Query-Efficient and Scalable Black-Box Adversarial Attacks on Discrete Sequential Data via Bayesian Optimization

Deokjae Lee[1], Seungyong Moon[1], Junhyeok Lee[1], Hyun Oh Song[1]

[1]Seoul National University

QR to Our Github !

Summary

- We propose a *Blockwise Bayesian Attack* (BBA), a novel query-efficient and scalable black-box attack framework based on *Bayesian Optimization* (BO).
- We propose a post-optimization technique which reduces the perturbation size.
- BBA achieves higher attack success rate (ASR) with considerably less modification rate (MR) and fewer required queries (Qrs) on all experiments we consider.

Adversarial Attack on Discrete Sequential Data

- Neural networks on discrete sequential data have exhibited vulnerability against adversarial examples that are artificially crafted to fool the networks by adding perturbations imperceptible to humans.

Adversarial Examples of Our Method (BBA)

- An Example on Document-Level Classification Dataset (Yelp) against BERT
  - ▷ **Original Sequence:**
    Food is *fantastic* and exceptionally *clean* ! My only complaint is I went there with my 2 small children and they were showing a very inappropriate R rated movie ! (LABEL: Positive)

    ↓ Our Method (BBA)

  - ▷ **Adversarial Sequence:**
    Food is *gorgeous* and exceptionally *unpolluted* ! My only complaint is I went there with my 2 small children and they were showing a very inappropriate R rated movie ! (LABEL: Negative)
- An Example on Protein Classification Dataset (EC50) against AWD-LSTM
  - ▷ **Original Sequence:**
    MATPWRRALLMILASQVVTLVKCLEDDDVPEEWLLLHVVQGQIGAGNYSYLRLNHEGKIILRMQSLRGDAD
    LYVSDSTPHPSFDDYELQSVTCGQDVVSIPAHFQRPVGIGIYGHPSHHESDFEMRVYYDRTVDQYPFGEAA
    YFTDPTGASQQQAYAPEEAAQEEESVLWTILISILKLVLEILF (LABEL: Non-Enzyme)

    ↓ Our Method (BBA)

  - ▷ **Adversarial Sequence:**
    MATPWRRALLMRLASQVVTLVKCLEDDDVPEEWLLLHVVQGQIGAGNYSYLRLNHEGKIILRMQSLRGDAD
    LYVSDSTPHPSFDDYELQSVTCGQDVVSIPAHFQRPVGIGIYGHPSHHESDFEMRVYYDWTVDWYPFGEAA
    YFTDPTGASQQQAYAPEEAAQEEESVLWTILISILKLVLEILF (LABEL: Enzyme)

Black-Box Attacks

- Adversary can only observe the predicted class probabilities on inputs with a limited number of queries to the network.
- Black-box attack is challenging but is a more realistic scenario since, for many commercial systems [1, 2], the adversary can only query input sequences and receive their prediction scores with restricted resources such as time and cost.

Objective

- We focus on black-box adversarial attacks on discrete sequential data.
- We aim to propose a new framework that finds adversarial examples with smaller perturbation size using less number of queries compared to existing methods.

Notation

- $s = [w_0, \ldots, w_{l-1}] \in \mathcal{X}^l$ is an original sequence and y is its corresponding label.
- $\mathcal{C}(w_i) \subseteq \mathcal{X}$ is a set of semantically similar candidates of w_i.
- $f_\theta(s) : \mathcal{X}^l \to \mathbb{R}^{|\mathcal{Y}|}$ is a target classifier.
- $d_H(s, s')$ is the Hamming distance between two sequences.
- $\mathcal{L}(f_\theta(s'), y) \triangleq \max_{y' \in \mathcal{Y}, y' \neq y} f_\theta(s')_{y'} - f_\theta(s')_y$ is the attack criterion.

Problem Formulation

- Following prior works, we consider the product space of synonym sets $\prod_{i=0}^{l-1} \mathcal{C}(w_i)$ as the attack space and aim to find an adversarial example s' that minimize the Hamming distance $d_H(s, s')$ [3, 4, 5].
- Note that we mainly consider 3 types of $\mathcal{C}$ corresponding to 3 word substitution methods based on Embedding [3], WordNet [4], and HowNet [5].
- Formally, the objective can be written as the following optimization problem:

$$\begin{aligned} &\underset{s' \in \prod_{i=0}^{l-1} \mathcal{C}(w_i)}{\text{minimize}} \quad d_H(s, s') \\ &\text{subject to} \quad \mathcal{L}(f_\theta(s'), y) \geq 0. \end{aligned}$$

Blockwise Bayesian Attack Framework

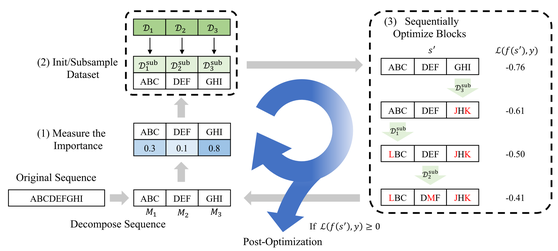


Figure 1: The overall process of BBA. A green arrow with a dataset $\mathcal{D}_k^{\text{sub}}$ denotes the Bayesian optimization step for the block M_k using D_k^{sub} as the initial dataset.

- BBA divides the problem into two steps.
  - ▷ **Finding an adversarial sequence.**
    BBA conducts BO to maximize $\mathcal{L}(f_\theta(\cdot), y)$ until finding an adversarial sequence.
  - ▷ **Post-optimization process.**
    BBA reduces the Hamming distance between the perturbed sequence and the original sequence while maintaining feasibility.
- BBA devises several techniques for the scalability.
  - ▷ **Block decomposition.**
    Solve the high query complexity problem (curse of dimensionality) in BO when the input is long.
  - ▷ **History subsampling.**
    Solve the high computational complexity problem in BO w.r.t. the number of evaluations.

Surrogate Model and GP Parameter Fitting

- BBA uses a categorical kernel with automatic relevance determination of the form

$$K^{\text{cate}}(s^{(1)}, s^{(2)}) = \sigma_f^2 \prod_{i=0}^{l-1} \exp\left(-\frac{\mathbf{1}[w_i^{(1)} \neq w_i^{(2)}]}{\beta_i}\right)$$

  to automatically determine the degree to which each input dimension is important.
- The surrogate model g can be written by

$$g(X) \sim \mathcal{N}(\eta, K^{\text{cate}}(X, X; \{\beta_i\}, \sigma_f^2) + \sigma_n^2 I).$$

- BBA updates GP parameters to the maximizer of the posterior probability

$$p(\eta, \{\beta_i\}, \sigma_f^2, \sigma_n^2 \mid \mathcal{D}).$$

Block Decomposition (BD)

- BBA divides the input sequence of length l into $\lceil l/m \rceil$ disjoint blocks $\{M_k\}_{i=0}^{\lceil l/m \rceil - 1}$ of length m.
- At the start of each iteration, BBA assigns an importance score to each block.
- Then, BBA sequentially optimizes each block in order of decreasing importance score.
- Importance score.
  - ▷ **First iter:** change in objective after deleting the block.
  - ▷ **Remaining iters:** $\sum_{i \in M_k} 1/\beta_i$.
- BO on each block has a bounded search space size. Hence, BD can solve the high query complexity problem.

History Subsampling (HS)

- Fitting the GP model requires the matrix inversion of the kernel, whose computational complexity is $\mathcal{O}(n^3)$ where n is the number of evaluations so far.
- To this end, BBA only uses a subset of evaluations to fit GP model.
- BBA adopts Subset of Data (SoD) method with Farthest Point Clustering (FPC) for the subsampling technique and achieves $\mathcal{O}(1)$ computational complexity w.r.t. the total number of evaluations.

Post-Optimization Process

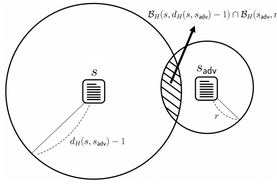


Figure 2: The reduced attack space used in the post-optimization process.

- Post-optimization process finds an adversarial sequence with a smaller MR.
- BBA repeatedly conducts BO on
  $\overbrace{\mathcal{B}_H(s, d_H(s, s_{\text{adv}}) - 1)}^{\text{establish smaller MR}} \cap \overbrace{\mathcal{B}_H(s_{\text{adv}}, r)}^{\text{optimize near } s_{\text{adv}}}$
  to find a new s_{adv} with a smaller MR.
- If we find a new adversarial sequence, we replace the current adversarial sequence with the new sequence and repeat the step above until the halting condition holds.

Results

$\mathcal{C}$	Model	Method	ASR (%)	MR (%)	Qrs
WordNet	BERT-base	PWWS	57.1	18.3	367
		BBA	**77.4**	**17.8**	**217**
	LSTM	PWWS	78.3	16.4	336
		BBA	**83.2**	**15.4**	**190**
Embedding	BERT-base	TF	84.7	24.9	346
		BBA	**96.0**	**18.9**	**154**
	LSTM	TF	94.9	17.3	228
		BBA	**98.5**	**16.6**	**142**
HowNet	BERT-base	PSO	67.2	21.2	65860
		BBA	**70.8**	**15.5**	**5176**
	LSTM	PSO	71.0	19.7	44956
		BBA	**71.9**	**13.7**	**3278**

(a) AG's News

$\mathcal{C}$	Model	Method	ASR (%)	MR (%)	Qrs
WordNet	BERT-base	PWWS	94.3	7.6	1036
		BBA	**99.2**	**7.4**	**486**
		LSH	92.6	9.5	389
		BBA	**98.8**	**8.8**	**271**
Embedding	BERT-base	TF	93.5	11.1	461
		BBA	**99.8**	**9.6**	**319**
		LSH	94.7	8.9	550
		BBA	**99.8**	**8.6**	**403**
HowNet	BERT-base	PSO	**98.8**	10.6	86611
		BBA	**98.8**	**8.2**	**283**
		LSH	93.9	8.0	533
		BBA	**98.2**	**7.4**	**353**

(b) Yelp Polarity

Table 1: The main attack results on text classification datasets. 'WordNet', 'Embedding', and 'HowNet' denote the type of $\mathcal{C}$. BBA significantly outperforms all the baseline methods in all the evaluation metrics for all datasets and victim models we consider. We note that PWWS, TF, and LSH are greedy-based methods and PSO is an evolutionary method.

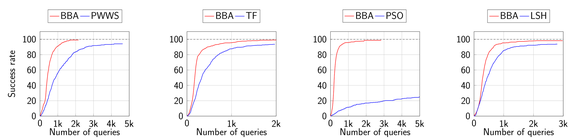
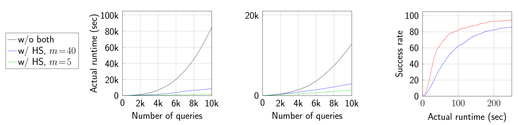


Figure 3: The cumulative distribution of the number of queries required for the attack methods against a BERT-base model on the Yelp dataset. We use the HowNet based word substitution when comparing our method against LSH. The results show that BBA finds successful adversarial texts using fewer queries than the baseline methods.



(a) 123rd text (b) 348th text

Figure 4: The cumulative runtime versus the number of queries plot. HS in the legend denotes history subsampling, and $m = k$ in the legend denotes block decomposition with the block size k. We select the texts from the Yelp dataset.

Figure 5: The cumulative distribution of the actual runtime required for the attack methods against XLNet-large on Yelp. The result shows that BBA consistently finds successful texts faster than PWWS. Refer to **Table 3** below for the detailed attack results.

	Level 0			Level 1			Level 2		
Method	ASR	MR	Qrs	ASR	MR	Qrs	ASR	MR	Qrs
TF	83.8	3.2	619	85.8	3.0	584	89.6	2.5	538
BBA	**99.8**	**2.9**	**285**	**99.8**	**2.3**	**293**	**100.0**	**2.0**	**231**

Table 2: Attack results against AWD-LSTM models on the protein classification dataset EC50 level 0, 1, and 2.

Dataset	Model	Method	ASR (%)	MR (%)	Qrs
Yelp	XLNet-large	PWWS	94.5	10.8	1107
		BBA	**98.2**	**9.4**	**485**

Table 3: Attack results against XLNet-large on Yelp. We use $\mathcal{C}$ based on WordNet.

References

[1] Amazon Comprehend. https://aws.amazon.com/comprehend, 2022.

[2] Google Cloud NLP. https://cloud.google.com/natural-language, 2022.

[3] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *AAAI*, 2020.

[4] S. Ren, Y. Deng, K. He, and W. Che. Generating natural language adversarial examples through probability weighted word saliency. In *ACL*, 2019.

[5] Y. Zang, C. Yang, F. Qi, Z. Liu, M. Zhang, Q. Liu, and M. Sun. Word-level textual adversarial attacking as combinatorial optimization. In *ACL*, 2020.