Using Deep Neural Networks to Automate Large Scale Statistical Analysis for Big Data Applications

Rongrong Zhang, Wei Deng, Michael Zhu   Department of Statistics, Purdue University

Contribution

- We proposed and developed the neural model selector and parameter estimator to automate two major tasks in the Statistical Analysis(SA) process, which are model selection and parameter estimation.
- Simulation study shows that the neural selector and estimator can be properly trained with systematically simulated labeled data, and further demonstrate excellent prediction performance.
- The idea and proposed framework can be further extended to automate the entire SA process and have the potential to revolutionize how SA is performed in big data analytics.

Method

- Suppose $\mathcal{M} = \{M_k : 1 \leq k \leq K\}$ is a collection of K prespecified models/distributions. Let $f(y|\theta_k, M_k)$ be the density function of model M_k, where θ_k is the scalar parameter.
- A random sample of size N is from one of the models, but we do not know the data-generating model and its parameter. The goal of statistical analysis is to identify the model and further estimate the model parameter.
- The procedures for model selection and parameter estimation can be considered mappings from the sample to a model and a value of the model parameter

$$G : \{y_j\} \to \begin{pmatrix} G_1(\{y_j\}) \\ G_2(\{y_j\}) \end{pmatrix} \in \mathcal{M} \times \Theta$$

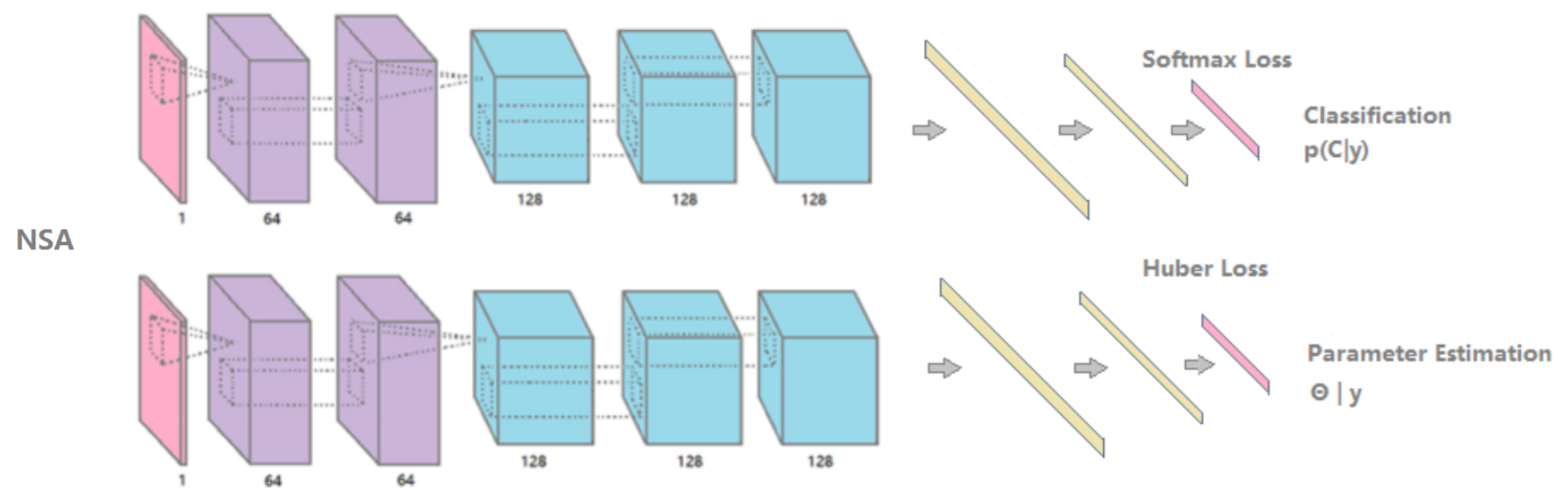
where G_1 is the model selection mapping and G_2 is the parameter estimation mapping.

- We propose to use CNNs to approximate G_1 and G_2.
- $K = 50$ probability distributions are taken from the textbook Casella and Berger (2002) and some R packages.
- Training data were systematically generated by placing an equally space grid over the prespecified parameter space, and then generating the multiple samples of size $N = 100, 400$ and 900.
- In total, we have generated roughly 400 thousand training samples, 100 thousand validation samples, and 50 thousand test samples for each sample size.
- The Huber loss is employed in training of neural estimator to improve the robustness against outliers generated from models with long tails.

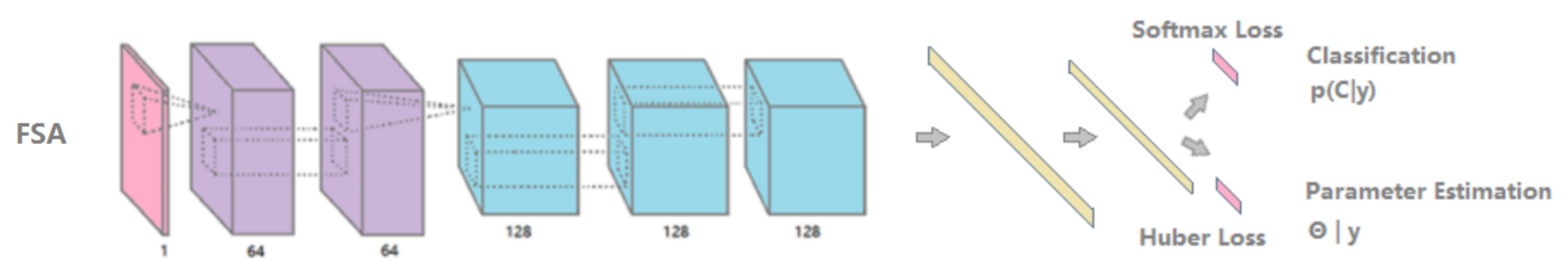
SA Architectures

Interplay architectures of the neural model selector and parameter estimator

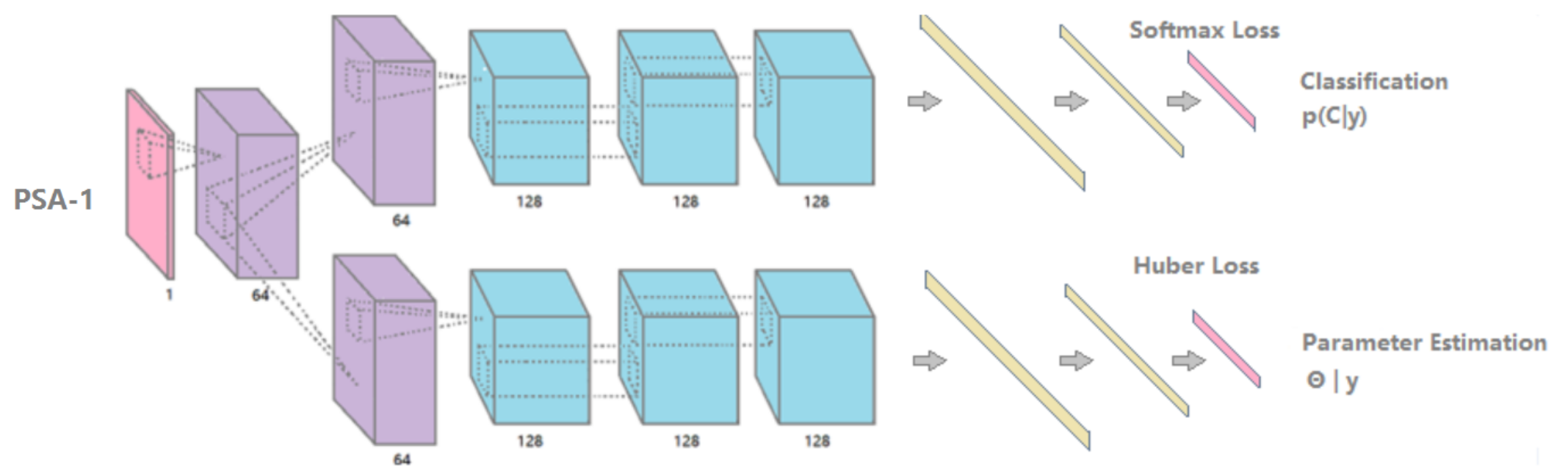
The first SA architecture uses two separate CNNs for the model selector and the parameter estimator, respectively, which we refer to as the Non-Shared Architecture (NSA).



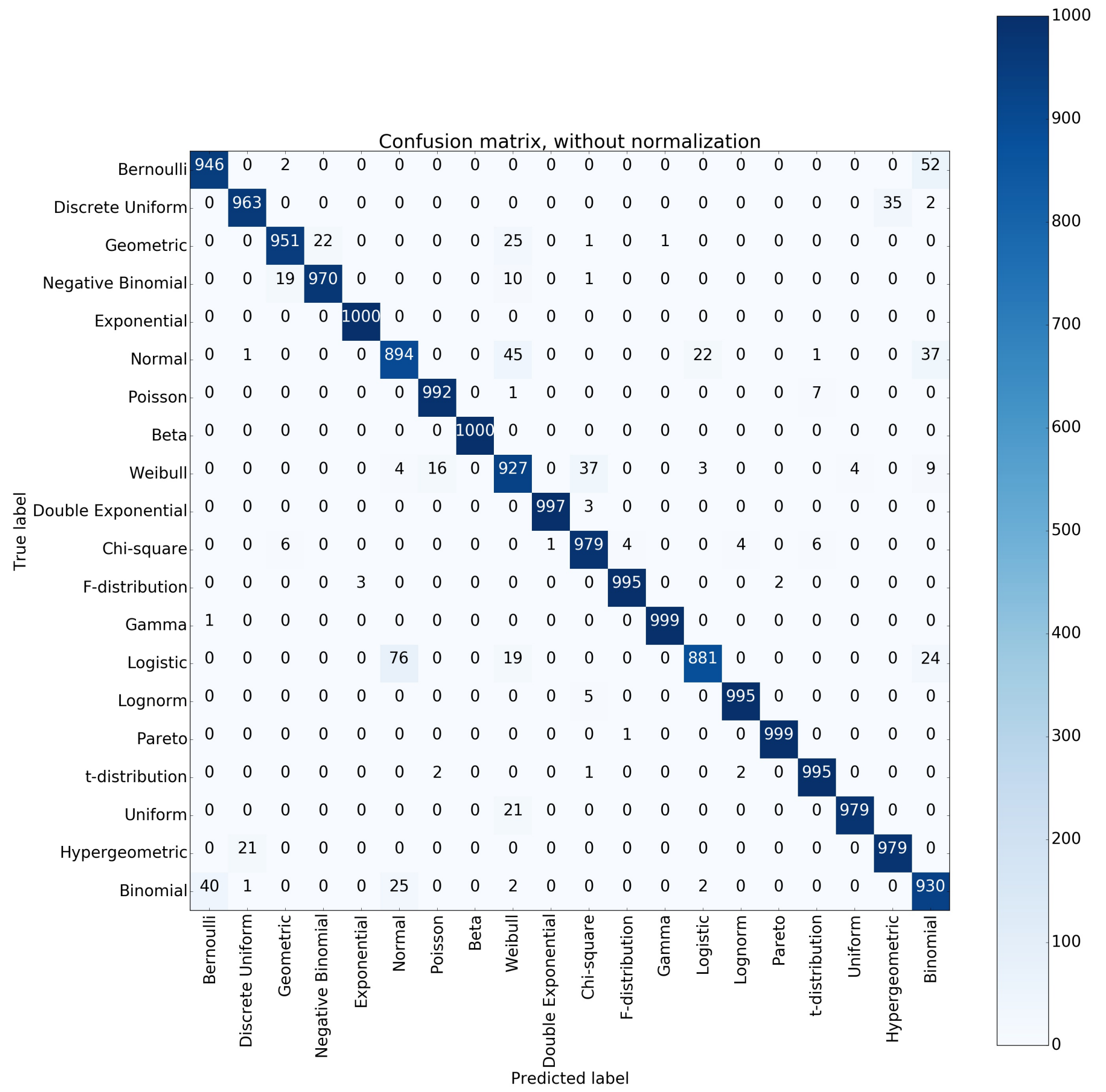
The second SA architecture uses one single CNN for both G_1 and G_2, and they part their ways only at the output layer. We refer to this architecture as the Fully Shared Architecture (FSA).



The third architecture uses two partially joint CNNs for G1 and G2, respectively. The two CNNs can share from one to all common convolutional and fully connected layers. We refer to this architecture as the Partially Shared Architecture (PSA).



The confusion matrix based on large CNN and PSA-5 neural model selector on test dataset with K = 20 distributions.

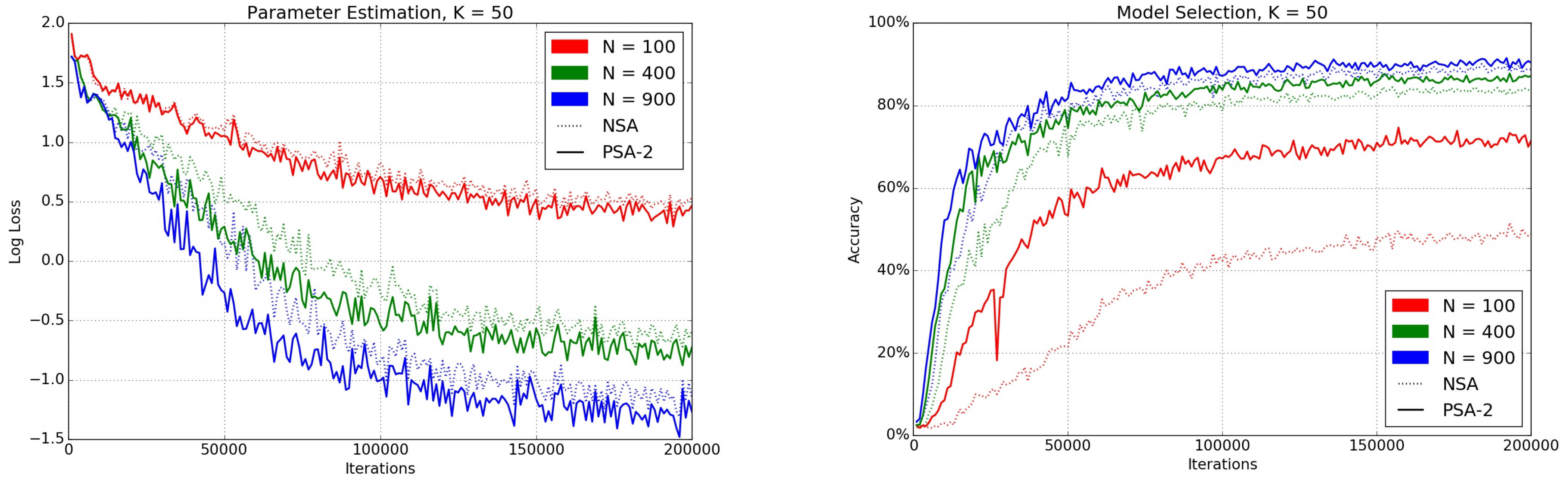


	Bernoulli	Discrete Uniform	Geometric	Negative Binomial	Exponential	Normal	Poisson	Beta	Weibull	Double Exponential	Chi-square	F-distribution	Gamma	Logistic	Lognorm	Pareto	t-distribution	Uniform	Hypergeometric	Binomial
Bernoulli	946	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	52
Discrete Uniform	0	963	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	2
Geometric	0	0	951	22	0	0	0	0	25	0	1	0	1	0	0	0	0	0	0	0
Negative Binomial	0	0	19	970	0	0	0	0	10	0	1	0	0	0	0	0	0	0	0	0
Exponential	0	0	0	0	1000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Normal	0	1	0	0	0	894	0	0	45	0	0	0	0	22	0	0	1	0	0	37
Poisson	0	0	0	0	0	0	992	0	1	0	0	0	0	0	0	0	7	0	0	0
Beta	0	0	0	0	0	0	0	1000	0	0	0	0	0	0	0	0	0	0	0	0
Weibull	0	0	0	0	0	4	16	0	927	0	37	0	0	3	0	0	0	4	0	9
Double Exponential	0	0	0	0	0	0	0	0	0	997	3	0	0	0	0	0	0	0	0	0
Chi-square	0	0	6	0	0	0	0	0	0	1	979	4	0	0	4	0	6	0	0	0
F-distribution	0	0	0	0	3	0	0	0	0	0	0	995	0	0	0	2	0	0	0	0
Gamma	1	0	0	0	0	0	0	0	0	0	0	0	999	0	0	0	0	0	0	0
Logistic	0	0	0	0	0	76	0	0	19	0	0	0	0	881	0	0	0	0	0	24
Lognorm	0	0	0	0	0	0	0	0	0	0	5	0	0	0	995	0	0	0	0	0
Pareto	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	999	0	0	0	0
t-distribution	0	0	0	0	0	0	2	0	0	0	1	0	0	0	2	0	995	0	0	0
Uniform	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	979	0	0
Hypergeometric	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	979	0
Binomial	40	1	0	0	0	25	0	0	2	0	0	0	0	2	0	0	0	0	0	930

Results

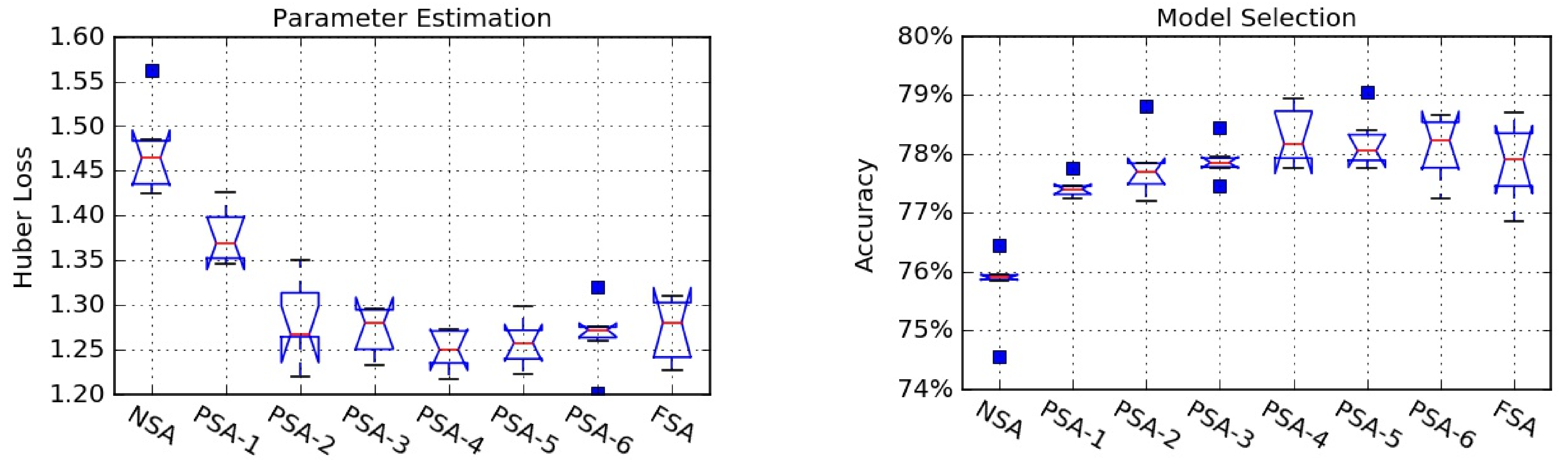
Impact of SA architectures on learning rate

Medium PSA-2 neural parameter estimator and model selector outperformed medium NSA architecture in terms of the learning rate and prediction accuracy.



How many layers should be shared?

We consider the scenario with 50 models, 100 sample size, and the large CNN architecture, and vary the SA architectures from NSA to FSA. The left panel presents the boxplots of the Huber loss of the parameter estimator, whereas the right panel presents the boxplots of accuracy of the model selector under various SA architectures.



We apply the three conventional model selection methods, the KS distance, BIC, and Bayes factor to the test datasets under the scenario with 20 models, and compare their performances with that of the trained neural model selector. Table below reports the accuracy of the three statistical methods as well as the trained neural model selector under various sample sizes.

	$N = 100$		$N = 400$		$N = 900$	
	Top-1	Top-2	Top-1	Top-2	Top-1	Top-2
KS distance	72.5%	83.2%	83.3%	85.0%	84.7%	85.0%
BIC	69.9%	74.6%	74.7%	75.0%	75.0%	75.0%
Bayes factor	75.5%	84.8%	77.8%	83.3%	70.0%	75.0%
Neural selector	92.1%	99.2%	96.4%	99.7%	97.9%	99.7%

Future Work

- Extend the neural model selector and parameter estimator to models with multiple parameters as well as regression models involving a large number of explanatory variables;
- Investigate how CNNs or other DNNs can be used to automate other tasks such as hypotheses testing and diagnostics of the SA process.
- **Contact Information:** zhan1602@purdue.edu