ICLR Paper

Abstract

Introduction

Program Representation with Abstract Syntax Translation

Variational Autoencoder

Limitations of VAE:

1. It is non-trivial to implement hard constrains with a VAE using RNNs or LSTMs.
2. VAEs learn an approximation of the posterior distribution P(Z|X), which is Q(Z|X). Because of this, reconstructed data x' is not identical to the original data x.

Metropolis-Hastings (MH) Algorithm

The Case for MH:

1. MCMC can guarantee to satisfy hard constrains by adding indicator functions to the stationary distribution. This was proposed in the paper "Constrained Sentence Generation by MH Sampling" [Miao et al].
2. By setting the stationary distribution of the Markov Chain to P(Z|X), we can guarantee that after convergence we obtain P(Z|X), we can guarantee that after convergence we obtain P(Z|X) and hence can sample from the true posterior distribution. This idea was first introduced in the paper "Markov Chain Monte Carlo & Variational Inference: Bridging the Gap" [Salinas et al].

Technical Approach

Data (V, E, RT, FP) for AST G(V, E)
Let A be the set of all APIs, RT be the set of all return types, and FP be the set of formal parameters in the vocabulary.
Let S be the set of control structures, S = {DSubTree, DBranch, DLoop, DExcept, DStop}
Let AST x be represented by directed graph G(V, E).

$V = A \cup S$
$E = \{\text{sibling}, \text{child}\}$

VAE

Encoder: Q(Z| X, RT, FP)
Decoder: P(X| Z, RT, FP)

<u>Metropolis-Hastings</u>
<u>Required Inputs:</u>
C = list of apis or control structures that must appear in the program in order for it to be valid.

$c_i \in A$ or $c_i \in S$ and $|C| = n$

rt = return type, $|rt| = 1$, $rt \in RT$

fp = formal parameters, $fp \in FP$

<u>Optional Inputs:</u>
e = list of apis or control structures S \ {DSubTree} that cannot appears in the program to be considered valid.

$e_i \in V \setminus \{DSubTree\}$ and $e_i \neq c_j \ \forall i,j$

min_L = minimum length of program to be valid
max_L = maximum length of the program to be valid, |c| < max_L

<u>Initialization:</u>

x_0 is the initial program

$x_0[0] = DSubTree$

$x_0[1] \ldots x_0[n] \in C$

$Z_0 \sim Q(Z \mid x_0, rt, fp)$

If min_L > |C| then nodes as added to the tree using the *insertion* proposal till min_L = |C|.

Problem Statement

We do Component-wise Metropolis-Hastings to sample from the multivariate posterior distribution. In Component-wise Metropolis-Hastings, we loop over each dimension and sample each dimension independently from the others. These are called component-wise updates. Each proposal distribution is univariate, working in only one dimension. The proposal for the i-th dimension is either accepted or rejected, while all other dimensions are held fixed. Here, we describe proposals for both the program space dimension and latent space dimension.

Proposals for Program Space

Let g be the proposal distribution
x be the AST at time t where $|x| = n$ and
$x = w_1 \dots w_n$, $w_i \in V$

<u>*Insertion*</u>

Insert node w_c at position m where $1 < m \leq n$

$$g_{insert}(x' | x) = \begin{cases} 0 \text{ if } x' \neq w_1, \dots, w_{m-1}, w_c, w_m, \dots, w \\ \text{softmax}\left(P(w_1, \dots, w_{m-1}, w^*, w_m, \dots w_n \mid Z_t, z \right. \\ \qquad \left. \theta P) \; \forall \, w^* \in V\right) \text{ if } \\ \qquad x' = w_1, \dots, w_{m-1}, w_c, w_m, \dots, w \end{cases}$$

<u>if w_c is DBranch, DLoop or DExcept</u> :
use insert proposal to add required nodes till a DStop node is added.
Say nodes w_a and w_b are added after w_c such that $x''' = w_1, \dots, w_{m-1}, w_c, w_a, w_b, w_m, w_{m+1} \dots w_n$

$$g_{insert}(x''' | x) = g_{insert}(x''' | x'') \cdot g_{insert}(x'' | x') \cdot g_{insert}(x' | x$$

where $x'' = w_1, \dots, w_{m-1}, w_c, w_a, w_m, w_{m+1}, \dots, w_n$

<u>*Replacement*</u>

We wish to replace w_m with a new node w^* at position m where $1 < m \leq n$

$$g_{replace}(x'|x) = \begin{cases} 0 \text{ if } x = [w_1, \ldots, w_{m-1}, w^*, w_{m+1}, \ldots, w_n] \\ \text{softmax}(P(w_1, \ldots, w_{m-1}, w^*, w_{m+1}, \ldots, w_n | Z_t \; r_t, f_p) \; \forall \, w^* \in V) \text{ if} \\ x = [w_1, \ldots, w_{m-1}, w^*, w_{m+1}, \ldots, w_n] \end{cases}$$

Swap

Let $w_s, w_t \in x$ where $1 < s \leq n$ and $1 < t \leq n$ and $s \neq t$.

Let $x = [w_1, \ldots, w_t, w_s, \ldots, w_n]$

$$g_{swap}(x'|x) = \begin{cases} 1 \text{ if } x' = [w_1, \ldots, w_s, w_t, \ldots w_n] \\ 0 \text{ if } x' \neq [w_1, \ldots, w_s, w_t, \ldots, w_n] \end{cases}$$

Constraint Growth

Insert node after w_m

$$g_{\text{insert after constraint}}(x'|x) = \begin{cases} 0 & \text{if } x' = w_1, \ldots, w_m, w_{c_0}, \ldots w_n \text{ and } w_m \notin C \\ \text{softmax}(P(w_1, \ldots, w_m, w_c, \ldots w_n \mid Z_t, r_t, f_p) \; \forall w_{c_0} \in V, & \text{if } x' = w_1, \ldots, w_m, w_c, \ldots, w_n \text{ (} w_m \in C \end{cases}$$

Randomly choose how many nodes to add after w_c, k. This gives new AST x^k.

$$g_{\text{constraint\_growth}}(x^k|x) =$$

$$g_{\text{insertafter constraint}}(x^0|x) \cdot g_{\text{insert}}(x^1|x^0) \ldots g_{\text{insert}}(x^k|x^{k-1}$$

where:

$$x^i = w_1, \ldots, w_m, w_{c_0} \ldots w_{c_i}, w_{m+1}, \ldots w_n \;\; \forall 0 \leq i \leq$$

Deletion

Delete node w_m from x where $1 < m \leq n$

$$g_{delete}(x'|x) = \begin{cases} 1 & \text{if } x' = [w_1, \ldots, w_{m-1}, w_{m+1}, \ldots, w_n] \\ 0 & \text{if } x' \neq [w_1, \ldots, w_{m-1}, w_{m+1}, \ldots, w_n] \end{cases}$$

(Copied from CGMH paper)

Deleting multiple nodes (only to reverse insertion of control structures or constraint growth proposal):

Say the original AST is $X = w_1 \ldots w_n$ and nodes w_a and w_b were added such that

$$x'' = w_1, \ldots, w_a, \ldots w_n$$
$$x' = w_1, \ldots, w_a, w_b, \ldots w_n$$

Then,

$$g_{delete}(x'|x) = g_{delete}(x'|x'') \cdot g_{delete}(x''|x)$$

Proposals for Latent Space

Independent Proposal

The proposal for the latent state does not depend on the current latent state.

At time $t+1$, x_{t+1} is already computed.

The latent space is given by

$$\mu_{t+1}, \sigma_{t+1} \leftarrow Q(Z_{t+1} | x_{t+1}, rt, fp)$$

$$z_{t+1} \sim Z_{t+1} = \mathcal{N}(\mu_{t+1}, \sigma_{t+1})$$

$$g(z_{t+1} | z_t) = g(z_{t+1}) = \mathcal{N}(\mu_{t+1}, \sigma_{t+1})$$

Then, the acceptance rate is

$$A^{*}(z_{t+1}|z_t) = \frac{P_{\substack{\text{independent}\\\text{proposal}}} \cdot g(z_t|z_{t+1}) \cdot \pi(x_{t+1}, z}{P_{\substack{\text{independent}\\\text{proposal}}} \cdot g(z_{t+1}|z_t) \cdot \pi(x_{t+1}, z_t)}$$

$$= \frac{f_{z_t}(z_t) \cdot \pi(x_{t+1}, z_{t+1})}{f_{z_{t+1}}(z_{t+1}) \cdot \pi(x_{t+1}, z_t)}$$

$$\pi(x_{t+1}, z_{t+1}) = \pi(x_{t+1}|z_{t+1}) \cdot \pi(z_{t+}$$

$$\frac{\cancel{f_{z_t}(z_t)} \cdot P(x_{t+1}|z_{t+1}) \cdot \cancel{f_{z_{t+1}}(z_{t+1})}}{\cancel{f_{z_{t+1}}(z_{t+1})} \cdot P(x_{t+1}|z_t) \cdot \cancel{f_{z_t}(z_t}}$$

$$= \frac{f_{z_t}(z_t) \cdot \cancel{P(x_{t+1})} \cdot Q(z_{t+1}|x_{t+1}}{f_{z_{t+1}}(z_{t+1}) \cdot \cancel{P(x_{t+1})} \cdot Q(z_t|x_{t+}}$$

Stationary Distribution

$$\pi(x, Z) \propto P(x, Z \mid rt, jp) \cdot \overline{X}(x) \cdot \overline{E}(x) \cdot \overline{L}(x) \cdot V(x$$

where,

$P(x, Z \mid rt, jp)$ is the joint distribution given by the VA

$$\overline{X}(x) = x^0_{c_0} \cdot X^1_{c_1} \cdot \ldots \cdot x^{n-1}_{c_{n-1}}$$

$x^i_{c_i}$ is an indicator function where

$$x^i_{c_i}(x) = \begin{cases} 1 & \text{if } c_i \in x \\ 0 & \text{if } c_i \notin x \end{cases}$$

$$\overline{E}(x) = E^0_{e_i} \cdot E^1_{e_1} \cdot \ldots \cdot E^{m-1}_{e_{m-1}}$$

$E^i_{c_i}$ is an indicator function where

$$E^i_{c_i}(x) = \begin{cases} 1 & \text{if } e_i \notin x \\ 0 & \text{if } e_i \in x \end{cases}$$

$$\overline{L}(x) = L_{min} \cdot L_{max}$$

where L_{min} and L_{max} are indicator functio

$$L_{min}(x) = \begin{cases} 1 & \text{if } |x| \geq min \\ 0 & \text{if } |x| < min \end{cases}$$

$$L_{max}(x) = \begin{cases} 1 & \text{if } |x| \leq max \\ 0 & \text{if } |x| > max \end{cases}$$

V is an indicator function where

$$V(x) = \begin{cases} 1 & \text{if } x \text{ is a valid AST} \\ 0 & \text{if } x \text{ is a valid AST} \end{cases}$$

Acceptance Rate

At time $t+1$, x' is the AST candidate

$$A^*_{insert}(x'|x_t) = \frac{P_{delete} \cdot g_{delete}(x_t|x') \cdot \pi(x', Z_t)}{P_{insert} \cdot g_{insert}(x'|x_t) \cdot \pi(x_t, Z_t)}$$

$$A^*_{delete}(x'|x_t) = \frac{P_{insert} \cdot g_{insert}(x_t|x') \cdot \pi(x', Z_t)}{P_{delete} \cdot g_{delete}(x'|x_t) \cdot \pi(x_t, Z_t)}$$

$$A^*_{swap}(x'|x_t) = \frac{P_{swap} \cdot g_{swap}(x_t|x') \cdot \pi(x', Z_t)}{P_{swap} \cdot g_{swap}(x'|x_t) \cdot \pi(x_t, Z_t)}$$

$$= \frac{\pi(x', Z_t)}{\pi(x_t, Z_t)}$$

$$A^*_{replace}(x'|x_t) = \frac{P_{replace} \cdot g_{replace}(x_t|x') \cdot \pi(x', Z_t)}{P_{replace} \cdot g_{replace}(x'|x_t) \cdot \pi(x_t, Z_t)}$$

$$\approx \frac{\pi(w_m | x_{1 \ldots n}, Z, rt, dp) \cdot \pi(x', Z_t)}{\pi(w'_m | x'_{1 \ldots n}, Z, rt, dp) \cdot \pi(x_t, Z_t)}$$

$$= 1$$

$$A^*_{constraint\ growth}(x'|x_t) = \frac{P_{delete} \cdot g_{delete}(x_t|x') \cdot \pi(x', Z_t)}{P_{constraint\ growth} \cdot g_{constraint\ growth}(x'|x_t) \cdot \pi(x_t, Z_t)}$$

Experiments