Optimality, Accuracy, and Efficiency of an Exact Functional Test

Hien H. Nguyen[1,2] , **Hua Zhong**[1,3] and **Mingzhou Song**[1*]
[1]Department of Computer Science, New Mexico State University, Las Cruces, NM, USA
[2]Pennsylvania State University, Harrisburg, PA, USA
[3]Fred Hutchinson Cancer Research Center, Seattle, WA, USA
hzn5099@psu.edu, hzhong2@fredhutch.org, joemsong@cs.nmsu.edu

Abstract

Functional dependency can lead to discoveries of new mechanisms not possible via symmetric association. Most asymmetric methods for causal direction inference are not driven by the function-versus-independence question. A recent exact functional test (EFT) was designed to detect functionally dependent patterns model-free with an exact null distribution. However, the EFT lacked a theoretical justification, had not been compared with other asymmetric methods, and was practically slow. Here, we prove the functional optimality of the EFT statistic, demonstrate its advantage in functional inference accuracy over five other methods, and develop a branch-and-bound algorithm with dynamic and quadratic programming to run at orders of magnitude faster than its previous implementation. Our results make it practical to answer the exact functional dependency question arising from discovery-driven artificial intelligence applications. Software that implements EFT is freely available in the R package 'FunChisq' ($\geq$2.5.0) at https://cran.r-project.org/package=FunChisq

1 Introduction

As artificial intelligence becomes ubiquitous to capture data harboring dynamic, nonlinear, and non-monotonic relationships, the ability to tell functionally dependent patterns apart from independent patterns can enable one to contemplate on causality—a task important for reasoning. One application domain is biomarker discovery. For example, in aging research, one goal is to detect how longevity is a function of biomarker genes [Zhavoronkov and Mamoshina, 2019]. Numerous statistics have been proposed for association studies [Bewick *et al.*, 2003] and are still counting [Jiang and Wu, 2018; Leung and Drton, 2018; Pardy *et al.*, 2018]. These statistics often have dual optimality. At one extreme, an independent pattern minimizes such a statistic; at the other extreme, a perfect pattern maximizes the statistic. Here minimization and maximization can be switched. A statistic is characterized by the type of perfect pattern on which it optimizes. If a perfect pattern must belong to a family of parametric equations, the statistic is model based; otherwise, the statistic is model free. We focus on statistics derived from contingency tables for pattern discovery, because they alleviate one from the requirement of parametric equations and most of them are model free.

If a perfect pattern optimizing a statistic between X and Y requires either Y being a function of X or X being a function of Y, we call the statistic symmetrically functionally optimal. Model-free symmetric functional optimality underlies statistics of many well-known tests, including Pearson chi-squared test [Pearson, 1922], G-test/mutual information [McDonald, 2014], and Fisher exact test [Fisher, 1922]. Although rarely discussed, such a property is rudimentary to the success of these long-lived tests of association. However, due to symmetry over X and Y, these tests cannot provide evidence for the direction of functional dependency.

To assess functional dependency, one can design tests with asymmetric functional optimality—the test statistic is optimized if and only if Y is a function of X. Two methods in the literature claim such a property: conditional entropy [Cover and Thomas, 2006] and exact functional test (EFT) [Zhong and Song, 2019]. Conditional entropy $H(Y|X)$ is minimized to zero if and only if Y is a function of X. However, no statistical test is associated with $H(Y|X)$. One may attempt to derive the exact p-value for conditional entropy. However, it can be proven that the exact p-values of $H(Y|X)$ and $H(X|Y)$ are always equal at fixed row and column sums, making the corresponding exact test symmetrical. In contrast, EFT is the only known exact test to detect asymmetric functional dependency. The test calculates a p-value by an exact distribution of its test statistic under the null hypothesis that X and Y are independent. Although empirical evaluation showed that EFT promotes functional patterns by demoting non-functional patterns [Zhong and Song, 2019], a theoretical explanation was not offered regarding the optimality of EFT for inferring functional dependency. Other asymmetric methods are founded on principles different from functional optimality. The Kruskal-Wallis test [Kruskal and Wallis, 1952], as the rank-based version of ANOVA, detects differences in the conditional mean of Y given X. Methods for causal inference from discrete data focus on telling the direction only ($X \to Y$ versus $Y \to X$), such as digital regres-

*Contact Author

sion (DR) [Peters *et al.*, 2010], causal inference via stochastic complexity (CISC) [Budhathoki and Vreeken, 2017], and hidden compact representation (HCR) [Cai *et al.*, 2018].

This state-of-the-art of discrete functional inference needs either theoretical justification, clarification on the context of accuracy, or improvement in efficiency. To fill these gaps, we prove the asymmetric functional optimality of EFT to clarify the foundation of functional dependency. We further evaluated the accuracy of EFT in differentiating functional from independent patterns, and it outperformed five other asymmetric methods on simulated patterns with and without noise. Most importantly, EFT is robust to the column (Y) marginal distribution of contingency tables, while three other methods are heavily influenced by the deviation of the Y marginal from a uniform distribution.

Although the previous EFT implementation used a branch-and-bound strategy with quadratic programming to trim down enumerations [Zhong and Song, 2019], it is still not as practical as other exact tests such as Fisher exact test, which is implemented using dynamic programming to avoid recalculating bounds for sub-tables with identical row and column sums [Mehta and Patel, 1983]. Marrying quadratic and dynamic programming, we develop an algorithm to run faster than the previous EFT implementation by several orders of magnitude, comparable to Fisher exact test.

Therefore, we offer the only exact test known to us that is theoretically optimal, empirically accurate, and practically fast to detect functional ($f : X \to Y$) versus independent ($X \perp Y$) patterns on discrete data.

2 The Optimality

Let O be a contingency table with r rows and s columns. Let X and Y be the row and column variables, respectively. Let O_{ij} be a non-negative integer representing the count in the cell at row i and column j. The sum of row i is denoted by R_i, and the sum of column j is denoted by C_j. Let N be the sum of counts in all cells of the table.

The asymmetric test statistic of EFT is computed from the observed table O [Zhong and Song, 2019] by

$$\chi^2_f(O) = \sum_{i=1}^{r}\sum_{j=1}^{s} \frac{(O_{ij} - R_i/s)^2}{R_i/s} - \sum_{j=1}^{s} \frac{(C_j - N/s)^2}{N/s} \quad (1)$$

Under the null hypothesis that X and Y are statistically independent, it was shown earlier that the test statistic asymptotically follows a chi-squared distribution [Zhang and Song, 2013], hence the symbol χ^2. This null distribution is inexact.

Let $\mathcal{S}$ be the set of all feasible $r \times s$ contingency tables with the same row sums and column sums as O:

$$\mathcal{S} = \{A : A_{i\cdot} = R_i \text{ and } A_{\cdot j} = C_j\}$$

where $A_{i\cdot}$ is the sum of row i and $A_{\cdot j}$ is the sum of column j in a contingency table A. Given row and column sums, under the null hypothesis of $X \perp Y$, the probability of observing $A \in \mathcal{S}$ is multivariate hypergeometric [Freeman and Halton, 1951], constituting the exact null distribution:

$$\Pr(A) = \frac{\prod_{i=1}^{r} R_i! \cdot \prod_{j=1}^{s} C_j!}{N! \cdot \prod_{i=1}^{r}\prod_{j=1}^{s} A_{ij}!} \quad (2)$$

The exact statistical significance, or p-value, associated with table O is the sum of probabilities of all tables having a test statistic no less than $\chi^2_f(O)$:

$$p\text{-value} = \sum \Pr(A), \text{ where } A \in \mathcal{S},\ \chi^2_f(A) \geq \chi^2_f(O) \quad (3)$$

A test statistic T on the joint observations of random variables X and Y is called asymmetrically functionally optimal if T is maximized if and only if Y is a function of X. We show that the functional chi-squared statistic $\chi^2_f(O)$ is asymmetrically functionally optimal at fixed marginal sums.

Lemma 1. *Given an $r \times s$ contingency table O of a total of N counts with fixed row and column sums $\{R_i\}$ ($i \in \{1, \ldots, r\}$) and $\{C_j\}$ ($j \in \{1, \ldots, s\}$), the functional chi-squared test statistic $\chi^2_f(O)$ is asymmetrically functionally optimal with the maximum value of $N - \sum_{j=1}^{s} \frac{C_j^2}{N}$.*

Proof. We establish the upper bound of $\chi^2_f(O)$:

$$\frac{1}{s}\chi^2_f(O) = \sum_{i=1}^{r}\sum_{j=1}^{s} \frac{O_{ij}^2}{R_i} - \sum_{j=1}^{s} \frac{C_j^2}{N} \quad (4)$$

$$\leq \sum_{i=1}^{r} \frac{R_i^2}{R_i} - \sum_{j=1}^{s} \frac{C_j^2}{N} = N - \sum_{j=1}^{s} \frac{C_j^2}{N} \quad (5)$$

The bound is reached if and only if each row has only one non-empty cell or equivalently Y is a function of X. □

Theorem 1. *The p-value of EFT is minimized when Y is functionally dependent on X, provided that such a functional dependency is feasible with observed row and column sums.*

Proof. By definition, the p-value of EFT is minimized if $\chi^2_f(O)$ is largest. That means O is the most extreme table, and the p-value in that case is the hypergeometric probability $\Pr(O)$. If feasible, $\chi^2_f(O)$ is largest if Y is a function of X (Lemma 1). If such a function is infeasible for given row or column sums, the most extreme table deviates the most from the null (independent) table, given row and column sums. □

Being asymmetrically functionally optimal, EFT indicates whether Y is more likely functionally dependent on X against X and Y being independent. This is the basis for functional inference. The Pearson's chi-squared statistic [Pearson, 1922] achieves symmetric functional optimality. Its upper bound $N(\min(s, r) - 1)$ [Cramér, 1999] is attained if and only if Y is a function of X or X is a function of Y. The Fisher exact test statistic $1/\Pr(O)$ is also symmetrically functionally optimal. However, symmetric statistics do not promote functions from X to Y over those from Y to X, blind to the direction.

3 The Accuracy

A previous study [Zhong and Song, 2019] showed that EFT promotes functional patterns more than the symmetric Fisher exact test, while desirably demoting independent patterns equally with the Fisher exact test. However, no other asymmetric methods were evaluated. Thus, we compare EFT and

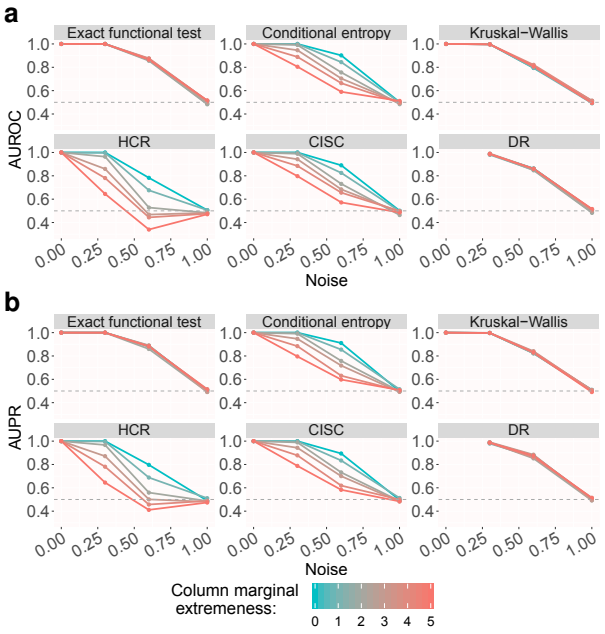

Figure 1: The exact functional test exhibited outstanding robustness to column marginal non-uniformity in contrast to five other methods. EFT maintained good performance in detecting functional from independent patterns, but conditional entropy deteriorated as the column marginal deviated from the uniform distribution. (a) AUROC and (b) AUPR as a function of noise level at six levels of column marginal non-uniformity from 0 (uniform) to 5 (most non-uniform).

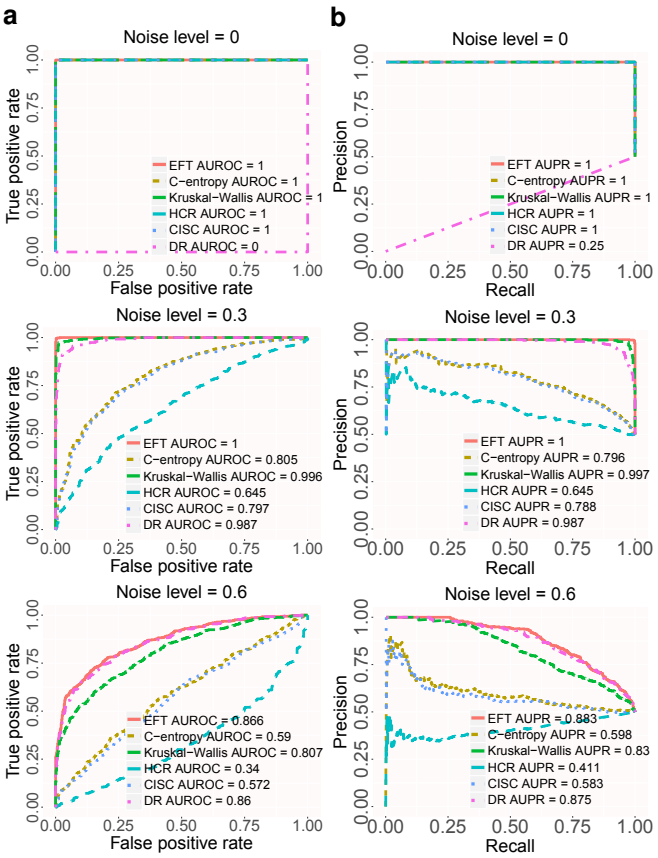

Figure 2: The exact functional test outperforms five other methods in telling apart functional from independent patterns. (a) ROC and (b) PR curves on data of highly non-uniform column marginals ($\tau = 5$).

five other asymmetric methods, including the Kruskal-Wallis test [Kruskal and Wallis, 1952], conditional entropy [Cover and Thomas, 2006], DR [Peters *et al.*, 2010], CISC [Budhathoki and Vreeken, 2017], and HCR [Cai *et al.*, 2018].

The non-parametric Kruskal-Wallis test evaluates whether two or more groups are equal in mean rank. DR determines that X causes Y, if there is an additive noise model $Y=f(X)+\epsilon$, but not vice versa, where ϵ is the noise variable. CISC tells apart the cause and effect by identifying the direction with the lowest approximated Kolmogorov complexity. HCR uses a two-stage process to obtain a compact description of the causal mechanism involved, including mapping the cause variable to a hidden representation, and generating the effect variable from the hidden representation.

We generated various contingency tables by a pattern simulator [Sharma *et al.*, 2017]. In a *functional pattern*, Y functionally depends on X but Y is not a constant function of X; in an *independent pattern*, X and Y are statistically independent with column (Y) marginal distributions varying from being uniform to non-uniform. We randomly generated 12,000 3×3 functional tables of sample size 100 with uniform row (X) marginal distributions. Then we also generated 12,000 independent tables of sample size 100 at six column marginal extremeness levels (τ=0,1,2,3,4,5). The extremeness is controlled by the column sum ratio, set as 1^τ:2^τ:3^τ for 3×3 tables. Column marginals are uniform at τ=0, and become most non-uniform when τ is 5. We evaluate the accuracy of EFT and the five other methods on distinguishing the two pattern types at four noise levels 0, 0.3, 0.6 and 1 using the house noise model [Zhang *et al.*, 2015]. Figure 1 shows the area under the ROC curve (AUROC) and the area under the PR curve (AUPR) as a function of increasing noise levels for each method. As the column marginal distribution deviates from being uniform, we observe decreased AUROC and AUPR of conditional entropy, HCR, and CISC. Meanwhile, EFT, Kruskal-Wallis test, and DR performed well regardless of marginal distributions. At zero noise, however, DR performed poorly with 0 AUROC and 0.25 AUPR in distinguishing functional from independent patterns. Figure 2 shows ROC and PR curves under the most non-uniform column marginal distribution (with $\tau = 5$) at three noise levels. EFT performed the best at all noise levels in telling apart functional from independent patterns, while conditional entropy and recent causal inference methods CISC and HCR performed poorly. Our findings indicate that column marginal non-uniformity could give rise to spurious patterns to which not all methods are robust. Thus direction-only inference alone may be inadequate to prioritize directional from random patterns. Non-uniform marginals do manifest in the real world: single-cell genomics measurements are often skewed to zero due to low RNA capture efficiency [Chen *et al.*, 2019]. As such, robustness to marginal non-uniformity is a strength of EFT for an increased accuracy in functional inference.

4 The Efficiency

Branch-and-bound is often used in exact tests, where the problem of identifying extreme tables for p-value calculation is equivalent to finding extreme paths in a directed acyclic graph. The number of feasible tables in the EFT grows exponentially in sample size N and polynomially in table size $r \times s$. Zhong and Song [2019] designed a branch-and-bound algorithm for EFT. It uses quadratic programming to compute upper and lower bounds of the test statistic to decide whether it is necessary to continue enumerating a table. However, it is still slow for practical use because bounds are inexact and re-computed many times for sub-tables with identical marginal sums. On the other hand, dynamic programming was used with branch-and-bound for Fisher exact test [Mehta and Patel, 1983; Mehta and Patel, 1986; Clarkson *et al.*, 1993] to compute tight bounds regardless of the observed test statistic, potentially incurring unnecessary work for many branches.

To alleviate this hurdle, we develop Algorithm 1 EFT-DQP to combine dynamic and quadratic programming in branch-and-bound. With three steps, it uses quadratic programming to prune the feasible-table network and dynamic programming to generate tight bounds for network traversal, leading to massively reduced search space and improved efficiency.

4.1 A Network Encoding all Feasible Tables

We transform the feasible table set $\mathcal{S}$ into a directed acyclic graph of sub-tables. Each table in the reference set is mapped to a unique path from the only source node to the only sink node in the network. The network consists of $r+1$ layers of nodes. Each node encodes a set of sub-tables with required partial column sums. The number of columns in a sub-table is always s. Each edge encodes values of a row in a table, equal to differences between partial column sums of the two nodes connected by the edge. Traversing an edge from one layer to the next is equivalent to enumerating a row in a table.

Figure 3 illustrates a network encoding all 3×3 tables feasible for row sums $\{4, 2, 3\}$ and column sums $\{2, 3, 4\}$. Layers in the network from top to bottom are numbered $k = r, r-1, \ldots, 0$. An edge from layer k to $k-1$ corresponds to row k of a table. A node in layer k is represented by $(k, C_{1k}, \ldots, C_{sk})$, where $C_{1k}, \ldots, C_{sk}$ are partial column sums of $k \times s$ sub-tables mapping to rows 1 to k in a full table. The source node is $(r, C_{1r}, \ldots, C_{sr}) = (r, C_1, \ldots, C_s)$. The sink node is $(0, 0, \ldots, 0)$. The full network is defined recursively by specifying all nodes $(k-1, C_{1,k-1}, \ldots, C_{s,k-1})$ pointed to by outgoing edges of node $(k, C_{1k}, \ldots, C_{sk})$. The range of $C_{j,k-1}$ given $C_{1,k-1}$, ..., $C_{j-1,k-1}$ is

$$\begin{aligned} &\max\left(0, C_{jk} - R_k + \sum_{l=1}^{j-1}(C_{lk} - C_{l,k-1})\right) \\ &\leq C_{j,k-1} \leq \min\left(C_{jk}, \sum_{l=1}^{k-1} R_l - \sum_{l=1}^{j-1} C_{l,k-1}\right) \end{aligned} \tag{6}$$

4.2 Lengths and Weights of Edges and Paths

Let A be a feasible table corresponding to a path π from source to sink. We define the length and weight of path π by the null probability and test statistic of table A, respectively. Row k of A maps to an edge from node $n = (k, C_{1k}, \ldots, C_{sk})$ to node $m = (k-1, C_{1,k-1}, \ldots, C_{s,k-1})$ on path π: $A_{kj} = C_{jk} - C_{j,k-1}$ $(j = 1, \ldots, s)$. We define this *edge's length* by EDGELENGTH(n, m):

$$R_k! / \prod_{j=1}^{s} [(C_{jk} - C_{j,k-1})!] = R_k! / \prod_{j=1}^{s} [A_{kj}!] \tag{7}$$

Proportional to null probability $\Pr(A)$ (Eq. (2)), we define the *path length* as the product of edge lengths along path π:

$$\text{PATHLENGTH}(\pi) = \frac{\prod_{i=1}^{r} R_i!}{\prod_{i=1}^{r}\prod_{j=1}^{s} A_{ij}!} \tag{8}$$

Test statistic $\chi^2_f(A)$ in Eq. (1) can be written as

$$\chi^2_f(A) = s\left(\sum_{i=1}^{r}\sum_{j=1}^{s}\frac{A_{ij}^2}{R_i} - \sum_{j=1}^{s}\frac{C_j^2}{N}\right) \tag{9}$$

Since s, R_i, C_j, and N are fixed and only relative test statistics are needed for p-value calculation in Eq. (3), we define the first term inside the parentheses as the test statistic $T(A) = \sum_{i=1}^{r}\sum_{j=1}^{s}\frac{A_{ij}^2}{R_i}$. We define the *weight of the edge* from

Algorithm 1 EFT-DQP(Observed table O)

```
1  // Step 1. Build & prune network by quadratic bounds:
2  Initialize the SourceNode
3  for each layer k from layer r (top) to layer 1
4     for each node n in layer k
5        if QLB(n) + min_π{PATHWEIGHT(π)} < T(O)
6           if QUB(n) + max_π{PATHWEIGHT(π)} ≥ T(O)
7              Generate child nodes using Equation (6)
8              Compute quadratic bounds for the child nodes
9  // Step 2. Find tight bounds by dynamic programming:
10 for each node n in layer 1
11    LB(n) = UB(n) = EDGEWEIGHT(n, SinkNode)
12 for each layer k from layer 2 to layer r
13    for each node n in layer k
14       // m is a child node of n
15       LB(n) = min_m{LB(m) + EDGEWEIGHT(n, m)}
16       UB(n) = max_m{UB(m) + EDGEWEIGHT(n, m)}
17 // Step 3. Traverse the network using tight bounds:
18 p-value = 0
19 for each layer k from layer r (top) to layer 1
20    for each node n in layer k
21       for each path π to node n
22          if PATHWEIGHT(π) + UB(n) < T(O)
23             Abandon all branches below node n
24          elseif PATHWEIGHT(π) + LB(n) ≥ T(O)
25             p-value += PATHLENGTH(π) · LENGTHTOSINK(n)
26          else // extending path π to each child node
27             for each child node m
28                π' = addNodeToPath(π, m)
29                PATHWEIGHT(π') = PATHWEIGHT(π)+
30                         EDGEWEIGHT(n, m)
31                PATHLENGTH(π') = PATHLENGTH(π)·
32                         EDGELENGTH(n, m)
33 return p-value
```

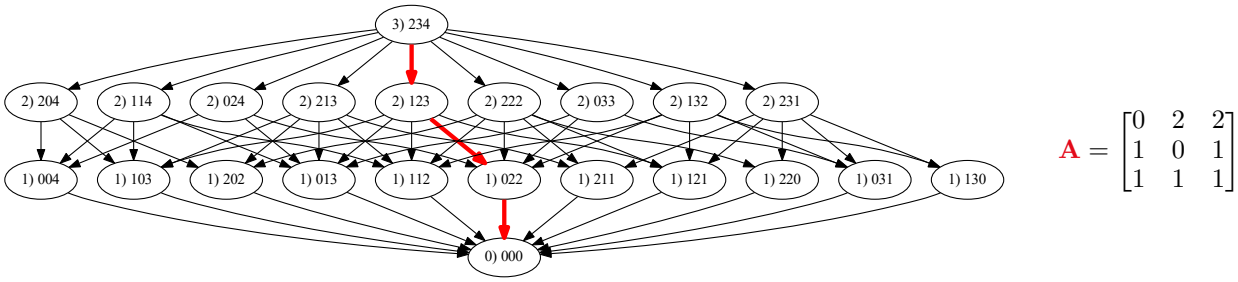

Figure 3: The branch-and-bound network. A directed acyclic graph where paths from the source node to the sink node represent all possible tables with row sums {4,2,3} and column sums {2,3,4}. Each edge maps to a row in a table, but the edges along a path are in the reverse row order. The highlighted path corresponds to table **A** on the right, with the 1st, 2nd, and 3rd edges along the path encoding row three (1,1,1), row two (1,0,1), and row one (0,2,2) of **A**, respectively.

node $n = (k, C_{1k}, \ldots, C_{sk})$ to $m = (k-1, C_{1,k-1}, \ldots, C_{s,k-1})$ by EDGEWEIGHT(n, m):

$$\sum_{j=1}^{s}(C_{jk} - C_{j,k-1})^2/R_k = \sum_{j=1}^{s} A_{kj}^2/R_k \qquad (10)$$

We define the *weight of a path* as the sum of edge weights along the path. So the weight of path π for table A is exactly $T(A)$. The p-value associated with $\chi_f^2(O)$ is equal to

$$p = \sum \Pr(A), \text{ where } A \in \mathcal{S} \text{ and } T(A) \geq T(O)$$

Equivalently, the p-value is the sum of lengths of all paths whose weights are no less than $T(O)$.

4.3 Network Pruning via Quadratic Programming

To avoid traversing the full network, equivalent to enumerating every feasible table, Step 1 of the EFT-DQP algorithm creates a pruned network using quadratic programming bounds. When a branch leads to a subset of tables that are all more or all less extreme than the observed table, any further branches are pruned from the network.

Adapting the quadratic bounds in [Zhong and Song, 2019], we use a quadratic upper bound

$$\text{QUB}(k, C_{1k}, \ldots, C_{sk}) = \sum_{i=1}^{k}\sum_{j=1}^{s} \frac{{A_{ij}^*}^2}{R_i} \qquad (11)$$

where $A_{ij}^* = \min\{U_j, R_i\}$ for j=1 or $\min\{U_j, R_i - \sum_{m=1}^{j-1} A_{im}^*\}$ for j>1 and $\{U_j\}$ is decreasingly sorted $\{C_{1k}, \ldots, C_{sk}\}$. A quadratic lower bound is

$$\text{QLB}(k, C_{1k}, \ldots, C_{sk}) = \sum_{i=1}^{k}\sum_{j=1}^{s} \frac{{A_{ij}^{**}}^2}{R_i} \qquad (12)$$

where A_{ij}^{**}=$\min\{L_j, R_i/s\}$ for j=1 or $\min\{L_j, (R_i - \sum_{m=1}^{j-1} A_{im}^{**})/(s-j+1)\}$ for j>1 and $\{L_j\}$ is increasingly sorted $\{C_{1k}, \ldots, C_{sk}\}$. As a trade-off between tightness and efficiency, both bounds are correct but not always tight.

The current node $n = (k, C_{1k}, \ldots, C_{sk})$ will *not* be expanded when one of two conditions is satisfied:

$$\text{QLB}(n) + \min_{\pi \in \Pi}\{\text{PATHWEIGHT}(\pi)\} \geq T(O) \qquad (13)$$

$$\text{QUB}(n) + \max_{\pi \in \Pi}\{\text{PATHWEIGHT}(\pi)\} < T(O) \qquad (14)$$

where Π represents all sub-paths from source to n. If the condition in Eq. (13) is satisfied, each and every child node of n contributes to tables heavier than O. If the condition in Eq. (14) is satisfied, each and every child node of n belongs to tables lighter than O. In either case, the branches beyond the current node n are pruned.

4.4 Network Traversal via Dynamic Programming

Step 2 in the EFT-DQP algorithm computes tight bounds by dynamic programming on the pruned network. We define LB(node), the exact lower bound for the weight of a sub-path, as the smallest weight among all the sub-paths from this node to sink node. Similarly, we define the exact upper bound UB(node) as the greatest weight among all the sub-paths from this node to the sink node. For each node in bottom layer 1, the initial bounds are $\text{UB}(1, C_{11}, \ldots, C_{s1}) = \text{LB}(1, C_{11}, \ldots, C_{s1}) = (C_{11}^2 + \cdots + C_{s1}^2)/R_1$. For any node $n = (k, C_{1k}, \ldots, C_{sk})$ in layer $k > 1$, the recurrence equations for lower and upper bounds are

$$\text{LB}(n) = \min_m\{\text{LB}(m) + \text{EDGEWEIGHT}(n, m)\} \qquad (15)$$

$$\text{UB}(n) = \max_m\{\text{UB}(m) + \text{EDGEWEIGHT}(n, m)\} \qquad (16)$$

where $m = (k-1, C_{1,k-1}, \ldots, C_{s,k-1})$ is a child node of n and belongs to layer $k-1$.

The recurrence is evidently correct as the definitions are exhaustive. Encouragingly, as a sub-table of required row and column sums can show up in many paths, the two bounds need only computed once. Reusing of bounds for the same sub-table in many paths leads to massive saving in time.

Although dynamic programming can store the bounds in a multidimensional array, it will be inefficient as most entries in the array can be empty due to network pruning. We thus create a hash table with sub-table column marginal sums as key and bounds as value to improve space efficiency.

In Step 3, we use upper bounds to stop enumeration of a partial table that is no longer possible to be more extreme than the observed table O, or use lower bounds to keep all tables that contain the partially enumerated one which guarantees no less extreme than O. We enumerate one row at a time. This step starts from the source node $(r, C_{1r}, \ldots, C_{sr})$. Let the current node be $n = (k, C_{1k}, \ldots, C_{sk})$. The accumulated length and weight of a sub-path π from source to the current

node n are PATHLENGTH(π):

$$\prod_{i=k+1}^{r} \frac{R_i!}{(C_{1i} - C_{1,i-1})! \cdots (C_{si} - C_{s,i-1})!} \tag{17}$$

and PATHWEIGHT(π):

$$\sum_{i=k+1}^{r} \frac{(C_{1i} - C_{1,i-1})^2 + \cdots + (C_{si} - C_{s,i-1})^2}{R_i} \tag{18}$$

We abandon the branch beyond the current node n if

$$\text{PATHWEIGHT}(\pi) + \text{UB}(n) < T(O) \tag{19}$$

We keep all sub-paths from current node n to sink if

$$\text{PATHWEIGHT}(\pi) + \text{LB}(n) \geq T(O) \tag{20}$$

The sum of lengths of all such sub-paths from $n = (k, C_{1k}, \ldots, C_{sk})$ to sink equals

$$\text{LENGTHTOSINK}(n) = \frac{(R_1 + \cdots + R_k)!}{C_{1k}! \cdots C_{sk}!} \tag{21}$$

If condition (20) holds, the extended length of the current path π will be updated as PATHLENGTH(π)·LENGTHTOSINK(n).

4.5 Empirical Evaluation of the EFT Speedup

To compare EFT-DQP and the previous EFT quadratic programming (EFT-QP) implementation [Zhong and Song, 2019], we measured their runtime on contingency tables at increasing dimensions and sample sizes (Figure 4). At the table size of 3×3, both methods ran fast, because quadratic programming has lower overhead than dynamic programming. Remarkably, as the table size increases, the runtime benefit of EFT-DQP becomes overwhelming. In 5×5 tables of sample size 40 (Figure 4), the speedup of EFT-DQP over EFT-QP can be three orders of magnitude or higher.

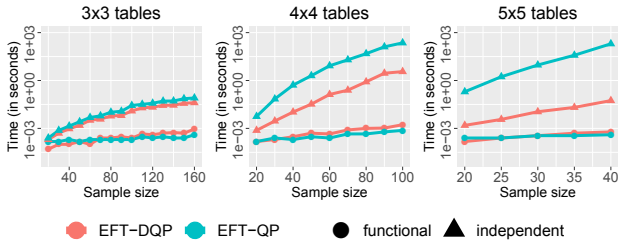

Figure 4: EFT-DQP runs remarkably faster than the previous EFT-QP approach. On functional tables, both algorithms are fast and comparable in runtime. On independent tables, the runtime is more substantial. As the table size and sample size increase, the benefit of EFT-DQP becomes evident on 4×4 tables and extraordinary on 5×5 tables: it is 1500-fold faster than EFT-QP at sample size 40.

5 Discussion

We focus on the question whether one quantity is a function of another (f:$X \to Y$) against them being independent ($X \perp Y$). Other recent causal inference methods answer a different question whether the direction is from X to Y or Y to X.

Measuring the information content of Y conditioned on X, conditional entropy $H(Y|X)$ is both asymmetric and unconditionally functionally optimal. However, it subjects to false positives arising from independent patterns represented by tables with a column (Y) marginal distribution deviating from being uniform. In contrast to conditional entropy, EFT's asymmetric functional optimality is conditioned on the Y marginal distribution. Such conditional optimality, or "imperfectness," is also shared by Fisher exact test (though it is symmetric). Surprisingly, this "imperfectness" counterintuitively frees EFT from the influence of non-uniform Y marginal distributions. Our simulation study suggests that EFT showed substantial statistical robustness to variations in Y marginal distribution. We illustrate this fundamental difference between EFT and conditional entropy on two tables:

$$\begin{bmatrix} 1 & 2 & 15 \\ 2 & 4 & 30 \\ 4 & 8 & 60 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 15 \\ 2 & 30 & 4 \\ 60 & 8 & 4 \end{bmatrix}$$

The first table is an independent table with a highly non-uniform Y marginal distribution. The second table is a strong functional table with a more uniform Y marginal. Conditional entropy computed on both tables takes the same value of 0.5566, failing to distinguish them. In contrast, EFT returns the p-values of 1 and 2.05×10^{-26} respectively for the two tables, correctly recognizing a perfect independent pattern (1st table) and promoting a functional pattern (2nd table).

In practice, one could argue to add a preprocessing step to filter out patterns of extreme Y marginal distributions before applying a direction-only method. However, as the performance of three such methods degrades continuously as Y marginal non-uniformity increases (Figure 1), it can be tricky to decide a threshold to use for filtering. Meanwhile, EFT, balancing between functional dependency and marginals, does not need such a preprocessing step.

It remains an open question whether a test can be effective in detecting both functional (against independent) and directional relationships. Such a test would be convenient to discover directional patterns not spuriously arising from noise.

6 Conclusions

We have established the EFT as an effective method for detecting functional patterns based on its theoretical asymmetric functional optimality, favorable empirical performance over alternative methods, and a practically fast algorithm. We have also argued that functional optimality is the underlying principle of widely used association tests, including Pearson chi-squared test, Fisher exact test, G-test/mutual information, and conditional entropy. Such a principle was not explicitly stated in the past. EFT is the only asymmetric exact test based on this principle. Recent causal inference methods are not suitable for testing functional versus independent patterns. Therefore, our work contributes a practical statistical inference method to assess evidence for functional dependency increasingly important in artificial intelligence.

Acknowledgements

This work was supported by US National Science Foundation grant 1661331.

References

[Bewick *et al.*, 2003] Viv Bewick, Liz Cheek, and Jonathan Ball. Statistics review 8: Qualitative data–tests of association. *Critical Care*, 8(1):46, Dec 2003.

[Budhathoki and Vreeken, 2017] Kailash Budhathoki and Jilles Vreeken. MDL for causal inference on discrete data. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 751–756, 2017.

[Cai *et al.*, 2018] Ruichu Cai, Jie Qiao, Kun Zhang, Zhenjie Zhang, and Zhifeng Hao. Causal discovery from discrete data using hidden compact representation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2666–2674, 2018.

[Chen *et al.*, 2019] Geng Chen, Baitang Ning, and Tieliu Shi. Single-cell RNA-seq technologies and related computational data analysis. *Frontiers in Genetics*, 10:317–317, 04 2019.

[Clarkson *et al.*, 1993] Douglas B. Clarkson, Yuan-An Fan, and Harry Joe. A remark on Algorithm 643: FEXACT: An algorithm for performing Fisher's exact test in $r \times c$ contingency tables. *ACM Trans. Math. Softw.*, 19(4):484–488, December 1993.

[Cover and Thomas, 2006] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, Hoboken, New Jersey, 2nd edition, 2006.

[Cramér, 1999] Harald Cramér. *Mathematical Methods of Statistics*. Princeton University Press, Princeton, New Jersey, 9th edition, 1999.

[Fisher, 1922] R. A. Fisher. On the interpretation of χ^2 from contingency tables, and the calculation of P. *Journal of the Royal Statistical Society*, 85(1):87–94, January 1922.

[Freeman and Halton, 1951] G. H. Freeman and J. H. Halton. Note on an exact treatment of contingency, goodness of fit and other problems of significance. *Biometrika*, 38(1/2):141–149, 1951.

[Jiang and Wu, 2018] Hangjin Jiang and Qiongli Wu. Robust dependence measure for detecting associations in large data set. *Acta Mathematica Scientia*, 38(1):57–72, 2018.

[Kruskal and Wallis, 1952] William H. Kruskal and W. Allen Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621, 1952.

[Leung and Drton, 2018] Dennis Leung and Mathias Drton. Testing independence in high dimensions with sums of rank correlations. *The Annals of Statistics*, 46(1):280–307, 2018.

[McDonald, 2014] John H. McDonald. *Handbook of Biological Statistics*. Sparky House Publishing, Baltimore, Maryland, 3rd edition, 2014.

[Mehta and Patel, 1983] Cyrus R. Mehta and Nitin R. Patel. A network algorithm for performing Fisher's exact test in $r \times c$ contingency tables. *Journal of the American Statistical Association*, 78(382):427–434, 1983.

[Mehta and Patel, 1986] Cyrus Mehta and Nitin Patel. Algorithm 643. FEXACT: A FORTRAN subroutine for Fisher's exact test on unordered $r \times c$ contingency tables. *ACM Trans. Math. Softw.*, 12:154–161, 09 1986.

[Pardy *et al.*, 2018] Christopher Pardy, Sally Galbraith, and Susan R. Wilson. Integrative exploration of large high-dimensional datasets. *The Annals of Applied Statistics*, 12(1):178–199, 2018.

[Pearson, 1922] Karl Pearson. On the χ^2 test of goodness of fit. *Biometrika*, 14:186–191, 1922.

[Peters *et al.*, 2010] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Identifying cause and effect on discrete data using additive noise models. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 597–604, 2010.

[Sharma *et al.*, 2017] Ruby Sharma, Sajal Kumar, Hua Zhong, and Mingzhou Song. Simulating noisy, nonparametric, and multivariate discrete patterns. *The R Journal*, 9(2):366–377, 2017.

[Zhang and Song, 2013] Yang Zhang and Mingzhou Song. Deciphering interactions in causal networks without parametric assumptions. *arXiv preprint*, Molecular Networks:arXiv:1311.2707, 2013.

[Zhang *et al.*, 2015] Yang Zhang, Z Lewis Liu, and Mingzhou Song. ChiNet uncovers gene rewired transcription subnetworks in tolerant yeast for advanced biofuels conversion. *Nucleic Acids Research*, 43(9):4393–4407, 2015.

[Zhavoronkov and Mamoshina, 2019] Alex Zhavoronkov and Polina Mamoshina. Deep aging clocks: The emergence of AI-based biomarkers of aging and longevity. *Trends Pharmacol Sci*, 40(8):546–549, Aug 2019.

[Zhong and Song, 2019] Hua Zhong and Mingzhou Song. A fast exact functional test for directional association and cancer biology applications. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16(3):818–826, May 2019.