

Correctness and Performance Charts

Peter Occil

1 Correctness and Performance Charts

This version of the document is dated 2025-01-27.

The following charts show the correctness of many of the algorithms in “**Bernoulli Factory Algorithms**¹” and show their performance in terms of the number of bits they use on average. For each algorithm, and for each of 100 λ values evenly spaced from 0.0001 to 0.9999:

- 500 runs of the algorithm were done. Then...
- The number of bits used by the runs were averaged, as were the return values of the runs (since the return value is either 0 or 1, the mean return value will be in the interval $[0, 1]$). The number of bits used included the number of bits used to produce each coin flip, assuming the coin flip procedure for λ was generated using the `Bernoulli#coin()` method in *bernoulli.py*, which produces that probability in an optimal or near-optimal way.

For each algorithm, if a single run was detected to use more than 5000 bits for a given λ , the entire data point for that λ was suppressed in the charts below.

In addition, for each algorithm, a chart appears showing the minimum number of input coin flips that any fast Bernoulli factory algorithm will need on average to simulate the specified function, based on work by Mendo (2019)^[^1]. Note that some functions require a growing number of coin flips as λ approaches 0 or 1. Note that for the 2014, 2016, and 2019 algorithms—

- an ϵ of $1 - (x + c) * 1.001$ was used (or 0.0001 if ϵ would be greater than 1), and
- an ϵ of $(x - c) * 0.9995$ for the subtraction variants.

Points with invalid ϵ values were suppressed. For the low-mean algorithm, an m of $\max(0.49999, xc1.02)$ was used unless noted otherwise.

1.1 The Charts

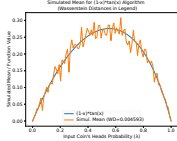
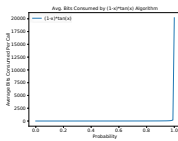
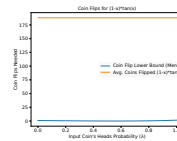
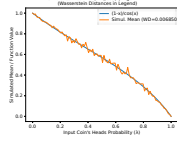
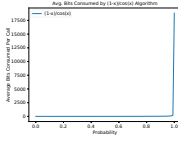
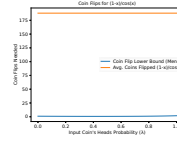
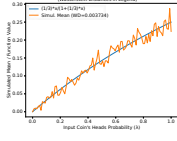
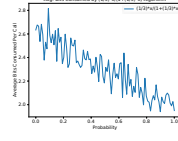
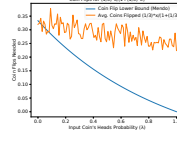
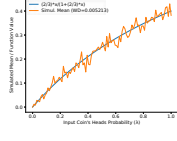
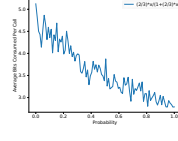
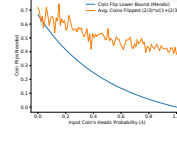
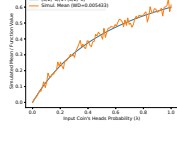
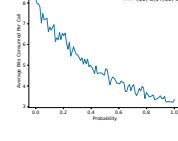
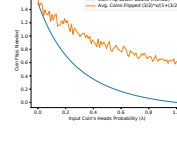
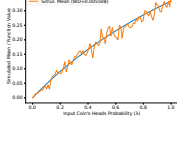
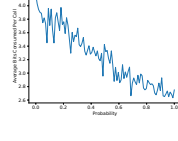
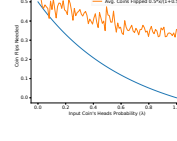
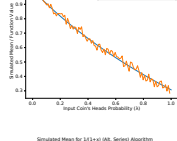
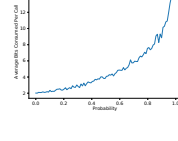
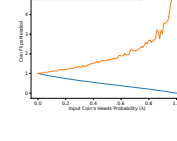
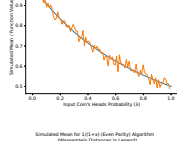
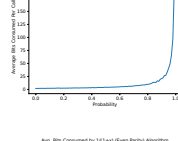
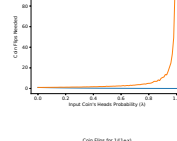
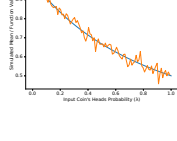
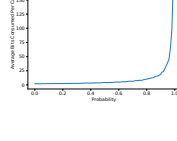
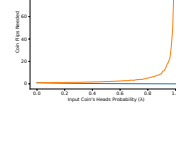
Algorithm	Simulated Mean	Average Bits Consumed	Coin Flips
$(1-x)*\tan(x)$			
$(1-x)/\cos(x)$			
$(1/3)*x/(1+(1/3)*x)$			
$(2/3)*x/(1+(2/3)*x)$			
$(3/2)*x/(1+(3/2)*x)$			
$0.5*x/(1+0.5*x)$			
$1 - \ln(1+x)$ (Alt. Series)			
$1/(1+x)$ (Alt. Series)			
$1/(1+x)$ (Even Parity)			

Figure 10 consists of three subplots showing the performance of the proposed algorithm on the Stanford-Dataset. The top-left plot shows the execution time in seconds versus the input data size in MB. The top-right plot shows the average run completed by the 32-bit algorithm versus the problem size. The bottom plot shows the error rate in percentage versus the input data size in MB.

Top-Left Plot: Execution Time (s) vs Input Data Size (MB)

Input Data Size (MB)	Exact (s)	Approx (s)
0.5	~0.00015	~0.00015
1.0	~0.00015	~0.00015
1.5	~0.00015	~0.00015
2.0	~0.00015	~0.00015
2.5	~0.00015	~0.00015
3.0	~0.00015	~0.00015
3.5	~0.00015	~0.00015
4.0	~0.00015	~0.00015
4.5	~0.00015	~0.00015
5.0	~0.00015	~0.00015

Top-Right Plot: Avg. Run Completed by 32-bit algorithm vs Problem Size

Problem Size	Exact (%)	Approx (%)
0.5	~0.00015	~0.00015
1.0	~0.00015	~0.00015
1.5	~0.00015	~0.00015
2.0	~0.00015	~0.00015
2.5	~0.00015	~0.00015
3.0	~0.00015	~0.00015
3.5	~0.00015	~0.00015
4.0	~0.00015	~0.00015
4.5	~0.00015	~0.00015
5.0	~0.00015	~0.00015

Bottom Plot: Error Rate (%) vs Input Data Size (MB)

Input Data Size (MB)	Exact (%)	Approx (%)
0.5	~0.00015	~0.00015
1.0	~0.00015	~0.00015
1.5	~0.00015	~0.00015
2.0	~0.00015	~0.00015
2.5	~0.00015	~0.00015
3.0	~0.00015	~0.00015
3.5	~0.00015	~0.00015
4.0	~0.00015	~0.00015
4.5	~0.00015	~0.00015
5.0	~0.00015	~0.00015

Figure 1 consists of three subplots. Subplot (a) is a scatter plot titled 'Scatter Plot of True AUC vs. Predicted AUC (10000 Samples)'. The x-axis is 'Predicted AUC' and the y-axis is 'True AUC', both ranging from 0.0 to 1.0. Data points are clustered along the diagonal line y=x. Subplot (b) is a line plot titled 'Average True AUC vs. Avg. BIC Coefficient (10000 Samples)'. The x-axis is 'Avg. BIC Coefficient' (0.00 to 0.50) and the y-axis is 'Average True AUC' (0.00 to 1.00). Multiple colored lines show a general downward trend. Subplot (c) is a line plot titled 'CAPRA vs. CAPRA (10000 Samples)'. The x-axis is 'CAPRA' (0.00 to 1.00) and the y-axis is 'CAPRA' (0.00 to 1.00). All lines are horizontal at y=1.0.

Figure 1 consists of three subplots. Subplot (a) is a scatter plot titled 'Observed Values in Logarithm' on the y-axis and 'Input Data in Logarithm (X)' on the x-axis, both ranging from 0.0 to 1.0. It shows a strong positive linear correlation between observed and predicted values. Subplot (b) is a line graph titled 'Avg. RMSE by Probability' on the y-axis (ranging from 0 to 75) and 'Probability' on the x-axis (ranging from 0.0 to 1.0). It displays multiple lines representing different data series, showing fluctuations in RMSE across probabilities. Subplot (c) is a line graph titled 'Cost Function by Input Variable' on the y-axis (ranging from 0 to 1000) and 'Input Data in Logarithm (X)' on the x-axis (ranging from 0.0 to 1.0). It shows the cost function for various input variables, with most variables having a low, stable cost function, except for 'Cost Clog' which shows a sharp increase as the input variable approaches 1.0.

[illegible]

Figure 1 consists of three subplots labeled (a), (b), and (c). Subplot (a) is titled 'Mean Squared Error (MSE) vs. Input Correlation Coefficient (rho)'. The x-axis is 'Input Correlation Coefficient (rho)' ranging from 0.0 to 0.8. The y-axis is 'Mean Squared Error (MSE)' ranging from 0.0 to 0.4. It shows four data series: '2016 AAdv. v+1.1' (blue line with circles), '2016 AAdv. v+1.2' (green line with circles), '2016 AAdv. v+1.3' (orange line with circles), and '2016 AAdv. v+1.4' (red line with circles). All series show a decreasing trend as rho increases. Subplot (b) is titled 'Average Squared Error (ASE) vs. Probability'. The x-axis is 'Probability' ranging from 0.0 to 0.7. The y-axis is 'Average Squared Error (ASE)' ranging from 0.0 to 600. It shows four data series: '2016 AAdv. v+1.1' (blue line with circles), '2016 AAdv. v+1.2' (green line with circles), '2016 AAdv. v+1.3' (orange line with circles), and '2016 AAdv. v+1.4' (red line with circles). The ASE values are generally low, with some fluctuations. Subplot (c) is titled 'Cost Function vs. Input Correlation Coefficient (rho)'. The x-axis is 'Input Correlation Coefficient (rho)' ranging from 0.0 to 0.8. The y-axis is 'Cost Function' ranging from 0.0 to 100. It shows four data series: '2016 AAdv. v+1.1' (blue line with circles), '2016 AAdv. v+1.2' (green line with circles), '2016 AAdv. v+1.3' (orange line with circles), and '2016 AAdv. v+1.4' (red line with circles). The cost function values are generally low, with some fluctuations.

Figure 1 consists of three subplots comparing the proposed method (blue line) with the state-of-the-art (orange line) across different frequencies (0.0 to 0.5).

- Left Subplot: Empirical Mean**
 - Y-axis: Spectral Power (0.0 to 0.4)
 - X-axis: Frequency (0.0 to 0.5)
 - Legend:
 - Empirical Mean of 2018 A&A: $\alpha = 0.2$ (0.000497)
 - Empirical Mean of 2018 A&A: $\alpha = 0.3$ (0.000497)
 - Empirical Mean of 2018 A&A: $\alpha = 0.4$ (0.000497)
 - Empirical Mean of 2018 A&A: $\alpha = 0.5$ (0.000497)
 - Empirical Mean of 2018 A&A: $\alpha = 0.6$ (0.000497)
 - Empirical Mean of 2018 A&A: $\alpha = 0.7$ (0.000497)
 - Empirical Mean of 2018 A&A: $\alpha = 0.8$ (0.000497)
 - Empirical Mean of 2018 A&A: $\alpha = 0.9$ (0.000497)
 - Empirical Mean of 2018 A&A: $\alpha = 1.0$ (0.000497)
- Middle Subplot: Avg. Bias Coefficient**
 - Y-axis: Avg. Bias Coefficient (0 to 250)
 - X-axis: Frequency (0.0 to 0.5)
 - Legend:
 - 2018 A&A: $\alpha = 0.2$
 - 2018 A&A: $\alpha = 0.3$
 - 2018 A&A: $\alpha = 0.4$
 - 2018 A&A: $\alpha = 0.5$
 - 2018 A&A: $\alpha = 0.6$
 - 2018 A&A: $\alpha = 0.7$
 - 2018 A&A: $\alpha = 0.8$
 - 2018 A&A: $\alpha = 0.9$
 - 2018 A&A: $\alpha = 1.0$
- Right Subplot: Cost Function**
 - Y-axis: Cost Function (0 to 10)
 - X-axis: Frequency (0.0 to 0.5)
 - Legend:
 - 2018 A&A: $\alpha = 0.2$
 - 2018 A&A: $\alpha = 0.3$
 - 2018 A&A: $\alpha = 0.4$
 - 2018 A&A: $\alpha = 0.5$
 - 2018 A&A: $\alpha = 0.6$
 - 2018 A&A: $\alpha = 0.7$
 - 2018 A&A: $\alpha = 0.8$
 - 2018 A&A: $\alpha = 0.9$
 - 2018 A&A: $\alpha = 1.0$

Figure 1 consists of three subplots illustrating the performance of the proposed algorithm. The left subplot shows the Normalized Mean Squared Error (MSE) versus Input Class Probabilities (x) for 2000 Samples per Class and 2000 Samples per Class - 10%. The middle subplot shows the Average Bit Count per Element versus Probabilities for 2004 data with 100, 1000, and 5000 elements. The right subplot shows the Cost (bits per element) versus Input Class Probabilities (x) for 2004 data with 100, 1000, and 5000 elements, along with the Average Cost for 1000 elements.

[illegible]

Figure 1 consists of four subplots (a, b, c, d) showing the performance of the proposed algorithm. Subplot (a) is a Receiver Operating Characteristic (ROC) curve for the proposed algorithm, plotting True Positive Rate (TPR) on the y-axis (0.0 to 1.0) against False Positive Rate (FPR) on the x-axis (0.0 to 1.0). The curve is a solid blue line, showing a steep increase in TPR as FPR increases, indicating good performance. Subplot (b) is an Average ROC curve for the proposed algorithm, plotting Average True Positive Rate (ATPR) on the y-axis (0.0 to 1.0) against Average False Positive Rate (AFPR) on the x-axis (0.0 to 1.0). The curve is a solid blue line, showing a steep increase in ATPR as AFPR increases, indicating good performance. Subplot (c) is an Average ROC curve for the proposed algorithm, plotting Average True Positive Rate (ATPR) on the y-axis (0.0 to 1.0) against Average False Positive Rate (AFPR) on the x-axis (0.0 to 1.0). The curve is a solid blue line, showing a steep increase in ATPR as AFPR increases, indicating good performance. Subplot (d) is an Average ROC curve for the proposed algorithm, plotting Average True Positive Rate (ATPR) on the y-axis (0.0 to 1.0) against Average False Positive Rate (AFPR) on the x-axis (0.0 to 1.0). The curve is a solid blue line, showing a steep increase in ATPR as AFPR increases, indicating good performance.

[illegible]

Figure 1 consists of three subplots, (a), (b), and (c), each showing the CAGR (Y-axis) versus the input CAGR's means Probability (X-axis). The X-axis for all plots ranges from 0.0 to 1.0. The Y-axis for (a) ranges from 0 to 400, for (b) from 0 to 1400, and for (c) from 0 to 1400. Each plot compares the proposed model (blue line) with other models (colored lines) across different input CAGR means and probabilities. The proposed model generally shows a more stable and accurate prediction compared to the other models, especially in the wet weather scenario.

(a) Dry Weather (CAGR vs. Prob)

Legend:

- Prop. (Blue line)
- 2014-16 (Green line)
- Mean of 2014-16, $\mu=0$ (Red line)
- Mean of 2014-16, $\mu=0.000001$ (Orange line)
- Mean of 2014-16, $\mu=0.000001$ (Yellow line)
- Low Mean, $\mu=0$, $\mu=0$ (Purple line)
- Low Mean, $\mu=0$, $\mu=0$ (Pink line)
- Low Mean, $\mu=0$, $\mu=0$ (Light Blue line)

(b) Wet Weather (CAGR vs. Prob)

Legend:

- 2014-16, $\mu=0$ (Green line)
- Low Mean, $\mu=0$, $\mu=0$ (Purple line)
- Low Mean, $\mu=0$, $\mu=0$ (Pink line)
- Low Mean, $\mu=0$, $\mu=0$ (Light Blue line)
- Prop. (Blue line)

(c) CAGR (CAGR vs. Prob)

Legend:

- Prop. (Blue line)
- 2014-16, $\mu=0$ (Green line)
- Low Mean, $\mu=0$, $\mu=0$ (Purple line)
- Low Mean, $\mu=0$, $\mu=0$ (Pink line)
- Low Mean, $\mu=0$, $\mu=0$ (Light Blue line)
- Prop. (Blue line)

[illegible]

Figure 1 consists of two line plots comparing the proposed method (red line) with several existing methods (blue, green, orange, purple, brown lines) across different input Cor's mean's probability (x) values (0.0, 0.1, 0.2, 0.3, 0.4, 0.5).

Left Plot: Oxidized Weight (mg) vs Input Cor's mean's probability (x)

The y-axis represents Oxidized Weight (mg) from 0 to 700. The x-axis represents Input Cor's mean's probability (x) from 0.0 to 0.5. The legend includes:

- Prop. (red line)
- 2018 Liu, et al. (blue line)
- 2018 Liu, et al. (green line)
- 2018 Liu, et al. (orange line)
- 2018 Liu, et al. (purple line)
- 2018 Liu, et al. (brown line)
- 2018 Liu, et al. (black line)
- 2018 Liu, et al. (grey line)
- 2018 Liu, et al. (dark blue line)
- 2018 Liu, et al. (dark green line)
- 2018 Liu, et al. (dark orange line)
- 2018 Liu, et al. (dark purple line)
- 2018 Liu, et al. (dark brown line)
- 2018 Liu, et al. (dark grey line)
- 2018 Liu, et al. (dark dark blue line)
- 2018 Liu, et al. (dark dark green line)
- 2018 Liu, et al. (dark dark orange line)
- 2018 Liu, et al. (dark dark purple line)
- 2018 Liu, et al. (dark dark brown line)
- 2018 Liu, et al. (dark dark grey line)

Right Plot: Cor's Mean vs Input Cor's mean's probability (x)

The y-axis represents Cor's Mean from 0 to 200. The x-axis represents Input Cor's mean's probability (x) from 0.0 to 0.5. The legend includes:

- Prop. (red line)
- 2018 Liu, et al. (blue line)
- 2018 Liu, et al. (green line)
- 2018 Liu, et al. (orange line)
- 2018 Liu, et al. (purple line)
- 2018 Liu, et al. (brown line)
- 2018 Liu, et al. (black line)
- 2018 Liu, et al. (grey line)
- 2018 Liu, et al. (dark blue line)
- 2018 Liu, et al. (dark green line)
- 2018 Liu, et al. (dark orange line)
- 2018 Liu, et al. (dark purple line)
- 2018 Liu, et al. (dark brown line)
- 2018 Liu, et al. (dark grey line)
- 2018 Liu, et al. (dark dark blue line)
- 2018 Liu, et al. (dark dark green line)
- 2018 Liu, et al. (dark dark orange line)
- 2018 Liu, et al. (dark dark purple line)
- 2018 Liu, et al. (dark dark brown line)
- 2018 Liu, et al. (dark dark grey line)

[illegible]

Figure 10 displays the performance of the proposed model for the 2014 and 2015 datasets, comparing the proposed model (Proposed) against the baseline (Baseline) across various metrics.

The figure is organized into three rows, each corresponding to a different dataset: 2014, 2015, and 2016. Each row contains three subplots:

- Top Row (2014):**
 - Spilloverd Means (Histograms: Correlation vs. Argument):** A scatter plot showing the relationship between Input Cor'n Weak Probability (a) and Spilloverd Means. The y-axis ranges from 0.0 to 1.0, and the x-axis ranges from 0.0 to 0.5. The plot shows a strong positive correlation.
 - Avg. Bits Consumed:** A line plot showing the average bits consumed (y-axis, 0 to 1000) versus Probability (x-axis, 0.0 to 0.5). The plot shows a significant increase in bits consumed as probability increases.
 - Cor'n Bits for 2014, 100000:** A line plot showing the correlation bits (y-axis, 0 to 1000) versus Input Cor'n Weak Probability (x-axis, 0.0 to 0.5). The plot shows a sharp increase in correlation bits as probability increases.
- Middle Row (2015):**
 - Spilloverd Means (Histograms: Correlation vs. Argument):** A scatter plot showing the relationship between Input Cor'n Weak Probability (a) and Spilloverd Means. The y-axis ranges from 0.0 to 1.0, and the x-axis ranges from 0.0 to 0.5. The plot shows a strong positive correlation.
 - Avg. Bits Consumed:** A line plot showing the average bits consumed (y-axis, 0 to 1000) versus Probability (x-axis, 0.0 to 0.5). The plot shows a significant increase in bits consumed as probability increases.
 - Cor'n Bits for 2015, 100000:** A line plot showing the correlation bits (y-axis, 0 to 1000) versus Input Cor'n Weak Probability (x-axis, 0.0 to 0.5). The plot shows a sharp increase in correlation bits as probability increases.
- Bottom Row (2016):**
 - Spilloverd Means (Histograms: Correlation vs. Argument):** A scatter plot showing the relationship between Input Cor'n Weak Probability (a) and Spilloverd Means. The y-axis ranges from 0.0 to 1.0, and the x-axis ranges from 0.0 to 0.5. The plot shows a strong positive correlation.
 - Avg. Bits Consumed:** A line plot showing the average bits consumed (y-axis, 0 to 1000) versus Probability (x-axis, 0.0 to 0.5). The plot shows a significant increase in bits consumed as probability increases.
 - Cor'n Bits for 2016, 100000:** A line plot showing the correlation bits (y-axis, 0 to 1000) versus Input Cor'n Weak Probability (x-axis, 0.0 to 0.5). The plot shows a sharp increase in correlation bits as probability increases.

Figure 10 displays four plots related to the performance of the proposed model for the Hepatitis B dataset. The top-left plot shows the Actual Values (Y-axis) versus Predicted Values (X-axis) for the input class weights probability (0.0 to 1.0). The top-right plot shows the Average ROC Curve for the input class weights probability (0.0 to 1.0) for the years 2014 to 2020. The bottom-left plot shows the Average ROC Curve for the input class weights probability (0.0 to 1.0) for the years 2014 to 2020. The bottom-right plot shows the Cost vs. Input Class Weights Probability (0.0 to 1.0) for the years 2014 to 2020.

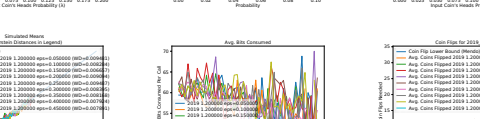


Figure 1 displays a 3x3 grid of plots showing the relationship between input and output variables for different models and datasets. The rows represent different models: 'Simplified Model', 'Aug. 8th Consumed', and 'Call Risk for 2015, 1,00000'. The columns represent different datasets: 'Simplified Dataset in Legend', 'Aug. 8th Consumed', and 'Call Risk for 2015, 1,00000'. Each plot shows a scatter of data points with a fitted curve. The x-axis for all plots is 'Input Call's Netw. Probability (%)' ranging from 0.00 to 0.20. The y-axis for the first column is 'Output Call's Netw. Probability (%)' ranging from 0.0 to 1.0. The y-axis for the second column is 'Avg. 8th Consumed' ranging from 60 to 90. The y-axis for the third column is 'Call Risk Number' ranging from 0 to 10000. The plots show that the output probability is generally higher than the input probability, and the average 8th consumed and call risk number are also higher than the input probability.

Figure 1 displays six plots related to the 2019 COVID-19 dataset, organized into three rows and two columns. The top row shows 'GenStatistical Measures' (Histograms, Scatter Plot, and Correlation Matrix). The middle row shows 'Avg. All Compartment' (Histograms, Scatter Plot, and Correlation Matrix). The bottom row shows 'Avg. All Compartment' (Histograms, Scatter Plot, and Correlation Matrix). Each plot includes a legend for the 2019 COVID-19 dataset.

Top Row: GenStatistical Measures

- Histograms (Difference in Legend):** Shows the distribution of 'Input Case Incidence Probability (%)' for the 2019 COVID-19 dataset. The x-axis ranges from 0.00 to 0.35, and the y-axis ranges from 0.0 to 1.0.
- Scatter Plot:** Shows the relationship between 'Input Case Incidence Probability (%)' (x-axis, 0.00 to 0.35) and 'Output Case Incidence Probability (%)' (y-axis, 0.0 to 1.0). The data points are clustered along the diagonal line.
- Correlation Matrix:** Shows the correlation between 'Input Case Incidence Probability (%)' and 'Output Case Incidence Probability (%)'. The correlation is 1.000000.

Middle Row: Avg. All Compartment

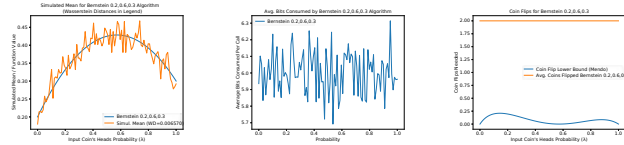
- Histograms (Difference in Legend):** Shows the distribution of 'Input Case Incidence Probability (%)' for the 2019 COVID-19 dataset. The x-axis ranges from 0.00 to 0.35, and the y-axis ranges from 0.0 to 1.0.
- Scatter Plot:** Shows the relationship between 'Input Case Incidence Probability (%)' (x-axis, 0.00 to 0.35) and 'Output Case Incidence Probability (%)' (y-axis, 0.0 to 1.0). The data points are clustered along the diagonal line.
- Correlation Matrix:** Shows the correlation between 'Input Case Incidence Probability (%)' and 'Output Case Incidence Probability (%)'. The correlation is 1.000000.

Bottom Row: Avg. All Compartment

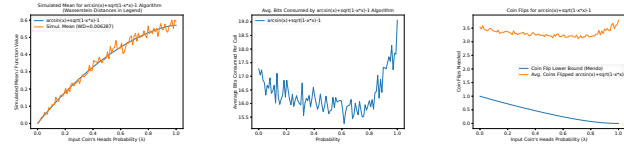
- Histograms (Difference in Legend):** Shows the distribution of 'Input Case Incidence Probability (%)' for the 2019 COVID-19 dataset. The x-axis ranges from 0.00 to 0.35, and the y-axis ranges from 0.0 to 1.0.
- Scatter Plot:** Shows the relationship between 'Input Case Incidence Probability (%)' (x-axis, 0.00 to 0.35) and 'Output Case Incidence Probability (%)' (y-axis, 0.0 to 1.0). The data points are clustered along the diagonal line.
- Correlation Matrix:** Shows the correlation between 'Input Case Incidence Probability (%)' and 'Output Case Incidence Probability (%)'. The correlation is 1.000000.

Figure 1 consists of three subplots. The top-left plot shows the output GCM's network probability (y-axis, 0.0 to 0.3) versus the input GCM's network probability (x-axis, 0.00 to 0.90). It contains several colored lines (green, blue, red, orange, purple) that all start at (0,0) and increase monotonically, with some lines being steeper than others. The top-right plot shows the average path length (y-axis, 100 to 150) versus probability (x-axis, 0.00 to 0.90). It contains several colored lines that generally decrease as probability increases, with some lines showing a more pronounced dip around 0.5 probability. The bottom plot shows the average clustering coefficient (y-axis, 0 to 20) versus the input GCM's network probability (x-axis, 0.00 to 0.90). It contains several colored lines that generally increase as probability increases, with some lines showing a more pronounced increase around 0.5 probability.

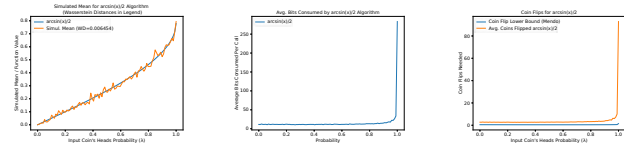
Bernstein
0.2,0.6,0.3



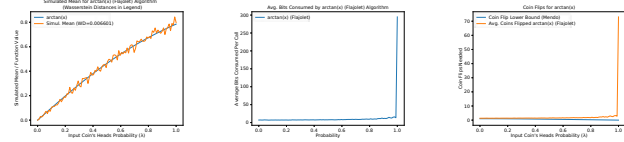
$\arcsin(x) + \sqrt{1-x^2} - 1$



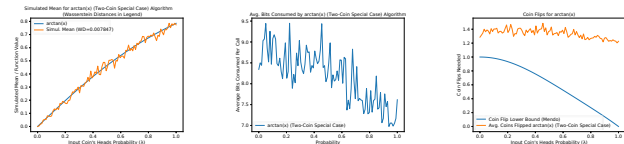
$\arcsin(x)/2$



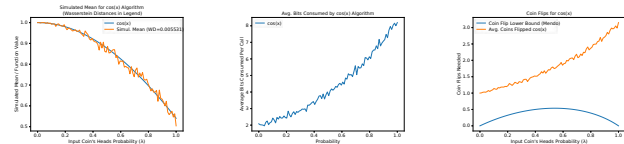
$\arctan(x)$
(Flajolet)



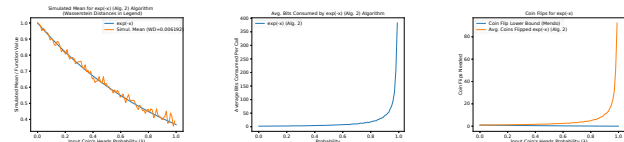
$\arctan(x)$ (Two-Coin Special Case)



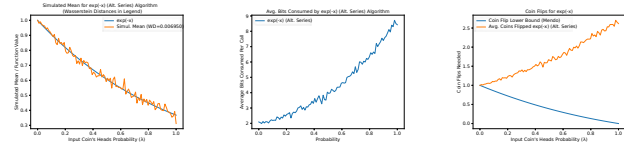
$\cos(x)$



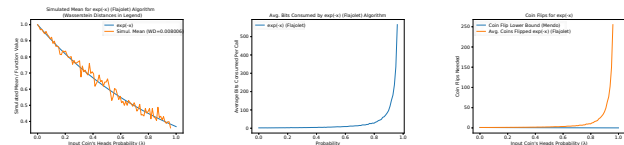
$\exp(-x)$ (Alg. 2)



$\exp(-x)$ (Alt. Series)



$\exp(-x)$ (Flajolet)



$\exp(x) * (1-x)$

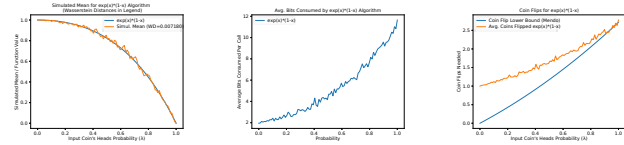


Figure 1 consists of three subplots labeled (a), (b), and (c), comparing the proposed algorithm with Reg-1.

- (a) Simulated Mean for Reg-1 vs Proposed Algorithm:** The y-axis is 'Simulated Mean for Reg-1 vs Proposed Algorithm' (0.0 to 1.0) and the x-axis is 'Iteration' (0.0 to 1.0). The 'Proposed Algorithm' (blue line) shows a smooth, nearly linear increase from 0.0 to 1.0. The 'Reg-1' (orange line) shows a similar trend but with significant fluctuations, especially in the middle range of iterations.
- (b) Avg. RMS Convergence Cost:** The y-axis is 'Avg. RMS Convergence Cost' (0 to 10) and the x-axis is 'Iteration' (0.0 to 1.0). The 'Proposed Algorithm' (blue line) remains very low, near 0, until iteration 0.8, where it spikes sharply to approximately 10. The 'Reg-1' (orange line) remains consistently low, near 0, throughout all iterations.
- (c) Cost Function:** The y-axis is 'Cost Function' (0 to 10) and the x-axis is 'Iteration' (0.0 to 1.0). The 'Proposed Algorithm' (blue line) remains very low, near 0, until iteration 0.8, where it spikes sharply to approximately 10. The 'Reg-1' (orange line) remains consistently low, near 0, throughout all iterations.

Figure 10 consists of three subplots showing the performance of the proposed algorithm on the 'coin flip' dataset. The x-axis for all plots is 'Number of Samples' ranging from 0.0 to 1.0.

- Left Plot:** The y-axis is 'Number of Iterations (Log Scale)' ranging from 0 to 10. The legend indicates 'Coin Flip' (orange line with dots) and 'Coin Flip - Lower Bound (Proposed)' (blue line). The orange line starts at approximately 2.5 and increases to about 9.5, while the blue line starts at approximately 1.5 and increases to about 8.5.
- Middle Plot:** The y-axis is 'Log of Number of Iterations' ranging from 0.0 to 1.0. The legend indicates 'Coin Flip' (orange line with dots) and 'Coin Flip - Lower Bound (Proposed)' (blue line). The orange line starts at approximately 0.5 and increases to about 0.9, while the blue line starts at approximately 0.4 and increases to about 0.8.
- Right Plot:** The y-axis is 'Coeff. Number' ranging from 0.0 to 1.0. The legend indicates 'Coin Flip' (orange line with dots) and 'Coin Flip - Lower Bound (Proposed)' (blue line). The orange line starts at approximately 0.9 and decreases to about 0.7, while the blue line starts at approximately 0.8 and decreases to about 0.6.

Figure 1 consists of three subplots labeled (a), (b), and (c), each showing performance metrics for different algorithms across input correlations from 0.0 to 1.0.

- Subplot (a): Standard Error (SE) vs. Input Correlation**
 The y-axis is 'Standard Error (SE) x 10^4' ranging from 0 to 1.5. The x-axis is 'Input Correlation (Input Correlation)' ranging from 0.0 to 1.0. The legend includes:
 - Estimated Mean for points 1/3/5 algorithm (black line with dots)
 - Importance Sampling (red line with dots)
 - Black Monte Carlo (blue line with dots)
 - point 1/3/5 (orange line with dots)
 The 'point 1/3/5' algorithm shows the highest SE, increasing from ~0.5 to ~1.4. The other three algorithms show much lower SE, starting near 0.5 and decreasing to near 0 as correlation increases.
- Subplot (b): Avg. Bias Consumed vs. Input Correlation**
 The y-axis is 'Average Bias Consumed' ranging from 0 to 1600. The x-axis is 'Input Correlation' ranging from 0.0 to 1.0. The legend includes:
 - point 1/3/5 (orange line with dots)
 The 'point 1/3/5' algorithm shows a sharp drop in bias from ~1500 at correlation 0.0 to near 0 at correlation 0.2, remaining at 0 for higher correlations.
- Subplot (c): CPU Time vs. Input Correlation**
 The y-axis is 'CPU Time (seconds)' ranging from 0 to 400. The x-axis is 'Input Correlation (Input Correlation)' ranging from 0.0 to 1.0. The legend includes:
 - Coin Flip for points 1/3/5 (blue line with dots)
 - Coin Flip for case-based (black line with dots)
 - Coin Flip for input points 1/3/5 (red line with dots)
 All three algorithms show a sharp drop in CPU time from ~350 seconds at correlation 0.0 to near 0 seconds at correlation 0.2, remaining at 0 for higher correlations.

Figure 10 consists of three subplots showing the performance of the proposed algorithm compared to the baseline algorithm. The x-axis for all plots is 'input Cor'n's sparsity (0.0 to 1.0)'. The y-axis for all plots is 'Reconstruction Error (RMSE)'.

- Left Plot:** Titled 'Reconstructed Mean for sparse L10 algorithm'. The y-axis ranges from 0.0 to 1.0. The legend indicates 'proposed' (blue line) and 'baseline' (orange line). Both lines show a decreasing trend, with the proposed algorithm consistently performing better (lower RMSE) than the baseline.
- Middle Plot:** Titled 'Avg. RMSE Consumed by sparse L10 algorithm'. The y-axis ranges from 0 to 300. The legend indicates 'proposed' (blue line) and 'baseline' (orange line). The proposed algorithm's RMSE is significantly lower (around 10) compared to the baseline's RMSE (around 250).
- Right Plot:** Titled 'Cost Edge for sparse L10'. The y-axis ranges from 0 to 10. The legend indicates 'proposed' (blue line) and 'baseline' (orange line). The proposed algorithm's cost is significantly lower (around 1) compared to the baseline's cost (around 10).

Figure 1 consists of three subplots labeled (a), (b), and (c).

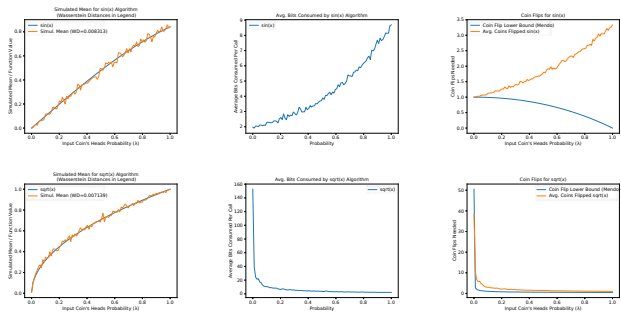
- (a) ROC curve for average AUC against input Cori's records probability:** The x-axis is 'input Cori's records probability (0.0-1.0)' and the y-axis is 'Data point's True rate (0.0-1.0)'. A red line represents the 'Actual, Mean (0.9141-0.998863)' and a blue line represents the 'Ideal Mean for average AUC against input Cori's records probability'. The red line is above the blue line, indicating better performance.
- (b) Mean time consumed by parallel algorithm against probability:** The x-axis is 'Probability' and the y-axis is 'Mean Time Consumed (Sec) (0.0-45)'. A blue line represents 'Parallel, 100' and a red line represents 'Serial, 100'. The blue line is significantly lower than the red line, indicating faster execution time for the parallel algorithm.
- (c) Cori flag for parallel algorithm against probability:** The x-axis is 'input Cori's records probability (0.0-1.0)' and the y-axis is 'Cori flag (0.0-1.0)'. A red line represents 'Cori flag for parallel, 100' and a blue line represents 'Avg. Cori's flag parallel, 100'. The red line is above the blue line, indicating a higher Cori flag for the parallel algorithm.

[illegible]

Figure 1 consists of four plots arranged in a 2x2 grid, showing the performance of different methods. The top-left plot shows the Discretized Area Under the Curve (AUC) for the 'area' variable against the Input Data's Feature Probability (x). The top-right plot shows the Average Area Under the Curve (AUC) for the 'area' variable against the Probability. The bottom-left plot shows the Average Area Under the Curve (AUC) for the 'area' variable against the Probability. The bottom-right plot shows the CDF of the Error for the 'area' variable against the Input Data's Feature Probability (x).

Figure 10 consists of three subplots. The left plot shows the Standard Error (Y-axis, 0.0 to 0.4) versus Frequency (Hz) (X-axis, 0.0 to 1.0). It compares the proposed model at 5.0 Hz (blue line with circles) and 5.5 Hz (orange line with circles). Both models show a linear increase in error with frequency, with the 5.5 Hz model having a slightly higher error. The middle plot shows the Magnitude of Transfer Function $G(s)$ (Y-axis, 0.0 to 1.0) versus Frequency (Hz) (X-axis, 0.0 to 1.0). The proposed model at 5.0 Hz (blue line) shows a sharp drop in magnitude from 1.0 to near 0.0 at a frequency of approximately 0.1 Hz. The right plot shows the Cost Function (Y-axis, 0.0 to 1200) versus Frequency (Hz) (X-axis, 0.0 to 1.0). It compares the proposed model at 5.0 Hz (blue line) with a lower bound (orange line) and a reference model (green line). The proposed model's cost function drops sharply from approximately 1100 to near 0.0 at a frequency of approximately 0.1 Hz, while the lower bound and reference model remain near 0.0.

$\sin(x)$



\sqrt{x}

1. <https://peteroupc.github.io/bernoulli.md>