. Clearly, even though connecting A and B's accounts network does not allow us to perfectly infer the inter-entity transaction, there is still a reduction in the estimation error when the two separate loops in the individual subgraphs become one loop in the quadruple entry graph.

Observation 7 *When the inter-entity transaction resides in loops within each firm, the inter-entity transaction cannot be perfectly inferred, but the the estimation error is still smaller than using the stand-alone double entry matrix.*

We further perform simulations with randomly generated numbers in this case, and examine the difference between the purchase of inventory and PPE when firm B's structure is not a spanning tree. We compare the Eculidean distance for all transactions and for the inter-entity transaction only. We follow the same

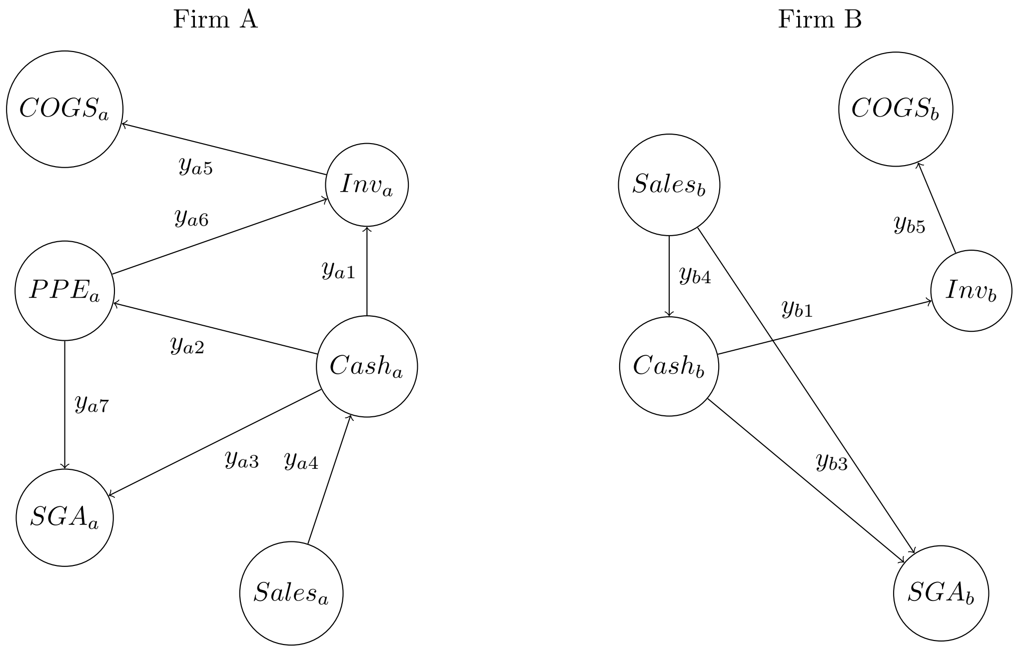


Figure 5

parameters as illustrated in the example in Table 1, but keeping the same variance for PPE and inventory transactions. We label the Eculidean distance for all transactions from the double entry matrix as e_0, and e_1 and e_2 from the quadruple entry matrix with inventory and PPE inter-entity transactions respectively. We also label the distance for the single transaction as e_0^{Inv} and e_0^{PPE} when using the double entry matrix, and the Eculidean distance for the inter-entity transaction from the quadruple entry matrix as e_1^{Inv} and e_2^{PPE} respectively. The results are presented in Table 2.

Variance		e_0	e_1	e_2	T-stat $(e_0 - e_1)$	T-stat $(e_1 - e_2)$		
INV PPE	$\frac{1}{4}Ln(8)$ $\frac{1}{4}Ln(8)$	3.632	2.752	2.494	27.165	6.086		
INV PPE	$\frac{1}{2}Ln(8)$ $\frac{1}{2}Ln(8)$	10.232	7.801	6.957	20.954	5.075		
Variance		e_0^{Inv}	e_0^{PPE}	e_1^{Inv}	e_2^{PPE}	T-stat $(e_0^{Inv} - e_1^{Inv})$	T-stat $(e_0^{PPE} - e_2^{PPE})$	T-stat diff-diff
INV PPE	$\frac{1}{4}Ln(8)$ $\frac{1}{4}Ln(8)$	1.646	3.292	0.941	1.013	33.54	28.68	26.50
INV PPE	$\frac{1}{2}Ln(8)$ $\frac{1}{2}Ln(8)$	4.538	9.077	2.593	2.832	24.713	21.809	20.293

Table 2: Simulation results: Case 2

Case 3: the left-nullspace complication Suppose B is A's customer. That is, y_{b1} is the same amount as y_{a4}. We assume the same types of transactions for B $y_b = (y_{b1}, y_{b3}, y_{b4}, y_{b5}) = (10, 6, 12, 9)$, with a prior of $\bar{y}_b = (10, 6, 12, 9)$. Firm B's financial statement accounts balance is $x_b = B * y_b = (-4, 1, -12, 9, 6)$. The two firms' subgraphs are connected as in Figure 7.

A's connection to B's accounts does not move/break any of the existing loops in A. The sales transaction

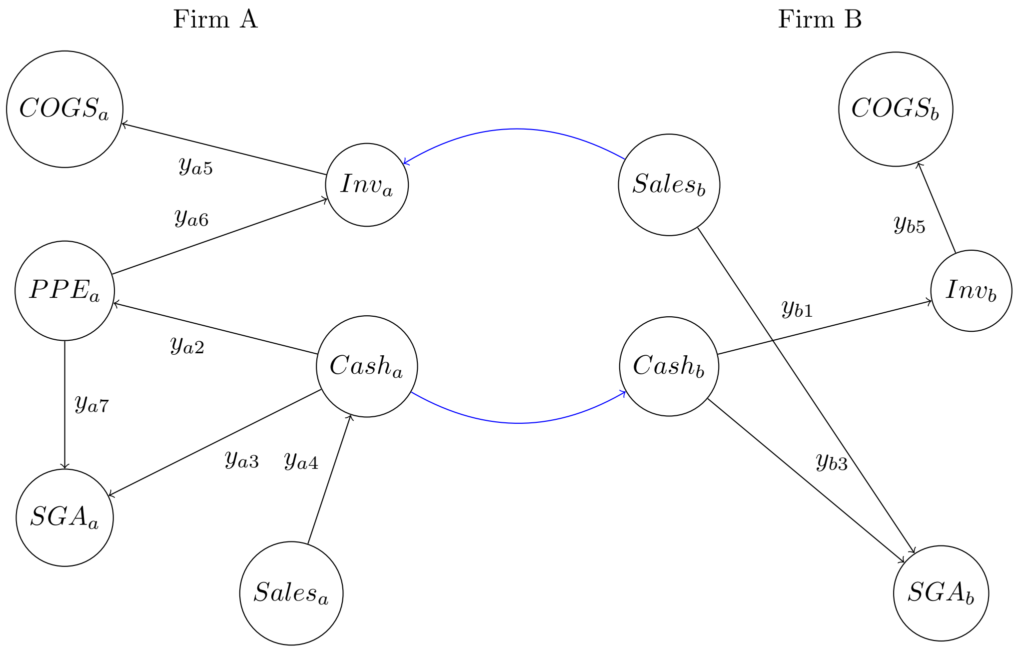


Figure 6

can be perfectly inferred from A's stand-alone financial accounts, which suggest that including the counterparty B should not improve the inference of the other transactions, given that the loops remain. We use the quadruple entry matrix and obtain the best guess of the transactions. As expected, the inferred transactions $\hat{y}_a = (7.625, 9.25, 1.125, 10, 5, 1.375, 1.875)$, exactly the same as the estimation from A's double entry matrix alone.

Notice that in this scenario, the total graph after linking the inter-entity transaction to different firms' accounts is still a disconnected graph with two subgraphs. The reason is that the account $Sales_a$ is 1-edge connected through the edge formed by this inter-entity transaction in the original graph of firm A. That is, if we remove the edge $Sales_a- > Cash_a$ in A, the single node ($Sales_a$) becomes disconnected to the remaining components in Graph A. Similarly, the subset of nodes (Inv_b, $COGS_b$) is 1-edge connected through the inter-entity transaction in the original graph of firm B. When we remove the connecting edge $Cash_b- > Inv_b$, the subset $(Inv_b, COGS_b)$ in B becomes disconnected to the remaining components in Graph B. Thus drawing the edge from $Sales_a$ to Inv_b from the quadruple entry bookkeeping connects $Sales_a$ and $(Inv_b, COGS_b)$ in B, but the subset of three nodes $(Sales_a, Inv_b, COGS_b)$ is disconnected from the rest of the quadruple entry graph. Meanwhile, the other two accounts involved in the quadruple entry bookkeeping, $Cash_a$ and $Cash_b$, still remain connected to the original graph because each is more than 1-edge connected. Furthermore, when we draw the edge between these two, it is clear that the two subsets from Graph A and B are connected through the edge $Cash_b- > Cash_a$. Note that if the edge is a cut-edge, it cannot be in any loop of the original graph.

The quadruple entry matrix M_{ab} in this example can be represented below.

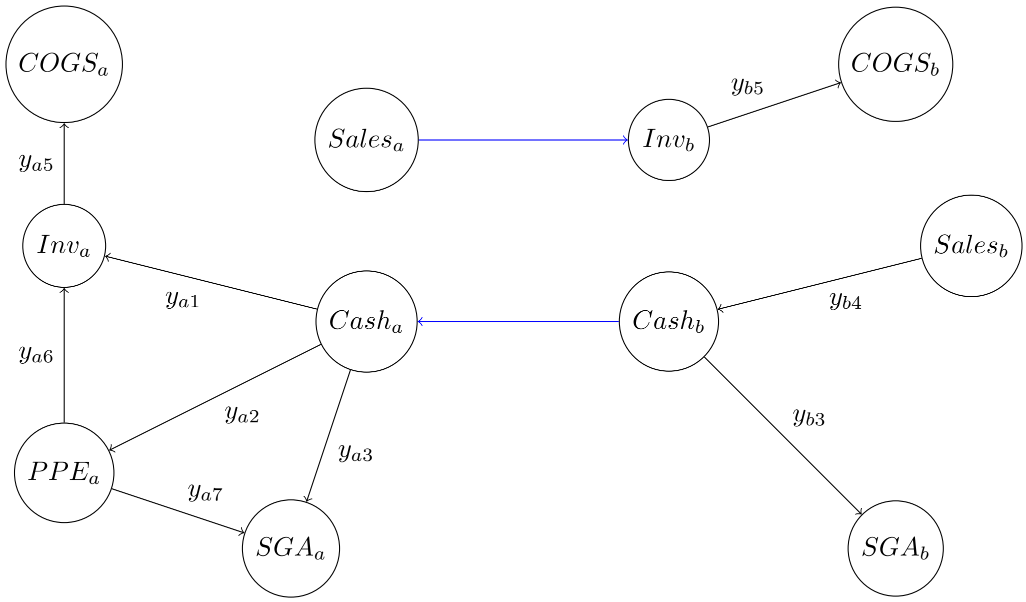


Figure 7

$$
\begin{array}{c|c}
 & \begin{array}{cccccccccc} y_{a1} & y_{a2} & y_{a3} & y_{ab} & y_{a5} & y_{a6} & y_{a7} & y_{b3} & y_{b4} & y_{b5} \end{array} \\
\begin{array}{c} Cash_a \\ Inv_a \\ PPE_a \\ Sales_a \\ COGS_a \\ SGA_a \\ Cash_b \\ Inv_b \\ Sales_b \\ COGS_b \\ SGA_b \end{array} &
\left[\begin{array}{ccccccc|ccc}
-1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{array}\right]
\end{array}
$$

The nullspace of this quadruple entry matrix has a dimension of 2, which is clear from the two loops in the graph. The left nullspace, however, has a dimension of 3. Why? First of all, the digraph is disconnected into two connected subgraphs, which means the left nullspace has at least two dimensions, which comes from the fact that the sum of all accounts in each separate original graph A and B is zero (as well as the sum of all accounts from both firms). However, one more dimension is added because of the disconnected graph from the quadruple entry link. Essentially one can add up only the three accounts $Sales_a$, Inv_b and $COGS_b$ from the disconnected set, and still get zero. The basis of the left nullspace is thus given by

$$
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0
\end{pmatrix}.
$$

In this quadruple entry matrix M_{ab}, the column space's dimension is $n_{ab} - 2 = 8$ given the nullspace's dimension of 2, the same as the row space's dimension, which is $m_{ab} - 3 = 8$, given the left nullspace's dimension of 3.

Observation 8 *If the edge from the inter-entity transaction is a cut-edge in either of the original firm's digraph, the new graph that represents the quadruple entry link of the accounts is disconnected. The left nullspace increases by one when two firms are linked through the inter-entity transaction.*

4.3 Simulation with multiple firms

In the last section, we are interested in the inference of transactions for a target firm, with a simplified framework that the target's firm external transaction of the same type is exclusively with another counter-part firm. In this section, we are interested in the multi-edge transactions we discussed in Section 2, that is, for the same type of transaction, there are multiple counter-parties. For example, the firm may sell to different customers or buy from different suppliers. In a large scale economy where there are n firms, each firm may have its own connected neighbors, while indirectly connected to other firms in the economy through the network. Suppose we are interested in knowing the economic transaction flows among all firms in the economy, the quadruple entry matrix of multi-firm accounts allows one to make the inference based on all firms' financial statements, if available. This task is impossible in the double entry matrix of a single-firm, because the best we can obtain is the single transaction amount that aggregates all edges direct to different firms.

In this section, we focus on the inference of the transactions among the multiple firms. For simplicity, we assume each firm only incurs two types of transactions: cash sales or cash purchase (cost of good sold).Therefore each firm has only three accounts: $Cash$, $Sales$, and $COGS$. We perform simulation by first generating a random network of firms' connections to each other. The network of firms are assumed to be represented by a matrix where each row and column of the matrix represents the entity (firm) in the economy, and each element in the matrix represents the amount of sales from the firm in i^{th} row to the firm in j^{th} column. From the network of flows among firms, we create the quadruple entry accounts network among all firms. Each transaction is recorded by the i^{th} firm as sales and by the j^{th} firm as purchases. By aggregating all sales and purchases respectively, we obtain each firm's $Cash$, $Sales$, and $COGS$ balances. Then we use the quadruple entry matrix that links all the connected firms' accounts to make inference about the inter-company transactions.

As often observed by researchers, most of the real networks are often large and sparse. Hence we start our simulation assuming the transactions among firms form a sparse network, which simplifies our analysis and computing. But we vary the degree of sparsity (the density of connections) to see how the network structure affects the inference results. We simulate an economy of 50 firms, and vary the density degree of the sparse matrix from 0 to 1.

The first set of simulation is based on the complete economy. In other words, all financial statements are available and we can include all firms in the economy in our estimation sample to infer all possible links among the firms. We measure the normalized Eculidean Distance of the estimated transactions and the actual transactions, $\sqrt{(\hat{y}-y)^2/n}$, and plot it against the density score of the network, measured by the proportion of the non-zero elements in the matrix. The result is presented in Figure 8. When the matrix is too sparse (roughly with a density score below 0.01), we see that transactions can be relatively inferred more precisely, and that increasing the connections increase the estimation errors. But when the density reaches a certain (small) amount, increasing the density actually reduces the estimation errors. The more connections we observe among firms, the easier to infer the transactions among each other. We also observe from the figure that after the density score reaches around 20%, the decrease in the estimation error is not significant when the density level further increases. In a sample of 50 firms, 20% density implies that each firm on average is connected to about 10 other firms. The inference results we get in such an economy is roughly as good as what we can get in the economy there are more connections among firms.

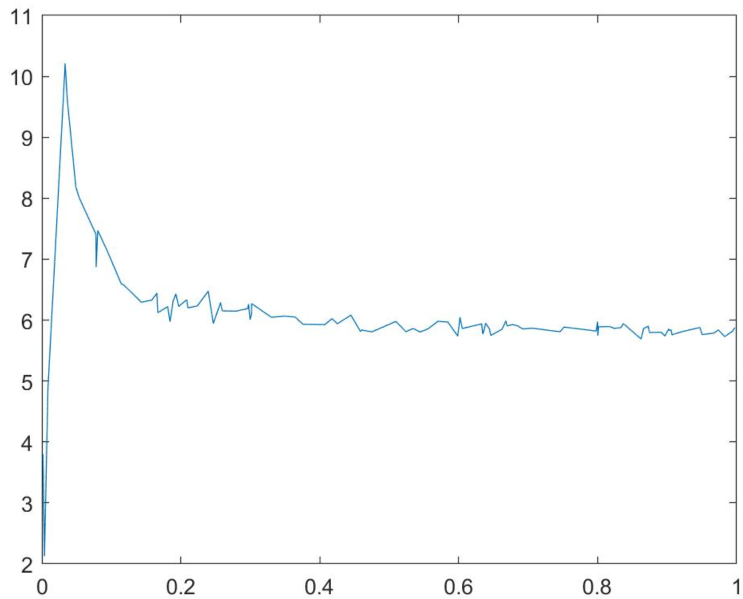
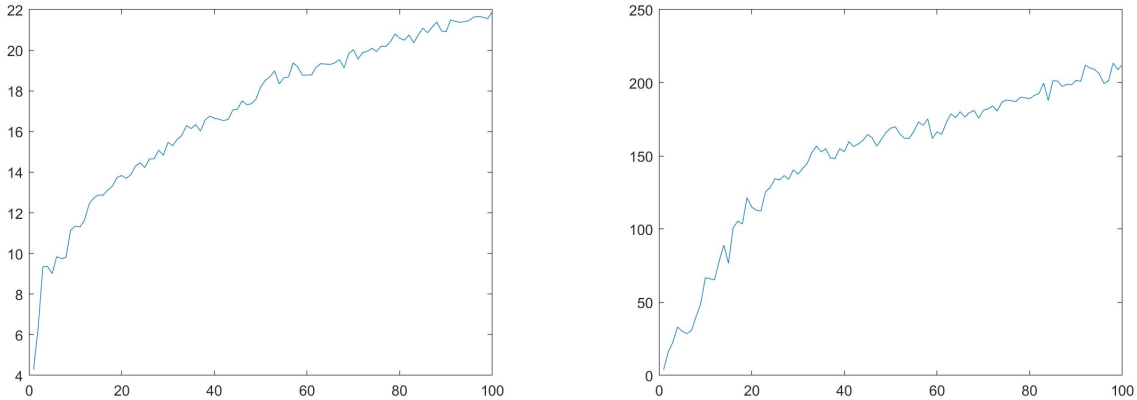


Figure 8: Normalized Eculidean Distance and Density of Network

The second set of simulation is based on the economy with partial data. In reality, not all firms' financial statements are publicly available. Even though there are more firms that transact with each other, we can only find a group of target firms with publicly available information, while the rest of the firms are unknown. We simulate this case to resemble the economy with private firms. We generate the underlying network among all 50 firms similar to the first case, but we assume that 20 firms are unknown to the public (named as "private sector") and the remaining 30 firms are our target firms. We aggregate each target firms' transactions with the private sector's firms and represent it as one single transaction. We provide the estimation results in Figure 9 for the estimated transactions including the private sector as one single firm (Figure a), and the estimated transactions for the target firms only (Figure b). The results are opposite to the first case: increasing density of the network makes it harder to infer the true transactions if part of the economy is missing from the estimation sample.

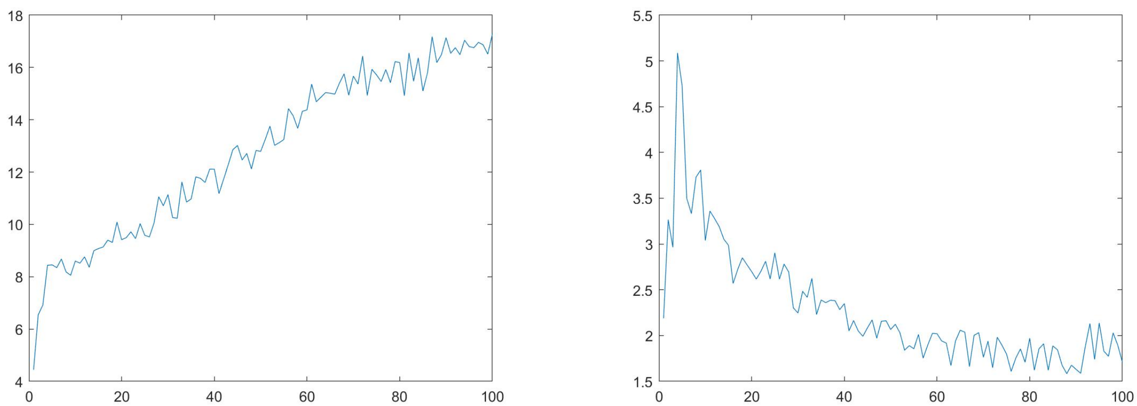
Suppose we are able to observe an aggregate measure of the private sector, for example, the household income and consumption, how would our estimation results change? We estimate the transactions with the private sector treated as one single firm, but the financial accounts balance are available (in aggregate manner). The results are somewhat different from Figure 9. While the estimation errors of all transactions still increase with the density of the network, the estimation errors of inter-entity transactions among the target firms decrease as the density increases, as shown in Figure 10.



(a) Inference of all transactions

(b) Target firms only

Figure 9



(a) Inference of all transactions

(b) Target firms only

Figure 10

5 conclusions

The paper explores the properties of the quadruple entry bookkeeping system for a collection of entities, each practicing double-entry bookkeeping system. Based on the preliminary investigation, we conclude the representations of the system add more insights beyond those discovered by studying the double-entry bookkeeping system of a single entity. The potential applications also offer promising new questions beyond double-entry bookkeeping such as inferring inter-entity links between entities. The framework offers potential path to an understanding of an inter-entity network based on transaction-level data.

References

Arya, A., Fellingham, J. C., Glover, J. C., Schroeder, D. A., Strang, G., 2000a. Inferring transactions from financial statements. Contemporary Accounting Research 17, 366–385.

Arya, A., Fellingham, J. C., Mittendorf, B., Schroeder, D. A., 2004. Reconciling financial information at varied levels of aggregation. Contemporary Accounting Research 21, 303–324.

Arya, A., Fellingham, J. C., Schroeder, D. A., 2000b. Estimating transactions given balance sheets and an income statement. Issues in Accounting Education 15, 393–411.

Butterworth, J. E., 1972. The accounting system as an information function. Journal of Accounting Research pp. 1–27.

Fellingham, J., 2018. The double entry system of accounting. Accounting, Economics, and Law: A Convivium 8.

Ijiri, Y., 1965. On the generalized inverse of an incidence matrix. Journal of the Society for Industrial and Applied Mathematics 13, 827–836.

Ijiri, Y., 2014. The beauty of double-entry bookkeeping and its impact on the nature of accounting information. Economie Notes by Monte dei Paschi di Siena 22, 265–285.