Learning about Language with Normalizing Flows

Graham Neubig
Language Technologies Institute, Carnegie Mellon University

Chunting Zhou  Junxian He

Di Wang, Xuezhe Ma, Daniel Spokoyny, Taylor Berg-Kirkpatrick
Learning about Language?
Learning about Language?

• Syntactic structure
Learning about Language?

• Syntactic structure

The cat sat on a green wall
Learning about Language?

- **Syntactic structure**

| The cat sat on a green wall |
| DT | NN | VBD | IN | DT | JJ | NN |

Parts-of-speech: **DT**  **NN**  **VBD**  **IN**  **DT**  **JJ**  **NN**
Learning about Language?

- Syntactic structure

```plaintext
The cat sat on a green wall

Parts-of-speech:  DT  NN  VBD  IN  DT  JJ  NN

Dependency:    →   →   →   →   →
```
Learning about Language?

- **Syntactic structure**

  The cat sat on a green wall

  Parts-of-speech: DT NN VBD IN DT JJ NN

  Dependency:

- **Cross-lingual correspondences**
Learning about Language?

- **Syntactic structure**

<table>
<thead>
<tr>
<th>English</th>
<th>Parts-of-speech:</th>
<th>Dependency:</th>
</tr>
</thead>
<tbody>
<tr>
<td>The cat sat on a green wall</td>
<td>DT NN VBD IN DT JJ NN</td>
<td></td>
</tr>
</tbody>
</table>

- **Cross-lingual correspondences**

<table>
<thead>
<tr>
<th>English</th>
<th>Japanese</th>
</tr>
</thead>
<tbody>
<tr>
<td>a cat green on sat the wall</td>
<td>のは上壁猫緑座った</td>
</tr>
</tbody>
</table>
Learning about Language?

- Syntactic structure
  
  The cat sat on a green wall

  Parts-of-speech: DT NN VBD IN DT JJ NN

  Dependency:

- Cross-lingual correspondences

  a cat green on sat the wall

  の は 上 壁 貓 緑 座った
Supervised Approaches

John passes the ball uphill to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.
Supervised Approaches

Supervised Learning

X \[ \theta \] Y

John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.

John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.

John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.

John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.

John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.
Supervised Approaches

Supervised Learning

\[ X \quad \theta \quad Y \]

John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.

John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.

John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.

John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.

John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.

John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.
Unsupervised Approaches

\[ X \]

---

Language Technologies Institute
Unsupervised Approaches

- Learning language models \( P(X) \)
Unsupervised Approaches

- Learning language models $P(X)$
- Learning continuous features from language models (e.g. word2vec, skipthought, BERT)
Unsupervised Approaches

- Learning language models $P(X)$
- Learning continuous features from language models (e.g. word2vec, skipthought, BERT)
- But how do we turn this into interpretable structure?
Unsupervised Approaches

- Learning language models $P(X)$
- Learning continuous features from language models (e.g. word2vec, skipthought, BERT)
- But how do we turn this into interpretable structure?
- How do we do it while taking advantage of continuous features?
Latent Variable Approaches

Unsupervised

X

θ

Y

X

Y
Latent Variable Approaches

Unsupervised

\[ X \rightarrow \theta \rightarrow Y \]

\[ ? \rightarrow X \]

\[ ? \rightarrow Y \]

\[ ? \rightarrow ? \]
Density Matching for Bilingual Word Embedding

Chunting Zhou, Xuezhe Ma, Di Wang, Graham Neubig
(NAACL 2019)
Bilingual Word Embedding
Bilingual Word Embedding

apple  pear  professor  school
dog  canine  piazza
cat  planet  earth
basketball
Bilingual Word Embedding
Bilingual Word Embedding

• Map word embeddings from different languages into a single vector space
Bilingual Word Embedding

- Map word embeddings from different languages into a single vector space
  - Cross-lingual transfer
Bilingual Word Embedding

- Map word embeddings from different languages into a single vector space
- Cross-lingual transfer
- Cross-lingual NLP tasks
Previous Work on Unsupervised BWE
Previous Work on Unsupervised BWE

• Unsupervised methods of minimization some form of distance between distributions of discrete vector sets:
Previous Work on Unsupervised BWE

• Unsupervised methods of minimization some form of distance between distributions of discrete vector sets:
Previous Work on Unsupervised BWE

• Unsupervised methods of minimization some form of distance between distributions of discrete vector sets:

  (A) X  cat feline  Y  profondo  auto gatto deep
  (B)  Y  WX
  (C)  Y  WX
  (D)  Y  WX  cat gatto

• No direct probabilistic interpretation, not a "typical" unsupervised generative model
Density Mapping for Bilingual Word Embedding (DeMa-BWE)
Density Mapping for Bilingual Word Embedding (DeMa-BWE)

Japanese Space

English Space

mapping function

dog

canine

cat

bird
Density Mapping for Bilingual Word Embedding (DeMa-BWE)

Japanese Space

English Space

• Mapping function is learned with normalizing flow
Normalizing Flows
Normalizing Flows

\[ X \sim P(X) \]
Normalizing Flows

\[ X \sim P(X) \]

\[ Z \sim N(0, I) \]
Normalizing Flows

\[ X \sim P(X) \]

\[ X = f_\theta^{-1}(Z) \]

\[ Z = f_\theta(X) \]

\[ Z \sim N(0, I) \]
Normalizing Flows

\[ X = f^{-1}_\theta(Z) \]
\[ Z = f_\theta(X) \]
\[ X \sim P(X) \]
\[ Z \sim N(0, I) \]

Change of variable formula:

\[
p_\theta(x) = p_Z(f_\theta(x)) \left| \det \left( \frac{\partial f_\theta(x)}{\partial x} \right) \right|
\]
Normalizing Flows

\[ X = f_{\theta}^{-1}(Z) \]
\[ Z = f_{\theta}(X) \]

\[ X \sim P(X) \]
\[ Z \sim N(0, I) \]

Change of variable formula:

\[ p_{\theta}(x) = p_{Z}(f_{\theta}(x)) \left| \det \left( \frac{\partial f_{\theta}(x)}{\partial x} \right) \right| \]

Intuitively, prevents degenerative mapping of everything to zero vector.
Normalizing Flows

\[ X \sim P(X) \]

\[ Z = f_\theta(X) \]

\[ X = f_\theta^{-1}(Z) \]

\[ Z \sim N(0, I) \]

**Change of variable formula:**

\[ p_\theta(x) = p_Z(f_\theta(x)) \left| \det \left( \frac{\partial f_\theta(x)}{\partial x} \right) \right| \]

Intuitively, prevents degenerative mapping of everything to zero vector

**Normalizing Flow:** A series of such invertible transformations \( f \)
DeMa-BWE: Preliminaries

Japanese Space

鸟
猫
犬

Mapping function

ing English Space

dog
canine

mapping function

cat
bird
DeMa-BWE: Preliminaries

Japanese Space

English Space

mapping function

dog

canine

cat

Notations:
DeMa-BWE: Preliminaries

Notations:

\( \mathbf{x} \in \mathbb{R}^d, \quad \mathbf{y} \in \mathbb{R}^d \) : denote vectors in the src and tgt embedding space
DeMa-BWE: Preliminaries

Notations:

\[ \mathbf{x} \in \mathbb{R}^d, \quad \mathbf{y} \in \mathbb{R}^d \] : denote vectors in the src and tgt embedding space

\[ x_i, \quad y_j \] : denote an actual word in src and tgt vocabularies
DeMa-BWE: Preliminaries

Notations:

\[ \mathbf{x} \in \mathbb{R}^d, \quad \mathbf{y} \in \mathbb{R}^d \]: denote vectors in the src and tgt embedding space

\[ x_i, \quad y_j \]: denote an actual word in src and tgt vocabularies

\[ f_{xy}, \quad f_{yx} \]: denote src->tgt, and tgt-src mapping functions
Prior Distribution

- Assumption on the monolingual word embedding space: Gaussian mixture model
Prior Distribution

- Assumption on the monolingual word embedding space: Gaussian mixture model

\[ p(x) = \sum_{i \in \{1, \ldots, N_x\}} \pi(x_i) \tilde{p}(x|x_i) \]
Prior Distribution

- Assumption on the monolingual word embedding space: Gaussian mixture model

\[
p(x) = \sum_{i \in \{1, \ldots, N_x\}} \pi(x_i)\tilde{p}(x|x_i)
\]

\[
\tilde{p}(x|x_i) = \mathcal{N}(x|x_i, \sigma_x^2 I)
\]
DeMa-BWE: Density Matching
DeMa-BWE: Density Matching

- Sampling a continuous vector from the GMM
DeMa-BWE: Density Matching

• Sampling a continuous vector from the GMM

\[ x_i \sim \pi(x_i) \quad x \sim \tilde{p}(x|\mathbf{x}_i) \]
DeMa-BWE: Density Matching

- Sampling a continuous vector from the GMM
  \[ x_i \sim \pi(x_i) \quad \mathbf{x} \sim \tilde{p}(\mathbf{x}|x_i) \]

- Apply the mapping function \( f_{xy} \) to obtain the transformed vector in the target space.
DeMa-BWE: Density Matching

- Sampling a continuous vector from the GMM
  \[ x_i \sim \pi(x_i) \quad x \sim \tilde{p}(x|x_i) \]

- Apply the mapping function \( f_{xy} \) to obtain the transformed vector in the target space.
  \[ f_{xy}(\cdot) = W_{xy} \]
DeMa-BWE: Density Matching

- Sampling a continuous vector from the GMM
  \[ x_i \sim \pi(x_i) \quad x \sim \tilde{p}(x|x_i) \]

- Apply the mapping function  \( f_{xy} \) to obtain the transformed vector in the target space.
  \[ f_{xy}(\cdot) = W_{xy} \]

- Computing the density of \( x \) in the mapped target space
DeMa-BWE: Density Matching

- Sampling a continuous vector from the GMM
  
  \[ x_i \sim \pi(x_i) \quad x \sim \tilde{p}(x|x_i) \]

- Apply the mapping function \( f_{xy} \) to obtain the transformed vector in the target space.
  
  \[ f_{xy}(\cdot) = W_{xy}. \]

- Computing the density of \( x \) in the mapped target space
  
  \[ \log p(x; W_{xy}) = \log p(y) + \log |\text{det}(W_{xy})| \]
DeMa-BWE: Density Matching

- Sampling a continuous vector from the GMM
  \[ x_i \sim \pi(x_i) \quad x \sim \tilde{p}(x|x_i) \]

- Apply the mapping function \( f_{xy} \) to obtain the transformed vector in the target space.
  \[ f_{xy}(\cdot) = W_{xy} \cdot \]

- Computing the density of \( x \) in the mapped target space
  \[ \log p(x; W_{xy}) = \log p(y) + \log |\text{det}(W_{xy})| \]

  minimize: \( \text{KL}(p(x) || p(x; W_{xy})) \)
DeMa-BWE: Density Matching

- Sampling a continuous vector from the GMM

\[ x_i \sim \pi(x_i) \quad \text{and} \quad x \sim \tilde{p}(x|x_i) \]

- Apply the mapping function \( f_{xy} \) to obtain the transformed vector in the target space.

\[ f_{xy}(\cdot) = W_{xy}. \]

- Computing the density of \( x \) in the mapped target space

\[ \log p(x; W_{xy}) = \log p(y) + \log |\det(W_{xy})| \]

- Objective: minimize: \( KL(p(x)||p(x; W_{xy})) \)
DeMa-BWE: Density Matching

- Sampling a continuous vector from the GMM
  \[ x_i \sim \pi(x_i) \quad \textit{and} \quad x \sim \tilde{p}(x | x_i) \]

- Apply the mapping function \( f_{xy} \) to obtain the transformed vector in the target space.
  \[ f_{xy}(\cdot) = W_{xy}. \]

- Computing the density of \( x \) in the mapped target space
  \[ \log p(x; W_{xy}) = \log p(y) + \log |\text{det}(W_{xy})| \]

- Objective: minimize: \( \text{KL}(p(x)||p(x; W_{xy})) \)
  \[ \mathcal{L}_{xy} = \mathbb{E}_{x \sim p(x)}[\log p(y) + \log |\text{det}(W_{xy})|] \]
Method Details
Method Details

- **Weak Orthogonality Constraint**: Try to make sure that the transformation is close to orthogonal
Method Details

- **Weak Orthogonality Constraint:** Try to make sure that the transformation is close to orthogonal

\[
\mathcal{L}_{bt} = \mathbb{E}_{x_i \sim \pi(x_i), x \sim \tilde{p}(x|x_i)} \left[ g(W_{yx} W_{xy} x, x) \right] + \mathbb{E}_{y_j \sim \pi(y_j), y \sim \tilde{p}(y|x_j)} \left[ g(W_{yx} W_{xy} y, y) \right]
\]
Method Details

• **Weak Orthogonality Constraint:** Try to make sure that the transformation is close to orthogonal

\[ L_{bt} = E_{x_i \sim \pi(x_i), x \sim \tilde{p}(x|x_i)} \left[ g(Wyx W_{xy}x, x) \right] + E_{y_j \sim \pi(y_j), y \sim \tilde{p}(y|x_j)} \left[ g(W_{xy} W_{yx}y, y) \right] \]

• **Weak Supervision w/ Identical Strings:** Take advantage of the fact that identical strings are usually the same word in both languages
Method Details

• **Weak Orthogonality Constraint:** Try to make sure that the transformation is close to orthogonal

\[
\mathcal{L}_{bt} = \mathbb{E}_{x_i \sim \pi(x_i), x \sim \tilde{p}(x|x_i)} \left[ g(W_{yx} W_{xy} x, x) \right] + \mathbb{E}_{y_j \sim \pi(y_j), y \sim \tilde{p}(y|x_j)} \left[ g(W_{xy} W_{yx} y, y) \right]
\]

• **Weak Supervision w/ Identical Strings:** Take advantage of the fact that identical strings are usually the same word in both languages

\[
\mathcal{L}_{sup} = \sum_{v \in \mathcal{V}'_{id}} g(v_x W_{xy}^T, v_y) + g(v_y W_{yx}^T, v_x)
\]
Method Details

- **Weak Orthogonality Constraint:** Try to make sure that the transformation is close to orthogonal

\[
\mathcal{L}_{bt} = E_{x_i \sim \pi(x_i), x \sim \tilde{p}(x|x_i)} [g(W_{yx} W_{xy} x, x)] + E_{y_j \sim \pi(y_j), y \sim \tilde{p}(y|x_j)} [g(W_{xy} W_{yx} y, y)]
\]

- **Weak Supervision w/ Identical Strings:** Take advantage of the fact that identical strings are usually the same word in both languages

\[
\mathcal{L}_{sup} = \sum_{v \in W_{id}} g(v_x W_{xy}^T, v_y) + g(v_y W_{yx}^T, v_x)
\]

- **Alignment Selection Methods:** Use cross-domain similarity local scaling (CSLS)
Method Details

- **Weak Orthogonality Constraint**: Try to make sure that the transformation is close to orthogonal

\[ \mathcal{L}_{bt} = E_{x_i \sim \pi(x_i), x \sim \tilde{p}(x|x_i)} \left[ g(W_{yx} W_{xy} x, x) \right] + E_{y_j \sim \pi(y_j), y \sim \tilde{p}(y|x_j)} \left[ g(W_{xy} W_{yx} y, y) \right] \]

- **Weak Supervision w/ Identical Strings**: Take advantage of the fact that identical strings are usually the same word in both languages

\[ \mathcal{L}_{sup} = \sum_{v \in W_{id}} g(v_x W_{xy}^T, v_y) + g(v_y W_{yx}^T, v_x) \]

- **Alignment Selection Methods**: Use cross-domain similarity local scaling (CSLS)

\[ \text{CSLS}(x', y) = 2 \cos(x', y) - r_T(x') - r_S(y) \]
Experiments

• Dataset and Tasks
  • Bilingual Lexicon Induction Task: MUSE dataset (Conneau et al., 2017)
  • Cross-lingual Word Similarity Task: SemEval 2017

• Languages
  • Baseline languages: en - es, de, fr, ru, zh, ja
  • Morphologically rich languages: en - et, fi, el, hu, pl, tr
Main Results on BLI (close languages)

- Procrustes(R)
- MUSE (U+R)
- SL-unsup-ID
- DeMa-BWE

Bar chart showing performance metrics for different language pairs (en-de, de-en, en-es, es-en) across various methods. Metrics range from 70 to 85.
Main Results on BLI (distant languages)

- Procrustes(R)
- MUSE (U+R)
- DeMa-BWE

<table>
<thead>
<tr>
<th>Language Pair</th>
<th>Procrustes(R)</th>
<th>MUSE (U+R)</th>
<th>DeMa-BWE</th>
</tr>
</thead>
<tbody>
<tr>
<td>en-et</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>et-en</td>
<td>16.25</td>
<td>48.75</td>
<td>65</td>
</tr>
<tr>
<td>en-el</td>
<td>32.5</td>
<td>48.75</td>
<td>65</td>
</tr>
<tr>
<td>el-en</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>en-ja</td>
<td>32.5</td>
<td>48.75</td>
<td>65</td>
</tr>
<tr>
<td>ja-en</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Unsupervised Learning of Syntactic Structure w/ Invertible Neural Projections

Junxian He, Graham Neubig, Taylor Berg-Kirkpatrick
(EMNLP 2018)
HMM for Part-of-Speech Induction

The cat sat
Gaussian HMM for POS Induction

\[ x_i \sim \mathcal{N}(\mu_{z_i}, \Sigma_{z_i}) \]

[Lin et al. 2015]
Gaussian HMM for POS Induction

\[ x_i \sim \mathcal{N}(\mu_{z_i}, \Sigma_{z_i}) \]

[Lin et al. 2015]
Latent Embeddings w/ Neural Projection

\[
\begin{align*}
&z_1 \rightarrow z_2 \rightarrow z_3 \\
&e_1 \rightarrow e_2 \rightarrow e_3 \\
&f_\phi(e) \\
x_1 \rightarrow x_2 \rightarrow x_3
\end{align*}
\]
Latent Embeddings w/ Neural Projection

\[ z_i \sim \text{Markov Structure} \]
Latent Embeddings w/ Neural Projection

\[ z_i \sim \text{Markov Structure} \]

\[ e_i \sim \mathcal{N}(\mu_{z_i}, \Sigma_{z_i}) \]
Latent Embeddings w/ Neural Projection

\[ z_i \sim \text{Markov Structure} \]

\[ e_i \sim \mathcal{N}(\mu_{z_i}, \Sigma_{z_i}) \]

Neural Projector

\[ x_i = f_\phi(e_i) \]

\[ x_i \sim \text{Point mass at } f_\phi(e_i) \]
Latent Embeddings w/ Neural Projection

\[ z_i \sim \text{Markov Structure} \]

\[ e_i \sim \mathcal{N}(\mu_{z_i}, \Sigma_{z_i}) \]

\[ f^{-1}(e; \phi) \]

Neural Projector

\[ x_i = f_{\phi}(e_i) \]

\[ x_i \sim \text{Point mass at } f_{\phi}(e_i) \]
Dependency Model with Valence

[The cat stopped a dog in Paris] 

[Klein and Manning 2004]
Dependency Model with Valence

[Klein and Manning 2004]
Dependency Parse Induction from POS
Grammar Induction from Raw Text
Grammar Induction from Raw Text

The cat stopped a dog in Paris
Latent Embeddings w/ Neural Projection
Latent Embeddings w/ Neural Projection

\[ z_1 \xrightarrow{} e_1 \xrightarrow{} x_1 \]
\[ z_2 \xrightarrow{} e_2 \xrightarrow{} x_2 \]
\[ z_3 \xrightarrow{} e_3 \xrightarrow{} x_3 \]

\[ f_\phi(e) \]
Latent Embeddings w/ Neural Projection

\[ z_i \sim \text{Syntax Model} \]

\[ e_i \sim \mathcal{N}(\mu_{z_i}, \Sigma_{z_i}) \]

Neural Projector

\[ x_i = f_\phi(e_i) \]

Point mass at \( f_\phi(e_i) \)
Learning and Inference

\[ p(\mathbf{x}_i | z_i; \eta, \phi) \]
Learning and Inference

\[ p(x_i | z_i; \eta, \phi) \]

Gaussian embedding parameters
Learning and Inference

\[ p(x_i | z_i; \eta, \phi) \]

Projection parameters
Learning and Inference

\[ \text{dim}(x) = \text{dim}(e) \quad \text{and} \quad f \quad \text{is invertible} \]

\[ p(x_i | z_i; \eta, \phi) \]

\[ f^{-1}(e; \phi) \]
Learning and Inference

\[
\text{dim}(x) = \text{dim}(e) \text{ and } f \text{ is invertible}
\]

\[
p(x_i | z_i; \eta, \phi)
\]

\[
= p(f^{-1}_\phi(x_i) | z_i; \eta) \left| \det \frac{\partial f^{-1}}{\partial x_i} \right|
\]
Learning and Inference

\[ \text{dim}(x) = \text{dim}(e) \text{ and } f \text{ is invertible} \]

\[
p(x_i|z_i; \eta, \phi) = p(f^{-1}_\phi(x_i)|z_i; \eta) \left| \det \frac{\partial f^{-1}}{\partial x_i} \right|
\]

Determinant of Jacobian matrix
Learning and Inference

\begin{equation}
\text{dim}(x) = \text{dim}(e) \text{ and } f \text{ is invertible}
\end{equation}

\begin{align*}
p(x_i | z_i; \eta, \phi) &= p(f_{\phi}^{-1}(x_i) | z_i; \eta) \det \frac{\partial f_{\phi}^{-1}}{\partial x_i} \\
\text{Gaussian distribution} &\quad \text{Determinant of Jacobian matrix}
\end{align*}
Learning and Inference
Learning and Inference

\[
p(f_\phi^{-1}(x_i)|z_i; \eta) \left| \det \frac{\partial f^{-1}}{\partial x_i} \right|
\]
Learning and Inference

Example of Markov prior

$$\log p(x) = \log p_{\text{GHMM}}(f_\phi^{-1}(x))$$
Learning and Inference

Example of Markov prior

$$\log p(x) = \log p_{\text{GHMM}}(f^{-1}_\phi(x))$$
Learning and Inference

Example of Markov prior

\[
\log p(x) = \log p_{\text{GHMM}}(f^{-1}_\phi(x)) + \sum \log \left| \det \frac{\partial f^{-1}_\phi}{\partial x_i} \right|
\]
Example of Markov prior

\[
\log p(x) = \log p_{\text{GHMM}}(f^{-1}_\phi(x)) + \sum \log \left| \det \frac{\partial f^{-1}_\phi}{\partial x} \right|
\]

\(-\infty\) when \( f \) is not invertible
Learning with Inverse Projection

\[ f_{\phi}^{-1}(x_i) \]

\[ e_i \]

\[ x_i \]
Learning with Inverse Projection

\[ h_{i,l}^{(1)} = x_{i,l} \]

\[ h_{i,r}^{(1)} = x_{i,r} + g(x_{i,l}) \]

[Dinh et al. 2014]
Learning with Inverse Projection

\[ h_{i,l}^{(1)} = x_{i,l} \]

\[ h_{i,r}^{(1)} = x_{i,r} + g(x_{i,l}) \]

\[
\begin{pmatrix}
1 & 1 & 0 \\
1 & 1 & 1 \\
\vdots & 1 & 1 \\
1 & 1 & 1
\end{pmatrix}
\]

[Dinh et al. 2014]
Learning with Inverse Projection

\[ h_{i,l}^{(1)} = x_{i,l} \]

\[ h_{i,r}^{(1)} = x_{i,r} + g(x_{i,l}) \]

\[
\begin{pmatrix}
1 & 1 & 0 \\
1 & 1 & 1 \\
\vdots & 1 & 1 \\
1 & 1 & 1
\end{pmatrix}
\]

[Dinh et al. 2014]
Experiments

- Dataset: English Penn Treebank
- POS tagging
  Trained and tested on whole PTB
- Grammar induction
  Trained on sentences of length <= 10 in section 2-21
  Tested on sentences in section 23
Part-of-speech Induction

- Feature-based: 80.2%
- Neural: 80.8%
- Gaussian: 75.4%
- HM: 62.7%

- Feature-based: 72.1%
- Neural: 74.1%
- Gaussian: 68.5%
- HM: 53.8%
Part-of-speech Induction

Outperform feature-based SOTA
Dependency Parse Induction

- Directed % on len ≤ 10
  - Oracle POS
  - DM: 49.6
  - Gaussi: 55.4
  - Neural: 60.2

- Directed % on all
  - Oracle POS
  - DM: 35.8
  - Gaussi: 43.1
  - Neural: 47.9
Original Embedding Space

- adjective
- adverb
- noun singular
- noun proper
- noun plural
- verb base
- verb gerund
- verb past tense
- verb past participle
- verb 3rd singular
- cardinal number
Projected Embedding Space w/ Markov Prior

\[ z_i \rightarrow e_i \rightarrow x_i \]

\[ f^{-1}(e; \phi) \]
Projected Embedding Space w/ Markov Prior

\[ z_i \rightarrow e_i \rightarrow x_i \]

\[ f^{-1}(e; \phi) \]
Projected Embedding Space w/ DMV Prior

- adjective
- adverb
- noun singular
- noun proper
- noun plural
- verb base
- verb past tense
- verb past participle
- verb 3rd singular
- cardinal number
Projected Embedding Space w/ DMV Prior

- adjectival
- adverb
- noun singular
- noun proper
- noun plural
- verb base
- verb gerund
- verb past tense
- verb past participle
- verb 3rd singular
- cardinal number
Projected Embedding Space w/ DMV Prior

- adjective
- adverb
- noun singular
- noun proper
- noun plural
- verb base
- verb gerund
- verb past tense
- verb past participle
- verb 3rd singular
- cardinal number
Projected Embedding Space w/ DMV Prior

- adjectival
- adverb
- noun singular
- noun proper
- noun plural
- verb base
- verb gerund
- verb past tense
- verb past participle
- verb 3rd singular
- cardinal number
Projected Embedding Space w/ DMV Prior

- adjectival
- adverb
- noun singular
- noun proper
- noun plural
- verb base
- verb gerund
- verb past tense
- verb past participle
- verb 3rd singular
- cardinal number
Conclusion
Conclusion
Conclusion

• Normalizing flows for unsupervised learning

\[ X = f^{-1}_\theta(Z) \]
\[ Z = f_\theta(X) \]
Conclusion

- Normalizing flows for unsupervised learning

\[ X = f^{-1}_\theta(Z) \]
\[ Z = f_\theta(X) \]

- Learning of bilingual lexicons
Conclusion

- Normalizing flows for unsupervised learning

\[ X = f^{-1}_\theta(Z) \]
\[ Z = f_\theta(X) \]

- Learning of bilingual lexicons

- Learning of syntactic structure

The cat sat on a green wall
Thank You! Questions?

DeMa-BWE

https://github.com/violet-zct/DeMa-BWE

The cat sat on a green wall

https://github.com/jxhe/struct-learning-with-flow