Simulation 2

Mengqi Liu

July 2023

1 Setting

- $d \in \{5, 10, 20, 40\}$: number of nodes
- $s0 \in \{1, \mathrm{int}(d/2), d-1, 2d, 3d\}$: expected number of edges
- graph_type: ER, SF, BP
  - ER: Erdős-Rényi Graph randomly choose one from all graphs with d nodes and $s0$ edges + random orientation
  - SF: Generates a scale-free (distribution of node degrees follows a power-law distribution) graph with d nodes and $s0/d$ edges for each node based on the Barabasi-Albert model (grow in a Preferential-Attachment way)
  - BP: Bipartite random graphs with $0.2 * d$ bottom vertices, $0.8 * d$ top vertices and $s0$ edges in total.
- $n = 100$: number of samples, n=inf mimics population risk
- sem_type: gauss, exp, gumbel, uniform, logistic, poisson
- Here I use MCMC to simulate for 20 times to observe the effect of the NOTEARS algorithm (the code ran for three days...).
- Evaluation metrics: FDR, TPR, FPR, Hamming distance, nnz(number of edges predicted positive)

2 Discoveries

The simulation results are shown in Figure 1, 2, 3, 4 and 5.

Increasing d will weaken the effect of FDR control, especially with smaller $s0$. TPR is relatively stable with varying d. FPR and Hamming distance decrease as d increases. This may implies that the algorithm tends to be more conservative with increasing d and we could pay more attention to FDR robust control for relatively sparse DAG with small $s0$ and large d.

In addition to the case of large d and small $s0$, generally speaking, larger $s0$ will lead to worse performance for the same d, graph_type and sem_type. Worse performance refers to larger FDR, TPR and Hamming distance and smaller TPR.

The performance under different sem_type also reveals some characteristics of the algorithm. First, NOTEARS behaves very poorly under Poisson and logistic distribution as it always fails to identify any edge(nnz=0) in the graph no matter what d and $s0$ are. Second, NOTEARS has difficulties controlling FDR and TPR if data is generated by exp or gumbel distribution when $s0$ or d is relatively small. Third, NOTEARS performs badly when $s0$ is large under uniform distribution. Preliminary I conjecture: (1) NOTEARS may not work for discrete distributions.

(2) When d and $s0$ are relatively small, the algorithm is not robust for skewed distributions.
(3) When $s0$ is relatively large, they are not robust for distributions without obvious kurtosis.

Different graph_type will lead to different behaving patterns. SF and BP have larger optimal value of $s0$ in terms of controlling FDR while ER has smaller value. This actually confirms that for purely randomized data, the larger $s0$ is, the algorithm loses the FDR control effect. Besides, the algorithm doesn't work when $s0 \leq d/2$ for SF graphs, which may due to the irregularity in the early stage when graph is generated by the Barabasi-Albert model. But it is very surprising that compared with the more uniformly connected BP and ER graphs, the algorithm still has a similar performance in the scale-free SF graphs.

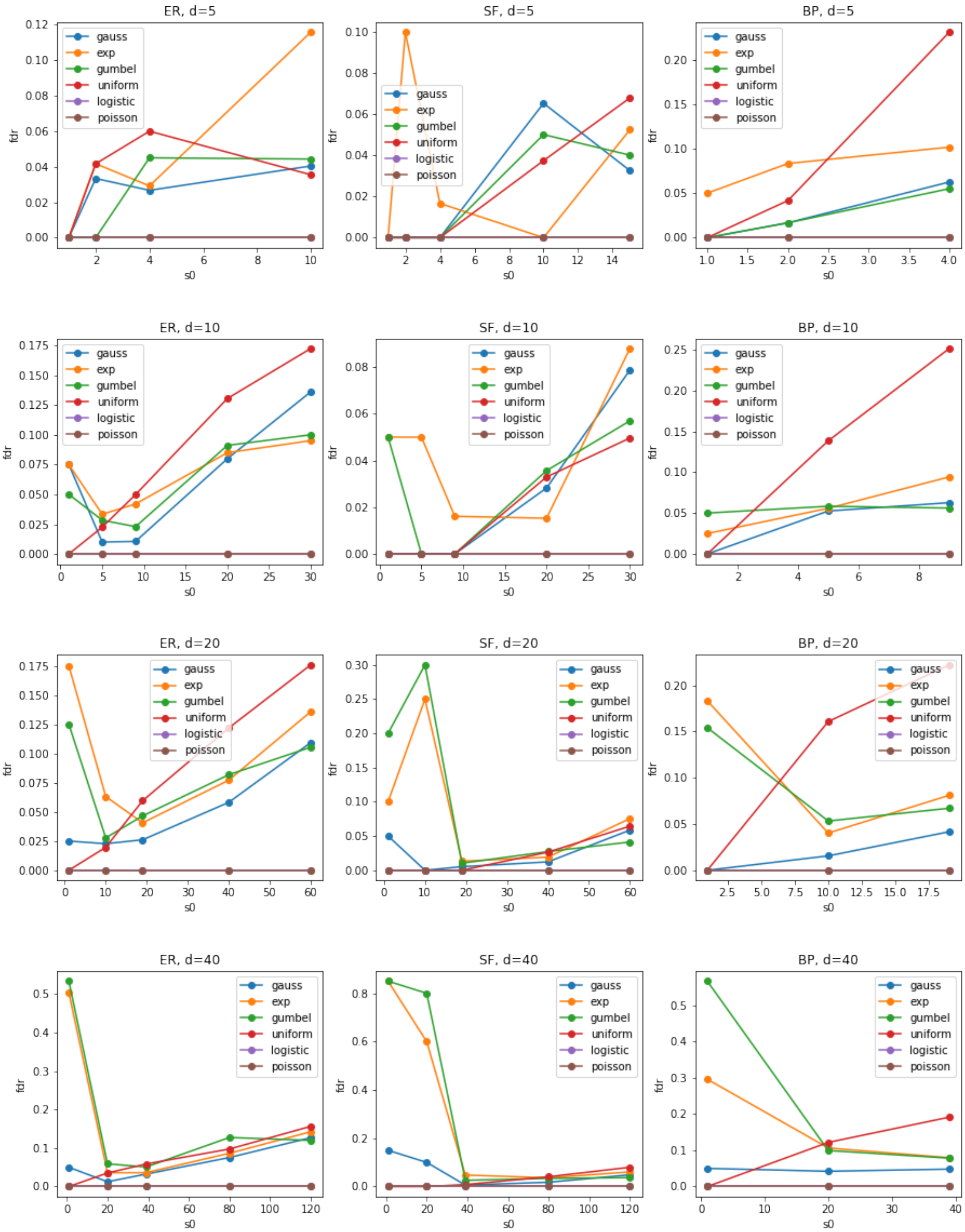

ER, d=5
SF, d=5
BP, d=5
ER, d=10
SF, d=10
BP, d=10
ER, d=20
SF, d=20
BP, d=20
ER, d=40
SF, d=40
BP, d=40
gauss
exp
gumbel
uniform
logistic
poisson
fdr
s0

Figure 1: FDR

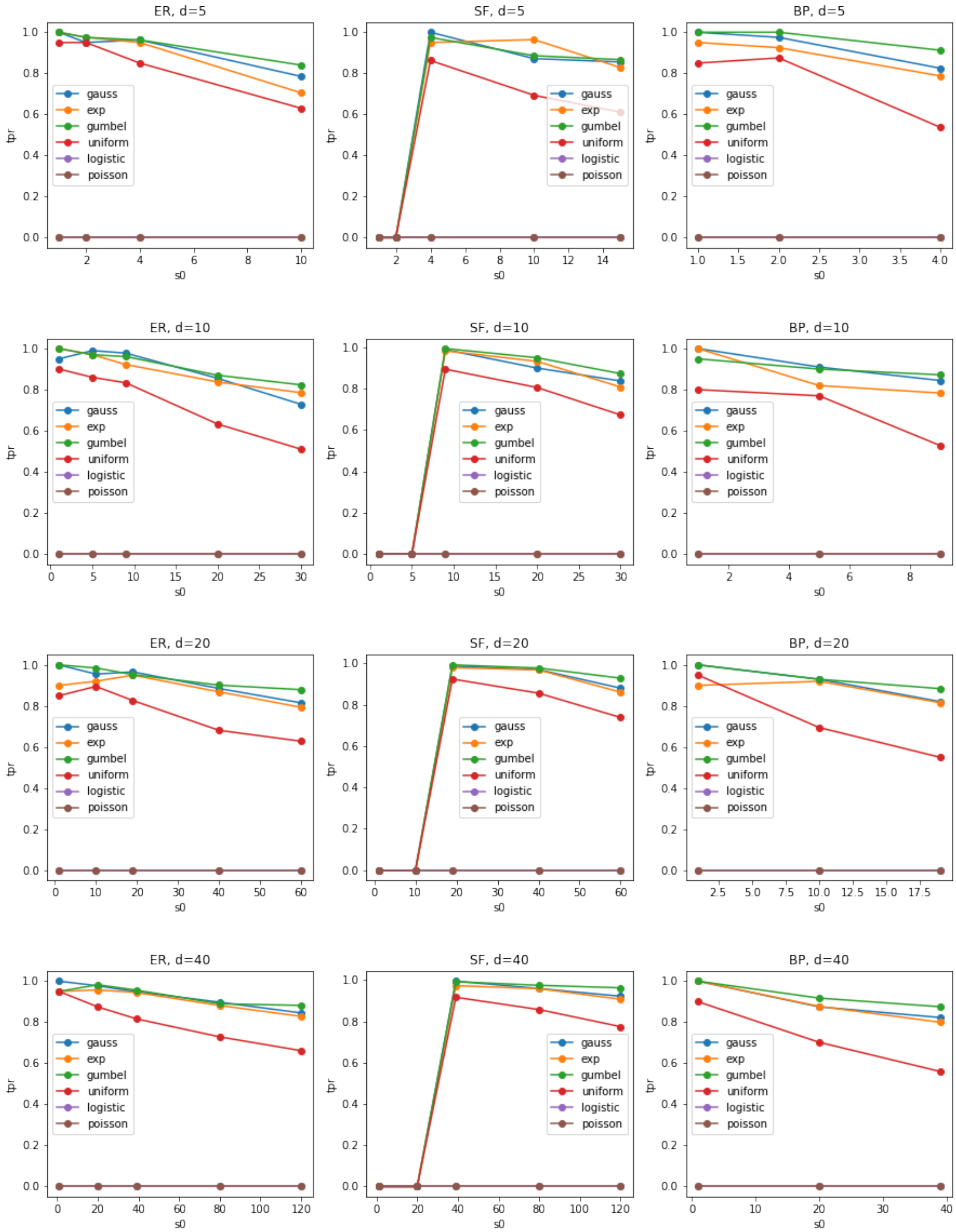

ER, d=5
SF, d=5
BP, d=5
ER, d=10
SF, d=10
BP, d=10
ER, d=20
SF, d=20
BP, d=20
ER, d=40
SF, d=40
BP, d=40
tpr
s0
gauss
exp
gumbel
uniform
logistic
poisson

Figure 2: TPR

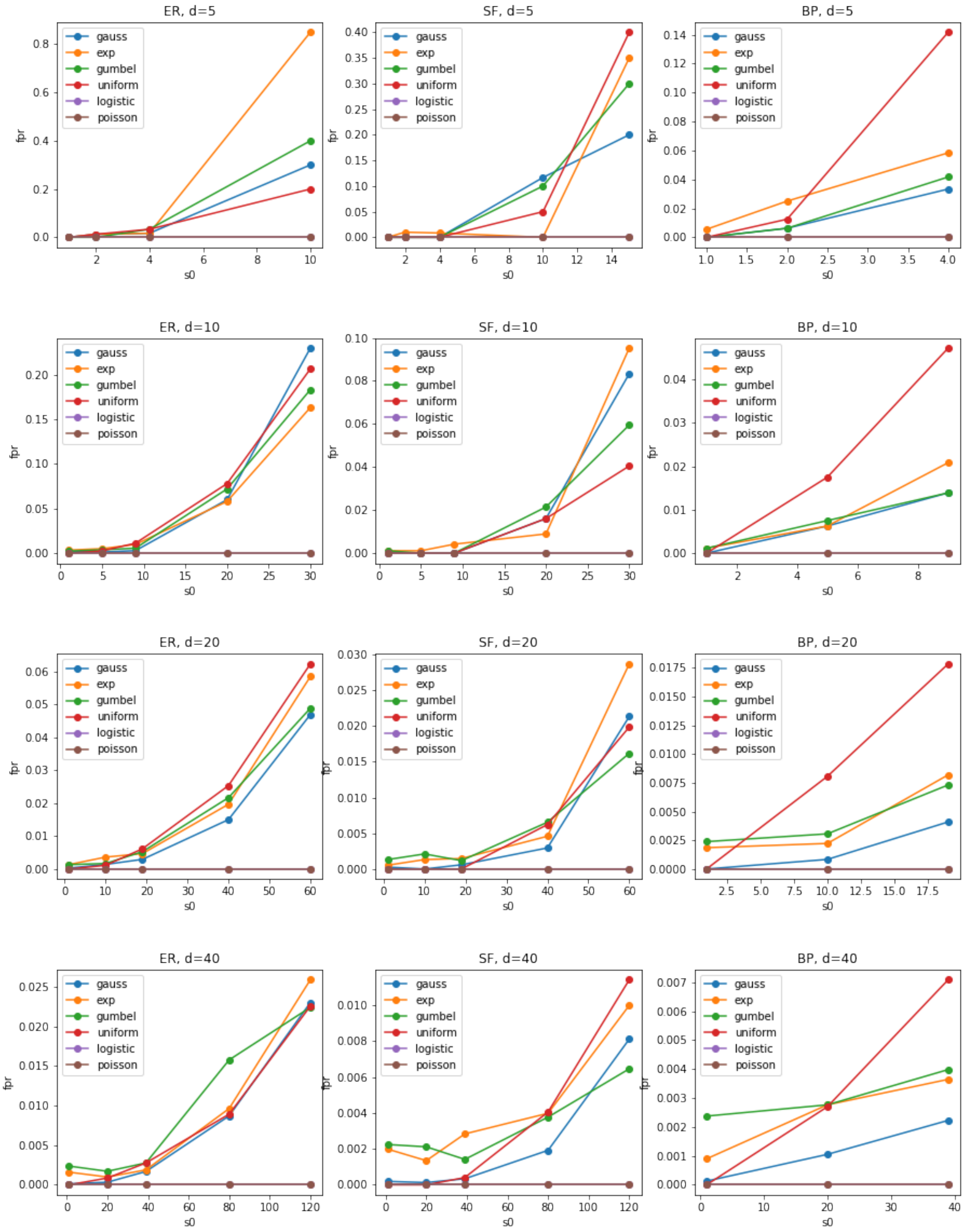

ER, d=5
SF, d=5
BP, d=5
ER, d=10
SF, d=10
BP, d=10
ER, d=20
SF, d=20
BP, d=20
ER, d=40
SF, d=40
BP, d=40
gauss
exp
gumbel
uniform
logistic
poisson
fpr
s0

Figure 3: FPR

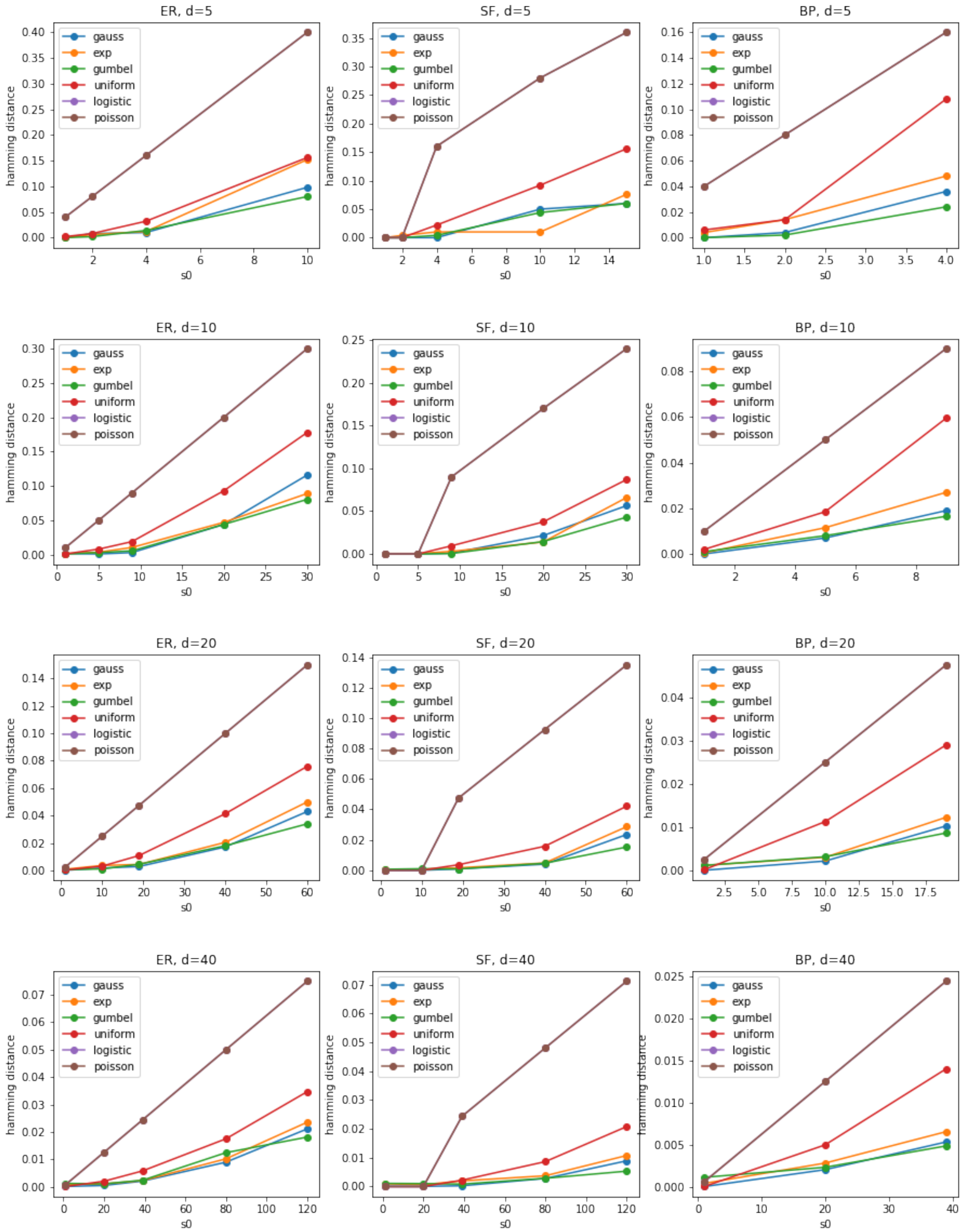
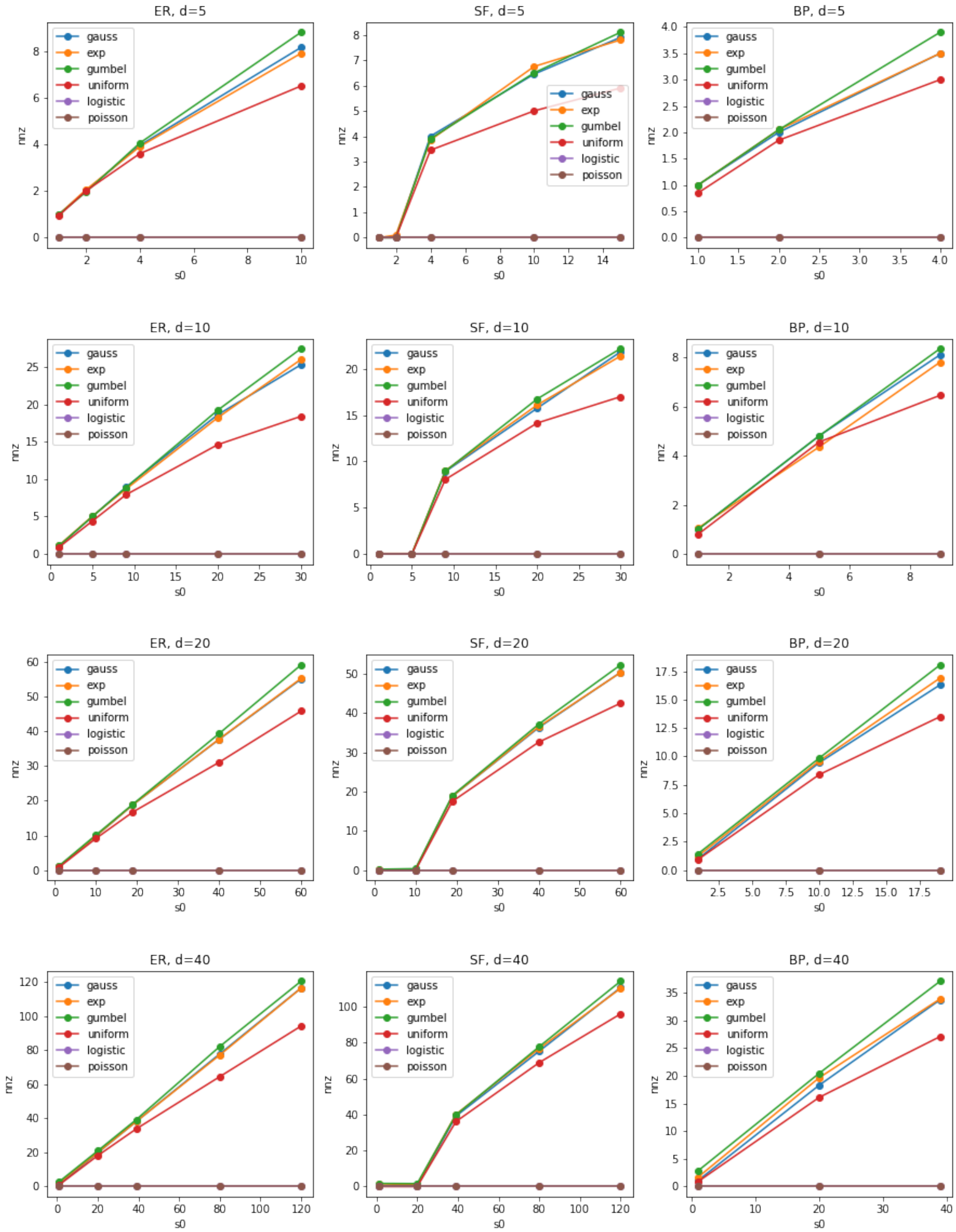


Figure 4: Hamming Distance



ER, d=5
SF, d=5
BP, d=5
ER, d=10
SF, d=10
BP, d=10
ER, d=20
SF, d=20
BP, d=20
ER, d=40
SF, d=40
BP, d=40
gauss
exp
gumbel
uniform
logistic
poisson
nnz
s0

Figure 5: nnz