Learning with Sparse Latent Structure

Vlad Niculae
Instituto de Telecomunicações

Work with: André Martins, Claire Cardie, Mathieu Blondel

github.com/vene/sparsemap  @vnfrombucharest
Structured Inference

...
Structured Inference

\[
\begin{align*}
\text{dog} & \text{ on } \text{ wheels} \\
\text{dog} & \text{ on } \text{ wheels} \\
\text{dog} & \text{ on } \text{ wheels}
\end{align*}
\]
Structured Inference

...
Latent Structured Inference

input

output

positive
neutral
negative
How to select an item from a set?
How to select an item from a set?
How to select an item from a set?

$c_1$

$c_2$

$\ldots$

$c_N$
How to select an item from a set?

\[
\theta \\
\quad \ C_1 \\
\quad \ C_2 \\
\quad \ldots \\
\quad \ C_N
\]
How to select an item from a set?

\[ \theta \]

\[
\begin{align*}
\theta & = f_1(\theta; \omega) \\
\theta & = f_2(p, \theta; \omega)
\end{align*}
\]

\[ \frac{\partial y}{\partial w} = ? \]

or, essentially, \[ \frac{\partial p}{\partial \theta} = ? \]
How to select an item from a set?

\[ \theta = f_1(x; w) \]

\[ y = f_2(p, x; w) \]
How to select an item from a set?

\[ \theta = f_1(x; w) \]

\[ y = f_2(p, x; w) \]

\[ \frac{\partial y}{\partial w} = ? \]
How to select an item from a set?

\[ \mathbf{\theta} = f_1(x; w) \]

\[ y = f_2(p, x; w) \]

\[ \frac{\partial y}{\partial w} = ? \quad \text{or, essentially,} \quad \frac{\partial p}{\partial \theta} = ? \]
Argmax

\[ \theta \]

\[ c_1 \]
\[ c_2 \]
\[ \ldots \]
\[ c_N \]

\[ p \]

\[ \frac{\partial p}{\partial \theta} = ? \]
\[ \frac{\partial p}{\partial \theta} = ? \]
$\text{Argmax}$

\[
\begin{align*}
\theta & \quad c_1 & \quad \ldots & \quad c_N \\
\theta & \quad c_2 & \quad \ldots & \quad c_N \\
\theta & \quad \ldots & \quad \ldots & \quad \ldots \\
\theta & \quad c_N & \quad \ldots & \quad \ldots \\
\end{align*}
\]

\[
\frac{\partial p}{\partial \theta} = ?
\]
Argmax

\[
\theta = \text{Argmax} \quad p
\]

\[
\frac{\partial p}{\partial \theta} = ?
\]
Argmax

\[ \theta \]

\[ c_1 \]

\[ c_2 \]

\[ \ldots \]

\[ c_N \]

\[ p \]

\[ \frac{\partial p}{\partial \theta} = ? \]
\[
\text{Argmax} \quad \theta \quad p
\]

\[
\begin{align*}
\partial p \\
\partial \theta = ?
\end{align*}
\]

\[
\theta \\
\begin{align*}
c_1 \\
c_2 \\
\ldots \ldots \\
c_N
\end{align*}
\]

\[
p \\
\begin{align*}
\quad \\
\quad \\
\quad \\
\quad \\
\quad
\end{align*}
\]
$$\arg\max_{\theta} \, c_1, c_2, \ldots, c_N$$

$$\frac{\partial p}{\partial \theta} = ?$$
Argmax

\[
\frac{\partial p}{\partial \theta} = ?
\]
\[
\text{Argmax} \\
\frac{\partial p}{\partial \theta} = 0
\]
Argmax vs. Softmax

\[ p_j = \frac{\exp(\theta_j)}{Z} \]

\[
\frac{\partial p}{\partial \theta} = \text{diag}(p) - pp^\top
\]
Variational Form of Argmax

\[ \Delta = \{ p \in \mathbb{R}^N : p \geq 0, \ 1^T p = 1 \} \]
Variational Form of Argmax

$$\Delta = \{ p \in \mathbb{R}^N : p \geq 0, \ 1^\top p = 1 \}$$
Variational Form of Argmax

\[ \Delta = \{ p \in \mathbb{R}^N : p \geq 0, \ 1^\top p = 1 \} \]
Variational Form of Argmax

$$\Delta = \{ p \in \mathbb{R}^N : p \geq 0, \ 1^\top p = 1 \}$$

- $N = 2$
- $p = [0, 1]$
- $p = [1, 0]$
Variational Form of Argmax

\[ \Delta = \{ p \in \mathbb{R}^N : p \geq 0, \ 1^\top p = 1 \} \]

\[ p = [0, 1] \]
\[ p = [1/2, 1/2] \]
\[ p = [1, 0] \]

\[ N = 2 \]
Variational Form of Argmax

\[ \Delta = \{ p \in \mathbb{R}^N : p \geq 0, \ 1^\top p = 1 \} \]
Variational Form of Argmax

\[ \Delta = \{ p \in \mathbb{R}^N : p \geq 0, \ 1^\top p = 1 \} \]
Variational Form of Argmax

\[ \Delta = \{ p \in \mathbb{R}^N : p \geq 0, \ 1^\top p = 1 \} \]
Variational Form of Argmax

$$\Delta = \{ p \in \mathbb{R}^N : p \geq 0, \ 1^T p = 1 \}$$

For $N = 2$:
- $p = [0, 1]$
- $p = [\frac{1}{2}, \frac{1}{2}]$
- $p = [1, 0]$

For $N = 3$:
- $p = [0, 0, 1]$
- $p = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$
- $p = [0, 1, 0]$
Variational Form of Argmax

$$\max_j \theta_j = \max_{p \in \Delta} p^\top \theta$$

Fundamental Thm. Lin. Prog. (Dantzig et al, 55)

$$N = 2$$

$$N = 3$$
Variational Form of Argmax

\[
\max_j \theta_j = \max_{p \in \Delta} p^\top \theta
\]

Fundamental Thm. Lin. Prog. (Dantzig et al, 55)

\[\theta = [0.2, 1.4]\]

\[N = 2\]

\[N = 3\]
Variational Form of Argmax

\[
\max_j \theta_j = \max_{p \in \Delta} p^\top \theta
\]

Fundamental Thm. Lin. Prog. (Dantzig et al, 55)

\[ \theta = [0.2, 1.4] \]

\[ N = 2 \]

\[ N = 3 \]
Variational Form of Argmax

\[
\max_j \theta_j = \max_{p \in \Delta} p^\top \theta
\]

Fundamental Thm. Lin. Prog. (Dantzig et al, 55)

\[\theta = [0.2, 1.4]\]
\[p^* = [0, 1]\]

\[N = 2\]
\[N = 3\]
Variational Form of Argmax

$$\max_j \theta_j = \max_{\mathbf{p} \in \Delta} \mathbf{p}^\top \mathbf{\theta}$$

Fundamental Thm. Lin. Prog. (Dantzig et al, 55)
Variational Form of Argmax

\[ \max_j \theta_j = \max_{p \in \Delta} p^\top \theta \]

Fundamental Thm. Lin. Prog. (Dantzig et al, 55)

\[ \theta = \begin{bmatrix} .2, 1.4 \end{bmatrix} \]

\[ p^* = \begin{bmatrix} 0, 1 \end{bmatrix} \]

\[ N = 2 \]

\[ \theta = \begin{bmatrix} .7, .1, 1.5 \end{bmatrix} \]

\[ N = 3 \]
Variational Form of Argmax

\[ \max_j \theta_j = \max_{p \in \Delta} p^\top \theta \]

Fundamental Thm. Lin. Prog.  
(Dantzig et al, 55)
Smoothed Max Operators

\[
\max_{\Omega}(\theta) = \max_{p \in \Delta} p^\top \theta - \Omega(p)
\]
Smoothed Max Operators

\[
\max_\Omega (\theta) = \max_{p \in \Delta} \quad p^\top \theta - \Omega(p)
\]

- \text{argmax: } \Omega(p) = 0

(Niculae & Blondel, 17)
Smoothed Max Operators

\[
\max_{\Omega}(\theta) = \max_{p \in \Delta} p^\top \theta - \Omega(p)
\]

- argmax: \( \Omega(p) = 0 \)
- softmax: \( \Omega(p) = \sum_j p_j \log p_j \)

(Niculae & Blondel, 17)
Smoothed Max Operators

$$\max_{\Omega} (\theta) = \max_{p \in \Delta} p^\top \theta - \Omega(p)$$

- **argmax**: $\Omega(p) = 0$
- **softmax**: $\Omega(p) = \sum_j p_j \log p_j$
- **sparsemax**: $\Omega(p) = \frac{1}{2} \| p \|_2^2$

(Martins & Astudillo, 16)
fusedmax ?!
Smoothed Max Operators

\[
\max_{\Omega} (\theta) = \max_{p \in \Delta} p^\top \theta - \Omega(p)
\]

- argmax: \( \Omega(p) = 0 \)
- softmax: \( \Omega(p) = \sum_j p_j \log p_j \)
- sparsemax: \( \Omega(p) = \frac{1}{2} \| p \|_2^2 \)

(Niculae & Blondel, 17)
Smoothed Max Operators

$$\max_{\Omega} \left( \theta \right) = \max_{p \in \Delta} p^\top \theta - \Omega(p)$$

- argmax: \( \Omega(p) = 0 \)
- softmax: \( \Omega(p) = \sum_j p_j \log p_j \)
- sparsemax: \( \Omega(p) = \frac{1}{2} \| p \|_2^2 \)
- fusedmax: \( \Omega(p) = \frac{1}{2} \| p \|_2^2 + \sum_j |p_j - p_{j-1}| \)
- oscarmax: \( \Omega(p) = \frac{1}{2} \| p \|_2^2 + \sum_{i,j} \max(p_i, p_j) \)
Structured Inference

finally
Structured Inference is essentially a (very high-dimensional) argmax

There are exponentially many structures ($\theta$ cannot be in memory!)
Structured Inference is essentially a (very high-dimensional) argmax.
Structured Inference is essentially a (very high-dimensional) argmax.

There are exponentially many structures \((\theta \text{ cannot fit in memory!})\)

input \(x\)  \(\theta\)  \(p\)  output \(y\)
Factorization Into Parts

\[ \theta = A^\top \eta \]
Factorization Into Parts

\[ \theta = A^\top \eta \]

\[
\begin{array}{c|ccc}
\ast \rightarrow \text{dog} & 1 & 0 & 0 \\
\text{on} \rightarrow \text{dog} & 0 & 1 & 1 \\
\text{wheels} \rightarrow \text{dog} & 0 & 0 & 0 \\
\hline
\ast \rightarrow \text{on} & 0 & 1 & 1 \\
\text{dog} \rightarrow \text{on} & 1 & \ldots & 0 & 0 & \ldots \\
\text{wheels} \rightarrow \text{on} & 0 & 0 & 0 \\
\hline
\ast \rightarrow \text{wheels} & 0 & 0 & 0 \\
\text{dog} \rightarrow \text{wheels} & 0 & 1 & 0 \\
\text{on} \rightarrow \text{wheels} & 1 & 0 & 1 \\
\end{array}
\]

\[
\begin{array}{c}
\eta = \begin{bmatrix}
.1 \\
.2 \\
-.1 \\
.3 \\
.8 \\
.1 \\
-.3 \\
.2 \\
-.1
\end{bmatrix}
\end{array}
\]
Factorization Into Parts

\[ \theta = A^\top \eta \]

\[
\begin{array}{c c c c c c}
\star \rightarrow \text{dog} & 1 & 0 & 0 & 0 \\
\text{on} \rightarrow \text{dog} & 0 & 1 & 1 & 1 \\
\text{wheels} \rightarrow \text{dog} & 0 & 0 & 0 & 0 \\
\hline
\star \rightarrow \text{on} & 0 & 1 & 1 & 1 \\
\text{dog} \rightarrow \text{on} & 1 & \ldots & 0 & 0 & \ldots \\
\text{wheels} \rightarrow \text{on} & 0 & 0 & 0 & 0 \\
\hline
\star \rightarrow \text{wheels} & 0 & 0 & 0 & 0 \\
\text{dog} \rightarrow \text{wheels} & 0 & 1 & 0 & \ldots \\
\text{on} \rightarrow \text{wheels} & 1 & 0 & 1 & \\
\end{array}
\]

\[
\begin{array}{c c c c}
\eta = \begin{bmatrix}
.1 \\
.2 \\
-.1 \\
.3 \\
.8 \\
.1 \\
-.3 \\
.2 \\
-.1
\end{bmatrix}
\end{array}
\]

\[
\begin{array}{c c c c c c}
\text{dog} \rightarrow \text{hond} & 1 & 0 & 0 \\
\text{dog} \rightarrow \text{op} & 0 & 1 & 1 \\
\text{dog} \rightarrow \text{wielen} & 0 & 0 & 0 \\
\hline
\text{on} \rightarrow \text{hond} & 0 & 0 & 0 \\
\text{on} \rightarrow \text{op} & 1 & \ldots & 0 & 0 & \ldots \\
\text{on} \rightarrow \text{wielen} & 0 & 0 & 0 & 0 \\
\hline
\text{wheels} \rightarrow \text{hond} & 0 & 1 & 0 \\
\text{wheels} \rightarrow \text{op} & 0 & 0 & 0 \\
\text{wheels} \rightarrow \text{wielen} & 1 & 0 & 1 & \\
\end{array}
\]

\[
\begin{array}{c c c c c}
\eta = \begin{bmatrix}
.1 \\
.2 \\
-.1 \\
.3 \\
.8 \\
.1 \\
-.3 \\
.2 \\
-.1
\end{bmatrix}
\end{array}
\]
\[ \arg\max p_{2}\Delta p^{\top}\theta \]

\[ \text{sparsemax} \]

\[ \arg\max p_{2}\Delta p^{\top}\theta - \frac{1}{2}\|p\|_{2}^{2} \]

\[ \text{MAP} \]

\[ \arg\max \mu_{2}M\mu^{\top}\eta + e^{H(\mu)} \]

\[ \text{SparseMAP} \]

\[ \arg\max \mu_{2}M\mu^{\top}\eta - \frac{1}{2}\|\mu\|_{2}^{2} \]

\[ \text{e.g. dependencyparsing} \]

\[ \text{the Hungarian algorithm} \]

\[ \text{e.g. dependencyparsing} \]

\[ \text{the Matrix-Tree theorem} \]

\[ \#P\text{-complete! (Valiant, 79)} \]
arg max $p^\top \theta$ so

sparse max $p^\top \theta - \frac{1}{2} \|p\|^2$

MAP $\arg \max \mu^\top \eta$ marginals

SparseMAP $\arg \max \mu^\top \eta - \frac{1}{2} \|\mu\|^2$

e.g. dependence parsing!

matching!

the Hungarian algorithm
e.g. dependence parsing!

the Matrix-Tree theorem

#P-complete!

(Valiant, 79)
$\text{argmax } \arg \max_{p \in \Delta} p^T \theta$

$\text{SparseMAP } \arg \max_{\mu \in M} \mu^T \eta - \frac{1}{2} \| \mu \|_2^2$

$\text{Sparsemax } \arg \max_{p \in \Delta} p^T \theta + H(p)$
\[ \text{argmax}_{p \in \Delta} \arg \max \ p^T \theta \]

\[ \text{MAP}_{\mu \in \mathcal{M}} \arg \max \mu^T \eta \]
\[ \text{argmax} \arg \max_{p \in \Delta} p^T \theta \]
\[ \text{MAP} \arg \max_{\mu \in \mathcal{M}} \mu^T \eta \]

\text{e.g. dependency parsing} \rightarrow \text{max. spanning tree}
\text{matching} \rightarrow \text{the Hungarian algorithm}
### Argmax

\[
\text{argmax } \arg \max_{\mathbf{p} \in \Delta} \mathbf{p}^T \mathbf{\theta}
\]

### Softmax

\[
\text{softmax } \arg \max_{\mathbf{p} \in \Delta} \mathbf{p}^T \mathbf{\theta} + H(\mathbf{p})
\]

### MAP

\[
\text{MAP } \arg \max_{\mu \in \mathcal{M}} \mu^T \eta
\]
argmax $\arg \max_{p \in \Delta} p^T \theta$

softmax $\arg \max_{p \in \Delta} p^T \theta + H(p)$

MAP $\arg \max_{\mu \in \mathcal{M}} \mu^T \eta$

marginals $\arg \max_{\mu \in \mathcal{M}} \mu^T \eta + \tilde{H}(\mu)$

e.g. dependency parsing

max. spanning tree

the Hungarian algorithm

e.g. the Matrix-Tree theorem

#P-complete

(Valiant, 79)
\[
\text{argmax } \arg \max_{p \in \Delta} p^T \theta \\
\text{softmax } \arg \max_{p \in \Delta} p^T \theta + H(p)
\]

\[
\text{MAP } \arg \max_{\mu \in \mathcal{M}} \mu^T \eta \\
\text{marginals } \arg \max_{\mu \in \mathcal{M}} \mu^T \eta + \tilde{H}(\mu)
\]

**e.g. dependency parsing → the Matrix-Tree theorem**

**matching → \#P-complete!** (Valiant, 79)
- **argmax** \( \operatorname{arg \, max} \ p^\top \theta \) \( p \in \Delta \)
- **softmax** \( \operatorname{arg \, max} \ p^\top \theta + H(p) \) \( p \in \Delta \)
- **sparsemax** \( \operatorname{arg \, max} \ p^\top \theta - \frac{1}{2} \| p \|^2 \) \( p \in \Delta \)

- **MAP** \( \operatorname{arg \, max} \ \mu^\top \eta \) \( \mu \in M \)
- **marginals** \( \operatorname{arg \, max} \ \mu^\top \eta + \tilde{H}(\mu) \) \( \mu \in M \)

- *Examples*: dependency parsing (e.g. max. spanning tree matching) and the Hungarian algorithm.

- *Theorems*: the Matrix-Tree theorem and #P-completeness (Valiant, 79).
argmax \ \arg\ \max_{p \in \Delta} \ p^T \theta

softmax \ \arg\ \max_{p \in \Delta} \ p^T \theta + H(p)

sparsemax \ \arg\ \max_{p \in \Delta} \ p^T \theta - \frac{1}{2}\|p\|^2

MAP \ \arg\ \max_{\mu \in M} \ \mu^T \eta

marginals \ \arg\ \max_{\mu \in M} \ \mu^T \eta + \tilde{H}(\mu)

SparseMAP \ \arg\ \max_{\mu \in M} \ \mu^T \eta - \frac{1}{2}\|\mu\|^2

(Niculae, Martins, Blondel, Cardie, 18)
SparseMAP Inference Solution

\[ \mu^* = \arg \max_{\mu \in M} \mu^T \eta - \frac{1}{2} \| \mu \|^2 \]

\[ = 0.6 \cdot 0 + 0.4 \cdot 0 \]

\[ = A \rho^* \text{ with very sparse } \rho^* \in \Delta^N \]
Algorithms for SparseMAP

\[ \mu^* = \arg \max_{\mu \in \mathcal{M}} \mu^T \eta - \frac{1}{2} \| \mu \|^2 \]
Algorithms for SparseMAP

\[ \mu^* = \arg \max_{\mu \in \mathcal{M}} \mu^T \eta - \frac{1}{2} \| \mu \|^2 \]

Greedy Conditional Gradient (Frank-Wolfe) algorithms
Algorithms for Sparse MAP

$$\mu^* = \arg \max_{\mu \in \mathcal{M}} \mu^T \eta - \frac{1}{2} \| \mu \|^2$$

Greedy Conditional Gradient (Frank-Wolfe) algorithms

- select a new corner of $\mathcal{M}$
Algorithms for SparseMAP

\[ \mu^* = \arg \max_{\mu \in \mathcal{M}} \mu^T \eta - \frac{1}{2} \| \mu \|^2 \]

Greedy Conditional Gradient (Frank-Wolfe) algorithms

- select a new corner of \( \mathcal{M} \)
- update the (sparse) coefficients of \( p \)
**Algorithms for SparseMAP**

\[ \mu^* = \arg \max_{\mu \in \mathcal{M}} \mu^T \eta - \frac{1}{2} \| \mu \|^2 \]

**Greedy Conditional Gradient (Frank-Wolfe) algorithms**

- select a new corner of \( \mathcal{M} \)
- update the (sparse) coefficients of \( p \)
  - Update rules: vanilla, away-step, pairwise
Algorithms for SparseMAP

$$\mathbf{\mu}^* = \arg \max_{\mathbf{\mu} \in \mathcal{M}} \mathbf{\mu}^\top \mathbf{\eta} - \frac{1}{2}\|\mathbf{\mu}\|^2$$

Greedy Conditional Gradient (Frank-Wolfe) algorithms

- select a new corner of $\mathcal{M}$
- update the (sparse) coefficients of $p$
  - Update rules: vanilla, away-step, pairwise
  - Quadratic objective: Active Set (Min-Norm Point)
Algorithms for SparseMAP

\[ \mu^* = \arg \max_{\mu \in \mathcal{M}} \mu^T \eta - \frac{1}{2} \| \mu \|^2 \]

Greedy Conditional Gradient (Frank-Wolfe)

- select a new corner
- update the (sparse) coefficients
- update rules: vanilla, away-step, pairwise
- Quadratic objective:
  Active Set (Min-Norm Point)

Active Set achieves finite & linear convergence!
Algorithms for SparseMAP

\[ \mu^* = \arg \max_{\mu \in \mathcal{M}} \mu^T \eta - \frac{1}{2} \|\mu\|^2 \]

Greedy Conditional Gradient (Frank-Wolfe) algorithms

- select a new corner of \( \mathcal{M} \)
- update the (sparse) coefficients of \( p \)
  - Update rules: vanilla, away-step, pairwise
  - Quadratic objective: \textbf{Active Set} (Min-Norm Point)

Backward pass

\[ \frac{\partial \mu}{\partial \eta} \] is sparse
Algorithms for Sparse MAP

\[ \mu^* = \arg \max_{\mu \in \mathcal{M}} \mu^\top \eta - \frac{1}{2} \| \mu \|^2 \]

Greedy Conditional Gradient (Frank-Wolfe) algorithms

- select a new corner of \( \mathcal{M} \)
- update the (sparse) coefficients of \( p \)
  - Update rules: vanilla, away-step, pairwise
  - Quadratic objective: **Active Set** (Min-Norm Point)

Backward pass

\[ \frac{\partial \mu}{\partial \eta} \text{ is sparse} \]

computing \( \left( \frac{\partial \mu}{\partial \eta} \right)^\top dy \) takes \( O(\dim(\mu) \nnz(p^*)) \)
Algorithms for SparseMAP

\[ \mu^* = \arg \max_{\mu \in \mathcal{M}} \mu^T \eta - \frac{1}{2} \|\mu\|^2 \]

- **Greedy Condiional Gradient (Frank-Wolfe)** algorithms
  - select a new corner of \( \mathcal{M} \)
  - update the (sparse) coefficients of \( p \)
    - Update rules: vanilla, away-step, pairwise
    - Quadratic objective: **Active Set** (Min-Norm Point)

- **Active Set** achieves finite & linear convergence!
- Completely modular: just add MAP

\[ \frac{\partial \mu}{\partial \eta} \text{ is sparse} \]

\[ \text{computing } \left( \frac{\partial \mu}{\partial \eta} \right)^T dy \]

\[ \text{takes } O(\dim(\mu) \text{ nnz}(p^*)) \]
Structured Attention & Graphical Models
Structured Attention & Graphical Models
Structured Attention for Alignments

NLI

premise: A gentleman overlooking a neighborhood situation.

hypothesis: A police officer watches a situation closely.

input (P, H)

output

A gentleman overlooking a situation

A police officer watches a situation closely

(Model: ESIM)
Structured Attention for Alignments

NLI

Premise: A gentleman overlooking a neighborhood situation.
Hypothesis: A police officer watches a situation closely.

Input: \((P, H)\)

Output: entails, contradicts, neutral

(Model: ESIM)
NLI

premise: A gentleman overlooking a neighborhood situation.
hypothesis: A police officer watches a situation closely.

input

(P, H)

output

(Model: ESIM)
Structured Attention for Alignments

NLI

premise: A gentleman overlooks a neighborhood situation.
hypothesis: A police officer watches a situation closely.

input (P, H)

output

entails
contradicts
neutral

(Proposed model: global matching)
SNLI

- Softmax: 86.5%
- Matching: 86%
- Sequence: 87%

MultiNLI

- Softmax: 76%
- Matching: 76.5%
- Sequence: 75.5%
a gentleman overlooking a neighborhood situation.
a police officer watches a situation closely.
A police officer watches a situation closely.

A gentleman overlooking a neighborhood situation.
Dynamically inferring the computation graph
The bears eat the pretty ones (Tai & al, 15)
The bears eat the pretty ones (Tai & al, 15)
The bears eat the pretty ones (Tai & al, 15)
Dependency TreeLSTM

(Tai & al, 15)
The bears eat the pretty ones (Tai & al, 15)
The bears eat the pretty ones (Tai & al, 15)
The bears eat the pretty ones

(Tai & al, 15)
Latent Dependency TreeLSTM

\[ p(y_j|x) = \sum h_2 H_p(y_j|h, x) p(h_j|x) \]

The bears eat the pretty ones

(Niculae, Martins, Cardie, 18)
The bears eat the pretty ones

\[ p(y|x) = \sum_{h \in \mathcal{H}} p(y \mid h, x) \ p(h \mid x) \]
Structured Latent Variable Models

\[ p(y \mid x) = \sum_{h \in \mathcal{H}} p(y \mid h, x) p(h \mid x) \]
Structured Latent Variable Models

\[ p(y \mid x) = \sum_{h \in \mathcal{H}} p_{\phi}(y \mid h, x) p_{\pi}(h \mid x) \]
Structured Latent Variable Models

\[ p(y | x) = \sum_{h \in \mathcal{H}} p_\phi(y | h, x) p_{\pi}(h | x) \]

e.g., a TreeLSTM defined by \( h \)
Structured Latent Variable Models

\[ p(y | x) = \sum_{h \in \mathcal{H}} \psi(y | h, x) \pi(h | x) \]

e.g., a TreeLSTM defined by \( h \)

parsing model, using some score \( \pi(h; x) \)

How to define \( \pi(h | x) \)?

\[ \frac{\partial p(y | x)}{\partial \pi} = \sum_{h \in \mathcal{H}} \psi(y | h, x) \frac{\partial \pi(h | x)}{\partial \pi} = \begin{cases} 1 & \text{if } h = h^\star \\ 0 & \text{else} \end{cases} \]

\[ \arg\max \psi(h | x) \propto \exp \text{score}_\pi(h; x) \]

SparseMAP e.g., a TreeLSTM defined by \( h \)

\[ \text{Exponentially large sum!} \]
Structured Latent Variable Models

$p(y \mid x) = \sum_{h \in \mathcal{H}} p_{\phi}(y \mid h, x) p_{\pi}(h \mid x)$

- sum over all possible trees
- e.g., a TreeLSTM defined by $h$
- parsing model, using some score $\pi(h; x)$

Exponentially large sum!
Structured Latent Variable Models

$p(y | x) = \sum_{h \in H} p_{\phi}(y | h, x) p_{\pi}(h | x)$

How to define $p_{\pi}$?

- **idea 1**: sum over all possible trees
- **idea 2**: $p_{\pi}(h | x) \propto \exp(\text{score}_{\pi}(h; x))$
- **idea 3**: e.g., a TreeLSTM defined by $h$

Parsing model, using some score $\pi(h; x)$
Structured Latent Variable Models

\[ p(y | x) = \sum_{h \in \mathcal{H}} p_\phi(y | h, x) p_\pi(h | x) \]

How to define \( p_\pi \)?

- **Idea 1**: Sum over all possible trees
- **Idea 2**: 
  \[ \sum_{h \in \mathcal{H}} \frac{\partial p(y | x)}{\partial p_\pi(h | x)} = 1 \text{ if } h = h^\star \text{ else } 0 \]
- **Idea 3**: 
  \[ \text{argmax}_{p_\pi(h | x)} \propto \exp \text{score}_\pi(h; x) \]

- Example: A TreeLSTM defined by \( h \)
- Parsing model, using some score \( \pi(h; x) \)

Exponentially large sum!
Structured Latent Variable Models

$p(y | x) = \sum_{h \in \mathcal{H}} p_{\phi}(y | h, x) p_{\pi}(h | x)$

How to define $p_{\pi}$?

- idea 1: sum over all possible trees
- idea 2: $\arg\max h \in \mathcal{H} \{ p_{\pi}(h | x) \}$
- idea 3: $\sum_{h \in \mathcal{H}} \frac{\partial p(y | x)}{\partial \pi}$

$\text{e.g.},$ a TreeLSTM defined by $h$

parse model, using some score $\pi(h; x)$

Exponentially large sum!
Structured Latent Variable Models

$p(y \mid x) = \sum_{h \in \mathcal{H}} p_{\phi}(y \mid h, x) p_{\pi}(h \mid x)$

How to define $p_{\pi}$?

- **Idea 1**: $p_{\pi}(h \mid x) = 1$ if $h = h^*$ else 0
  - argmax
- **Idea 2**
- **Idea 3**

e.g., a TreeLSTM defined by $h$

sum over all possible trees

$p_{\pi}(h \mid x)$ parsing model, using some score $\pi(h; x)$

Exponentially large sum!
Structured Latent Variable Models

\[ p(y \mid x) = \sum_{h \in \mathcal{H}} p_{\phi}(y \mid h, x) \ p_{\pi}(h \mid x) \]

How to define \( p_{\pi} \)?

- **idea 1**: \( p_{\pi}(h \mid x) = 1 \) if \( h = h^* \) else 0
  \[ \arg\max \]

- **idea 2**: e.g., a TreeLSTM defined by \( h \)

- **idea 3**: parsing model, using some \( \text{score}_{\pi}(h; x) \)

\[ \sum_{h \in \mathcal{H}} \frac{\partial p(y \mid x)}{\partial \pi} \]
Structured Latent Variable Models

\[ p(y \mid x) = \sum_{h \in \mathcal{H}} p_{\phi}(y \mid h, x) p_{\pi}(h \mid x) \]

How to define \( p_{\pi} \)?

**Idea 1**

\[ p_{\pi}(h \mid x) = 1 \text{ if } h = h^* \text{ else } 0 \]

**Idea 2**

\[ \arg\max_{h \in \mathcal{H}} \frac{\partial p(y \mid x)}{\partial \pi} \]

**Idea 3**

E.g., a TreeLSTM defined by \( h \)

sum over all possible trees

e.g., a TreeLSTM defined by \( h \)

Parsing model, using some score \( \pi(h; x) \)

Exponentially large sum!
Structured Latent Variable Models

$p(y | x) = \sum_{h \in \mathcal{H}} p_{\phi}(y | h, x) p_{\pi}(h | x)$

How to define $p_\pi$?

idea 1 $p_{\pi}(h | x) = 1$ if $h = h^*$ else 0

argmax

idea 2 $p_{\pi}(h | x) \propto \exp(\text{score}_\pi(h; x))$

softmax

idea 3

e.g., a TreeLSTM defined by $h$

parsing model, using some $\text{score}_\pi(h; x)$

sum over all possible trees

Exponentially large sum!
Structured Latent Variable Models

\[ p(y | x) = \sum_{h \in \mathcal{H}} p_{\phi}(y | h, x) p_{\pi}(h | x) \]

How to define \( p_{\pi} \)?

- **Idea 1**: \( p_{\pi}(h | x) = 1 \) if \( h = h^* \) else 0. 
  - Sum over all possible trees
  - \( \arg\max \)
  - \( \smiley \)
  - \( \frown \)

- **Idea 2**: \( p_{\pi}(h | x) \propto \exp(\text{score}_{\pi}(h; x)) \)
  - \( \sum_{h \in \mathcal{H}} \frac{\partial p(y | x)}{\partial \pi} \)
  - Softmax
  - \( \smiley \)

- **Idea 3**: Other models, e.g., a TreeLSTM defined by \( h \)

- Parsing model, using some score \( \pi(h; x) \)

- Exponentially large sum!
Structured Latent Variable Models

$$p(y \mid x) = \sum_{h \in \mathcal{H}} p_{\phi}(y \mid h, x) p_{\pi}(h \mid x)$$

How to define $p_{\pi}$?

- **idea 1**: $p_{\pi}(h \mid x) = 1$ if $h = h^*$ else 0
  - argmax
  - sum over all possible trees

- **idea 2**: $p_{\pi}(h \mid x) \propto \exp\left( \text{score}_{\pi}(h; x) \right)$
  - softmax
  - exponentially large sum!

- **idea 3**: e.g., a TreeLSTM defined by $h$
  - parsing model, using some $\text{score}_{\pi}(h; x)$
Structured Latent Variable Models

\[ p(y \mid x) = \sum_{h \in \mathcal{H}} p_{\phi}(y \mid h, x) p_{\pi}(h \mid x) \]

How to define \( p_{\pi} \)?

- **idea 1**: \( p_{\pi}(h \mid x) = 1 \) if \( h = h^* \) else 0
  - argmax

- **idea 2**: \( p_{\pi}(h \mid x) \propto \exp(\text{score}_{\pi}(h; x)) \)
  - softmax

- **idea 3**: SparseMAP
  - Exponentially large sum!
SparseMAP Inference

\[ p(y_j | x) = 0.7 \phi(y_j) + 0.3 \phi(y_j) \]

is not a tree itself.
SparseMAP Inference

\[ p(y_j | x) = 0.7 \cdot p(\phi(y_j | \cdot)) + 0.3 \cdot p(\phi(y_j | \cdot)) + 0 \cdot p(\phi(y_j | \cdot)) + \ldots \]
SparseMAP Inference

\[ p(y \mid x) = 0.7 p_\phi(y \mid \text{node}) + 0.3 p_\phi(y \mid \text{node}) + \ldots \]
SparseMAP Inference

\[
p(y | x) = 0.7 \, p_\phi(y | \text{root}) + 0.3 \, p_\phi(y | \text{not a tree}) + \ldots
\]

is not a tree itself: \( p(y | x) \neq p_\phi(y | \text{root}) \)!
<table>
<thead>
<tr>
<th>Accuracy</th>
<th>LTR</th>
<th>Flat</th>
<th>Latent</th>
</tr>
</thead>
<tbody>
<tr>
<td>85%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>84%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>83%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>82%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The bears eat the pretty ones

Left-to-right: regular LSTM
* The bears eat the pretty ones

Flat: bag-of-words–like

\[
p(\text{y}_j | \text{P}, \text{H}) = \frac{\sum_h \text{P}^2(\text{H}(\text{P}))}{\sum_h \text{H}^2(\text{H})} p(\phi(\text{y}_j | \text{h}_\text{P}, \text{h}_\text{H})) p(\pi(\text{h}_\text{P} | \text{P})) p(\pi(\text{h}_\text{H} | \text{H}))
\]
<table>
<thead>
<tr>
<th>Accuracy</th>
<th>LTR</th>
<th>Flat</th>
<th>CoreNLP</th>
</tr>
</thead>
<tbody>
<tr>
<td>85%</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>84%</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>83%</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>82%</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>81%</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>80%</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

The bears eat the pretty ones

CoreNLP: off-line parser
<table>
<thead>
<tr>
<th>Accuracy (binary)</th>
<th>LTR</th>
<th>Flat</th>
<th>CoreNLP</th>
<th>Latent</th>
</tr>
</thead>
<tbody>
<tr>
<td>85%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>84%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>83%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>82%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Sentiment classification (SST)**

<table>
<thead>
<tr>
<th>Accuracy (binary)</th>
<th>LTR</th>
<th>Flat</th>
<th>CoreNLP</th>
<th>Latent</th>
</tr>
</thead>
<tbody>
<tr>
<td>85%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>84%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>83%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>82%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80%</td>
<td>LTR</td>
<td>Flat</td>
<td>CoreNLP</td>
<td>Latent</td>
</tr>
</tbody>
</table>

**Natural Language Inference (SNLI)**

<table>
<thead>
<tr>
<th>Accuracy (3-class)</th>
<th>LTR</th>
<th>Flat</th>
<th>CoreNLP</th>
<th>Latent</th>
</tr>
</thead>
<tbody>
<tr>
<td>82%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81.8%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81.6%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81.4%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81.2%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80.8%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80.6%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Sentence pair classification**

\[ p(y \mid P, H) = \sum_{h_P \in \mathcal{H}(P)} \sum_{h_H \in \mathcal{H}(H)} p_\phi(y \mid h_P, h_H) p_\pi(h_P \mid P) p_\pi(h_H \mid H) \]
Sentiment classification (SST)

<table>
<thead>
<tr>
<th>Accuracy</th>
<th>LTR</th>
<th>Flat</th>
<th>CoreNLP</th>
<th>Latent</th>
</tr>
</thead>
<tbody>
<tr>
<td>85%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>84%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>83%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>82%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Natural Language Inference (SNLI)

<table>
<thead>
<tr>
<th>Accuracy</th>
<th>LTR</th>
<th>Flat</th>
<th>CoreNLP</th>
<th>Latent</th>
</tr>
</thead>
<tbody>
<tr>
<td>82%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81.8%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81.6%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81.4%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81.2%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80.8%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80.6%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Reverse dictionary lookup

given word description, predict word embedding (Hill et al, 17)

instead of \( p(y \mid x) \), we model \( \mathbb{E}_{p_\pi} g(x) = \sum_{h \in \mathcal{H}} g(x; h) p_\pi(h \mid x) \)
Sentiment classification (SST)

<table>
<thead>
<tr>
<th>Accuracy (binary)</th>
<th>LTR</th>
<th>Flat</th>
<th>CoreNLP</th>
<th>Latent</th>
</tr>
</thead>
<tbody>
<tr>
<td>85%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>84%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>83%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>82%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>81%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>80%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Natural Language Inference (SNLI)

<table>
<thead>
<tr>
<th>Accuracy (3-class)</th>
<th>LTR</th>
<th>Flat</th>
<th>CoreNLP</th>
<th>Latent</th>
</tr>
</thead>
<tbody>
<tr>
<td>82%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>81.8%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>81.6%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>81.4%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>81.2%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>81%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>80.8%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>80.6%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Reverse dictionary lookup

<table>
<thead>
<tr>
<th>Accuracy@10 (definitions)</th>
<th>LTR</th>
<th>Flat</th>
<th>Latent</th>
</tr>
</thead>
<tbody>
<tr>
<td>38%</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>36%</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>34%</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>32%</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>30%</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Accuracy@10 (concepts)</th>
<th>LTR</th>
<th>Flat</th>
<th>Latent</th>
</tr>
</thead>
<tbody>
<tr>
<td>38%</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>36%</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>34%</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>32%</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>30%</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Syntax vs. Composition Order

CoreNLP parse, \( p = 21.4\% \)

★ lovely and poignant .
Syntax vs. Composition Order

$p = 22.6\%$

★ lovely and poignant .

CoreNLP parse, $p = 21.4\%$

★ lovely and poignant .
Syntax vs. Composition Order

$p = 22.6\%$

* lovely and poignant .

CoreNLP parse, $p = 21.4\%$

* lovely and poignant .

$\star a$ deep and meaningful film .

$p = 15.33\%$

$\star a$ deep and meaningful film .

$p = 15.27\%$

* a deep and meaningful film .

CoreNLP parse, $p = 0\%$

$\star a$ deep and meaningful film .
Conclusions

Differentiable & sparse structured inference

Generic, extensible algorithms

Interpretable structured attention

Dynamically-inferred computation graphs

Catch us at EMNLP:
BlackboxNLP, Thursday 11:00 & EMNLP, Friday 15:36 (3B)

vlad@vene.ro  github.com/vene/sparsemap
https://vene.ro  @vnfrombucharest
Extra slides
Acknowledgements

This work was supported by the European Research Council (ERC StG DeepSPIN 758969) and by the Fundação para a Ciência e Tecnologia through contract UID/EEA/50008/2013.

Some icons by Dave Gandy and Freepik via flaticon.com.
a gentleman overlooking a neighborhood situation.

a police officer watches a situation closely.

a police officer watches a situation closely.
Structured Output Prediction

SparseMAP

\[ L_A(\eta, \tilde{\mu}) = \max_{\mu \in M} \left\{ \eta^T \mu - \frac{1}{2} \| \mu \|^2 \right\} \]
\[ - \eta^T \tilde{\mu} + \frac{1}{2} \| \tilde{\mu} \|^2 \]

Instance of a structured Fenchel-Young loss, like CRF, SVM, etc. [Blondel, Martins, Niculae '18]
Structured Output Prediction

SparseMAP

\[ L_A(\eta, \bar{\mu}) = \max_{\mu \in \mathcal{M}} \left\{ \eta^T \mu - \frac{1}{2} \| \mu \|^2 \right\} \]

\[ - \eta^T \bar{\mu} + \frac{1}{2} \| \bar{\mu} \|^2 \]

\[ \text{cost-SparseMAP} \]

\[ L^\rho_A(\eta, \bar{\mu}) = \max_{\mu \in \mathcal{M}} \left\{ \eta^T \mu - \frac{1}{2} \| \mu \|^2 + \rho(\mu, \bar{\mu}) \right\} \]

\[ - \eta^T \bar{\mu} + \frac{1}{2} \| \bar{\mu} \|^2 \]

Instance of a structured Fenchel-Young loss, like CRF, SVM, etc. [Blondel, Martins, Niculae ’18]
Dependency Parsing with bi-LSTM features

[Kiperwasser & Goldberg, 2016]
Unlabeled Accuracy (UAS)
Universal Dependencies dataset
Sparse Structured Output Prediction
As models train, inference gets sparser!

train: epoch 0

valid: epoch 0
Sparse Structured Output Prediction

Inference captures linguistic ambiguity!

They did a vehicle wrap for my Toyota Venza that looks amazing.
Sparse Structured Output Prediction

Inference captures linguistic ambiguity!

- the broccoli looks browned around the edges.