FDSampleRush: A Monte Carlo Tool for Sampling and Analysing the Distribution of Normal Forms of Relations with Functional Dependencies

Arsyad Kamili, Lee Jia Sheng, and Radhya Fawza

{arsyad.ik, jiashenglee, mradhyaf}@u.nus.edu

Abstract. The pursuit of database normalization, especially to higher forms like BCNF, has traditionally been guided by theoretical principles aimed at reducing redundancy and ensuring data consistency. However, the practical occurrence and distribution of these normal forms across databases with varying attribute numbers remain less quantified. To facilitate this, we developed FDSampleRush, which allows researchers to systematically generate and analyze the distribution of normal forms given functional dependencies within relational schemas. Through Monte Carlo simulations, this paper presents an analysis of distribution of normal forms of a relation with functional dependencies in a relational model: the study challenges the prevailing notion that BCNF is frequently achievable or necessary, particularly in complex schemas. We found that BCNF's occurrence diminishes with increasing attributes, while 3NF and 2NF maintain a relative stability. These insights prompt a reconsideration of the practical application of higher normal forms in database design, advocating for a normalization approach that balances theoretical rigor with empirical reality.

Keywords: Normalization · Functional Dependencies · Distribution of Normal Forms · Relational Model.

1 Introduction

In the evolving landscape of database technology, the normalization process stands as a pillar to ensure data integrity and minimize redundancy. Despite its established theoretical foundation, pioneered by the seminal works of Codd (1970) [1] and further refined by Boyce and Codd (1974) [2], the practical implications of normalization across varied real-world scenarios remain an area for exploration.

This paper aims to investigate the distribution of normal forms of a relation with functional dependencies and introduces FDSampleRush[1], an open source project tool designed to explore this gap by systematically generating relations

[1] https://github.com/KamiliArsyad/FDSampleRush

and functional dependencies to analyze the distribution of normal forms. Our work is motivated by the observation that, while the theoretical underpinnings of normalization are well-documented, the empirical distribution of normal forms under different data generation models has not been extensively explored. This gap in the literature presents an opportunity to approximate how theoretical normal forms manifest in practice using Monte-Carlo sampling methods, offering insights that could refine database design and optimization strategies.

Our contribution is twofold:

1. **Development of a User-Friendly Tool:** We introduce a versatile tool and framework, designed to facilitate the generation of random sets of FDs based on configurable distributions. This tool enables researchers and practitioners alike to simulate a wide array of database scenarios and to test the applicability and effectiveness of various normalization strategies under controlled conditions. The flexibility in configuring distributions allows for a comprehensive exploration of the space of functional dependencies, opening paths for a deeper understanding of normalization impacts in various contexts.
2. **Analysis on the need of performing BCNF Normalization in light of distribution of BCNF and 3NF:** Utilizing the developed tool, we performed an analysis aimed at challenging the practicality and need of achieving Boyce-Codd Normal Form (BCNF). Contrary to the assumption that BCNF normalization should be performed to preserve data integrity and reduce data redundancy, our exploratory findings suggest a more nuanced reality: the prevalence of relations naturally adhering to BCNF diminishes significantly with an increasing number of attributes. On the other hand, the incidence of 3NF and 2NF remains relatively stable and common, raising the question of whether the rigorous pursuit of BCNF is warranted or necessary, particularly in complex schemas with a large attribute set. This challenges the conventional wisdom surrounding normalization but also prompts a reevaluation of the relevancy and applicability of BCNF-related research in the face of complex, real-world database structures.

Our investigation, while preliminary, underscores the necessity of a pragmatic approach to database normalization, advocating for a balanced consideration of theoretical ideals and empirical realities. Through our contributions, we aim to push the discourse on refining normalization methodologies and to inspire future research endeavors that align more closely with the practicality and demands of contemporary database systems.

2 Background

A functional dependency (FD) defines a relationship between attributes in a relation. Let relation R be a relation with attributes $A_1, A_2, ..., A_n$. An FD $X \implies Y$ holds if the values of attributes in Y are uniquely determined by the values of attributes in X, where X and Y are sets of attributes chosen from $A_1, A_2, ..., A_n$ in any combination. A relation can have any number of FDs.

Normal forms define the restrictions on FDs within a relation, guiding the process of database normalization to reduce redundancy and dependency anomalies. A relation can be determined to be in a normal form when its functional dependencies fulfill certain properties. This paper focuses on the first four normal forms: First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF), and Boyce-Codd Normal Form (BCNF).

- **1NF:** All attributes in the relation are atomic values
- **2NF:**
  - Satisfies 1NF, and
  - No partial dependencies exist (i.e. for any FD $X \implies Y$, X is not a proper subset of a candidate key)
- **3NF:**
  - Satisfies 2NF, and
  - No transitive dependencies exist (i.e. for any FDs $X \implies Y$ and $Y \implies Z$, Y is a prime attribute)
- **BCNF:**
  - Satisfies 3NF, and
  - Every determinant is a superkey (i.e. for any FD $X \implies Y$, X is a superkey)

3 Related Work

The journey toward understanding and applying functional dependencies (FDs) within relational databases was pioneered by the work of E.F. Codd [1], who introduced the relational model and laid the groundwork for database normalization, and followed by some significant milestones. The theoretical enhancements by Boyce and Codd with the introduction of the Normal Form, commonly known as the BCNF, further refined our approach to eliminating redundancy and improving data integrity [2]. Bernstein's contributions expanded the theoretical landscape of dependency theory, providing algorithms for dependency inference and normalization, which are critical for optimizing database design [3].

With these theoretical advancements, the practical challenges of discovering and applying FDs in large, complex datasets have driven the development of innovative algorithms and tools. For instance, Akutsu et al. introduced a set cover-based approximation algorithm for inferring FDs, marking a shift towards algorithmic solutions tailored for large-scale data environments [4]. Building on this, Marcus and Lavine proposed a hybrid approach that combines greedy algorithms with Markov Chain Monte Carlo methods, showcasing the potential of probabilistic techniques in uncovering FDs within vast datasets efficiently [5].

Recent works by Ziheng Wei and Sebastian Link emphasize the significance of not just discovering FDs but also discerning their meaningfulness within the context of database normalization and schema design [6]. Their methodologies highlight the ongoing need for tools capable of navigating the complexity of real-world databases, suggesting a trend towards more nuanced and context-aware analysis of FDs.

Despite these advancements to approximate FDs on existing datasets, however, the empirical exploration of normalization forms across different data distributions itself remains notably absent. This gap is not trivial as the distribution of normalization forms on datasets and relations -randomly generated or otherwise, significantly influences database design decisions, performance optimization, and data integrity strategies.

Our project, FDSampleRush, seeks to address this void by providing a tool for systematically generating and analyzing the distribution of normalization forms across different random-uniformly generated sample relations and sets of FDs. By doing so, we aim to provide empirical insights that could challenge or confirm theoretical expectations, offering a new lens through which the database design process can be understood and improved.

4 Methodology

4.1 Sampling and Testing

Given a configurable $n \in \mathbf{Z}^+$ and time limit t, define $Choose(n, D_1, D_2)$ as the algorithm that samples random possible sets of functional dependencies Σ_i relation $\mathbf{R}$ with n attributes, and outputs the cumulative frequency distribution of each normal form. The pseudo-code written in Algorithm 1 explains the general mechanism of $Choose(n)$.

Algorithm 1 $Choose(n, D_1, D_2)$

```
$Result \leftarrow \{\}$
while time spent $<$ t do                ▷ Test different sets of functional dependencies
    Pick a random number $m \sim D_1$ where $0 \leq m \leq 2^{2n}$
    $counter \leftarrow 0$
    $\Sigma \leftarrow \{\}$
    while $counter < m$ do
        Generate a tuple of binary words $(w_1, w_2)$ where $w_1, w_2 \sim$ i.i.d. $D_2$ and len$(w_i) = n$
        $newFD \leftarrow (w_1 \rightarrow w_2)$
        if $newFD \in \Sigma$ then $continue$
        end if
        Append $newFD$ to $\Sigma$
        $counter \leftarrow counter + 1$
    end while
    $res \leftarrow$ TestNF$(\Sigma, R) \in \{$BCNF, 3NF, 2NF, 1NF$\}$
    Append $res$ to $Result$
end whilereturn $Result$
```

4.2 Technical Implementation

To accommodate for the exponential nature of candidate key checking algorithm, we perform some optimization strategies to efficiently compute the randomized generation process and the normal form checks of the sets of FDs.

Representation of Functional Dependency A functional dependency for a relation $\mathbf{R}(a_1, a_2, ..., a_n)$ with n attributes is represented as a pair of binary words of length n, (w_x, w_y) which translates as follows:

A binary word $w = b_{n-1}b_{n-2}b_{n-3}...b_0$ represents the existence of attributes of R where the i-th bit corresponds to whether attribute a_{i+1} is present. The tuple (w_x, w_y) thus corresponds to the functional dependency of B all the present attribute in w_y on A, all the present attribute in w_x; that is, $A \implies B$.

With this representation, performing attribute-related operations like subset attribute check or attribute set join/union becomes extremely fast computation-wise using simple binary logical operations.

5 Evaluation

FDSampleRush features core configurability, enabling users to customize distribution parameters, denoted as D_1 and D_2, for the generation of functional dependency sets. This flexibility allows for precise definitions of two critical dataset aspects: firstly, the cardinality of the set of functional dependencies relative to the number of attributes in a relation; and secondly, the distribution of attributes in the left-hand side X and right-hand side Y within each functional dependency, formulated as $X \implies Y$.

Utilizing these configurable distributions, we conducted sampling and analysis on two different experiments configuring the distribution definitions to determine the influence of attribute quantity and dependency structures on the probability of a relation conforming to various normalization forms.

Experiment 1: The first experiment employs FDSampleRush to analyze the impact of functional dependency distribution on database normalization. We configure the tool with the following distribution settings:

1. Uniform distribution for the cardinality of the set of functional dependencies, i.e., $Pr(|Z| = c) = \frac{1}{n}$, where Z is the set of functional dependencies and $c \in \{1, 2, \ldots, n\}$.
2. Binomial distribution for determining the number of attributes in X and Y independently for each functional dependency, i.e., the probability of choosing k_X attributes in X is $P(|X| = k_X) = \frac{\binom{n}{k_X}}{2^n}$ (similarly for k_Y).

To further clarify the effect of the number of attributes in the relation to the natural occurrence of normal form, we limit the checks to only BCNF checks (as the algorithm does not involve the expensive part - computing candidate keys) with the same distributions to yield the result shown in 2.

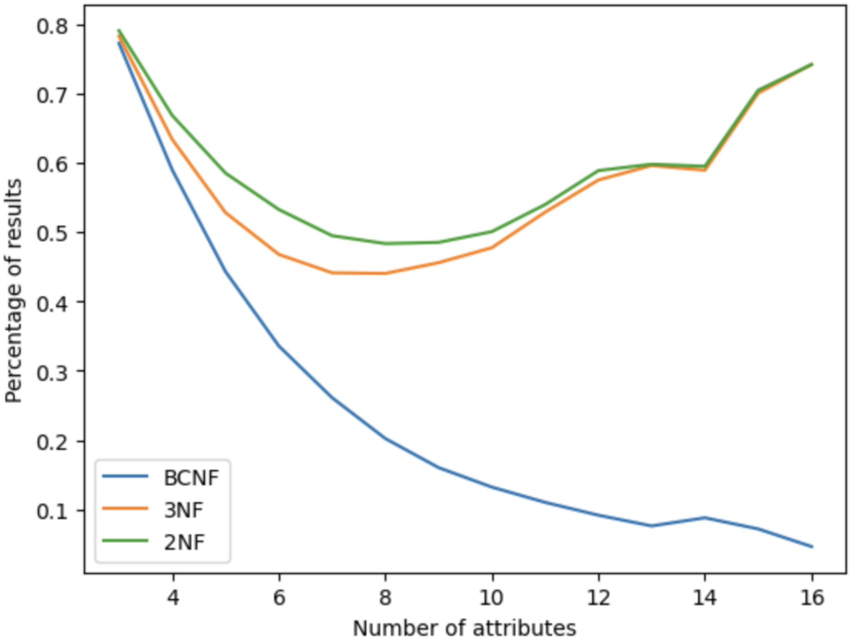

Fig. 1. Experiment 1: The fraction of normal forms in relation to the number of attributes, using a binomial distribution for attribute selection in functional dependencies.

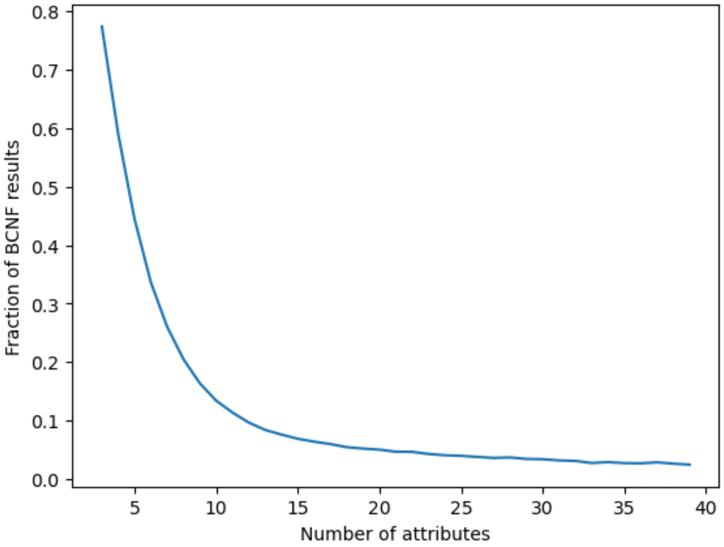

Fig. 2. The relationship between the number of attributes and the fraction of relations naturally occurring in BCNF, using the defined distributions for Experiment 1.

Experiment 2: The second experiment also utilizes FDSampleRush to conduct an analysis with a different configuration of the distribution settings:

1. Uniform distribution for the cardinality of the set of functional dependencies, consistent with the first experiment.
2. Uniform distribution for selecting the number of attributes in X and Y, where selection is made uniformly across the total number of attributes in the relation R.

In our data evaluation using FDSampleRush, both distribution pattern provided by the two experiments presents a monotonically decreasing pattern of BCNF of relation percentage over the number of attribute in the relations. The outcomes of Experiment 2, depicted in the accompanying graph (3), reinforce the findings from Experiment 1 showing the downward trend in BCNF -with the decline being less steep compared to Experiment 1, suggesting that the uniform distribution of attributes in a functional dependency offers a slightly more favorable environment for BCNF but does not significantly alter the overall trend. Additionally, it showed 3NF and 2NF declining in tandem with BCNF, albeit more gradually, suggesting these lower forms of normalization are also affected by increasing attributes but remain more attainable than BCNF in larger schemas.

Both experiments also showed that when relations have fewer attributes, there is a high coincidence between 2NF, 3NF, and BCNF; suggesting that, in simpler schemas, strict normalization is naturally occurring. However, as the number of attributes increases, a steep initial decline in the proportion of relations in BCNF is observed, indicating a challenge in maintaining BCNF with

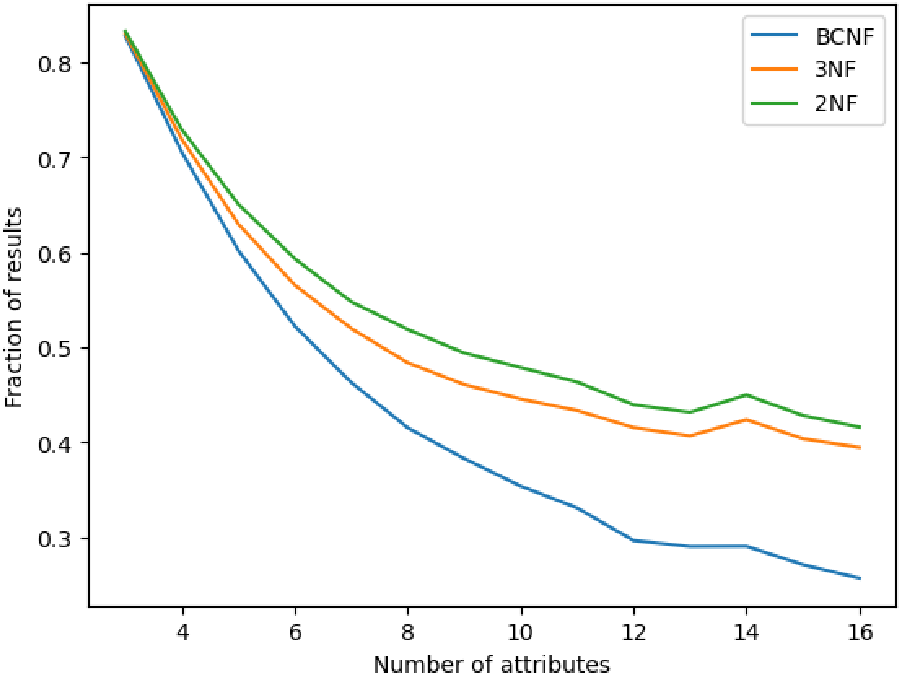

Fig. 3. Graph of the fraction of normal forms of relation against the number of attributes, employing a uniform distribution for attribute selection in functional dependencies.

more complex schemas. Notably, after around 12 attributes on Experiment 1, the differentiation between 2NF and 3NF narrows significantly, hinting at a potential convergence of these forms in more complex database structures following this distribution.

6 Conclusion

The experiments conducted shows an empirically testable insight: strict normalization, while theoretically desirable, may not always be practically necessary or feasible, especially as database schemas grow in complexity. The close alignment between 2NF and 3NF in relations with a larger number of attributes calls into question the necessity of aiming for BCNF, particularly when such efforts may not yield significant benefits over the simpler normal forms.

In light of these findings, our research pushes the need for a nuanced approach to database normalization—one that weighs the theoretical advantages of higher normal forms against the empirical realities of schema design and the diminishing likelihood of naturally achieving BCNF. We have demonstrated the use of FDSampleRush, with its Monte Carlo sampling capabilities, in performing the sampling and analysis of normal forms with respect to different probability distributions of functional dependencies. The tool provides a platform for future studies to further this line of inquiry and enhance the practical understanding of normalization processes. This research, grounded in probabilistic analysis, contributes to a growing body of knowledge that bridges the gap between normalization theory and its application in the field of database systems.

6.1 Future Work Suggestions

The development and initial analysis provided by FDSampleRush open several avenues for future research, particularly in the domain of empirical database studies and the testing of theoretical database normalization principles. Key among these potential directions are:

- **Dataset Generation for Empirical Testing of FD Discovery Algorithms:** FDSampleRush's capability to generate random sets of functional dependencies based on configurable distributions presents a novel opportunity for the empirical testing of database schemas. Future iterations of this work could focus on the generation of comprehensive datasets that simulate real-world complexities. These datasets could serve as benchmarks for evaluating the efficiency, scalability, and accuracy of current FD discovery algorithms and normalization tools, providing valuable insights into their practical applicability.
- **Analysis of Normalization Form Distributions:** Building upon the preliminary findings regarding the challenges of achieving optimal normalization, particularly in the context of BCNF, future work could delve deeper into the distribution of normalization forms across generated datasets. This

research could aim to identify patterns or thresholds where normalization efforts yield diminishing returns, thereby informing more pragmatic approaches to schema design.

- **Contribution to Open Source and Academic Communities:** By making FDSampleRush and its subsequent versions available as open-source tools, we aim to foster a collaborative environment where researchers and practitioners can contribute to its development. This collaborative effort would not only enhance the tool's capabilities but also ensure its relevance and utility in addressing the evolving challenges of database management and optimization.

References

1. E.F. Codd. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, 13(6):377–387, 1970.
2. E.F. Codd, and R.F. Boyce. Recent Investigations in Relational Database Systems. *Proc. IFIP Congress*, pages 1017–1021, 1974.
3. P.A. Bernstein. Synthesizing Third Normal Form Relations from Functional Dependencies. *ACM Transactions on Database Systems (TODS)*, 1(4):277–298, December 1976.
4. T. Akutsu, S. Miyano, and S. Kuhara. A simple greedy algorithm for finding functional relations: efficient implementation and average case analysis. *Theoretical Computer Science*, 292(2):481–495, 2003.
5. R. Marcus, and S. Lavine. An Efficient Algorithm and Monte Carlo Methods for Inferring Functional Dependencies. 2014.
6. Z. Wei, and S. Link. Towards the Efficient Discovery of Meaningful Functional Dependencies. 2023.