Randomized Greedy Search for Structured Prediction: Amortized Inference and Learning

Chao Ma[1] , **F A Rezaur Rahman Chowdhury**[2] , **Aryan Deshwal**[2] ,
Md Rakibul Islam[2] , **Janardhan Rao Doppa**[2] and **Dan Roth**[3]
[1]School of EECS, Oregon State University
[2]School of EECS, Washington State University
[3]Department of Computer and Information Science, University of Pennsylvania
machao@oregonstate.edu, {f.chowdhury, aryan.deshwal, mdrakibul.islam, jana.doppa}@wsu.edu,
danroth@seas.upenn.edu

A Appendix

A.1 Results of RGS Inference with $\alpha = 0$

In this section, we present the results of structured learning with RGS inference, where the starting outputs are completely random ($\alpha = 0$). We report two different accuracy metrics over testing examples: real accuracy of predicted outputs and generation accuracy (accuracy of the best output uncovered by the search process). Note that generation accuracy is greater than or equal to real accuracy. To account for randomness in RGS inference, we repeat each experiment 10 times, and report the mean and variance for each of the two accuracies.

Varying Number of Restarts R_{max}. Table 1 shows the Hamming accuracy with RGS(0) for different number of restarts $R_{max} \in \{1, 10, 20, 50, 100\}$. We make two observations. First, both prediction accuracy and generation accuracy improves with more number of restarts, and saturates at $R_{max} = 100$. With increased number of restarts, search process overcomes the local optima challenge and uncovers higher quality candidate outputs resulting in improved generation accuracy and prediction accuracy. Second, variance over accuracies decrease with more restarts improving the stability of results.

Higher-order Features. RGS allows to use higher-order features with negligible computational overhead. Therefore, we experiment by enriching the first-order representation. For sequence labeling, we add second-order (triplets) and third-order (quadruples) features. For coreference resolution, we add features over entity and entity-mention pair. For image segmentation, we add global features (e.g., normalized histogram for number of super-pixels with different labels). Table 2 shows the Hamming accuracy results with higher-order features and 50 restarts. Both prediction accuracy and generation accuracy improves with higher-order features.

Optimizing Non-decomposable Loss Functions. RGS allows learning to optimize arbitrary loss functions. Therefore, we train and test with different loss functions. For these experiments, we employ 50 restarts and highest order features for all datasets. Table 3 shows the results. As expected, we get the best accuracy when training and testing is done with the same loss function (i.e., diagonal entries).

Dataset	Features	Pred. Acc.	Gen. Acc.
a. Sequence Labeling			
HW-Small	First order	72.86±2.18E-6	94.76±1.64E-5
	Second order	86.86±7.55E-6	97.86±7.65E-5
	Third order	92.32±2.75E-5	98.11±1.1E-5
HW-Large	First order	85.59±6.74E-5	98.47±5.61E-5
	Second order	93.50±3E-4	98.90±6.48E-5
	Third order	97.83±2.16E-4	99.53±5.45E-5
Phoneme	First order	80.72±2.34E-6	97.75±8.32E-5
	Second order	81.72±7.15E-6	98.47±1.39E-5
	Third order	82.28±3.73E-6	98.41±6.8E-5
Stress	First order	76.18±7.89E-7	99.28±3.72E-5
	Second order	79.03±2.48E-6	99.50±8.75E-5
	Third order	80.84±1.91E-7	99.59±3.31E-5
b. Coreference Resolution			
ACE2005	First order	76.04±0E0	-
	First order + entity feat.	78.09±6.53E-3	80.19±5.23E-3
c. Image Segmentation			
MSRC	First order	80.59±5.98E-4	87.94±4.11E-4
	First order + global feat.	81.27±1.37E-3	87.52±1.16E-3

Table 2: Results of RGS(0) inference with higher-order features.

A.2 Experimental Setups for BiLSTM and Seq2Seq Baselines.

For sequence labeling problems, We include BiLSTM, BiLSTM-CRF, and Seq2Seq with beam search as baselines. Due to the space limit in the main paper, we present a more detailed setups here.

Experimental Setup for BiLSTM. We use SGD with momentum 0.9 and learning rate 0.1. L2 regularizer was employed. Hidden state units in LSTM: same as the input size. In BiLSTM, the final unary features are the concatenation of LSTM hidden states in two directions. We set a batch size of 20 for all datasets.

Experimental Setup for Seq2Seq. We employed the implementation from `https://github.com/JayParks/tf-seq2seq`. We used LSTM cell with attention type set to `bahdanau`. We set the hidden units to be the same dimension as the input token features. We set batch size to 20. Training was done with learning rate 0.01 and Adam optimizer.

Restart	1		10		20		50		100	
	Real Acc.	Gen. Acc.	Real Acc.	Gen. Acc.	Real Acc.	Gen. Acc.	Real Acc.	Gen. Acc.	Real Acc.	Gen. Acc.
a. Sequence Labeling										
HwSml	66.56±2.39E-4	78.61±2.06E-4	72.55±2.39E-5	91.25±3.63E-5	72.63±2.62E-5	92.96±4.54E-5	72.86±2.18E-6	94.76±1.64E-5	72.84±4.9E-7	95.75±1.83E-5
HwLrg	75.39±1.99E-3	85.04±1.96E-3	85.23±4.16E-4	96.74±3.46E-4	85.50±7.38E-4	97.23±1.44E-4	85.59±6.74E-5	98.47±5.61E-5	85.75±3.84E-5	98.89±2.84E-5
Phonm	80.15±6.49E-5	92.00±1.05E-4	80.60±5.18E-6	95.40±9.2E-5	80.68±3.06E-5	96.49±8.62E-5	80.72±2.34E-6	97.75±8.32E-5	80.42±1.37E-30	98.25±5.26E-5
Strs	73.97±4.67E-4	88.96±1.12E-3	76.39±3.16E-6	97.33±1.72E-4	76.17±8.86E-5	98.11±6.83E-5	76.18±7.89E-7	99.28±3.72E-5	76.33±1.37E-30	99.62±8.81E-6
b. Multi-label Classification										
Yeast	79.70±1.35E-4	89.38±2.65E-4	80.02±4.31E-7	96.62±1.53E-4	80.02±1.09E-4	97.91±6.63E-4	80.04±1.37E-30	99.04±7.22E-5	80.06±1.37E-30	99.53±4.78E-6
Bibtex	98.62±3.65E-8	99.33±3.06E-7	98.62±0E0	99.59±2.19E-7	98.62±0E0	99.64±7.15E-8	98.62±0E0	99.69±5.74E-8	98.62±0E0	99.72±1E-7
Bkmks	99.09±1.59E-6	91.06±2.44E-6	99.11±1.4E-7	97.21±1.15E-7	99.13±1.49E-7	99.24±1.42E-7	99.13±1.3E-7	99.54±1.53E-7	99.13±5.45E-8	99.74±3.34E-8
c. Coreference Resolution										
ACE05	69.12±5.41E-2	78.12±4.32E-2	71.45±1.01E-2	85.12±1.63E-2	75.15±1.09E-3	88.41±2.64E-3	76.92±2.99E-3	92.99±8.74E-4	76.99±4.52E-4	94.82±2.03E-4
d. Image Segmentation										
MSRC	64.19±5.79E-3	65.57±5.06E-3	76.18±2.28E-3	81.98±1.07E-3	78.27±7.33E-4	85.56±5.76E-4	80.59±5.98E-4	87.94±4.11E-4	81.35±1.2E-4	89.34±3.67E-4

Table 1: Results of RGS(0) inference with different number of random restarts.

a. Multi-label Classification

Name	Test on / Train on	Hamming	Example-F1	Example-Acc.
Yeast	Hamming	80.04±1.37E-30	59.30±1.37E-30	48.81±0E0
	Example-F1	75.88±1.37E-30	62.88±0E0	51.97±1.37E-30
	Example-Acc	76.44±1.37E-30	62.52±0E0	52.18±0E0
Bibtex	Hamming	98.62±0E0	39.94±0E0	33.56±1.3E-30
	Example-F1	98.00±6.28E-9	44.86±2.96E-6	34.93±3.17E-6
	Example-Acc	98.25±5.99E-10	44.13±3.98E-6	36.78±2.75E-6
Bookmarks	Hamming	99.13±1.3E-7	18.66±1.22E-7	17.91±8.56E-8
	Example-F1	98.13±1.16E-4	36.88±3.42E-4	26.85±1.61E-4
	Example-Acc	98.20±1.47E-4	31.29±3.05E-4	31.46±2.06E-4

b. Coreference Resolution

Name	Test on / Train on	Hamming	MUC	BCube
ACE2005	Hamming	78.09±6.53E-3	80.19±5.23E-3	76.52±1.63E-3
	MUC	75.65±5.31E-3	82.35±7.46E-3	72.58±3.97E-3
	BCUB	77.87±4.34E-3	77.79±5.24E-3	78.85±9.82E-3

Table 3: Results of RGS(0) with different train/test loss functions.

A.3 Results of Amortized RGS(α) for Test-Time Inference

No. of Testing Examples vs. Speedup Factor. We want to measure the speed of amortized RGS in producing outputs with same accuracy as RGS. Given a set of inputs D', let T_{RGS} and A_{RGS} be cumulative time and accuracy of RGS. If T_{ARGS} is the time taken by amortized RGS to reach the accuracy A_{RGS}, then speedup factor $\tau = T_{RGS}/T_{ARGS}$. Table 4 shows τ for varying number of testing examples. We make two observations. First, τ increases monotonically with number of testing examples. Second, we get speedup factor ranging from 3 to 10 on the entire testing set. Additionally, we see large τ for harder tasks (ACE2005, MSRC, Yeast-F1). Therefore, amortized RGS will be very beneficial for solving large number of inference problems and/or harder tasks.

A.4 Structured Learning using Amortized RGS Inference with Different Batch Sizes

We present results comparing structured learning with amortized RGS (Algorithm 3) and baseline RGS in terms of training time and accuracy. We train structured SVM with entire training set as batch D', which is the standard configuration employed in practice. We define speedup factor τ as T_{RGS}/T_{A-RGS}, where T_{A-RGS} and T_{RGS} stand for training time using amortized RGS and RGS inference solvers. Results for raw training time, accuracy, and speedup factor of A-RGS when compared to RGS are shown in Table 5. We make two observations. First, training time with amortized RGS is significantly less when compared to training with RGS and accuracies are almost same. Second, speedup factor τ is large for harder tasks (ACE2005 and MSRC).

	Yeast-F1	HW-Large	ACE2005	MSRC
T_{A-RGS} (milli seconds)				
Time	28±1.12E0	4812±1.79E+1	2294±2.10E+1	55355±3.17E+2
Speedup factor τ of amortized RGS inference				
100%D	10.24±1.61E-2	3.6±1.98E-2	4.12±1.81E-2	5.11±2.95E-1
90%D	5.01±2.75E-2	2.88±7.00E-3	3.15±2.38E-3	3.54±6.20E-2
80%D	3.23±8.70E-3	2.38±2.26E-2	2.55±1.31E-2	2.81±8.20E-2
70%D	2.46±6.81E-2	2.02±2.04E-3	2.13±1.47E-2	2.26±2.16E-1
60%D	2.14±6.80E-2	1.76±1.40E-2	1.83±7.12E-3	1.89±3.27E-1
50%D	1.73±7.10E-2	1.56±1.90E-2	1.61±3.00E-3	1.68±1.12E-1
40%D	1.49±5.40E-2	1.41±9.51E-3	1.43±8.17E-3	1.45±3.00E-1
30%D	1.35±2.13E-2	1.28±6.00E-3	1.29±4.68E-3	1.31±1.55E-1
20%D	1.19±6.17E-2	1.17±1.70E-2	1.17±9.21E-3	1.18±2.89E-1
10%D	1.06±3.69E-2	1.08±1.25E-3	1.08±7.44E-3	1.07±4.71E-2

Table 4: Speedup factor of amortized RGS inference vs. # of testing examples.

Datasets	RGS		Amortized RGS		Speedup factor
	Time (min.)	Acc.	Time (min.)	Acc.	
Structured SVM (100% training batch)					
Yeast-F1	17±0.71	63.35±1.61E-6	10±0.83	63.20±2.97E-6	1.69±9.40E-3
HW-Large	91±1.03	97.79±9.01E-4	32±1.16	97.51±3.13E-4	2.82±1.61E-3
ACE2005	179±8.83	77.58±7.85E-4	44±6.35	77.32±1.26E-3	4.01±1.57E-1
MSRC	279±10.3	83.47±6.06E-3	89±8.51	84.02±1.84E-3	3.13±4.28E-2
Structured SVM (50% training batch)					
Yeast-F1	12±0.99	63.35±5.93E-7	7±0.26	63.26±1.69E-6	1.62±7.65E-3
HW-Large	81±1.12	97.59±6.26E-5	36±0.20	97.21±2.96E-4	2.22±8.54E-4
ACE2005	127±7.37	77.63±3.91E-4	33±1.23	76.98±1.93E-3	3.79±2.57E-2
MSRC	216±8.29	83.63±4.29E-4	58±7.36	83.00±2.67E-3	3.70±1.07E-1
Structured SVM (25% training batch)					
Yeast-F1	13±0.31	63.69±6.9E-7	6±0.14	63.15±2.4E-6	2.12±8.22E-4
HW-Large	78±1.16	97.12±1.94E-5	31±0.83	97.00±1.62E-4	2.48±3.33E-3
ACE2005	111±7.45	77.02±1.32E-3	36±2.20	77.21±1.49E-3	3.03±4.84E-2
MSRC	188±9.41	83.99±2.93E-3	42±4.32	83.78±2.67E-3	4.45±8.67E-2
Structured SVM (Online)					
Yeast-F1	9±1.04	63.43±7.45E-7	6±0.15	63.22±1.6E-6	1.64±5.00E-3
HW-Large	71±2.81	97.48±5.2E-4	44±0.70	97.45±3.88E-4	1.58±9.75E-4
ACE2005	114±2.38	77.61±2.64E-4	27±0.46	77.37±1.16E-3	4.21±5.25E-3
MSRC	171±6.17	83.25±7.54E-4	39±5.63	83.32±3.77E-3	4.36±8.68E-2

Table 5: RGS training time and speedup factors on different datasets.