Sequence to Better Sequence: Continuous Revision of Combinatorial Structures

Jonas Mueller, David Gifford, Tommi Jaakkola

MIT Computer Science & Artificial Intelligence Laboratory

jonasmueller@csail.mit.edu

Introduction

Introduction

- Discrete sequence data is commonplace (eg. text, proteins/genes)

  sequence $x = (s_1, \ldots, s_T) \in \mathcal{X}$ where each symbol $s_t \in \mathcal{S}$ (discrete vocabulary)

Introduction

- Discrete sequence data is commonplace (eg. text, proteins/genes)

  sequence $x = (s_1, \ldots, s_T) \in \mathcal{X}$ where each symbol $s_t \in \mathcal{S}$ (discrete vocabulary)

- Tiny fraction of $\mathcal{X}$ represents sequences likely to naturally occur
  (ie. those which appear *realistic*)

Introduction

- Discrete sequence data is commonplace (eg. text, proteins/genes)

  sequence $x = (s_1, \ldots, s_T) \in \mathcal{X}$ where each symbol $s_t \in \mathcal{S}$ (discrete vocabulary)

- Tiny fraction of $\mathcal{X}$ represents sequences likely to naturally occur (ie. those which appear *realistic*)

- Each sequence x is associated with outcome $y \in \mathbb{R}$

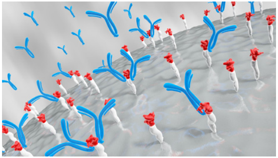
Introduction

- Discrete sequence data is commonplace (eg. text, proteins/genes)

  sequence $x = (s_1, \ldots, s_T) \in \mathcal{X}$ where each symbol $s_t \in \mathcal{S}$ (discrete vocabulary)

- Tiny fraction of $\mathcal{X}$ represents sequences likely to naturally occur (ie. those which appear *realistic*)

- Each sequence x is associated with outcome $y \in \mathbb{R}$



$\hookrightarrow y, \quad x =$ ASVKVSKC

Problem Setup

Problem Setup

- Dataset $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^n \stackrel{iid}{\sim} p_{XY}$ of sequence-outcome pairs
- $p_X =$ generative model of the *natural* sequences (unknown)

Problem Setup

- Dataset $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^n \overset{iid}{\sim} p_{XY}$ of sequence-outcome pairs
- $p_X =$ generative model of the *natural* sequences (unknown)

Goal: Given new sequence $x_0 \sim p_X$ (with unknown outcome), quickly identify a revision x^* with superior expected outcome

$$x^* = \underset{x \in \mathcal{C}_{x_0}}{\operatorname{argmax}} \ \mathbb{E}[Y \mid X = x]$$

$\mathcal{C}_{x_0} \subset \mathcal{X} =$ feasible set of natural sequences

Desiderata for our revision procedure: $x_0 \rightarrow x^*$

- Produces natural sequences

Desiderata for our revision procedure: $x_0 \to x^*$

- Produces natural sequences

$p_X(x^*)$ not too small

Desiderata for our revision procedure: $x_0 \to x^*$

- Produces natural sequences

$p_X(x^*)$ not too small

- Preserves intrinsic similarity

Desiderata for our revision procedure: $x_0 \to x^*$

- Produces natural sequences

  $p_X(x^*)$ not too small

- Preserves intrinsic similarity

  x^* and x_0 share similar underlying latent characteristics

Desiderata for our revision procedure: $x_0 \to x^*$

- Produces natural sequences

  $p_X(x^*)$ not too small

- Preserves intrinsic similarity

  x^* and x_0 share similar underlying latent characteristics

- Improves outcomes

Desiderata for our revision procedure: $x_0 \to x^*$

- Produces natural sequences

  $p_X(x^*)$ not too small

- Preserves intrinsic similarity

  x^* and x_0 share similar underlying latent characteristics

- Improves outcomes

  $$\mathbb{E}[Y \mid X = x^*] > \mathbb{E}[Y \mid X = x_0]$$

Desiderata for our revision procedure: $x_0 \to x^*$

- Produces natural sequences

  $p_X(x^*)$ not too small

- Preserves intrinsic similarity

  x^* and x_0 share similar underlying latent characteristics

- Improves outcomes

  $$\mathbb{E}[Y \mid X = x^*] > \mathbb{E}[Y \mid X = x_0]$$

- Computationally efficient

Desiderata for our revision procedure: $x_0 \to x^*$

- Produces natural sequences

  $p_X(x^*)$ not too small

- Preserves intrinsic similarity

  x^* and x_0 share similar underlying latent characteristics

- Improves outcomes

  $$\mathbb{E}[Y \mid X = x^*] > \mathbb{E}[Y \mid X = x_0]$$

- Computationally efficient

  Simple gradient optimization instead of discrete search

Related Work

- Do not require improved versions of a particular sequence
  (as in seq2seq/imitation learning)

Related Work

- Do not require improved versions of a particular sequence
  (as in seq2seq/imitation learning)

- Do not require any outcomes outside of given dataset
  (as in bandits/reinforcement learning)

Related Work

- Do not require improved versions of a particular sequence
  (as in seq2seq/imitation learning)
- Do not require any outcomes outside of given dataset
  (as in bandits/reinforcement learning)
- Combinatorial optimization commonly performed via search heuristics like genetic programming (evaluates minor changes in isolation)

Related Work

- Do not require improved versions of a particular sequence
  (as in seq2seq/imitation learning)

- Do not require any outcomes outside of given dataset
  (as in bandits/reinforcement learning)

- Combinatorial optimization commonly performed via search heuristics like genetic programming (evaluates minor changes in isolation)

- Gradient-optimization of inputs w.r.t. neural network predictions
  (mostly for conditional generation in the continuous image domain)

Related Work

- Do not require improved versions of a particular sequence
  (as in seq2seq/imitation learning)

- Do not require any outcomes outside of given dataset
  (as in bandits/reinforcement learning)

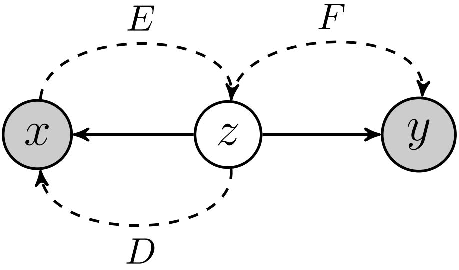
- Combinatorial optimization commonly performed via search heuristics like genetic programming (evaluates minor changes in isolation)

- Gradient-optimization of inputs w.r.t. neural network predictions
  (mostly for conditional generation in the continuous image domain)

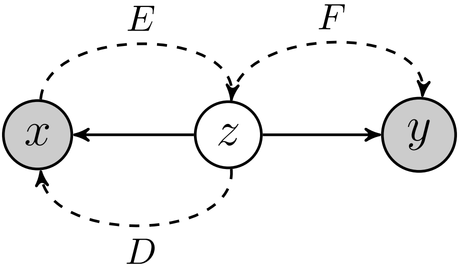
- Gomez-Bombarelli et al.[1] also utilize autoencoder representations to propose novel chemical structures via Bayesian optimization

[1] Gomez-Bombarelli, Duvenaud, Hernandez-Lobato, Aguilera-Iparraguirre, Hirzel, Adams, and Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *arXiv*, 2016

Probabilistic Generative Model



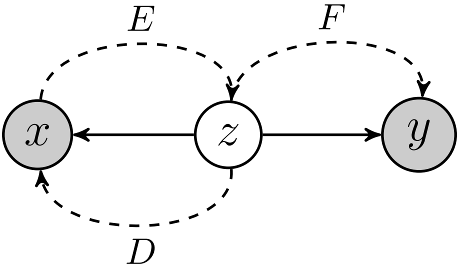
Probabilistic Generative Model



- Continuous latent factors $Z \in \mathbb{R}^d$ produce sequence X + outcome Y

  Prior: $p_Z = N(0, \mathbf{I})$

Probabilistic Generative Model

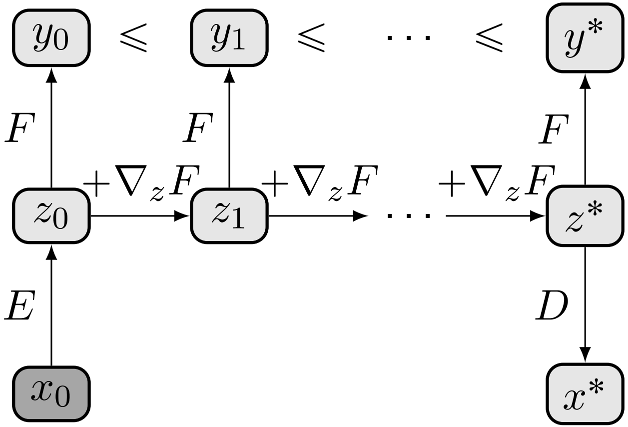


- Continuous latent factors $Z \in \mathbb{R}^d$ produce sequence X + outcome Y

  Prior: $p_Z = N(0, \mathbf{I})$

- Approximate inference maps F, E, D parameterized via three neural networks $\mathcal{F}, \mathcal{E}, \mathcal{D}$

Revision Framework



Variational Autoencoder (VAE)

- Generative model for sequences: $z \sim p_Z, \;\; x \sim \underbrace{p_D(x \mid z)}_{\text{parameterized by RNN } \mathcal{D}}$

Variational Autoencoder (VAE)

- Generative model for sequences: $z \sim p_Z, \;\; x \sim \underbrace{p_D(x \mid z)}_{\text{parameterized by RNN } \mathcal{D}}$

- Variational posterior approximation:
$$p(z \mid x) \propto \frac{p_D(x|z)}{p_Z(z)} \;\approx\; \underbrace{N(\mu_{z|x}, \mathrm{diag}(\sigma^2_{z|x}))}_{q_E(z \mid x) \text{ parameterized by RNN } \mathcal{E}}$$

Variational Autoencoder (VAE)

- Generative model for sequences: $z \sim p_Z, \;\; x \sim \underbrace{p_D(x \mid z)}_{\text{parameterized by RNN } \mathcal{D}}$

- Variational posterior approximation:
$$p(z \mid x) \propto \frac{p_D(x|z)}{p_Z(z)} \approx \underbrace{N(\mu_{z|x}, \mathrm{diag}(\sigma^2_{z|x}))}_{q_E(z \mid x) \text{ parameterized by RNN } \mathcal{E}}$$

- Learn parameters of $\mathcal{E}, \mathcal{D}$ using stochastic variational inference:
$$\log p_X(x) \geqslant -\left[\mathcal{L}_{\text{rec}}(x) + \mathcal{L}_{\text{pri}}(x)\right]$$
$$\mathcal{L}_{\text{rec}}(x) = -\mathbb{E}_{q_E(z|x)}\left[\log p_D(x \mid z)\right]$$
$$\mathcal{L}_{\text{pri}}(x) = \mathrm{KL}(q_E(z \mid x) || \, p_Z)$$

Variational Autoencoder (VAE)

- $\mathcal{E}, \mathcal{D} =$ standard language models with Gated Recurrent Unit[2]

[2] Cho, van Merrienboer, Gulcehre, Bahdanau, Bougares, Schwenk, and Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP*, 2014

Variational Autoencoder (VAE)

- $\mathcal{E}, \mathcal{D} =$ standard language models with Gated Recurrent Unit[2]
- $\mathcal{E}$ uses final hidden-state h_T to approximate posterior for $z \mid x$:

$$\mu_{z|x} = W_\mu h_T + b_\mu$$
$$\sigma_{z|x} = 1 \wedge \exp(-|W_\sigma v + b_\sigma|), \; v = \text{ReLU}(W_v h_T + b_v)$$

[2] Cho, van Merrienboer, Gulcehre, Bahdanau, Bougares, Schwenk, and Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP*, 2014

Variational Autoencoder (VAE)

- $\mathcal{E}, \mathcal{D} =$ standard language models with Gated Recurrent Unit[2]

- $\mathcal{E}$ uses final hidden-state h_T to approximate posterior for $z \mid x$:

$$\mu_{z|x} = W_\mu h_T + b_\mu$$
$$\sigma_{z|x} = 1 \wedge \mathsf{exp}(-|W_\sigma v + b_\sigma|), \; v = \mathsf{ReLU}(W_v h_T + b_v)$$

- We define:

$$E(x) = \operatorname*{argmax}_{z \in \mathbb{R}^d} \; q_E(z \mid x) \qquad \text{(MAP } Z\text{-estimate under encoder)}$$

[2] Cho, van Merrienboer, Gulcehre, Bahdanau, Bougares, Schwenk, and Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP*, 2014

Variational Autoencoder (VAE)

- $\mathcal{E}, \mathcal{D} =$ standard language models with Gated Recurrent Unit[2]
- $\mathcal{E}$ uses final hidden-state h_T to approximate posterior for $z \mid x$:

$$\mu_{z|x} = W_\mu h_T + b_\mu$$
$$\sigma_{z|x} = 1 \wedge \mathsf{exp}(-|W_\sigma v + b_\sigma|), \; v = \mathsf{ReLU}(W_v h_T + b_v)$$

- We define:

$$E(x) = \operatorname*{argmax}_{z \in \mathbb{R}^d} \; q_E(z \mid x) = \mu_{z|x}$$

(MAP Z-estimate under encoder)

[2] Cho, van Merrienboer, Gulcehre, Bahdanau, Bougares, Schwenk, and Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP*, 2014

Variational Autoencoder (VAE)

- $\mathcal{E}, \mathcal{D} =$ standard language models with Gated Recurrent Unit[2]

- $\mathcal{E}$ uses final hidden-state h_T to approximate posterior for $z \mid x$:

$$\mu_{z|x} = W_\mu h_T + b_\mu$$
$$\sigma_{z|x} = 1 \wedge \exp(-|W_\sigma v + b_\sigma|), \; v = \mathrm{ReLU}(W_v h_T + b_v)$$

- We define:

$$E(x) = \underset{z \in \mathbb{R}^d}{\operatorname{argmax}} \; q_E(z \mid x) = \mu_{z|x} \qquad \text{(MAP } Z\text{-estimate under encoder)}$$

$$D(z) = \underset{x \in \mathcal{X}}{\operatorname{argmax}} \; p_D(x \mid z) \qquad \text{(MAP } X\text{-estimate under decoder)}$$

[2] Cho, van Merrienboer, Gulcehre, Bahdanau, Bougares, Schwenk, and Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP*, 2014

Variational Autoencoder (VAE)

- $\mathcal{E}, \mathcal{D} =$ standard language models with Gated Recurrent Unit[2]

- $\mathcal{E}$ uses final hidden-state h_T to approximate posterior for $z \mid x$:

$$\mu_{z|x} = W_\mu h_T + b_\mu$$
$$\sigma_{z|x} = 1 \wedge \exp(-|W_\sigma v + b_\sigma|), \; v = \mathrm{ReLU}(W_v h_T + b_v)$$

- We define:

$$E(x) = \underset{z \in \mathbb{R}^d}{\operatorname{argmax}} \; q_E(z \mid x) \qquad \text{(MAP } Z\text{-estimate under encoder)}$$
$$= \mu_{z|x}$$

$$D(z) = \underset{x \in \mathcal{X}}{\operatorname{argmax}} \; p_D(x \mid z) \qquad \text{(MAP } X\text{-estimate under decoder)}$$

Greedily approximated via beam-search

[2] Cho, van Merrienboer, Gulcehre, Bahdanau, Bougares, Schwenk, and Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP*, 2014

Compositional Prediction of Outcomes

- Outcome map: $\underbrace{F(z)}_{\text{parameterized by feedforward net } \mathcal{F}} = \mathbb{E}[Y \mid Z = z]$

Compositional Prediction of Outcomes

- Outcome map: $\underbrace{F(z)}_{\text{parameterized by feedforward net } \mathcal{F}} = \mathbb{E}[Y \mid Z = z]$
- Taylor approximation: $F(E(x)) \approx \mathbb{E}[Y \mid X = x]$

Compositional Prediction of Outcomes

- Outcome map: $\underbrace{F(z)}_{\text{parameterized by feedforward net } \mathcal{F}} = \mathbb{E}[Y \mid Z = z]$

- Taylor approximation: $F(E(x)) \approx \mathbb{E}[Y \mid X = x]$

- Jointly train $\mathcal{E}$ and $\mathcal{F}$ with the loss:

$$\mathcal{L}_{\mathsf{mse}}(x, y) = [y - F(E(x))]^2$$

Enforcing Invariance

Bad Example: Suppose for $x \in \mathcal{X}$: $E(x) = z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \in \mathbb{R}^d$

$$\widehat{y} = F(z) = F(z_1) \quad \text{and} \quad \widehat{x} = D(z) = D(z_2)$$

Enforcing Invariance

Bad Example: Suppose for $x \in \mathcal{X}$: $E(x) = z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \in \mathbb{R}^d$

$\widehat{y} = F(z) = F(z_1)$ and $\widehat{x} = D(z) = D(z_2)$

- Avoid by bottlenecking latent dimensionality d

Enforcing Invariance

Bad Example: Suppose for $x \in \mathcal{X}$: $\quad E(x) = z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \in \mathbb{R}^d$

$$\widehat{y} = F(z) = F(z_1) \qquad \text{and} \qquad \widehat{x} = D(z) = D(z_2)$$

- Avoid by bottlenecking latent dimensionality d
- Add invariance loss to training objective:

$$\mathcal{L}_{\text{inv}} = \mathbb{E}_{z \sim p_Z} \big[\underset{\substack{\uparrow \\ \text{constant}}}{F}(z) - F(E(\underset{\substack{\uparrow \\ \text{constant}}}{D}(z))) \big]^2$$

Enforcing Invariance

Bad Example: Suppose for $x \in \mathcal{X}$: $E(x) = z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \in \mathbb{R}^d$

$$\widehat{y} = F(z) = F(z_1) \quad \text{and} \quad \widehat{x} = D(z) = D(z_2)$$

- Avoid by bottlenecking latent dimensionality d
- Add invariance loss to training objective:

$$\mathcal{L}_{\text{inv}} = \mathbb{E}_{z \sim p_Z}\Big[\underset{\substack{\uparrow \\ \text{constant}}}{F}(z) - F(E(\underset{\substack{\uparrow \\ \text{constant}}}{D}(z)))\Big]^2$$

- $\mathcal{L}_{\text{inv}} \to 0$ ensures outcome-predictions remain invariant to encoding-decoding variation

Jointly Learning Generative Model and Inference Maps

- Neural net parameters of F, q_E, p_D learned jointly

Jointly Learning Generative Model and Inference Maps

- Neural net parameters of F, q_E, p_D learned jointly

- Use stochastic gradient descent to minimize loss $\mathcal{L}$ over given data:

$$\mathcal{L}(x,y) = \mathcal{L}_{\text{rec}} + \lambda_{\text{pri}}\mathcal{L}_{\text{pri}} + \frac{\lambda_{\text{mse}}}{\sigma_Y^2}\mathcal{L}_{\text{mse}} + \frac{\lambda_{\text{inv}}}{\sigma_Y^2}\mathcal{L}_{\text{inv}}$$

$\mathcal{L}_{\text{rec}}(x) = -\mathbb{E}_{q_E(z|x)}\left[\log p_D(x \mid z)\right]$

$\mathcal{L}_{\text{pri}}(x) = \text{KL}(q_E(z \mid x) || \; p_Z)$

$\mathcal{L}_{\text{mse}}(x,y) = [y - F(E(x))]^2$

$\mathcal{L}_{\text{inv}} = \mathbb{E}_{z\sim p_Z}\big[F(z) - F(E(D(z)))\big]^2$

σ_Y^2 = (empirical) variance of outcomes

Jointly Learning Generative Model and Inference Maps

- Neural net parameters of F, q_E, p_D learned jointly
- Use stochastic gradient descent to minimize loss $\mathcal{L}$ over given data:

$$\mathcal{L}(x,y) = \mathcal{L}_{\text{rec}} + \lambda_{\text{pri}}\mathcal{L}_{\text{pri}} + \frac{\lambda_{\text{mse}}}{\sigma_Y^2}\mathcal{L}_{\text{mse}} + \frac{\lambda_{\text{inv}}}{\sigma_Y^2}\mathcal{L}_{\text{inv}}$$

$\mathcal{L}_{\text{rec}}(x) = -\mathbb{E}_{q_E(z|x)}\left[\log p_D(x \mid z)\right]$

$\mathcal{L}_{\text{pri}}(x) = \text{KL}(q_E(z \mid x) || \, p_Z)$

$\mathcal{L}_{\text{mse}}(x,y) = \left[y - F(E(x))\right]^2$

$\mathcal{L}_{\text{inv}} = \mathbb{E}_{z \sim p_Z}\left[F(z) - F(E(D(z)))\right]^2$

$\sigma_Y^2 =$ (empirical) variance of outcomes

- Start training with $\lambda_{\text{pri}} = \lambda_{\text{inv}} = 0$, slowly increase λ_{pri} and then λ_{inv}

Proposing Revisions

Revise Algorithm

Input: sequence $x_0 \in \mathcal{X}$, constant $\alpha \in (0, |2\pi\Sigma_{z|x_0}|^{-\frac{1}{2}})$

Output: revised sequence $x^* \in \mathcal{X}$

1) Use $\mathcal{E}$ to compute $q_E(z \mid x_0),\ E(x_0) = \mathbb{E}_{q_E}[z \mid x_0]$
2) Define $\mathcal{C}_{x_0} = \left\{z \in \mathbb{R}^d : q_E(z \mid x_0) \geqslant \alpha\right\}$ (ellipsoid)
3) Find $z^* \approx \underset{z \in \mathcal{C}_{x_0}}{\operatorname{argmax}}\ F(z)$ (gradient ascent w/ log-barrier penalty)
4) Return $x^* = D(z^*) \approx \underset{x \in \mathcal{X}}{\operatorname{argmax}}\, p_D(x \mid z^*)$ (greedy beam search)

Proposing Revisions

REVISE **Algorithm**

Input: sequence $x_0 \in \mathcal{X}$, constant $\alpha \in (0, |2\pi\Sigma_{z|x_0}|^{-\frac{1}{2}})$

Output: revised sequence $x^* \in \mathcal{X}$

1) Use $\mathcal{E}$ to compute $q_E(z \mid x_0),\ E(x_0) = \mathbb{E}_{q_E}[z \mid x_0]$
2) Define $\mathcal{C}_{x_0} = \left\{z \in \mathbb{R}^d : q_E(z \mid x_0) \geqslant \alpha\right\}$ (ellipsoid)
3) Find $z^* \approx \underset{z \in \mathcal{C}_{x_0}}{\operatorname{argmax}}\ F(z)$ (gradient ascent w/ log-barrier penalty)
4) Return $x^* = D(z^*) \approx \underset{x \in \mathcal{X}}{\operatorname{argmax}}\, p_D(x \mid z^*)$ (greedy beam search)

- We also propose alternative adaptive decoding biased toward x_0

Theoretical Results for $x^* = \textsc{Revise}(x_0)$

- If neural net approximations are exact, proposed revisions will satisfy:
  - x* associated with an expected outcome-increase

Theoretical Results for $x^* = \textsc{Revise}(x_0)$

- If neural net approximations are exact, proposed revisions will satisfy:
  - x^* associated with an expected outcome-increase
  - if x_0 appears natural (nontrivial likelihood under p_X), so does x^*

Theoretical Results for $x^* = \textsc{Revise}(x_0)$

- If neural net approximations are exact, proposed revisions will satisfy:
  - x^* associated with an expected outcome-increase
  - if x_0 appears natural (nontrivial likelihood under p_X), so does x^*
  - x^* and x_0 likely share similar latent characteristics Z

Theoretical Results for $x^* = \textsc{Revise}(x_0)$

- If neural net approximations are exact, proposed revisions will satisfy:
  - x^* associated with an expected outcome-increase
  - if x_0 appears natural (nontrivial likelihood under p_X), so does x^*
  - x^* and x_0 likely share similar latent characteristics Z
- We quantify proposed revisions' quality vs: accuracy in neural net approximations & marginal likelihood of x_0

Theoretical Results for $x^* = \textsc{Revise}(x_0)$

Theorem

With probability $\geqslant 1 - \delta$ *(over* $x_0 \sim p_X$*):*

$$p_X(x^*) \geqslant \frac{\alpha\gamma}{\eta} \cdot p_X(x_0)$$

Assuming with probability $\geqslant 1 - \delta$ (over $x \sim p_X$):

(A1) $p(z \mid x) \geqslant \gamma \cdot q_E(z \mid x)$ if $q_E(z \mid x) \geqslant \alpha$

(A2) $p(z^* \mid x^*) \leqslant \eta$ where $x^* = \textsc{Revise}(x)$

Theoretical Results for $x^* = \textsc{Revise}(x_0)$

Theorem

With probability $\geqslant 1 - \delta$ *(over* $x_0 \sim p_X$*):*

$$p_X(x^*) \geqslant \frac{\alpha\gamma}{\eta} \cdot p_X(x_0)$$

Assuming with probability $\geqslant 1 - \delta$ (over $x \sim p_X$):

(A1) $p(z \mid x) \geqslant \gamma \cdot q_E(z \mid x)$ if $q_E(z \mid x) \geqslant \alpha$

(A2) $p(z^* \mid x^*) \leqslant \eta$ where $x^* = \textsc{Revise}(x)$

- Replacing (A2) with Lipschitz condition on $p_D(x \mid z)$ $\implies$ similar result

Theoretical Results for $x^* = \textsc{Revise}(x_0)$

Theorem

With probability $\geqslant 1 - \delta - \kappa$ *(over* $x_0 \sim p_X$*):*

$$\Delta_{z^*} - \epsilon \leqslant F(z^*) - F(E(x_0)) \leqslant \Delta_{z^*} + \epsilon$$

where $\Delta_{z^*} = \mathbb{E}[Y \mid X = x^*] - \mathbb{E}[Y \mid X = x_0], \ \ \epsilon = \epsilon_{inv} + 2\epsilon_{mse}$

Assuming:

(A3) $p_X(x^*) \geqslant \kappa$ with probability $\geqslant 1 - \delta$ (over $x_0 \sim p_X$)

(A4) $|F(E(x)) - \mathbb{E}[Y|X = x]| \leqslant \epsilon_{\text{mse}}$ with probability $\geqslant 1 - \kappa$ (over $x \sim p_X$)

(A5) $|F(z) - F(E(D(z)))| \leqslant \epsilon_{\text{inv}}$ with probability $\geqslant 1 - \delta$ (over $z \sim p_Z$)

Theoretical Results for $x^* = \textsc{Revise}(x_0)$

Theorem

With probability $\geqslant 1 - \delta - \kappa$ *(over* $x_0 \sim p_X$*):*

$$\Delta_{z^*} - \epsilon \leqslant F(z^*) - F(E(x_0)) \leqslant \Delta_{z^*} + \epsilon$$

where $\Delta_{z^*} = \mathbb{E}[Y \mid X = x^*] - \mathbb{E}[Y \mid X = x_0], \;\; \epsilon = \epsilon_{inv} + 2\epsilon_{mse}$

Assuming:
(A3) $p_X(x^*) \geqslant \kappa$ with probability $\geqslant 1 - \delta$ (over $x_0 \sim p_X$)
(A4) $|F(E(x)) - \mathbb{E}[Y|X = x]| \leqslant \epsilon_{\text{mse}}$ with probability $\geqslant 1 - \kappa$ (over $x \sim p_X$)
(A5) $|F(z) - F(E(D(z)))| \leqslant \epsilon_{\text{inv}}$ with probability $\geqslant 1 - \delta$ (over $z \sim p_Z$)

- Previous theorem implies (A3)

Improving Sentence Positivity

- Data = 1M+ short sentences from BeerAdvocate reviews

Improving Sentence Positivity

- Data = 1M+ short sentences from BeerAdvocate reviews
- $y \in [0, 1]$: VADER sentiment compound score of each sentence[3]

[3] Hutto & Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *ICWSM*, 2014

Improving Sentence Positivity

- Data = 1M+ short sentences from BeerAdvocate reviews
- $y \in [0, 1]$: VADER sentiment compound score of each sentence[3]
- Apply methods to revise set of 1000 held-out sentences

[3] Hutto & Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *ICWSM*, 2014

Improving Sentence Positivity

Model	$\Delta_Y(x^*)$	$\Delta_L(x^*)$	$d(x^*, x_0)$
$\log\alpha = -10000$	**0.52** ±0.77	-8.8 ±6.5	2.6 ±3.3
$\log\alpha = -1$	0.31 ±0.50	**-7.6** ±5.8	1.7 ±2.6
$\lambda_{\text{inv}} = \lambda_{\text{pri}} = 0$	0.22 ±1.03	-10.2 ±7.0	3.3 ±3.4
SEARCH	0.19 ±0.56	-7.7 ±4.2	3.0 ±1.2

$\Delta_Y(x^*)$ = outcome improvement from revision (rescaled by std-dev of outcomes)

$\Delta_L(x^*) = \widehat{p}(x^*) - \widehat{p}(x_0)$

$d(x^*, x_0)$ = Levenshtein (edit) distance

Improving Sentence Positivity

Model	Sentence	$\Delta_Y(x^*)$	$\Delta_L(x^*)$
x_0	**this smells pretty bad.**	-	-
$\log\alpha = -10000$	smells pretty delightful!	+2.8	-0.5
$\log\alpha = -1$	i liked this smells pretty.	+2.5	-2.8
$\lambda_{\text{inv}} = \lambda_{\text{pri}} = 0$	pretty this smells bad!	-0.2	-3.1
SEARCH	wow this smells pretty bad.	+1.9	-4.6
x_0	**i like to support san diego beers.**	-	-
$\log\alpha = -10000$	i love to support craft beers!	+0.5	+1.6
$\log\alpha = -1$	i like to support craft beers!	+0.1	+2.6
$\lambda_{\text{inv}} = \lambda_{\text{pri}} = 0$	i like to support you know.	0	+3.7
SEARCH	i like to super support san diego.	+0.7	-2.9
x_0	**i'm not sure how old the bottle is.**	-	-
$\log\alpha = -10000$	i definitely enjoy how old is the bottle is.	+3.0	-3.6
$\log\alpha = -1$	i'm sure not sure how old the bottle is.	+2.5	-6.8
$\lambda_{\text{inv}} = \lambda_{\text{pri}} = 0$	i'm sure better is the highlights when cheers.	+3.3	-9.2
SEARCH	i 'm not sure how the bottle is love.	+2.3	-3.3

Revising Modern Text in the Language of Shakespeare

- Dataset of $\sim$100K short sentences

Revising Modern Text in the Language of Shakespeare

- Dataset of $\sim$100K short sentences
- Each is either from Shakespeare with label $y = 0.9$ or a more contemporary source (from NLTK) with label $y = 0.1$

Revising Modern Text in the Language of Shakespeare

- Dataset of $\sim$100K short sentences
- Each is either from Shakespeare with label $y = 0.9$ or a more contemporary source (from NLTK) with label $y = 0.1$
- Given new sentence, revise so that author is increasingly expected to be Shakespeare rather than contemporary source

Revising Modern Text in the Language of Shakespeare

# Steps	Decoded Sentence
x_0	**where are you, henry??**
100	where are you, henry??
1000	where are you, royal??
5000	where art thou now?
10000	which cannot come, you of thee?
x^*	where art thou, keeper??
x_0	**somewhere, somebody is bound to love us.**
100	somewhere, somebody is bound to love us.
1000	courage, honey, somebody is bound to love us!
5000	courage man; 'tis love that is lost to us.
10000	thou, within courage to brush and such us brush.
x^*	courage man; somebody is bound to love us.
x_0	**you are both the same size.**
100	you are both the same.
1000	you are both wretched.
5000	you are both the king.
10000	you are both these are very.
x^*	you are both wretched men.

Desiderata for our revision procedure

- Improves outcomes

Desiderata for our revision procedure

- Improves outcomes ✓

Desiderata for our revision procedure

- Improves outcomes ✓
- Produces natural sequences

Desiderata for our revision procedure

- Improves outcomes ✓
- Produces natural sequences ✓

Desiderata for our revision procedure

- Improves outcomes ✓
- Produces natural sequences ✓
- Preserves intrinsic similarity

Desiderata for our revision procedure

- Improves outcomes ✓
- Produces natural sequences ✓
- Preserves intrinsic similarity ✗

Desiderata for our revision procedure

- Improves outcomes ✓
- Produces natural sequences ✓
- Preserves intrinsic similarity ✗
- Computationally efficient

Desiderata for our revision procedure

- Improves outcomes ✓
- Produces natural sequences ✓
- Preserves intrinsic similarity ✗
- Computationally efficient ✓

Desiderata for our revision procedure

- Improves outcomes ✓
- Produces natural sequences ✓
- Preserves intrinsic similarity ✗
- Computationally efficient ✓

Ideas to improve method:

- Harness semantic similarity data to shape latent geometry[4]

[4] Mueller & Thyagarajan. Siamese Recurrent Architectures for Learning Sentence Similarity. *AAAI*, 2016

Desiderata for our revision procedure

- Improves outcomes ✓
- Produces natural sequences ✓
- Preserves intrinsic similarity ✗
- Computationally efficient ✓

Ideas to improve method:

- Harness semantic similarity data to shape latent geometry[4]
- Better generative model/prior[5] + variational inference strategy[6]

[4] Mueller & Thyagarajan. Siamese Recurrent Architectures for Learning Sentence Similarity. *AAAI*, 2016

[5] Yang et al. Improved Variational Autoencoders for Text Modeling using Dilated Convolutions. *ICML*, 2017

[6] Chen et al. Variational Lossy Autoencoder. *ICLR*, 2017