Bookkeeping Graphs: Computational Theory and Applications

Other titles in Foundations and Trends® in Accounting

Timeliness, Accuracy, and Relevance in Dynamic Incentive Contracts
Peter O. Christensen, Gerald A. Feltham, Christian Hofmann and Florin Sabac
ISBN: 978-1-63828-084-2

Entropy, Double Entry Accounting and Quantum Entanglement
John Fellingham, Haijin Lin and Doug Schroeder
ISBN: 978-1-63828-032-3

Foreign Currency: Accounting, Communication and Management of Risks
Trevor Harris and Shiva Rajgopal
ISBN: 978-1-68083-946-3

Audit Regulations, Audit Market Structure, and Financial Reporting Quality
Christopher Bleibtreu and Ulrike Stefani
ISBN: 978-1-68083-900-5

Accounting for Risk
Stephen Penman
ISBN: 978-1-68083-890-9

Evolution of U.S. Regulation and the Standard-Setting Process for Financial Reporting: 1930s to the Present
Stephen A. Zeff
ISBN: 978-1-68083-864-0

Bookkeeping Graphs: Computational Theory and Applications

Pierre Jinghong Liang
Carnegie Mellon University
and University of Hong Kong
liangj@andrew.cmu.edu

the essence of knowledge
Boston — Delft

Foundations and Trends® in Accounting

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

P. J. Liang. *Bookkeeping Graphs: Computational Theory and Applications*. Foundations and Trends® in Accounting, vol. 17, no. 2, pp. 77–172, 2023.

ISBN: 978-1-63828-165-8

Editorial Scope

Topics

Foundations and Trends® in Accounting publishes survey and tutorial articles in the following topics:

- Auditing
- Corporate Governance
- Cost Management
- Disclosure
- Event Studies/Market Efficiency Studies
- Executive Compensation
- Financial Reporting
- Management Control
- Performance Measurement
- Taxation

Information for Librarians

Foundations and Trends® in Accounting, 2023, Volume 17, 4 issues. ISSN paper version 1554-0642. ISSN online version 1554-0650. Also available as a combined paper and online subscription.

Contents

Bookkeeping Graphs: Computational Theory and Applications

Pierre Jinghong Liang

Carnegie Mellon University, USA and University of Hong Kong, Hong Kong; liangj@andrew.cmu.edu

ABSTRACT

This monograph first describes the graph or network representation of Double-Entry bookkeeping both in theory and in practice. The representation serves as the intellectual basis for a series of applied computational works on pattern recognition and anomaly detection in corporate journal-entry audit settings. The second part of the monograph reviews the computational theory of pattern recognition and anomaly detection built on the Minimum Description Length (MDL) principle. The main part of the monograph describes how the computational MDL theory is applied to recognize patterns and detect anomalous transactions in graphs representing the journal entries of a large set of transactions extracted from real-world corporate entities' bookkeeping data.

Pierre Jinghong Liang (2023), "Bookkeeping Graphs: Computational Theory and Applications", Foundations and Trends® in Accounting: Vol. 17, No. 2, pp 77–172. DOI: 10.1561/1400000070.

1

Introduction and Overview

This monograph grew out of a series of interdisciplinary research projects conducted primarily at Carnegie Mellon University starting in 2017 and, at the time of this writing, still on-going. While future work, like any research endeavor, remains unpredictable, a theme in both the nature of results and in how the work is conducted has emerged, which are recorded here as the main ideas in the monograph. The main ideas are:

1. Representing journal entries as graphs unleashes the power of modern computational graph-mining tools;

2. Academic and practical advances require interdisciplinary teams working closely with industry practitioners.

1.1 Main Idea No. 1: Power of Graph Representation

Double-entry bookkeeping remains a foundation of the financial infrastructure in any modern organization. Not surprisingly, it is one of the 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, 1993). High praises of the bookkeeping system articulated by Johann Wolfgang von Goethe and Arthur Cayley are well-celebrated. Its important role in the rise of capitalism has been raised by Sombart, Webber, and Schumpeter. More recently, interest in double-entry is evident in the works of Waymire and Basu (2008) and Basu and Waymire (2021)

As recognized long ago, a deep connection between linear algebra and double-entry bookkeeping exists; good sources are Ijiri (1967) and Ijiri (1993). By recording each transaction in two accounts, the double-entry system links all accounts of an entity together and in the process creates laws that govern 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 (2018).[1] Beyond its elegance, the structure proves useful in a variety of problem-solving scenarios.[2]

One such scenario, the one we take up in this monograph, involves solving the problem of pattern recognition and anomaly detection among large sets of journal entries within an entity. As proved useful in many applied tasks such as the analysis of the social network, recommendation systems, and telecommunication, analyzing graphs is unusually

[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, 2013). 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."

[2] 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).

useful and makes use of many well-developed and powerful analytic tools. Specifically, the different computational solutions reported in this monograph are unified by sharing a common underlying principle: the Minimum Description Length (MDL) principle. This principle, which originates in the 1970s proposed by Rissanen (1978), has its intellectual roots in the Komolgorov complexity concept in the 1960s (Kolmogorov, 1965, Solomonoff, 1964, and Chaitin, 1969). The original idea is that we can measure the patterns in any object (such as the number π) by the length of computer program that generates the object. A simple example is the very short expression developed by the amazing Indian mathematician Ramanujan who found the following formula around 1910. According to Faloutsos and Megalooikonomou (2007), the first million bits of fractional extension of π can be implemented by Ramanujan's formula

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{0}^{\infty} \frac{(4n)!(1103 + 26390n)}{(n!)^4 396^n}$$

which is probably the shortest and fastest converging formula for π according to Schroeder (2009). So while the first million bits of fractional extension of π appear order-less, it does have a pattern, which is distilled in the short computational program inherent in Ramanujan's formula. This is the core idea behind MDL, which has found wide use and success in modern machine learning. In our work, it has proven useful in analyzing graphs generated by the bookkeeping data. This is why we claim that *representing journal entries as graphs unleashes the power of modern computational graph-mining tools.*

1.2 Main Idea No. 2: Interdisciplinary Collaboration with Computational Scientists and with Industry Partners

The second lesson from conducting research projects on which the current monograph is based is the critical importance of interdisciplinary collaboration. Considering the sizeable distance between research environments of the accounting and computer science fields, an open-minded and sometimes creative, outside-the-box collaboration is indispensable in achieving any substantive, positive outcome. The success of these collaborations relies heavily on the following unique contributory sources:

- **Industry partners**: They are the sources of practical research questions and real-world data.
- **Computer scientists**: While not accounting experts, they possess computational theories and tools that are a must to handle the part of problem-solving that is not familiar to accounting researchers.
- **Accounting scholars**: While not computational experts, the conceptual understanding of bookkeeping and its mathematical representation in matrices and graphs serves as the linchpin connecting the practical problems posed by the industry partners and the problem-solving tools of the computer scientists.

In conducting this work that is a departure from typically social-science styled accounting research, which has been the dominant paradigm since the mid-1960s, accounting researchers are likely to return to their earlier management science roots. That is, it would be useful to:

- Adopt a worldview focusing on the information-processing role of bookkeeping devices such as Double-entry bookkeeping, as capturing economic activities in an efficient way,
- Focus on solving problems faced by practitioners in their daily work (such as how to design solutions to efficiently detect anomalous transactions in the general ledger data), and
- Deploy research methodologies more akin to engineering solutions such as information/coding theory, complexity theory, graph theory and computational tools such as graph mining.

In the end, these works reminded this author of the passages on foundational accounting questions discussed in a 2002 *Accounting Horizon* commentary by select accounting thinkers Joel Demski, John Fellingham, Yuji Ijiri and Shyam Sunder (see Demski *et al.*, 2002). Using the well-known Hatfield (1924) quote:

> "I am sure that all of us who teach accounting in universities suffer from the implied contempt of our colleagues, who look

> upon accounting as an intruder, a Saul among the prophets, a pariah whose very presence detracts somewhat from the sanctity of the academic halls." (page 1)

as the starting point, the commentary attempted to:

> "serves up a positive and ambitious outlook for accounting as a scholarly discipline. Hatfield reminds accountants of their proud heritage; Demski calls for renewed scholarly leadership. We think refocusing on foundational issues in both our educational and research endeavors will invigorate us as individuals as well as our discipline." (page 167)

The initial results shown in the work reported here give some comfort that double-entry bookkeeping, a human invention at least five-hundred-years old and the very foundation of modern accounting, still factors in a substantial way in building cutting-edge computational solutions to the challenging yet practical real-world problems confronting accounting researchers and practitioners.

1.3 Artificial Intelligence in Accounting: The Backdrop

Before proceeding, I provide my own perspective on the current transformation taking place in accounting practice and in academic research and education.

1.3.1 Rise of Machine Learning in Accounting

It is beyond the scope of this work to offer a long-form review of the intellectual history leading up to the current visible advances in applying data-driven tools, either labeled as data-mining, machine learning, digital transformation, or artificial intelligence (you name it!) to accounting practice, research, and education. Here we provide a perspective which may be useful in placing the work reported here into the large, dynamic picture of the changing accounting landscape. This landscape is central to the integration of these AI tools into much of the accounting enterprise (in practice, research or education) whenever and wherever any part of human labor can be replaced by an automated process with equal or higher efficiency.

One can trace back the competing approaches to applied problems we see today all the way to the divergent paths suggested by the AI pioneers in the fateful 1956 Summer Dartmouth workshop gathering, where the term AI is coined. Symbolic reasoning and early expert systems were encouraged by those with a strong theoretical starting point (by participants like, for example, Herb Simon), while inductive systems (by participants like, for example, Solomonoff) were also proposed, serving as early ideas underlying the future rise of machine learning.

Machine learning, with the aid of both faster computer hardware and the exponentially growing size of machine-readable datasets, is now leading the race to realize AI in many parts of the business society. One useful way to view its central function is saving labor costs, broadly defined. Given that accounting practice, research, and education are currently labor intensive, and have been for decades if not centuries, it is no surprise that accounting, like many other disciplines, would be suitable for an industry disruption given the promise of AI. Next, let us use the labor-cost saving theme to discuss the various roles accounting researchers can play in the pending disruption the entire profession must face.

1.3.2 Four Roles for Academic Accountants

One way to organize our thinking about AI and accounting is to group the enterprise into the following four distinct roles or activities for academic accountants.

- **Help save practitioners' labor costs** To achieve this goal, the academic researchers would create new AI tools to (better) solve existing or new accounting problems faced by accounting practitioners in their business environment every day. Labor costs are the primary cost of business for these accounting professionals so a major innovation theme has been replacing labor with machines. The work reported here and others, especially within the data-mining community, starting with those referenced in Margineantu *et al.* (2005) and the KDD workshop report, fall into this category. A recent work by Ding *et al.* (2019) illustrates this approach where machine learning techniques improves an accounting estimate using the data from insurance companies. In a framework-setting piece, Sun (2019) points to the potential for highly sophisticated machine learning tools like deep learning can bring to the practical work of corporate audits. In fact, one on-going collaborative effort currently at CMU is leveraging graph neural network, a deep learning method, to solve anomaly detection problems when the data is both complex (journal entries are high-dimensional objects) and massive (in terms of number of transactions). See Section 4 for a brief description.

One key distinction in this type of work is that new technologies are discovered and developed. That is, it is not typically the case that an off-the-shelf technology (algorithm) can be applied successfully to financial or accounting data. This is because most successful off-the-shelf technologies are not really robust. They may be highly successful but only in a specific application with a very specific task within a specific domain. As a result, when ML applications began to move into new areas beyond the traditional domains (such as the military or healthcare space), new challenges emerge. As an example, while graph mining has been quite popular, the bookkeeping graphs discussed in this monograph present unusual challenges in graph mining because of the uniqueness of the feature-set of the bookkeeping graphs. Within this category, interdisciplinary work, as emphasized earlier in this section, can be extremely important. Future challenges along these lines include the optimal integration of humans and the machine from a technical or engineering perspective.

- **Help save own and other researchers' labor costs** Much of the existing academic accounting research can be labor intensive and consequently the lack of labor may prevent research to ask or solve certain new research problems. Here we have opportunities to adapt and deploy existing AI tools to solve existing accounting research problems better. An obvious path is data gathering: images, speech, natural language, etc. A good example is the long and varied fundamental analysis literature recently invigorated by data mining and AI techniques. Classic works, like Ou and Penman (1989) and Nissim and Penman (2001) and its modern extensions such as Yan and Zheng (2017), focus almost entirely on accounting numbers to explain current and predict firm-level future outcome variables, like earnings and stock returns. Binz *et al.* (2020) also takes an explicit machine learning approach to consider non-linear relations between accounting ratios and returns. Another recent model built by Cao *et al.* (2021) incorporates corporate financial information, qualitative disclosure, and macroeconomic indicators. The recent literature on robo-analysts (Coleman *et al.*, 2020 and Grennan and Michaely, 2020) and the effect of AI-readership on corporate disclosure (Cao *et al.* 2020) are also ready examples here.[3]

- **Use saved up labor cost to address AI-induced new problems** While the promise of AI is allowing researchers to open their minds to new problems made possible only because of AI, the challenges are the thorny problems to the individual or society only brought about by the advances of AI. Like many disruptive technologies before it, AI brings up new problems that have not

[3]Another applying-ML-to-existing-research example is financial-text-as-data. For example, Li (2008) studied the statistical associations between the linguistic features of the annual report (10K filings) and its components, summarized as a Fog index, and numerical information reported in the same or future annual reports such as earnings numbers as well as the persistence of earnings over time. Later work follows this basic framework by extending the set of textual properties of primary accounting documents. The textual features include transparency measures (readability), tone (optimism), and self-serving attribution. Additional work links these extracted properties to economic variables such as book-to-market ratio, accounting accruals, return volatility, cost of capital, litigation, and impact of financial analysts' information processing efficiency. Li (2010a) is an excellent introductory summary. Most research approaches to extracting information from text involve a supervised machine learning model in specific examples like Kogan *et al.* (2009) and Frankel *et al.* (2016) who use support vector regressions, Li (2010b) who uses naive Bayesian model, Brown *et al.* (2020) who use a combination of a topic model and supervised regression, Ke *et al.* (2020) who use a multistep procedure involving a supervised model, and Garcia *et al.* (2021), who use multinomial inverse regression. For a recent example of applying text-regression to traditional topic of post-earnings-announcement-drift or PEAD, consider Meursault *et al.* (2021).

confronted us before. This affords new academic questions. Now that AI is used in society (firms, individuals, governments, etc.), how must we adapt and create new institutions or norms of behavior to minimize its destructive aspects? Here the question about optimal integration of human and machine may also emerge from less of an engineering but more social-economical perspective.[4]

- **Help save students' time learning the accounting tools** Cognitive science has a lot to say about how students learn. With better AI-based technology, instruction and learning can be improved in all areas of learning, including accounting. Research opportunities in accounting education also arise with the help of AI. At the practical level, with the simple fact that our students will graduate to jobs and societies with an increasing presence of AI, it is important to prepare our curriculum to better prepare students.

1.3.3 A Long Way to Go

Every major paradigm shift in the accounting history of thoughts has been accompanied by forces emanating from outside the accounting discipline, in addition to internal forces. These could be outside academic forces such as the rise of information economics within academic economics discipline, or business and societal changes, such as the rise of capital market and thus increased importance of external financial accounting, or the varying levels of general inflation. In this latest iteration, a societal-level driving force has been the marked advance in information technology which dramatically lowers the cost of storing and analyzing massive amounts of data.

What this monograph describes is only the beginning of an interdisciplinary approach to solve particular types of auditing problems faced by practitioners. The eventually successful solutions are likely to incorporate solutions from a host of interdisciplinary research efforts,

[4]For example, Cao *et al.* (2020) show a potential feedback mechanism: Higher AI-readership causes disclosure to be more catered to machine readers (than human readers) by avoiding words that are known to be perceived negatively by computational algorithms.

similar to ours, to address complex accounting and auditing problems beyond what a simple framework, like ours, can fully capture. We have a long way to go in building a robust, new theory of accounting which, like the iterations built by earlier generations of scholars, must respond positively to the environment and must incorporate the best of contemporary scientific ideas and tools into the existing best ideas in accounting thoughts.

Acknowledgements

I am grateful to Professor Jonathan Glover of Columbia University for inviting me to write this monograph. The writing process has further stimulated my interest in continuing my interdisciplinary collaborations. I also appreciate the detailed and constructive comments on the early drafts from Professor Glover and from an anonymous reviewer which greatly improved the monograph.

The monograph is built upon much work that I had participated in in collaboration with a number of colleagues at CMU and other institutions. As such, I wish to acknowledge their influence on my thinking and their contribution to the work reported here. Professor Leman Akoglu has been my primary collaborator here at CMU and, through her, I benefited from many other colleagues from the CMU's CS community: Professor Christos Faloutsos, students and post-docs Hung Nguyen, Dimitris Berberidis, Martin Ma, Sean Zhang, and Jeremy Lee. I also benefited from my colleagues based at CMU's Tepper School of Business, including Professors Zack Lipton, R. Ravi, Bryan Routledge, and doctoral students Aluna Wang, Sang Wu and Lavender Yang. Other Students in my CMU/Tepper doctoral courses where some of the monograph materials are taught have also helped me develop the ideas contained in this monograph.

Outside CMU, I wish to express much support of my work from Professors Rick Antle of Yale, John Fellingham of Ohio State, Haijin Lin of Houston, Hai Lu of Toronto, TJ Wong of Southern California, Sean Cao of Maryland, Jing Li and Pingyang Gao of HKU, Bin Ke of NUS, the Rutgers research group led by Miklos Vasarhelyi and Alexander Kogan, Allen Huang, and Haifeng You of HKUST. The earlier versions of the work related to this monograph have been presented at the University of Houston, University of California at San Diego, Columbia ADP project, SWUFE research institute led by Phil Dybvig, HKUST, University of Hong Kong, Georgetown, National Taiwan University, and Southern Methodist University. I appreciate the opportunities to share ideas with and learn from the participants there.

A part of monograph-writing was completed while I was on a professional leave of absence (LOA) from CMU and visiting HKU during the academic year September 2022 to August 2023. I wish to express my gratitude for the support from Faculty of Business and Economics at the University of Hong Kong (HKU-FBE), especially Accounting Department Chair Pingyang Gao and School Dean Hongbin Cai, for allowing me to concentrate on completing the monograph.

Special thanks to Zac Rolnik and Lucy Wiseman at Now Publishers for the patience and assistance which made the process smooth for me.

Pierre Jinghong Liang (梁景宏)
Pok Fu Lam, Hong Kong Island
Hong Kong
January 2023

Appendices

A

Technical Background on Entropy, Coding, and Code-Length

This section reviews the necessary theoretical ingredients for a computational theory of pattern recognition. Readers familiar with basic information and coding theory of Shannon (1948) may skip sections A.1, A.2, or A.3 respectively.[1]

A.1 Entropy and Information Theory

We now follow Cover (1999) in describing the basic definitions and theorems on entropy, efficient coding, and code length, all necessary ingredients for building a theory of pattern recognition based on minimum description length (MDL).

We begin with a definition of Entropy of a probability distribution.

Definition A.1 (Entropy). Let X be a discrete random variable with alphabet $\mathcal{X}$ and probability mass function $p(x) = Pr\{X = x\}, x \in \mathcal{X}$.

[1]For more complete treatments, consult Cover (1999) for basic information theory and coding, Li *et al.* (2019b) for more formal treatment of the theory and applications of Kolmogorov Complexity, and Grünwald (2007) for the specific application to Minimum description length (MDL) principle.

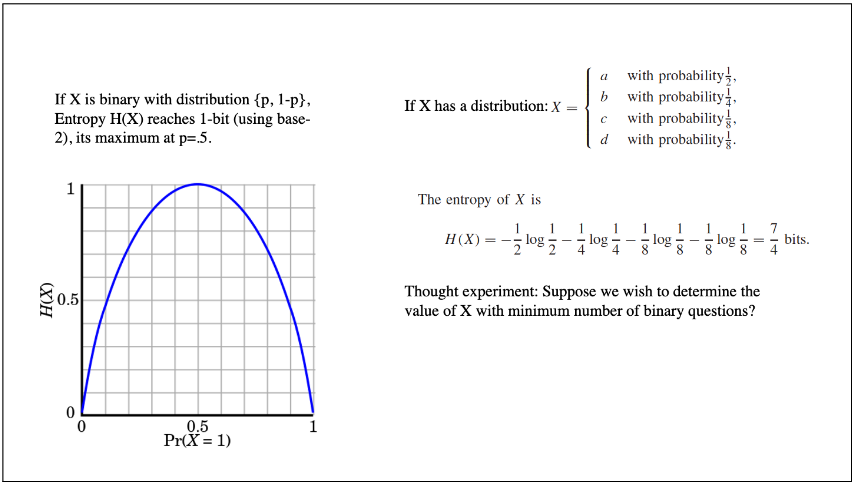


Figure A.1: Some Simple Examples of Entropy Calculation

The *entropy* $H(X)$ of X is

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) log_b \, p(x)$$

where b is the base of logarithm.

- Capital letter (X) denotes a random variable, lower case (x) denotes a particular realization or outcome, and fancy script $(\mathcal{X})$ denotes its alphabet (outcome/sample space);
- Entropy is a property of a distribution $p(x)$ (x may not be a number) but entropy itself is the expectation of a real random variable $g(X) = log_b \frac{1}{p(X)}$:

$$H(X) = \sum_{x \in \mathcal{X}} g(x) p(x) = E_p log_b \frac{1}{p(X)}$$

- If the log is to the base 2 (or e, or 10), entropy is expressed in bits (or nats or bans).

Figure A.1 shows a few examples of entropy.

Entropy Interpretations

- Entropy is a measure of the average uncertainty in a random variable.
- Entropy measures the number of bits on average required to describe the random variable.
- Consider a random variable that has a uniform distribution over 32 outcomes. To identify an outcome, we need a label that takes on 32 different values. Thus, 5-bit strings suffice as labels:

$$H(X) = -\sum_{i=1}^{32} \frac{1}{32} log_2 \frac{1}{32} = log_2 32 = 5(bits)$$

- Suppose we wish to send a message indicating which of the 8 horses won the race. Assume that the probabilities are:

$$\{1/2, 1/4, 1/8, 1/16, 1/64, 1/64, 1/64, 1/64\}$$

  with an $H(X) = 2$.

  – option-1: send an index of the winning horse: 3 bits for any of the horses

  – option-2: send $\{0, 10, 110, 1110, 111100, 111101, 111110, 111111\}$ achieves a lower expected description length 2 bits ($H(X) = 2$).

- Also the lower bound on the average number of questions needed to identify the variable in a game like “20 questions.”

Definitions: Joint and Conditional Entropy

Definition A.2 (Joint Entropy). The *joint entropy* $H(X, Y)$ of a pair of discrete random variables (X, Y) with a joint distribution $p(x, y)$ is defined by

$$H(X, Y) = -\sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} p(x, y) log\ p(x, y)$$

Suppose X and Y are distributed as follows:

$Y \backslash X$	1	2	3	4
1	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{32}$
2	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{32}$	$\frac{1}{32}$
3	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$
4	$\frac{1}{4}$	0	0	0

The entropy of X is

$$H(X) = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{4}\log\frac{1}{4} - \frac{1}{8}\log\frac{1}{8} - \frac{1}{8}\log\frac{1}{8} = \frac{7}{4} \text{ bits.}$$

The entropy of Y is H(Y) = 2 bits;

$$\begin{aligned} H(X|Y) &= \sum_{i=1}^{4} p(Y=i)H(X|Y=i) \\ &= \frac{1}{4}H\left(\frac{1}{2},\frac{1}{4},\frac{1}{8},\frac{1}{8}\right) + \frac{1}{4}H\left(\frac{1}{4},\frac{1}{2},\frac{1}{8},\frac{1}{8}\right) \\ &\quad + \frac{1}{4}H\left(\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4}\right) + \frac{1}{4}H(1,0,0,0) \\ &= \frac{1}{4}\times\frac{7}{4} + \frac{1}{4}\times\frac{7}{4} + \frac{1}{4}\times 2 + \frac{1}{4}\times 0 \\ &= \frac{11}{8} \text{ bits.} \end{aligned}$$

Notice: $H(Y|X) \neq H(X|Y)$,

But $H(X) - H(X|Y) = H(Y) - H(Y|X)$

Figure A.2: Some Simple Examples of Conditional Entropy Calculation

Definition A.3 (Conditional Entropy). If (X,Y) ∼ p(x,y), the *conditional entropy* $H(Y|X)$ is defined by

$$H(Y|X) = -\sum_{X\in\mathcal{X}} p(x)\sum_{y\in\mathcal{Y}} p(y|x)log\ p(y|x) = \sum_{X\in\mathcal{X}} p(x)H(y|X=x)$$

Theorem A.1 (Chain Rules).

$$H(X,Y) = H(X) + H(Y|X)$$

$$H(X,Y) = H(Y) + H(X|Y)$$

$$H(X,Y|Z) = H(X|Z) + H(Y|X,Z)$$

Figure A.2 illustrates some follow-up examples.

Definitions: Relative Entropy

Definition A.4 (Relative Entropy). The *relative entropy* or *Kullback-Leibler distance* between two probability mass function $p(x)$ and $q(x)$ is defined as

$$D(p||q) = \sum_{X\in\mathcal{X}} p(x)log\frac{p(x)}{q(x)} = E_p log\frac{p(X)}{q(X)}$$

Relative Entropy Interpretations

- $D(p||q)$ is always non-negative;
- But $D(p||q)$ is not a true distance because it may violate symmetry and triangle inequality for some probability distributions;
- $D(p||q)$ is a measure of the inefficiency of assuming distribution q when distribution p is true; and
- Under p, $H(p)$ is the average description length, but if we instead use q by mistake, $H(p) + D(p||q)$ is the average length.

Definitions: Mutual Information

Definition A.5 (Mutual Information)**.** The *mutual information* is the relative entropy between the joint distribution p(x,y) and the product distribution p(x)p(y):

$$I(X;Y) = D(p(x,y)||p(x)p(y)) = E_{p(x,y)} log \frac{p(X,Y)}{p(X)p(Y)}$$

Mutual Information Interpretations

- $I(X;Y)$ is a measure of the amount of information that one random variable (X) contains about another random variable (Y).
- $I(X;X) = H(X)$: the original entropy is sometimes referred to as *self-information.*
- The reduction in the uncertainty of X due to the knowledge of Y.
  - **Only true** on average. For a particular realization, say, $Y = y$, $H(X|Y = y) \leq$ or $\geq H(X)$.
  - For example, in a court case, specific new evidence might increase uncertainty, but on average evidence decreases uncertainty.

Suppose X and Y are distributed as follows:

Joint	X= 1	2	3	4	Marginal
Y=1	1/8	1/16	1/32	1/32	1/4
2	1/16	1/8	1/32	1/32	1/4
3	1/16	1/16	1/16	1/16	1/4
4	1/4	0	0	0	1/4
Marginal	1/2	1/4	1/8	1/8	1

Product	X= 1	2	3	4	Marginal
Y=1	1/8	1/16	1/32	1/32	1/4
2	1/8	1/16	1/32	1/32	1/4
3	1/8	1/16	1/32	1/32	1/4
4	1/8	1/16	1/32	1/32	1/4
Marginal	1/2	1/4	1/8	1/8	1

The entropy of X is

$$H(X) = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{4}\log\frac{1}{4} - \frac{1}{8}\log\frac{1}{8} - \frac{1}{8}\log\frac{1}{8} = \frac{7}{4} \text{ bits.}$$

The entropy of Y is H(Y) = 2 bits;

Calculate: $I(X;Y) = \frac{3}{8} bits = .375\ bits$

Check: $H(X) - H(X|Y) = \frac{7}{4} - \frac{11}{8} = \frac{3}{8} bits$

Need to verify:

$$\frac{3}{8} bits = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Figure A.3: Some Simple Examples of Mutual Information Calculation

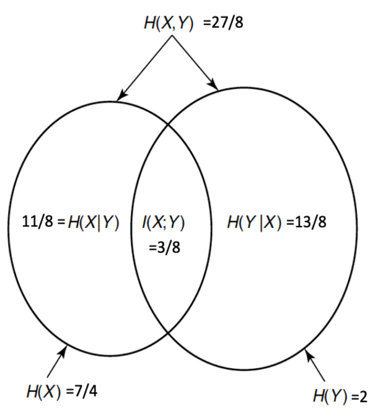


Figure A.4: Relation among entropy H, joint ($H(X,Y)$ and conditional ($H(X|Y)$ entropy, and mutual information ($I(X;Y)$)

Figure A.3 provides numerical examples of mutual information and Figure A.4 provides an illustration of the relation among entropy, joint and conditional entropy, and mutual information.

Theorem A.2 (Non-negativity of Mutual Information)**.** Let X, Y be any random variables

$$I(X; Y) \geq 0$$

with equality if and only if X and Y are independent.

Theorem A.3 (Conditional Independence)**.** Let X, Y, Z be any random variables

$$I(X; Y|Z) \geq 0$$

with equality if and only if X and Y are conditionally independent given Z.

A.2 Codes and Code Length

Prior to this point, we focus on the distribution function of $p(x)$. Recall $H(x)$ and other measures are all functions of $p(x)$, not x themselves. We did not pay any attention to what is in $\mathcal{X}$ other than the number of different $x \in \mathcal{X}$, or its cardinality $|\mathcal{X}|$.

Now we move to deal with objects in the set $\mathcal{X}$. If each $x \in \mathcal{X}$ is complicated in that each requires lots of resources to describe and transmit, it may be a good idea to compress x before transmitting them to others to save resources. In this sense, data compression is definitely an economic activity. Consider sending a voice or picture over long distances. The economy is to convert actual voices or picture segments into numerical strings (or codes) in such a way to minimize the total cost of the transmission. Two human tasks emerge: (1) picking a set of numerical strings to represent voice/picture segments; and (2) constructing sequences of strings in an efficient way. Figure A.5 illustrates the data compression and coding tasks.

Definitions: Codes and Length

Definition A.6 (Source Code)**.** Let $\mathcal{D}^*$ be the set of finite-length strings of symbols from a D-ary alphabet. Define a source code C be a discrete

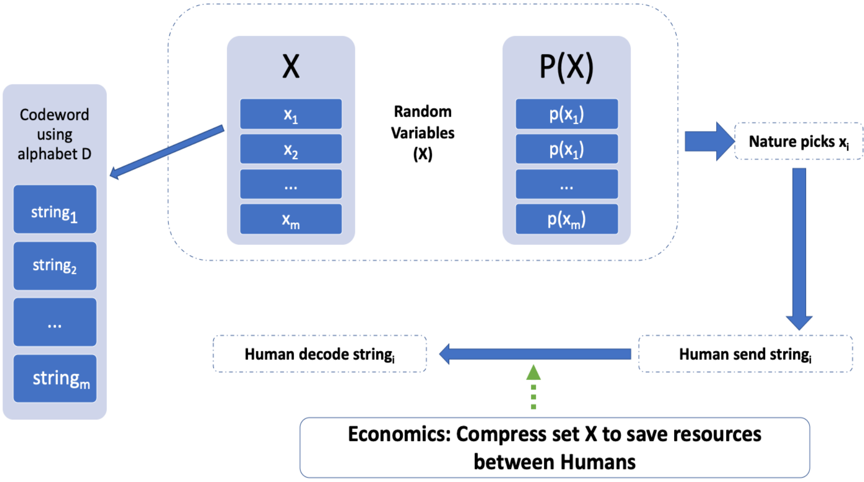


Figure A.5: Summary of Data Compression and Coding

random variable X mapping from $\mathcal{X}$ to $\mathcal{D}^*$. Call $C(x)$ as the codeword corresponding to x and $\ell(x)$ denote the length of $C(x)$.

Definition A.7 (Expected Code Length). Let $p(x)$ be probability mass function of $x \in \mathcal{X}$, the expected code length of a source code $C(x)$ for a discrete random variable X is given by

$$L(C) = \sum_{x \in \mathcal{X}} p(x)\ell(x)$$

Definition A.8 (nonsingular). A code $C(X)$ is *nonsingular* if every element of the range of X maps into a different string in $\mathcal{D}^*$; that is,

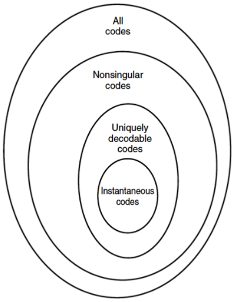
$$x \neq x' \implies C(x) \neq C(x')$$

Definition A.9 (Extension). The *extension* C^* of a code C is a mapping from the finite-length strings of $\mathcal{X}$ to finite-length strings of $\mathcal{D}$, defined by

$$C(x_1 x_2 ... x_n) = C(x_1)C(x_2)...C(x_n)$$

where $C(x_1)C(x_2)...C(x_n)$ indicates concatenation of the corresponding codewords.

Code Hierarchy:



If X has a distribution: $X = \begin{cases} a & \text{with probability } \frac{1}{2}, \\ b & \text{with probability } \frac{1}{4}, \\ c & \text{with probability } \frac{1}{8}, \\ d & \text{with probability } \frac{1}{8}. \end{cases}$

X	Singular	Nonsingular, But Not Uniquely Decodable	Uniquely Decodable, But Not Instantaneous	Instantaneous
a	0	0	10	0
b	0	010	00	10
c	0	01	11	110
d	0	10	110	111

Figure A.6: Code Hierarchy

Definition A.10 (Uniquely Decodable). A code $C(X)$ is *Uniquely Decodable* if the extension of $C(X)$ is nonsingular.

So any encoded C-string in a uniquely decodable code has only one possible source x-string producing it but one may have to look at the entire string to determine even the first symbol in the corresponding source string

Definition A.11 (Prefix Code). A code $C(X)$ is a *prefix* or *instantaneous* code if no codeword is a prefix of any other codeword.

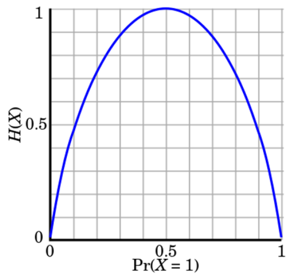
An instantaneous code can be decoded without reference to future codewords or it is *self-punctuating* or (in a case of bad naming) *Prefix-free*.

Figure A.6 illustrates the *Code Hierarchy* described above.

Examples of Source Codes and the Length Consider these two examples:

- A Two-state example: suppose $\mathcal{X} = \{red, blue\}$, here is a simple example of a source code: $C(red) = 00; C(blue) = 11$ with alphabet $\mathcal{D} = \{0, 1\}$.
- A Four-state example is shown in Figure A.7.

If X is binary with distribution {p, 1-p}, Entropy H(X) reaches 1-bit (using base-2), its maximum at p=.5.



If X has a distribution: $X = \begin{cases} a & \text{with probability } \frac{1}{2}, \\ b & \text{with probability } \frac{1}{4}, \\ c & \text{with probability } \frac{1}{8}, \\ d & \text{with probability } \frac{1}{8}. \end{cases}$

The entropy of X is

$$H(X) = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{4}\log\frac{1}{4} - \frac{1}{8}\log\frac{1}{8} - \frac{1}{8}\log\frac{1}{8} = \frac{7}{4} \text{ bits.}$$

Let's code X with C(a) = 0; C(b) = 10; C(c) = 110; C(d) = 111;
The expected L (C) = 1.75 bits = H(X)

Why not use C(b) = 1 to save code-length?

Answer: what if we need to code a sequence of X's:

For example: the bit string 0110111100110 is uniquely decoded as "acdbac"

Figure A.7: Code Hierarchy Examples

A.3 Kraft Inequality and Optimal Codes

We wish to construct instantaneous codes of minimum expected length to describe a given source. It is clear that we cannot assign short codewords to all source symbols and still be prefix-free. The set of codeword lengths possible for instantaneous codes is limited by the following inequality.

Theorem A.4 (Kraft Inequality). For any instantaneous code (prefix code) $C(X)$ over an alphabet of size D (that is $C : \mathcal{X} \to \mathcal{D}^*$ where $|\mathcal{X}| = m$), the codeword lengths $\ell_1, \ell_2, ..., \ell_m$ must satisfy the inequality

$$\sum_{x_i \in \mathcal{X}} D^{-\ell_i} \leq 1$$

Conversely, given a set of codeword lengths that satisfy this equality, there exists an instantaneous code with these word lengths.

From the Krafts theorem, any codeword set that satisfies the prefix condition has to have the corresponding set of *code-lengths* satisfy the Kraft inequality: finding codewords is the same as finding the *lengths* of codewords. So the problem of finding prefix codes with the minimum expected length becomes the same thing as finding/assigning a set of lengths $\ell_1, \ell_2, ..., \ell_m$ satisfying the Kraft inequality and whose expected length $L(C)$ is minimized.

Theorem A.5 (Optimal Prefix Code)**.** The expected length L of any instantaneous D-ary (such as binary, ternary, etc.) code for a random variable X is greater than or equal to the entropy $H_D(X)$; that is

$$L \geq H_D(X)$$

with equality if and only if $D^{-\ell_i} = p_i, \forall i \in \{1, 2, ..., m\}$.

To find the codes, solve a standard constrained optimization problem:

$$\min_{\ell_1,\ell_2,...,\ell_m} \sum p_i \ell_i \qquad \text{(minimize expected code length)}$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{X}} D^{-\ell_i} \leq 1 \qquad \text{(respecting Kraft Inequality)}$$

Assuming the constraint binds, use Lagrange multiplier approach:

$$\begin{aligned}
\mathcal{L} &= \sum p_i \ell_i + \lambda \left(\sum_{i \in \mathcal{X}} D^{-\ell_i} - 1 \right) \\
\frac{\partial \mathcal{L}}{\partial \ell_i} &= p_i - \lambda D^{-\ell_i} log_e D = 0 \\
D^{-\ell_i} &= \frac{p_i}{\lambda log_e D} \longrightarrow \lambda = 1/log_e D \\
p_i &= D^{-\ell_i} \longrightarrow \ell_i^* = -log_D p_i
\end{aligned}$$

Value function evaluated at the optimal ℓ_i^*: $L^* = \sum p_i \ell_i^* = H_D(X)$. This is a remarkable result. Optimal (data) compressing of $\mathcal{X}$ is linked to Entropy via efficient coding. This exhibits the enduring power of the Entropy concept.

Now how to find the optimal codes?

Definition A.12 (*D-adic*)**.** A probability distribution $p(X)$ is *D-adic* if each of the probabilities is equal to D^{-n} for some n.

Here is a procedure for finding an optimal code:

- The *D-adic* distribution that is closest (in the relative entropy sense) to the distribution of X.
- Construct the code by choosing the first available node in the sequence as in the proof of the Kraft inequality.

This procedure is not easy, since the search for the closest *D-adic* distribution is not obvious. Alternatives include a good suboptimal procedure (Shannon-Fano coding) and the a simple procedure called Huffman coding which actually finds THE optimal prefix code (for a known distribution).

A.4 Shannon-Fano Codes

Definition A.13 (Shannon Code). Let p_i denote the $Pr(X = x_i)$ where $\mathcal{X} = \{x_1, x_2, ..., x_m\}$. Shannon Code assigns codeword length to x_i with:

$$\ell_i = log\frac{1}{p_i}$$

Shannon coding may be much worse than the optimal code for some particular symbols. For example, consider two symbols, one of which occurs with probability 0.9999 and the other with probability 0.0001. Then, using Shannon Coding gives codeword lengths of 1 bit and 14 bits. The optimal codeword length is obviously one bit for both symbols. So the Shannon codeword for the infrequent symbol is much longer in the Shannon code than in the optimal code. Figure A.8 is an illustration of Shannon-Fano-Elias Coding.

Competitive Optimality of Shannon Code

- Consider the following two-person zero-sum game: Two people are given a probability distribution and are asked to design an instantaneous code for the distribution.
- A source symbol is drawn from this distribution, and the payoff to player A is 1 or -1, depending on whether the codeword of player A is shorter or longer than the codeword of player B. The payoff is zero for ties.

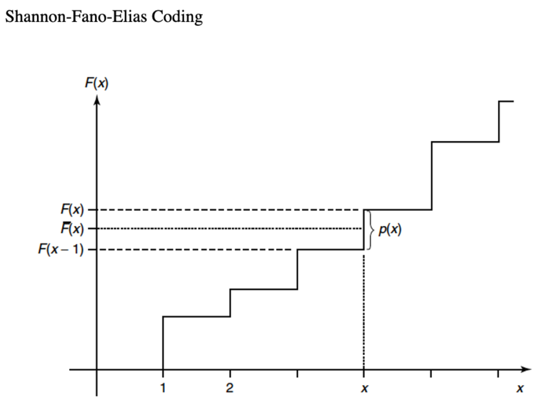


Figure A.8: The Basic Idea of Calibrating a Model Economy

Theorem A.6 (Competitive Optimality of Shannon Code)**.** Let $\ell(x) = log\frac{1}{q(x)}$ be the codeword lengths associated with the Shannon code, and let $\ell'(x)$ be the codeword lengths associated with any other uniquely decodable code. Then

$$Pr(\ell(X) \geq \ell'(X) + c) \leq \frac{1}{2^{c-1}}$$

- Hence, no other code can do much better than the Shannon code most of the time.

As a practical manner, Shannon-Fano-Elias coding is widely used in practice due to its ease of use, especially if expected coding length, not necessarily the codes themselves, is the key consideration. This is precisely the case in pattern recognition applications in machine learning. From here, we connect to Section 2.3 on the 1960s idea of Kolmogorov complexity, a giant discovery in its own right, but serves as the bridge between the original description length ideas of Shannon (1948) to the applied use of description length in pattern recognition inherent in the MDL principle of Rissanen (1978).

B

Formal Problem Statements and Solutions

This appendix provides the formal problem statements and analytic solutions.

B.1 Use MDL Approach to Detect Meta-data Anomalies

Definition B.1 (Bookkeeping Example). A database D is a collection of n *journal entries* where each entry has m column features (such as f_1 is effective date; f_2 is name of the approver; f_3 account debited;...). Each feature $f \in \mathcal{F}$ has a domain $dom(f)$ of possible values (e.g., there are 400 different accounts and 10 approvers). $arity(|\text{accounts debited}|) = 400$:

- The domains are not necessarily distinct between features: some accounts are both debited or credited; some approvers are also initiators.
- An item is a feature-value pair can be $\{$account-debited $=$ cash$\}$.
- An itemset (a pattern) is a pair $\{$account-debited $=$ cash; approver $=$ doe; $\ldots\}$.

Step 1: Define what a model is: The model is a two-column code-table (CT):

- The first column contains patterns (p), i.e., *itemsets* (F), ordered by descending by length and by support;
- The second column contains the codeword $code(p)$;
- $Usage$ of $p \in CT$: number of $t \in D$ containing p in their $cover$.

Table B.1: The Code Table for CompreX

Table 1: An illustrative database D and an example code table CT for a set of three features, F={f_1, f_2, f_3}.

Data	Code Table			
$f_1 f_2 f_3$	$p(F = v)$	$code(p)$	$usage(p)$	$L(code(p))$
a b x	a b x	0	4	1 bit
a b x	a c	10	2	2 bits
a b x	x	110	1	3 bits
a b x	y	111	1	3 bits
a c x				
a c y				

CompreX exploits correlation among some features by building multiple codes, probably smaller tables for each highly correlated group of features instead of a single table for all features (Table B.1).

Definition B.2 (Feature Partitioning)**.** A feature partitioning $\mathcal{P} = \{F_1, F_2, ..., F_k\}$ of a set of features $\mathcal{F}$ is a collection of subsets of $\mathcal{F}$ where:

- Each subset contains one or more features: $\forall F_i \in \mathcal{P}, F_i \neq \emptyset$;
- All subsets are pairwise disjoint: $\forall i \neq j, F_i \cap F_j = \emptyset$; and
- Every feature belongs to a subset: $\cup F_i = \mathcal{F}$.

Step 2: Data encoding scheme: designing a system to encode the patterns and to encode the data using such patterns, with prefix-free codes as basic ingredients.

Step 3: Search algorithm: The search space for finding the best code table for a given set of features, let alone for finding the optimal partitioning of features, is quite large:

– Finding the optimal code table for a set of $|F_i|$ features involves finding all the possible patterns with different value combinations up to length $|F_i|$ and choosing a subset of those patterns that would yield the minimum total cost on the database induced on F_i.

  - Furthermore, the number of possible partitioning of a set of m features is the well-known Bell number.

- While the search space is prohibitively large, it neither has a structure nor exhibits monotonicity properties which could help us in pruning. As a result, we resort to heuristics. Our approach builds the set of code tables in a greedy bottom-up, iterative fashion.
  - Start with $\mathcal{P} = \{f_1, f_2, ..., f_m\}$.
  - Calculate $IG(F_i, F_j) = H(F_i) + H(F_j) - H(F_i, F_j) = M(F_i, F_j)$ for a pair of feature-subsets of the partition.
  - See Akoglu *et al.* (2012) for details.

B.2 Use MDL Approach to Detect Graph Anomalies

Definition B.3 (Bookkeeping Example). A database $\mathcal{G}$ is a collection of J *journal entries* where each entry is represented as a graph $G_j = (V_j, E_j)$ with at least two nodes and one directed edge.

- A node $u \in V_j$ corresponds to an account such as *cash* or *accounts receivable*;
- A directed edge $(u, v) \in E_j$ corresponds to a credit to account u and a debit to account v;
- $m(u, v)$ represents the number of edges from u to v within a same journal entry G_j;
- $\cup_j V_j$ corresponds to the set of all accounts in the company's chart of accounts (COA);
- $\mathcal{T}$ denotes the set of account labels: $\mathcal{T} = \{assets, liabilities, equity\}$ for example.

Step 1: Define what a model is: The model is a two-column Motif-table ($MT \in \mathcal{MT}$):

- The first column contains small graph structures, i.e., *motifs* (g), a connected, directed, node-labeled, simple graph, with possible self-loops on the nodes.
- The second column contains the codeword $code_{MT}(g)$ (or c) with length $\ell(g)$.

Step 2: Data encoding scheme: Design an encoding scheme to encode the motifs table as well as graphs using the motifs using a given motif-table efficiently to convert each graph into their corresponding code-word.

Step 3: Search algorithm.

Definition B.4 (Formal Problem Statement). Given a set of J node-labeled, directed, multi-graphs in $\mathcal{F}$, find a motif table $MT \in \mathcal{MT}$ such that the total compression cost in bits given below is minimized:

$$\min_{MT \subset \mathcal{CT}} L(MT, \mathcal{G}) = L(MT) + \sum_{G_j \in \mathcal{G}} L(G_j|MT)$$

- The key idea of the motif table was to economize over frequencies of sub-graphs commonly used in lots of real journal entries (leading to patterns).

Use Compression to Detect Anomalies Compression based techniques are naturally suited for anomaly and rare instance detection. This is how we exploit the dictionary based compression framework for this task:

- In a given Motif table, the patterns with short code words, that is those that have high usage, represent the sub-graphs in the graph database that can effectively compress the majority of the data points.
- Consequently, the graphs in a graph database can be scored by their encoding cost for anomalousness.
- Formally, for every tuple $t \in D$, compute the anomaly score:

$$score(G_j) = L(G_j|MT) = \sum_{g \in \mathcal{M}: g \in cover(G_j)} L(code(g)|MT)$$

- The higher the score, the more likely it is "to arouse suspicion that it was generated by a different mechanism".

B.3 Use MDL Approach to Evaluate Account Classification

Problem Statement Given a large graph that is *node-labeled, directed, multi-graphs*, create a summary graph which is a representative summary that facilitates the visualization.

Definition B.5 (Formal: Summary Graph)**.** Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be a directed graph with multiplicity $m(e) \in \mathbf{N}$ and node type $l(u) \in \mathcal{T}$. A *Summary graph* is a graph $\mathcal{G}_s = \{\mathcal{V}_s, \mathcal{E}_s\}$ where every super node $v \in \mathcal{V}_s$ is annotated by four components:

- $l(v) \in \mathcal{T}$ is depicted by color;
- $|\mathcal{S}_v|$ denote the number of nodes it contains, depicted by size;
- The *glyph* $\mu(v) \in \mathcal{M}$ depicted by shape; and
- The representative multiplicity $m(v)$ of the edges it summarizes, depicted by a scalar inside the glyph.
- Each super edge $e \in \mathcal{E}_s$ is annotated by $m(e)$ be the representative multiplicity of the edges between super nodes it captures, depicted by a scalar on the super edge.

Now we define decomposition of a given summary graph.

Definition B.6 (Formal: Decompression)**.** A *Summary graph* $\mathcal{G}_s = \{\mathcal{V}_s, \mathcal{E}_s\}$ with the annotation above decompresses uniquely and unambiguously into $\mathcal{G}' = dec(\mathcal{G}_s)\{\mathcal{V}, \mathcal{E}'\}$ according to simple and intuitive rules:

- Every super node expands to the set of nodes it contains, all of which also inherit the super-node's type;
- The nodes are then connected according to the super-node's glyph (for out(in)-stars a node defined as the hub points to (is pointed by) all other nodes, for cliques all possible directed edges are added between the nodes, and for disconnected sets no edges are added);

- Super-edges expand to sets of edges that have the same direction. (If the source/target glyphs involved are not stars, all nodes contained in source glyph point to all nodes contained in target glyph. For stars, expanded incoming and out-going super-edges are only connected to the star's hub); and
- All expanded edges obtain their corresponding "parent" super-node or super-edge representative multiplicity.

Two-part MDL: TG-sum In summary, to apply the MDL principle to a learning task (i.e., the summary graph), we proceed in three main steps.

- **Step 1: Define what the model is:** The model here consists of a list of subsets (v's) of original nodes ($\mathcal{V}$) to merge, a list of glyphs (μ's) to design for a given graph $\mathcal{G}$.
- **Step 2a: How to encode summarization error given summary graph:** Define a suitable encoding scheme, designing a system to encode the patterns and to encode the data using such patterns, with prefix-free codes as basic ingredients.
- **Step 2b: How to encode a summary graph based on the model:** Design a search algorithm, allowing to identify in the data a collection of patterns that yield a good compression under the chosen encoding scheme.
- **Step 3: Search for the best model.**

  Definition B.7 (Formal Problem Statement)**.** Given a node-labeled, directed, multi-graph $\mathcal{G}$, find a summary graph $\mathcal{G}_s$ such that the encoding cost in bits given below is minimized:

$$\mathcal{G}_s := arg \min_{\mathcal{G}'_s} L(\mathcal{G}'_s) + L(\mathcal{G}|\mathcal{H})$$

$$\text{s.t.} \quad \mathcal{H} = dec(\mathcal{G}'_s)$$

B.4 Benchmark Comparisons

As is standard in algorithmic work, benchmark comparisons are conducted where the same data is subjected to investigation by alternate detection algorithms. The following Table B.2 lists competing algorithms used in each of the two approaches we developed specifically for the bookkeeping data. The three technical papers provide additional details for these benchmarks.

Table B.2: Benchmark Algorithms Used

CODEtect	TG-sum
SMT	Navlakha *et al.* (2008)
SUBDUE	SUBDUE
iForest	SNAP
iForest+G2V	CoSum
iForest+DGK	VoG
ENTROPY	GraSS
MULTI-EDGES	Liu *et al.* (2012b)

C

Software Codes

This appendix provides links to codes omitted in the main text. The codes for the algorithms described are available at:

- CompreX: http://www.andrew.cmu.edu/user/lakoglu/tools/CompreX_12_tbox.tar.gz
- CODEtect: https://bit.ly/2P0bPZQ
- TG-SUM: https://bit.ly/2UOX4u6

References

Akoglu, L., H. Tong, and D. Koutra. (2015). "Graph based anomaly detection and description: a survey". *Data mining and knowledge discovery*. 29(3): 626–688.

Akoglu, L., H. Tong, J. Vreeken, and C. Faloutsos. (2012). "Fast and reliable anomaly detection in categorical data". In: *Proceedings of the 21st ACM international conference on Information and knowledge management*. 415–424.

Arya, A., J. C. Fellingham, J. C. Glover, D. A. Schroeder, and G. Strang. (2000a). "Inferring transactions from financial statements". *Contemporary Accounting Research*. 17(3): 366–385.

Arya, A., J. C. Fellingham, B. Mittendorf, and D. A. Schroeder. (2004). "Reconciling financial information at varied levels of aggregation". *Contemporary Accounting Research*. 21(2): 303–324.

Arya, A., J. C. Fellingham, and D. A. Schroeder. (2000b). "Estimating transactions given balance sheets and an income statement". *Issues in Accounting Education*. 15(3): 393–411.

Bao, Y., B. Ke, B. Li, Y. Yu, and J. Zhang. (2019). "Detecting accounting fraud in publicly traded us firms using a machine learning approach. Available at SSRN 2670703."

Basu, S. and G. B. Waymire. (2021). "The evolution of double-entry bookkeeping". *Available at SSRN 3093303*.

Bay, S., K. Kumaraswamy, M. G. Anderle, R. Kumar, and D. M. Steier. (2006). "Large scale detection of irregularities in accounting data". In: *Sixth International Conference on Data Mining (ICDM'06)*. IEEE. 75–86.

Beneish, M. D. (1997). "Detecting GAAP violation: Implications for assessing earnings management among firms with extreme financial performance". *Journal of accounting and public policy*. 16(3): 271–309.

Beneish, M. D. (1999). "The detection of earnings manipulation". *Financial Analysts Journal*. 55(5): 24–36.

Berberidis, D., P. J. Liang, and L. Akoglu. (2020). "TG-sum: Summarizing Directed Multi-Type Multi-Graphs". *under Blind Review Submission*.

Berberidis, D., P. J. Liang, and L. Akoglu. (2022). "Summarizing Labeled Multi-Graphs". *arXiv preprint arXiv:2206.07674*.

Bertomeu, J., E. Cheynel, E. Floyd, and W. Pan. (2019). "Using machine learning to detect misstatements. Available at SSRN 3496297."

Binz, O., S. Katherine, and K. Stanridge. (2020). "What can analysts learn from artificial intelligence about fundamental analysis?"

Brown, N. C., R. M. Crowley, and W. B. Elliott. (2020). "What are you saying? Using topic to detect financial misreporting". *Journal of Accounting Research*. 58(1): 237–291.

Butterworth, J. E. (1972). "The accounting system as an information function". *Journal of Accounting Research*: 1–27.

Cao, S., W. Jiang, J. L. Wang, and B. Yang. (2021). "From man vs. machine to man+ machine: The art and AI of stock analyses". *Tech. rep.* National Bureau of Economic Research.

Cao, S., W. Jiang, B. Yang, and A. L. Zhang. (2020). "How to talk when a machine is listening: Corporate disclosure in the age of AI". *Tech. rep.* National Bureau of Economic Research.

Chaitin, G. J. (1966). "On the length of programs for computing finite binary sequences". *Journal of the ACM (JACM)*. 13(4): 547–569.

Chaitin, G. J. (1969). "On the length of programs for computing finite binary sequences: statistical considerations". *Journal of the ACM (JACM)*. 16(1): 145–159.

Coleman, B., K. J. Merkley, and J. Pacelli. (2020). "Man versus machine: A comparison of robo-analyst and traditional research analyst investment recommendations". *Working paper*.

Cover, T. M. (1999). *Elements of information theory*. John Wiley & Sons.

Cover, T. M. and J. A. Thomas. (2006). "Elements of information theory 2nd edition (wiley series in telecommunications and signal processing)". *Acessado em*.

Dechow, P. M., W. Ge, C. R. Larson, and R. G. Sloan. (2011). "Predicting material accounting misstatements". *Contemporary accounting research*. 28(1): 17–82.

Demski, J. S., J. C. Fellingham, Y. Ijiri, and S. Sunder. (2002). "Some thoughts on the intellectual foundations of accounting". *Accounting Horizons*. 16(2): 157–168.

Ding, K., B. Lev, X. Peng, T. Sun, and M. Vasarhelyi. (2019). "Machine learning improves accounting estimates. Review of Accounting Studies, forth."

Faloutsos, C. and V. Megalooikonomou. (2007). "On data mining, compression, and kolmogorov complexity". *Data mining and knowledge discovery*. 15(1): 3–20.

Fellingham, J. (2018). "The Double Entry System of Accounting". *Accounting, Economics, and Law: A Convivium*. 8(1).

Feltham, G. A. and J. A. Ohlson. (1995). "Valuation and clean surplus accounting for operating and financial activities". *Contemporary accounting research*. 11(2): 689–731.

Frankel, R., J. Jennings, and J. Lee. (2016). "Using unstructured and qualitative disclosures to explain accruals". *Journal of Accounting and Economics*. 62(2-3): 209–227.

Garcia, D., X. Hu, and M. Rohrer. (2021). "The colour of finance words". *Working paper*.

Grennan, J. and R. Michaely. (2020). "Artificial intelligence and high-skilled work: Evidence from Analysts". *Working paper*.

Grünwald, P. D. (2004). "A tutorial introduction to the minimum description length principle". *arXiv preprint math/0406077*.

Grünwald, P. D. (2007). *The minimum description length principle*. MIT press.

Hatfield, H. R. (1924). "Historical defense of bookkeeping". *Journal of Accountancy*. 37(4): 1.

Ijiri, Y. (1965). "On the generalized inverse of an incidence matrix". *Journal of the Society for Industrial and Applied Mathematics*. 13(3): 827–836.

Ijiri, Y. (1967). *The foundations of accounting measurement: A mathematical, economic, and behavioral inquiry*. Prentice-Hall.

Ijiri, Y. (1975). *Theory of accounting measurement*. No. 10. Amer Accounting Assn.

Ijiri, Y. (1993). "The beauty of double-entry bookkeeping and its impact on the nature of accounting information". *Economie Notes by Monte dei Paschi di Siena*. 22(2-1993): 265–285.

Ke, Z. T., B. Kelly, and D. Xiu. (2020). "Predicting returns with text data". *Working paper*.

Keogh, E., S. Lonardi, and C. A. Ratanamahatana. (2004). "Towards parameter-free data mining". In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 206–215.

Kogan, S., D. Levin, B. R. Routledge, J. S. Sagi, and O. A. Noah Smith. (2009). "Predicting risk from financial reports with regression". *Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL,* Boulder, Colorado: 272–280.

Kolmogorov, A. N. (1965). "Three approaches to the quantitative definition ofinformation'". *Problems of information transmission*. 1(1): 1–7.

Lee, M.-C., H. T. Nguyen, D. Berberidis, V. S. Tseng, and L. Akoglu. (2021). "GAWD: graph anomaly detection in weighted directed graph databases". In: *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. 143–150.

Li, F. (2008). "Annual report readability, current earnings, and earnings persistence". *Journal of Accounting and Economics*. 45(2-3): 221–247.

Li, F. (2010a). "Textual analysis of corporate disclosures: A survey of the literature". *Journal of Accounting Literature*. 29: 143–165.

Li, F. (2010b). “The Information Content of Forward-Looking Statements in Corporate Filings—A Naïve Bayesian Machine Learning Approach”. *Journal of Accounting Research*. 48(5): 1049–1102.

Li, J., P. J. Liang, and H. Hwang. (2019a). “Inter-entity bookkeeping networks: Representations and applications”. *Tech. rep.*

Li, M. *et al.* (2019b). *An introduction to Kolmogorov complexity and its applications*. Vol. 4. Springer.

Liang, P. J., A. Wang, L. Akoglu, and C. Faloutsos. (2022). “Pattern Recognition and Anomaly Detection in Bookkeeping Data”. *Carnegie Mellon University Working Papers*.

Liu, F. T., K. M. Ting, and Z.-H. Zhou. (2012a). “Isolation-based anomaly detection”. *ACM Transactions on Knowledge Discovery from Data (TKDD)*. 6(1): 1–39.

Liu, Z., J. X. Yu, and H. Cheng. (2012b). “Approximate homogeneous graph summarization”. *Information and Media Technologies*. 7(1): 32–43.

Ma, X., J. Wu, S. Xue, J. Yang, C. Zhou, Q. Z. Sheng, H. Xiong, and L. Akoglu. (2021). “A comprehensive survey on graph anomaly detection with deep learning”. *IEEE Transactions on Knowledge and Data Engineering*.

Margineantu, D., S. Bay, P. Chan, and T. Lane. (2005). “Data Mining Methods for Anomaly Detection KDD-2005 Workshop Report”. *SIGKDD Explor. Newsl.* 7(2): 132–136. DOI: 10.1145/1117454.1117473.

McGlohon, M., S. Bay, M. G. Anderle, D. M. Steier, and C. Faloutsos. (2009). “Snare: a link analytic system for graph labeling and risk detection”. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1265–1274.

Meursault, V., P. J. Liang, B. R. Routledge, and M. M. Scanlon. (2021). “PEAD. txt: Post-Earnings-Announcement Drift Using Text”. *Journal of Financial and Quantitative Analysis*: 1–50.

Myung, I. J., V. Balasubramanian, and M. A. Pitt. (2000). “Counting probability distributions: Differential geometry and model selection”. *Proceedings of the National Academy of Sciences*. 97(21): 11170–11175.

Navlakha, S., R. Rastogi, and N. Shrivastava. (2008). “Graph summarization with bounded error”. In: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. 419–432.

Nguyen, H. T., P. J. Liang, and L. Akoglu. (2022). “Detecting Anomalous Graphs in Labeled Multi-Graph Databases”. *ACM Transactions on Knowledge Discovery from Data (TKDD)*.

Nissim, D. and S. H. Penman. (2001). “Ratio analysis and equity valuation: From research to practice”. *Review of accounting studies*. 6(1): 109–154.

Noble, C. C. and D. J. Cook. (2003). “Graph-based anomaly detection”. In: *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 631–636.

Ou, J. A. and S. H. Penman. (1989). “Financial statement analysis and the prediction of stock returns”. *Journal of Accounting and Economics*. 11(4): 295–329.

Pacioli, L. (1494; 1994). *Summa de Arithmetica geometria proportioni: et proportionalita...* Paganino de paganini.

Penman, S. H. and S. H. Penman. (2007). *Financial statement analysis and security valuation*. Vol. 3. McGraw-Hill New York.

Perols, J. (2011). “Financial statement fraud detection: An analysis of statistical and machine learning algorithms”. *Auditing: A Journal of Practice & Theory*. 30. DOI: 10.2308/ajpt-50009.

Perols, J. L., R. M. Bowen, C. Zimmermann, and B. Samba. (2017). “Finding needles in a haystack: Using data analytics to improve fraud prediction”. *The Accounting Review*. 92. DOI: 10.2308/accr-51562.

Rissanen, J. (1978). “Modeling by shortest data description”. *Automatica*. 14(5): 465–471.

Rissanen, J. (1998). *Stochastic complexity in statistical inquiry*. Vol. 15. World scientific.

Schroeder, M. (2009). *Fractals, chaos, power laws: Minutes from an infinite paradise*. Courier Corporation.

Shannon, C. E. (1948). “A mathematical theory of communication”. *The Bell system technical journal*. 27(3): 379–423.

Solomonoff, R. J. (1964). “A formal theory of inductive inference. Part I”. *Information and control*. 7(1): 1–22.

Sun, T. (2019). "Applying deep learning to audit procedures: An illustrative framework". *Accounting Horizons*. 33. DOI: 10.2308/acch-52455.

Trudeau, R. J. (2013). *Introduction to graph theory*. Courier Corporation.

Waymire, G. B. and S. Basu. (2008). "Accounting is an evolved economic institution". *Foundations and Trends® in Accounting*. 2(1–2): 1–174.

Yan, X. and L. Zheng. (2017). "Fundamental analysis and the cross-section of stock returns: A data-mining approach". *Review of Financial Studies*. 30(4): 1382–1423.

Zhao, L., S. Sawlani, A. Srinivasan, and L. Akoglu. (2022). "Graph Anomaly Detection with Unsupervised GNNs". *arXiv:2210.09535*.