



Learning with Sparse Latent Structure

Vlad Niculae

Instituto de Telecomunicações

Work with: André Martins, Claire Cardie, Mathieu Blondel



github.com/vene/sparsemap

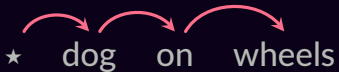


[@vnfrombucharest](https://twitter.com/vnfrombucharest)

Structured Prediction

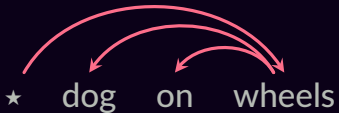
...

★ dog on wheels



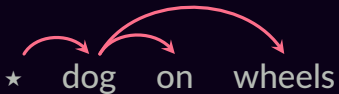
A diagram illustrating a sequence of transitions between words in the phrase "dog on wheels". Three red curved arrows point from left to right, connecting the words in sequence: from "dog" to "on", from "on" to "wheels", and from the star symbol "★" to "dog".

★ dog on wheels



A diagram illustrating multiple transitions from a single word to others in the phrase "dog on wheels". Three red curved arrows originate from the word "dog": one points to "on", one points to "wheels", and one points to the star symbol "★".

★ dog on wheels



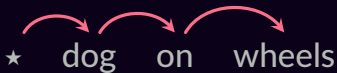
A diagram illustrating transitions from a single word to multiple words in the phrase "dog on wheels". Three red curved arrows originate from the word "dog": one points to "on", one points to "wheels", and one points to the star symbol "★".

...

Structured Prediction

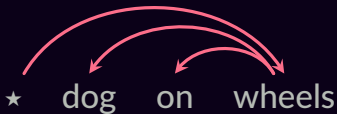
...

VERB PREP NOUN
dog on wheels



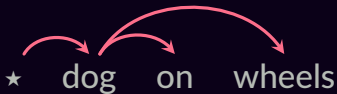
dog hond
on op
wheels wielen

NOUN PREP NOUN
dog on wheels



dog hond
on op
wheels wielen

NOUN DET NOUN
dog on wheels



dog hond
on op
wheels wielen

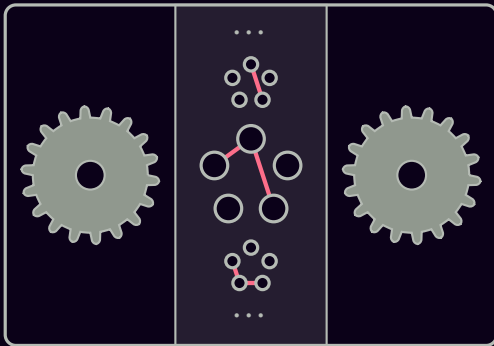
...

Structured Prediction

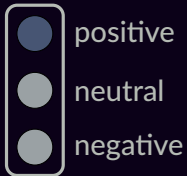


Latent Structure Models

input



output



record scratch

freeze frame

**How to select an item
from a set?**

How to select an item from a set?



...



How to select an item from a set?

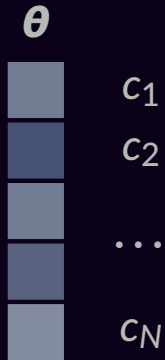
c_1

c_2

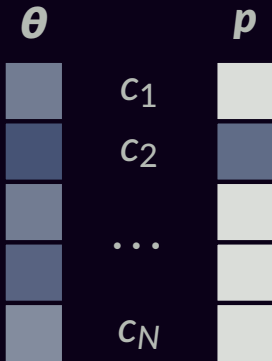
...

c_N

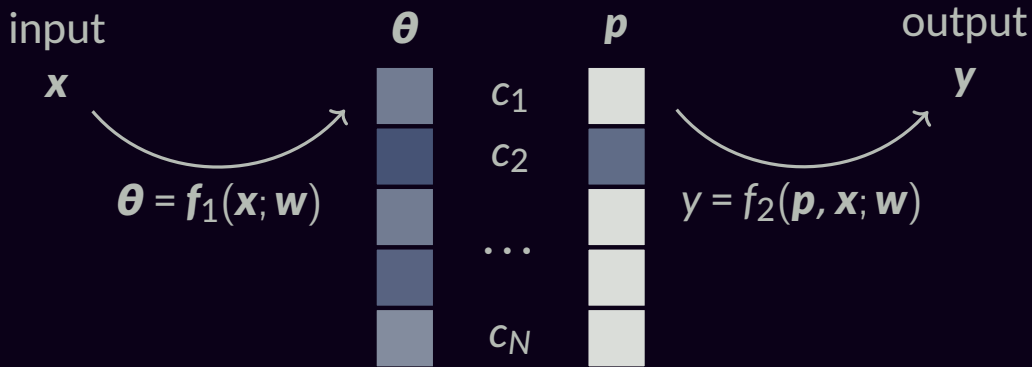
How to select an item from a set?



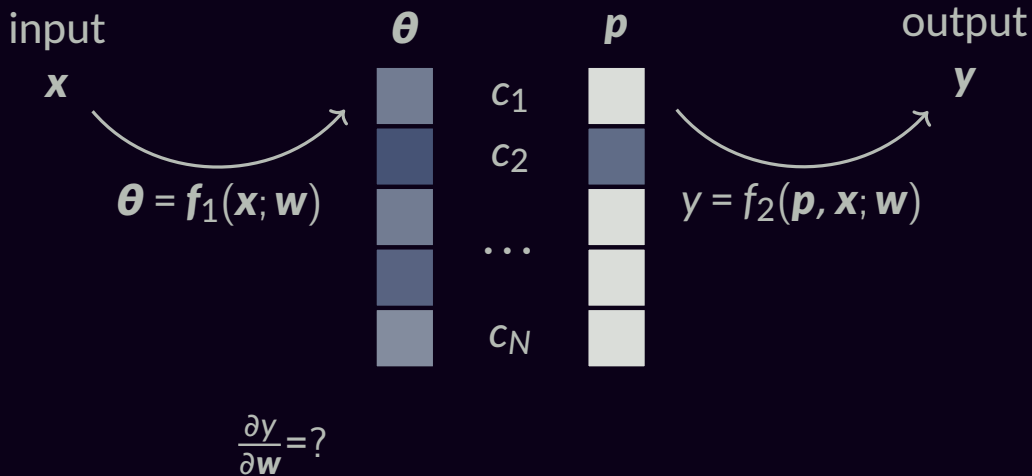
How to select an item from a set?



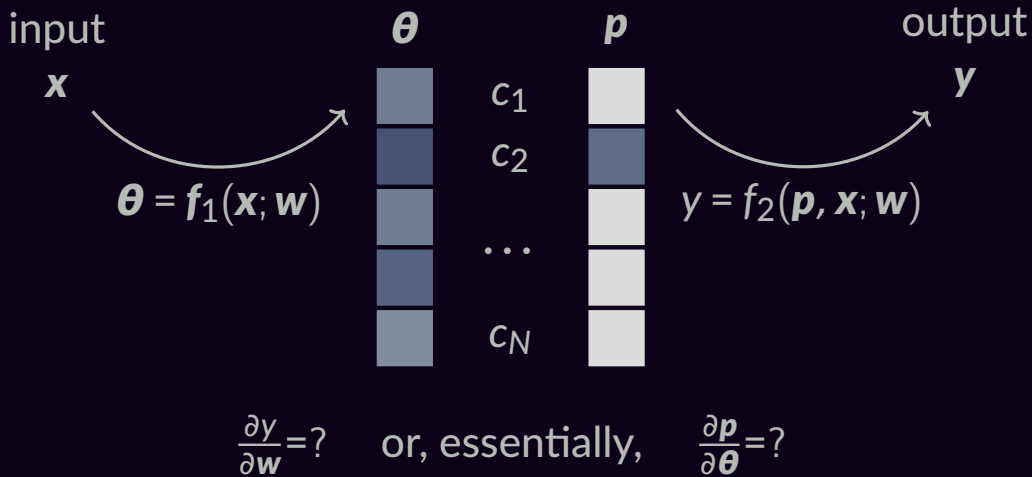
How to select an item from a set?



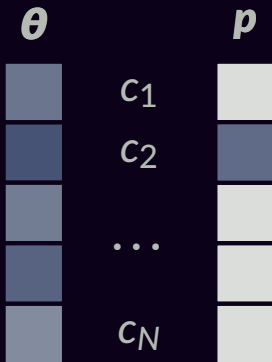
How to select an item from a set?



How to select an item from a set?

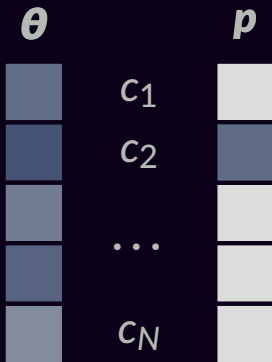


Argmax













$$\frac{\partial \mathbf{p}}{\partial \boldsymbol{\theta}} = ?$$

Argmax



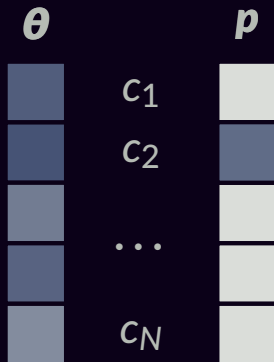
$$\frac{\partial \mathbf{p}}{\partial \boldsymbol{\theta}} = ?$$

Argmax

θ		p
	c_1	
	c_2	
	\dots	
		
	c_N	

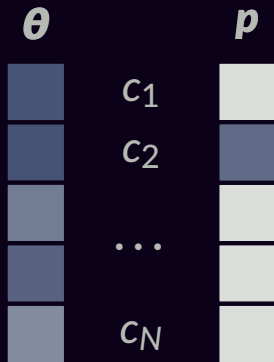
$$\frac{\partial \mathbf{p}}{\partial \boldsymbol{\theta}} = ?$$

Argmax



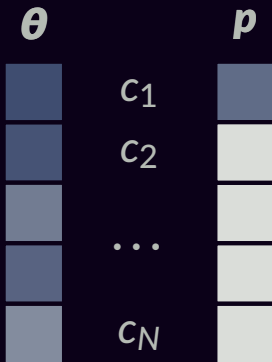
$$\frac{\partial \mathbf{p}}{\partial \boldsymbol{\theta}} = ?$$

Argmax



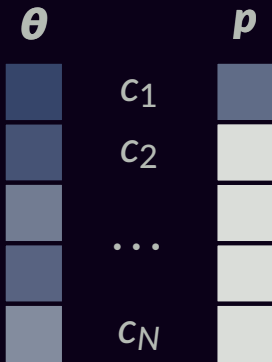
$$\frac{\partial \mathbf{p}}{\partial \boldsymbol{\theta}} = ?$$

Argmax



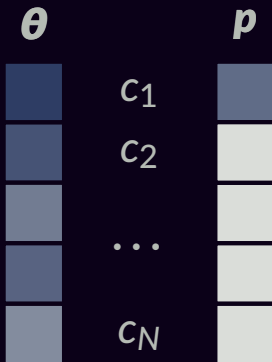
$$\frac{\partial \mathbf{p}}{\partial \boldsymbol{\theta}} = ?$$

Argmax



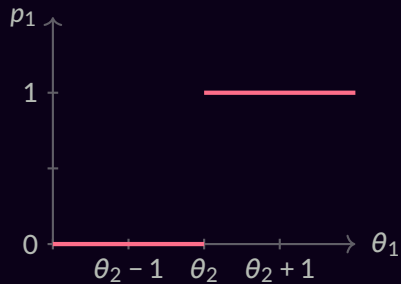
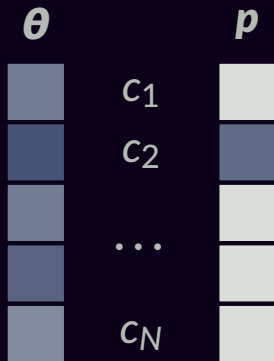
$$\frac{\partial \mathbf{p}}{\partial \boldsymbol{\theta}} = ?$$

Argmax



$$\frac{\partial \mathbf{p}}{\partial \boldsymbol{\theta}} = ?$$

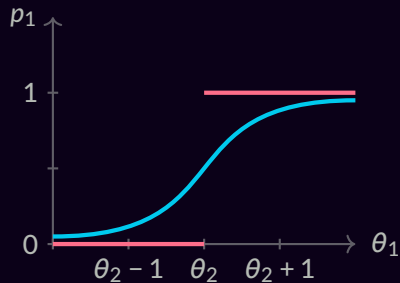
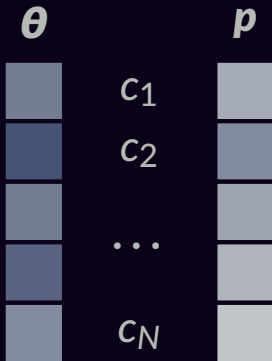
Argmax



$$\frac{\partial p}{\partial \theta} = \mathbf{0}$$

Argmax vs. Softmax

$$p_j = \exp(\theta_j)/Z$$



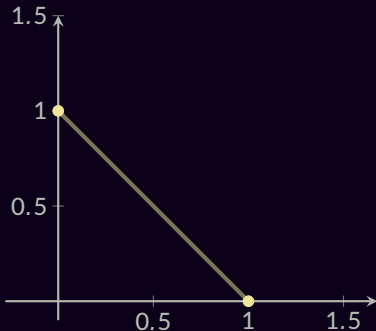
$$\frac{\partial \mathbf{p}}{\partial \boldsymbol{\theta}} = \text{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top$$

Variational Form of Argmax

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

Variational Form of Argmax

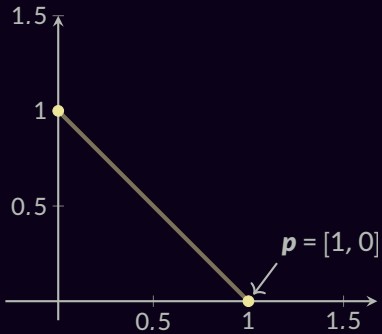
$$\Delta = \{\mathbf{p} \in \mathbb{R}^N : \mathbf{p} \geq \mathbf{0}, \mathbf{1}^\top \mathbf{p} = 1\}$$



$N = 2$

Variational Form of Argmax

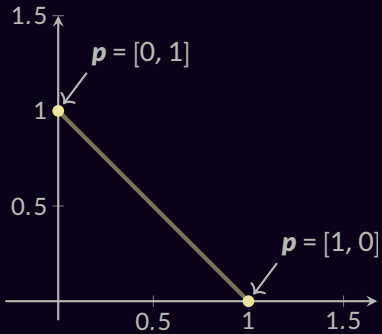
$$\Delta = \{\mathbf{p} \in \mathbb{R}^N : \mathbf{p} \geq \mathbf{0}, \mathbf{1}^\top \mathbf{p} = 1\}$$



$N = 2$

Variational Form of Argmax

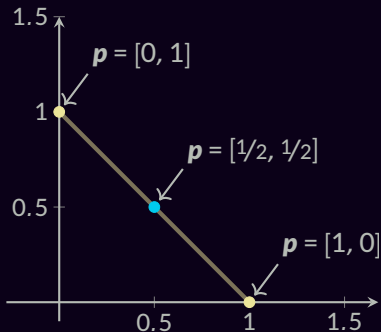
$$\Delta = \{\mathbf{p} \in \mathbb{R}^N : \mathbf{p} \geq \mathbf{0}, \mathbf{1}^\top \mathbf{p} = 1\}$$



$N = 2$

Variational Form of Argmax

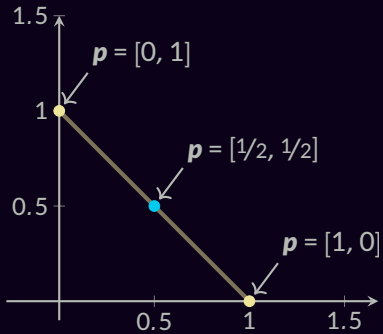
$$\Delta = \{\mathbf{p} \in \mathbb{R}^N : \mathbf{p} \geq \mathbf{0}, \mathbf{1}^\top \mathbf{p} = 1\}$$



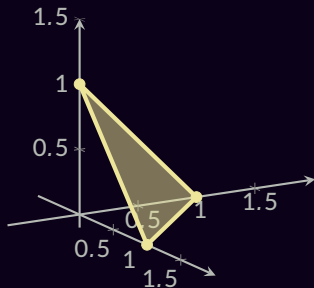
$N = 2$

Variational Form of Argmax

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



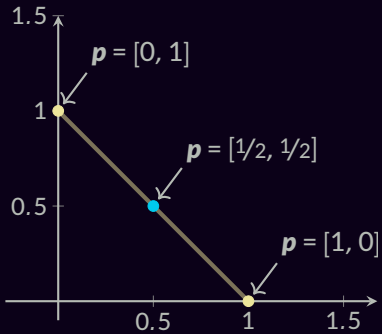
$N = 2$



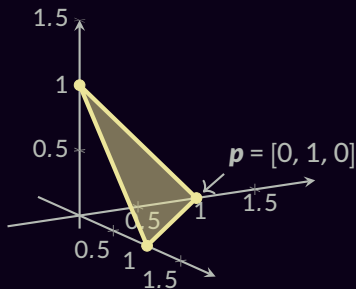
$N = 3$

Variational Form of Argmax

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



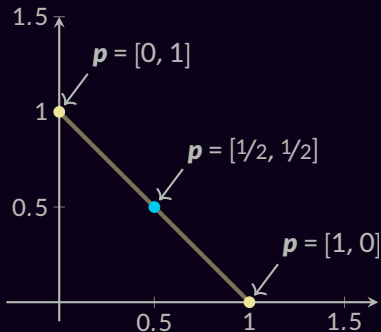
$N = 2$



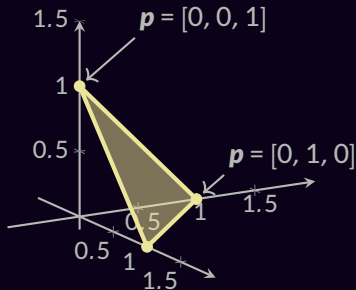
$N = 3$

Variational Form of Argmax

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



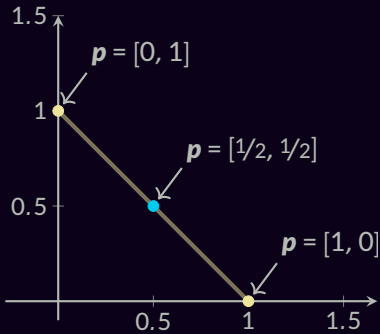
$N = 2$



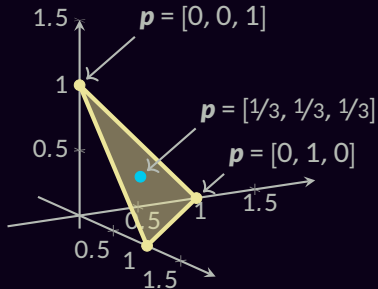
$N = 3$

Variational Form of Argmax

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



$N = 2$

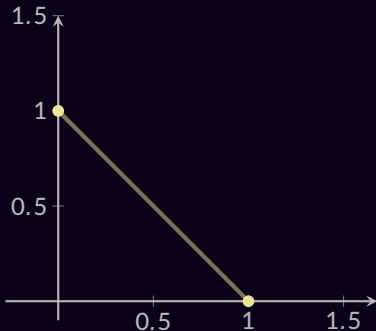


$N = 3$

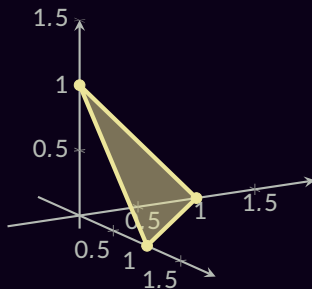
Variational Form of Argmax

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

Fundamental Thm. Lin. Prog.
(Dantzig et al., 1955)



$N = 2$

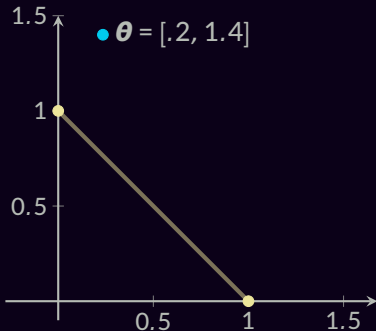


$N = 3$

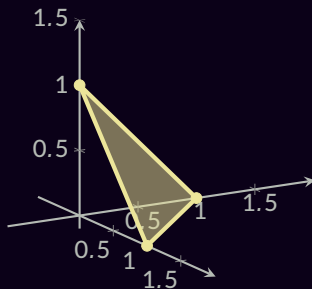
Variational Form of Argmax

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

Fundamental Thm. Lin. Prog.
(Dantzig et al., 1955)



$N = 2$

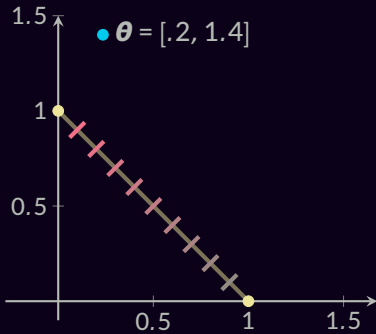


$N = 3$

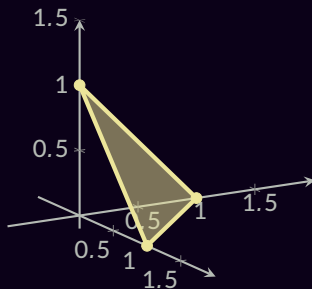
Variational Form of Argmax

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

Fundamental Thm. Lin. Prog.
(Dantzig et al., 1955)



$N = 2$

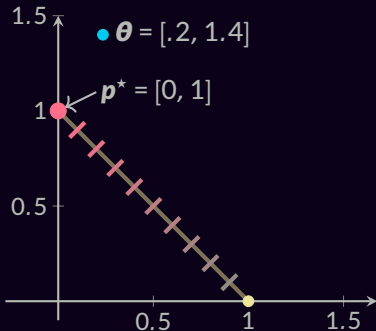


$N = 3$

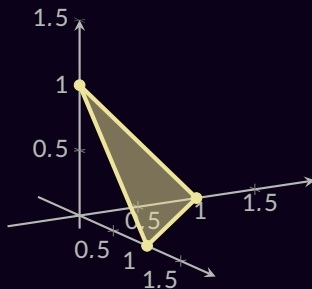
Variational Form of Argmax

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

Fundamental Thm. Lin. Prog.
(Dantzig et al., 1955)



$N = 2$

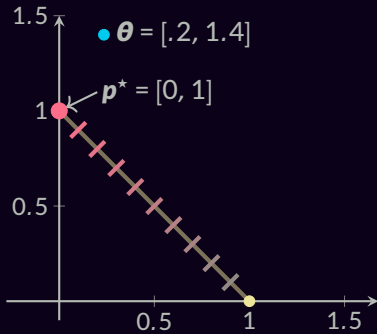


$N = 3$

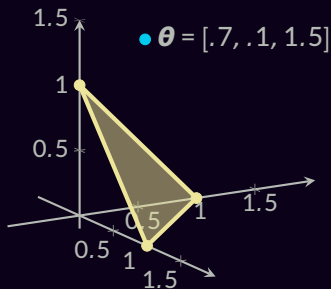
Variational Form of Argmax

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

Fundamental Thm. Lin. Prog.
(Dantzig et al., 1955)



$N = 2$

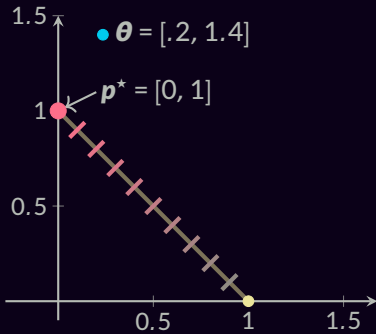


$N = 3$

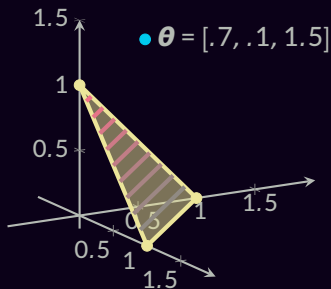
Variational Form of Argmax

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

Fundamental Thm. Lin. Prog.
(Dantzig et al., 1955)



$N = 2$

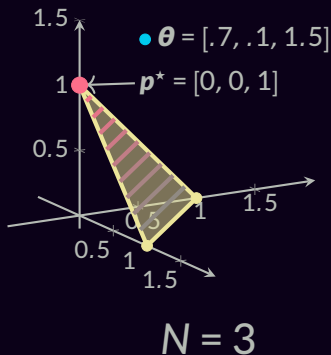
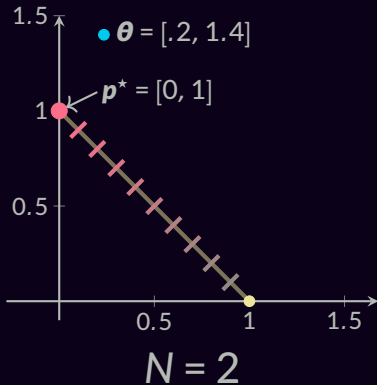


$N = 3$

Variational Form of Argmax

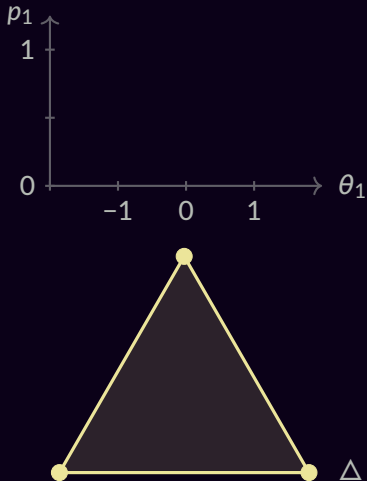
$$\max_j \theta_j = \max_{\mathbf{p} \in \Delta} \mathbf{p}^T \boldsymbol{\theta}$$

Fundamental Thm. Lin. Prog.
(Dantzig et al., 1955)



Smoothed Max Operators

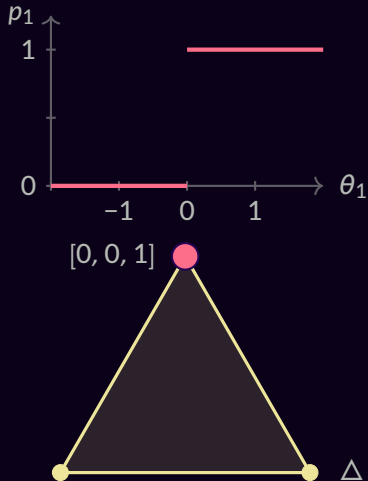
$$\boldsymbol{\pi}_{\Omega}(\boldsymbol{\theta}) = \arg \max_{\boldsymbol{p} \in \Delta} \boldsymbol{p}^{\top} \boldsymbol{\theta} - \Omega(\boldsymbol{p})$$



Smoothed Max Operators

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

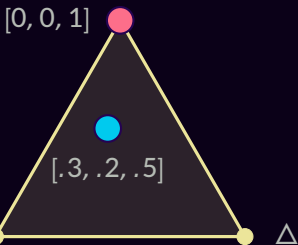
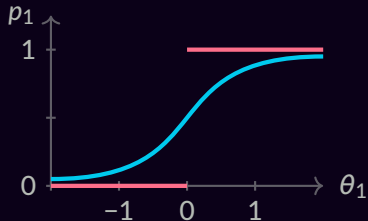
- argmax: $\Omega(\boldsymbol{p}) = 0$



Smoothed Max Operators

$$\boldsymbol{\pi}_{\Omega}(\boldsymbol{\theta}) = \arg \max_{\mathbf{p} \in \Delta} \mathbf{p}^{\top} \boldsymbol{\theta} - \Omega(\mathbf{p})$$

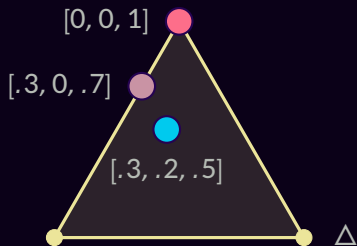
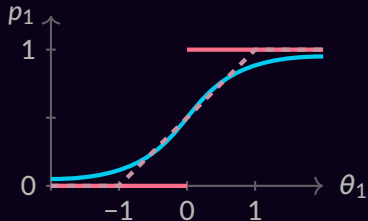
- argmax: $\Omega(\mathbf{p}) = 0$
- softmax: $\Omega(\mathbf{p}) = \sum_j p_j \log p_j$

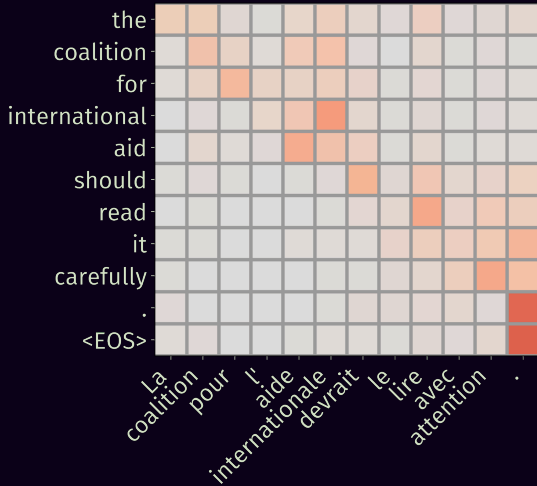


Smoothed Max Operators

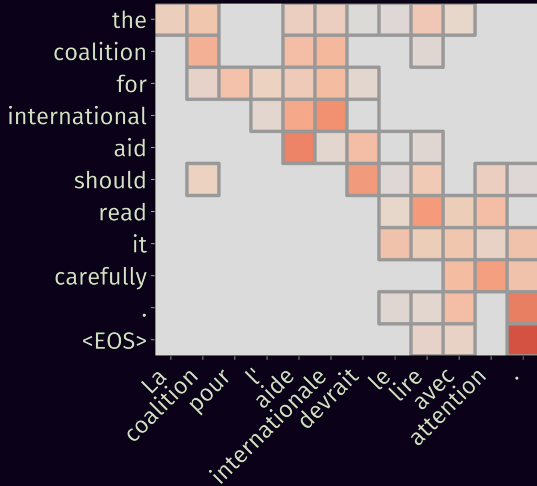
$$\boldsymbol{\pi}_{\Omega}(\boldsymbol{\theta}) = \arg \max_{\mathbf{p} \in \Delta} \mathbf{p}^{\top} \boldsymbol{\theta} - \Omega(\mathbf{p})$$

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

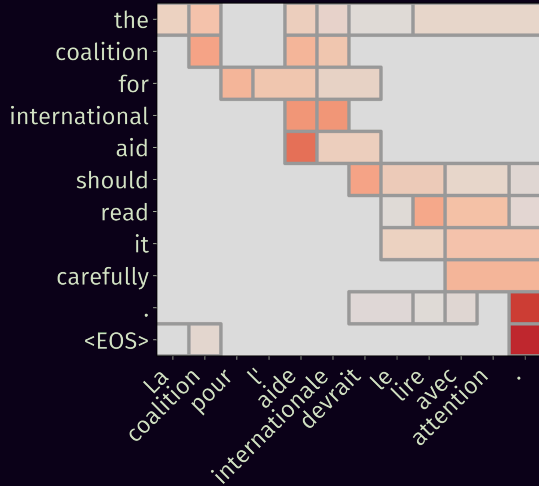




softmax



sparsemax

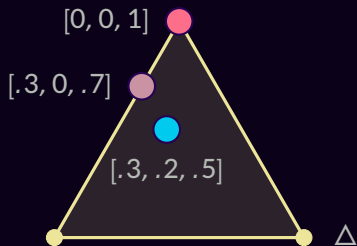
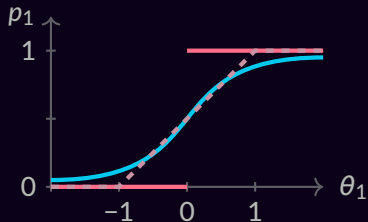


fusedmax ?!

Smoothed Max Operators

$$\boldsymbol{\pi}_{\Omega}(\boldsymbol{\theta}) = \arg \max_{\mathbf{p} \in \Delta} \mathbf{p}^{\top} \boldsymbol{\theta} - \Omega(\mathbf{p})$$

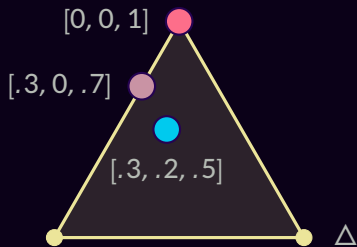
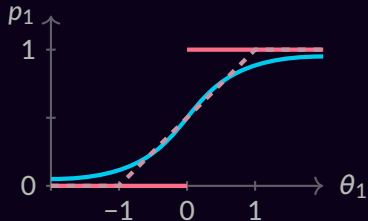
- argmax: $\Omega(\mathbf{p}) = 0$
- softmax: $\Omega(\mathbf{p}) = \sum_j p_j \log p_j$
- sparsemax: $\Omega(\mathbf{p}) = 1/2 \|\mathbf{p}\|_2^2$



Smoothed Max Operators

$$\boldsymbol{\pi}_{\Omega}(\boldsymbol{\theta}) = \arg \max_{\mathbf{p} \in \Delta} \mathbf{p}^{\top} \boldsymbol{\theta} - \Omega(\mathbf{p})$$

- argmax: $\Omega(\mathbf{p}) = 0$
- softmax: $\Omega(\mathbf{p}) = \sum_j p_j \log p_j$
- sparsemax: $\Omega(\mathbf{p}) = 1/2 \|\mathbf{p}\|_2^2$
- fusedmax: $\Omega(\mathbf{p}) = 1/2 \|\mathbf{p}\|_2^2 + \sum_j |p_j - p_{j-1}|$
- csparsesmax: $\Omega(\mathbf{p}) = 1/2 \|\mathbf{p}\|_2^2 + \iota(\mathbf{a} \leq \mathbf{p} \leq \mathbf{b})$

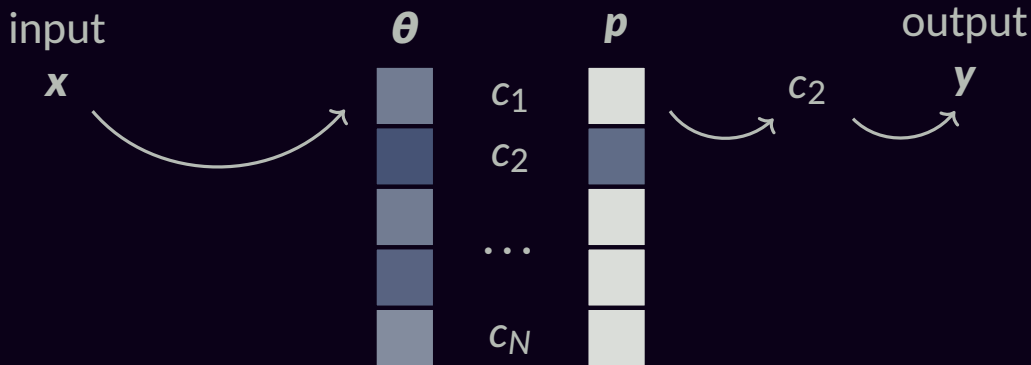


Structured Prediction

finally

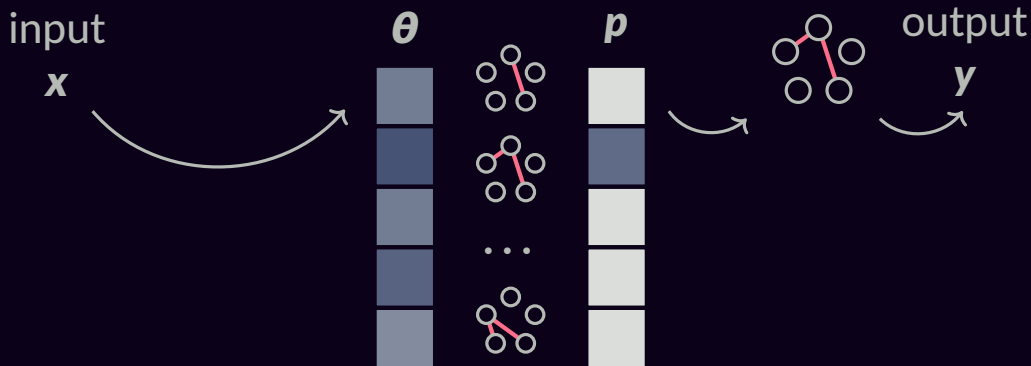
Structured Prediction

is essentially a (very high-dimensional) argmax



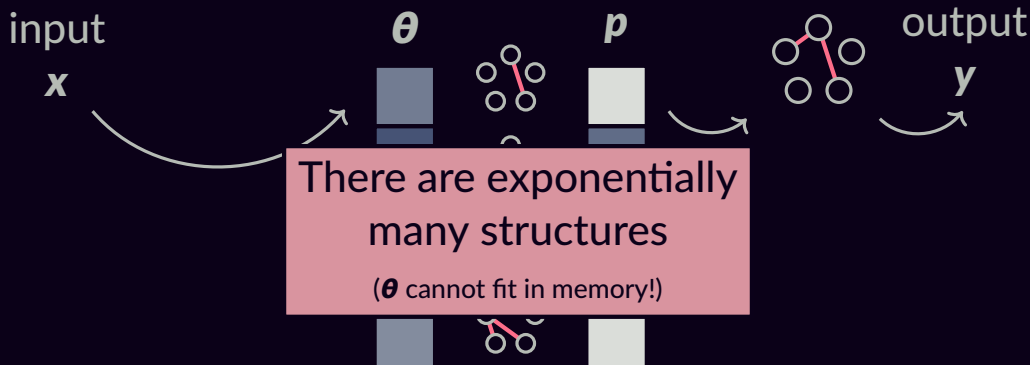
Structured Prediction

is essentially a (very high-dimensional) argmax



Structured Prediction

is essentially a (very high-dimensional) argmax



Factorization Into Parts

$$\theta = A^{\top} \eta$$

Factorization Into Parts

$$\theta = A^T \eta$$

★ dog on wheels

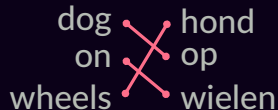


★→dog	1	0	0	$\eta = \begin{bmatrix} .1 \\ .2 \\ -.1 \\ .3 \\ .8 \\ .1 \\ -.3 \\ .2 \\ -.1 \end{bmatrix}$
on→dog	0	1	1	
wheels→dog	0	0	0	
★→on	0	1	1	
dog→on	1	...	0	
wheels→on	0	0	0	
★→wheels	0	0	0	
dog→wheels	0	1	0	
on→wheels	1	0	1	

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & \dots & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

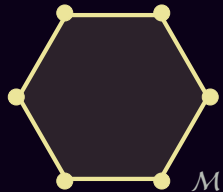
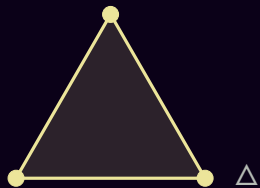
Factorization Into Parts

$$\theta = A^T \eta$$

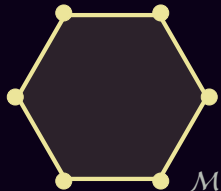
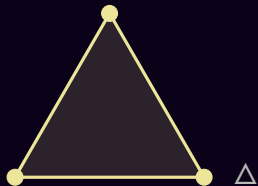


$$A = \begin{bmatrix} \star \rightarrow \text{dog} & 1 & 0 & 0 \\ \text{on} \rightarrow \text{dog} & 0 & 1 & 1 \\ \text{wheels} \rightarrow \text{dog} & 0 & 0 & 0 \\ \hline \star \rightarrow \text{on} & 0 & 1 & 1 \\ \text{dog} \rightarrow \text{on} & 1 & \dots & 0 & 0 & \dots \\ \text{wheels} \rightarrow \text{on} & 0 & 0 & 0 \\ \hline \star \rightarrow \text{wheels} & 0 & 0 & 0 \\ \text{dog} \rightarrow \text{wheels} & 0 & 1 & 0 \\ \text{on} \rightarrow \text{wheels} & 1 & 0 & 1 \end{bmatrix} \quad \eta = \begin{bmatrix} .1 \\ .2 \\ -.1 \\ \hline .3 \\ .8 \\ .1 \\ \hline -.3 \\ .2 \\ -.1 \end{bmatrix}$$

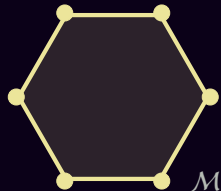
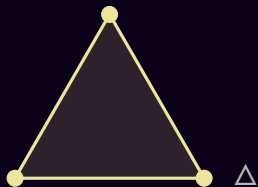
$$A = \begin{bmatrix} \text{dog} - \text{hond} & 1 & 0 & 0 \\ \text{dog} - \text{op} & 0 & 1 & 1 \\ \text{dog} - \text{wielen} & 0 & 0 & 0 \\ \hline \text{on} - \text{hond} & 0 & 0 & 0 \\ \text{on} - \text{op} & 1 & \dots & 0 & 0 & \dots \\ \text{on} - \text{wielen} & 0 & 1 & 1 \\ \hline \text{wheels} - \text{hond} & 0 & 1 & 0 \\ \text{wheels} - \text{op} & 0 & 0 & 0 \\ \text{wheels} - \text{wielen} & 1 & 0 & 1 \end{bmatrix} \quad \eta = \begin{bmatrix} .1 \\ .2 \\ -.1 \\ \hline .3 \\ .8 \\ .1 \\ \hline -.3 \\ .2 \\ -.1 \end{bmatrix}$$



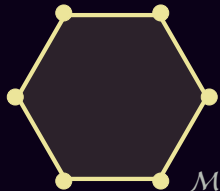
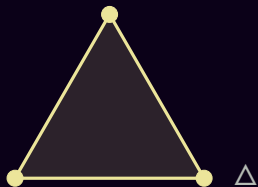
$$\mathcal{M} := \text{conv} \{ \mathbf{a}_y : y \in \mathcal{Y} \}$$



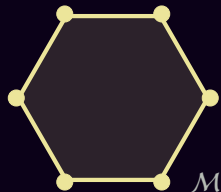
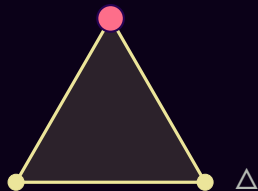
$$\begin{aligned}\mathcal{M} &:= \text{conv} \{ \mathbf{a}_y : y \in \mathcal{Y} \} \\ &= \{ \mathbf{A}\mathbf{p} : \mathbf{p} \in \Delta \}\end{aligned}$$



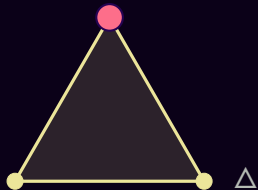
$$\begin{aligned}\mathcal{M} &:= \text{conv} \{ \mathbf{a}_y : y \in \mathcal{Y} \} \\ &= \{ \mathbf{A} \mathbf{p} : \mathbf{p} \in \Delta \} \\ &= \{ \mathbb{E}_{Y \sim \mathbf{p}} \mathbf{a}_Y : \mathbf{p} \in \Delta \}\end{aligned}$$



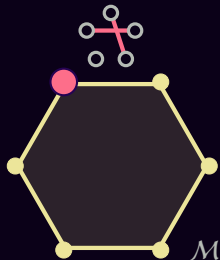
● $\operatorname{argmax}_{p \in \Delta} p^\top \theta$



• $\mathbf{argmax}_{p \in \Delta} \mathbf{argmax} p^T \boldsymbol{\theta}$



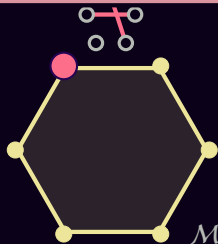
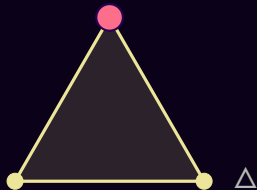
• $\mathbf{MAP} \mathbf{argmax}_{\mu \in \mathcal{M}} \boldsymbol{\mu}^T \boldsymbol{\eta}$



• $\text{argmax}_{p \in \Delta} p^T \theta$

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

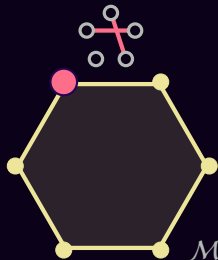
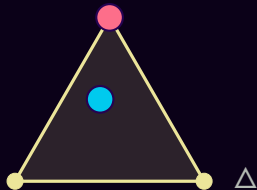
e.g. dependency parsing → **max. spanning tree**
 matching → **the Hungarian algorithm**



● **argmax** $\arg \max_{\mathbf{p} \in \Delta} \mathbf{p}^\top \boldsymbol{\theta}$

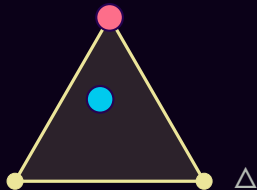
● **softmax** $\arg \max_{\mathbf{p} \in \Delta} \mathbf{p}^\top \boldsymbol{\theta} + H(\mathbf{p})$

● **MAP** $\arg \max_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta}$



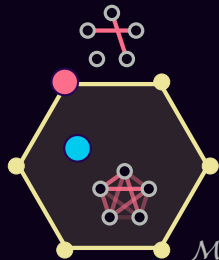
● **argmax** $\arg \max_{\mathbf{p} \in \Delta} \mathbf{p}^\top \boldsymbol{\theta}$

● **softmax** $\arg \max_{\mathbf{p} \in \Delta} \mathbf{p}^\top \boldsymbol{\theta} + H(\mathbf{p})$



● **MAP** $\arg \max_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta}$

● **marginals** $\arg \max_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} + \tilde{H}(\boldsymbol{\mu})$



● **argmax** $\arg \max_{p \in \Delta} p^\top \theta$

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

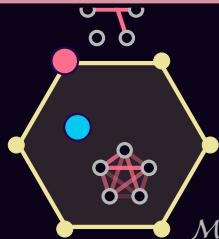
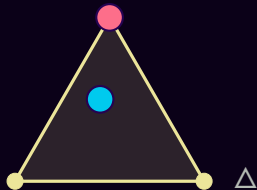
● **MAP** $\arg \max_{\mu \in \mathcal{M}} \mu^\top \eta$

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

e.g. sequence labelling \rightarrow forward-backward

(Rabiner, 1989)

As attention: (Kim et al., 2017)



● **argmax** $\arg \max_{\mathbf{p} \in \Delta} \mathbf{p}^\top \boldsymbol{\theta}$

● **softmax** $\arg \max_{\mathbf{p} \in \Delta} \mathbf{p}^\top \boldsymbol{\theta} + H(\mathbf{p})$

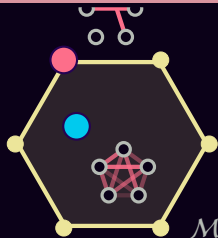
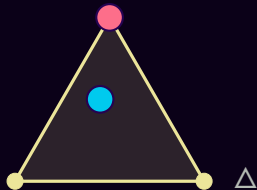
● **MAP** $\arg \max_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta}$

● **marginals** $\arg \max_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} + \tilde{H}(\boldsymbol{\mu})$

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

(Koo et al., 2007; D. A. Smith and N. A. Smith, 2007; McDonald and Satta, 2007)

As attention: (Liu and Lapata, 2018)



● **argmax** $\arg \max_{p \in \Delta} p^\top \theta$

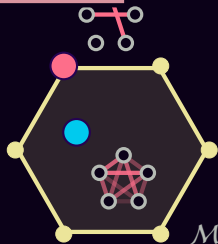
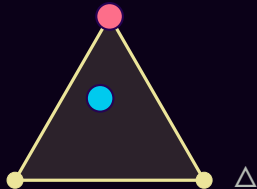
● **softmax** $\arg \max_{p \in \Delta} p^\top \theta + H(p)$

● **MAP** $\arg \max_{\mu \in \mathcal{M}} \mu^\top \eta$

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

e.g. matchings \rightarrow **#P-complete!**

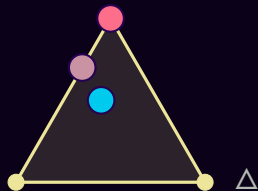
(Taskar, 2004; Valiant, 1979)



● **argmax** $\arg \max_{\mathbf{p} \in \Delta} \mathbf{p}^\top \boldsymbol{\theta}$

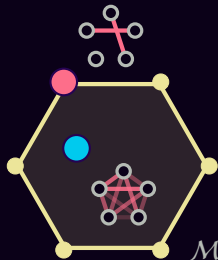
● **softmax** $\arg \max_{\mathbf{p} \in \Delta} \mathbf{p}^\top \boldsymbol{\theta} + H(\mathbf{p})$

● **sparsemax** $\arg \max_{\mathbf{p} \in \Delta} \mathbf{p}^\top \boldsymbol{\theta} - 1/2 \|\mathbf{p}\|^2$



● **MAP** $\arg \max_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta}$

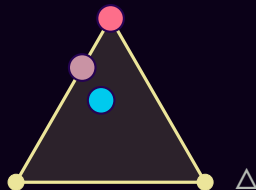
● **marginals** $\arg \max_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} + \tilde{H}(\boldsymbol{\mu})$



● **argmax** $\arg \max_{p \in \Delta} p^\top \theta$

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

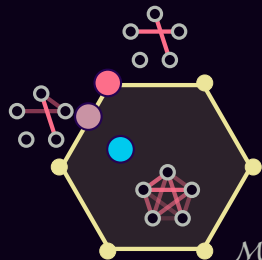
● **sparsemax** $\arg \max_{p \in \Delta} p^\top \theta - 1/2 \|p\|^2$



● **MAP** $\arg \max_{\mu \in \mathcal{M}} \mu^\top \eta$

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

● **SparseMAP** $\arg \max_{\mu \in \mathcal{M}} \mu^\top \eta - 1/2 \|\mu\|^2$



SparseMAP Solution

$$\boldsymbol{\mu}^{\star} = \arg \max_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^{\top} \boldsymbol{\eta} - 1/2 \|\boldsymbol{\mu}\|^2$$

$$= \begin{array}{c} \circ & \circ & \circ \\ & \diagup & \diagdown \\ \circ & \circ & \circ \end{array} = .6 \begin{array}{c} \circ & \circ & \circ \\ & \diagup & \diagdown \\ \circ & \circ & \circ \end{array} + .4 \begin{array}{c} \circ & \circ & \circ \\ & \diagup & \diagdown \\ \circ & \circ & \circ \end{array}$$

$$= \mathbf{A} \mathbf{p}^{\star} \text{ with very sparse } \mathbf{p}^{\star} \in \Delta^N$$

Algorithms for SparseMAP

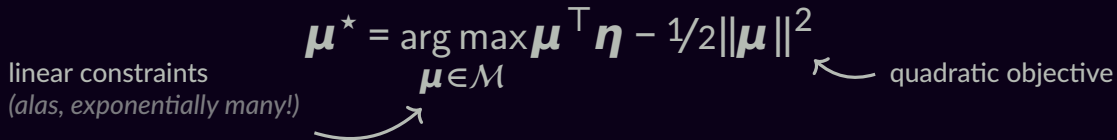
$$\boldsymbol{\mu}^{\star} = \arg \max_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^{\top} \boldsymbol{\eta} - 1/2 \|\boldsymbol{\mu}\|^2$$

Algorithms for SparseMAP

linear constraints
(*alas, exponentially many!*)

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

quadratic objective



Algorithms for SparseMAP

linear constraints
(*alas, exponentially many!*)

$$\boldsymbol{\mu}^{\star} = \arg \max_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^{\top} \boldsymbol{\eta} - 1/2 \|\boldsymbol{\mu}\|^2$$

quadratic objective

Conditional Gradient

(Frank and Wolfe, 1956; Lacoste-Julien and Jaggi, 2015)

Algorithms for SparseMAP

linear constraints
(*alas, exponentially many!*)

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

quadratic objective

Conditional Gradient

(Frank and Wolfe, 1956; Lacoste-Julien and Jaggi, 2015)

- select a new corner of \mathcal{M}

Algorithms for SparseMAP

linear constraints
(*alas, exponentially many!*)

$$\boldsymbol{\mu}^{\star} = \arg \max_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^{\top} \boldsymbol{\eta} - 1/2 \|\boldsymbol{\mu}\|^2$$

quadratic objective

Conditional Gradient

(Frank and Wolfe, 1956; Lacoste-Julien and Jaggi, 2015)

- select a new corner of \mathcal{M}


$$\mathbf{a}_{y^{\star}} = \arg \max_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^{\top} \underbrace{(\boldsymbol{\eta} - \boldsymbol{\mu}^{(t-1)})}_{\tilde{\boldsymbol{\eta}}}$$

Algorithms for SparseMAP

linear constraints
(*alas, exponentially many!*)

$$\boldsymbol{\mu}^{\star} = \arg \max_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^{\top} \boldsymbol{\eta} - 1/2 \|\boldsymbol{\mu}\|^2$$

quadratic objective



Conditional Gradient

(Frank and Wolfe, 1956; Lacoste-Julien and Jaggi, 2015)


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

Algorithms for SparseMAP

linear constraints
(*alas, exponentially many!*)

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

quadratic objective



Conditional Gradient

(Frank and Wolfe, 1956; Lacoste-Julien and Jaggi, 2015)

- select a new corner of \mathcal{M}
- update the (sparse) coefficients of \boldsymbol{p}
 - Update rules: vanilla, away-step, pairwise
 - Quadratic objective: **Active Set**
(Nocedal and Wright, 1999, Ch. 16.4 & 16.5)
(Wolfe, 1976; Vinyes and Obozinski, 2017)

Algorithms for SparseMAP

linear constraints
(*alas, exponentially many!*)

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

quadratic objective

Conditional Gradient

(Frank and Wolfe, 1956; Lacost

- select a new corner
- update the (sparse)

Active Set achieves
finite & linear convergence!

- Update rules: vanilla, away-step, pairwise
- Quadratic objective: **Active Set**

(Nocedal and Wright, 1999, Ch. 16.4 & 16.5)

(Wolfe, 1976; Vinyes and Obozinski, 2017)

Algorithms for SparseMAP

linear constraints
(*alas, exponentially many!*)

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

quadratic objective

Conditional Gradient

(Frank and Wolfe, 1956; Lacoste-Julien and Jaggi, 2015)

- select a new corner of \mathcal{M}
- update the (sparse) coefficients of \mathbf{p}
 - Update rules: vanilla, away-step, pairwise
 - Quadratic objective: **Active Set**
(Nocedal and Wright, 1999, Ch. 16.4 & 16.5)
(Wolfe, 1976; Vinyes and Obozinski, 2017)

Backward pass

$$\frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\eta}} \text{ is sparse}$$

Algorithms for SparseMAP

linear constraints
(*alas, exponentially many!*)

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

quadratic objective

Conditional Gradient

(Frank and Wolfe, 1956; Lacoste-Julien and Jaggi, 2015)

- select a new corner of \mathcal{M}
- update the (sparse) coefficients of \mathbf{p}
 - Update rules: vanilla, away-step, pairwise
 - Quadratic objective: **Active Set**
(Nocedal and Wright, 1999, Ch. 16.4 & 16.5)
(Wolfe, 1976; Vinyes and Obozinski, 2017)

Backward pass

$\frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\eta}}$ is sparse
computing $\left(\frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\eta}}\right)^\top d\mathbf{y}$
takes $\mathcal{O}(\dim(\boldsymbol{\mu}) \text{nnz}(\mathbf{p}^*))$

Algorithms for SparseMAP

linear constraints
(*alas, exponentially many!*)

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

quadratic objective

Condition Completely modular: just add MAP **pass**

(Frank and Wolfe, 1956)

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

(Nocedal and Wright, 1999, Ch. 16.4 & 16.5)

(Wolfe, 1976; Vinyes and Obozinski, 2017)

$\frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\eta}}$ is sparse
computing $\left(\frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\eta}}\right)^\top d\boldsymbol{\eta}$
takes $\mathcal{O}(\dim(\boldsymbol{\mu}) \text{nnz}(\mathbf{p}^*))$

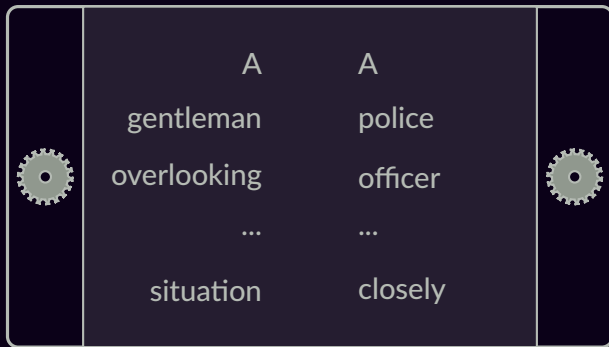
Structured Attention for Alignments

NLI

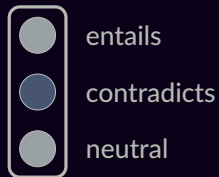
premise: A gentleman overlooking a neighborhood situation.
hypothesis: A police officer watches a situation closely.

input

(P, H)



output



(Model: ESIM (Chen et al., 2017))

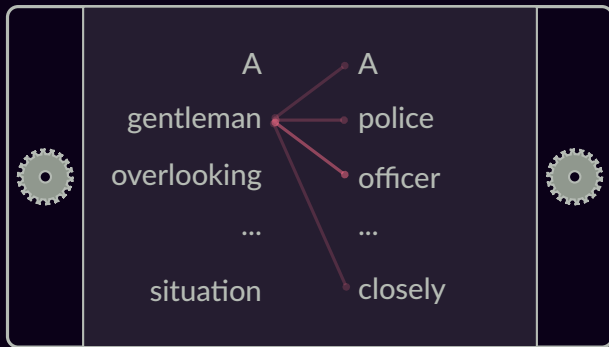
Structured Attention for Alignments

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 (Chen et al., 2017))

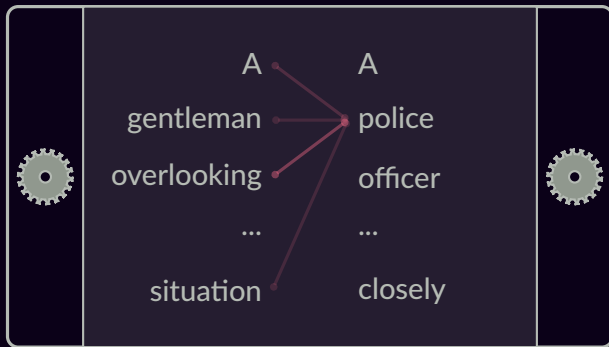
Structured Attention for Alignments

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 (Chen et al., 2017))

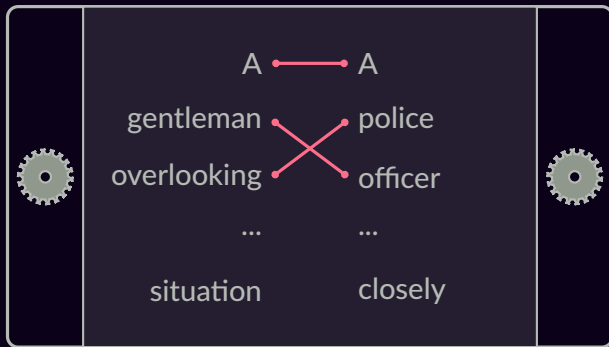
Structured Attention for Alignments

NLI

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

input

(P, H)



output



entails

contradicts

neutral

(Proposed model: global matching)

In code:

```
# Xp: (n_prem x k)
# Xh: (n_hypo x k)

Z = Xp @ Xh.t()

Up = softmax(Z, dim=1)
Uh = softmax(Z, dim=0)

Xp = cat([Xp, Up @ Xh])
Xh = cat([Xh, Uh.t() @ Xp])

# ...
```

In code:

```
# Xp: (n_prem x k)
# Xh: (n_hypo x k)
```

```
Z = Xp @ Xh.t()
```

```
Up = softmax(Z, dim=1)
Uh = softmax(Z, dim=0)
```

```
Xp = cat([Xp, Up @ Xh])
Xh = cat([Xh, Uh.t() @ Xp])
```

```
# ...
```

```
# Xp: (n_prem x k)
# Xh: (n_hypo x k)
```

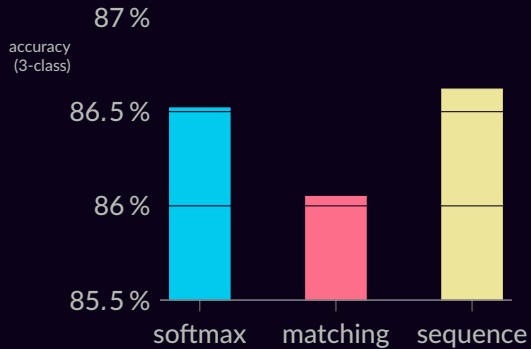
```
Z = Xp @ Xh.t()
```

```
U = sparsemap_matching(Z)
```

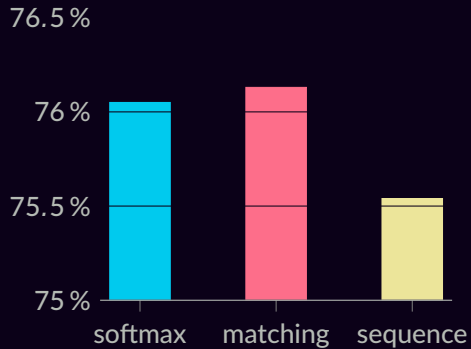
```
Xp = cat([Xp, U @ Xh])
Xh = cat([Xh, U.t() @ Xp])
```

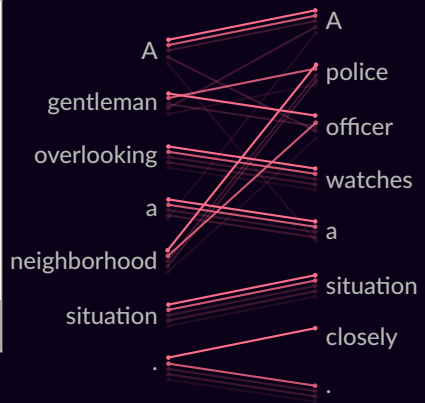
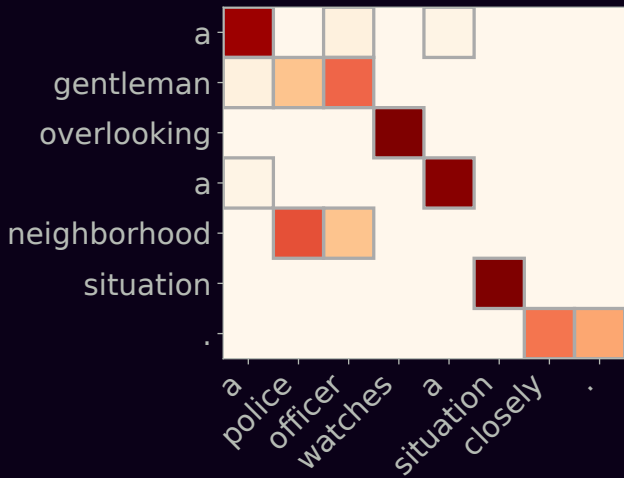
```
# ...
```

SNLI



MultiNLI





**Dynamically inferring
the computation graph**

Dependency TreeLSTM

(Tai et al., 2015)

closely related to GCNs, e.g.

(Kipf and Welling, 2017)

(Marcheggiani and Titov, 2017)



The bears eat the pretty ones

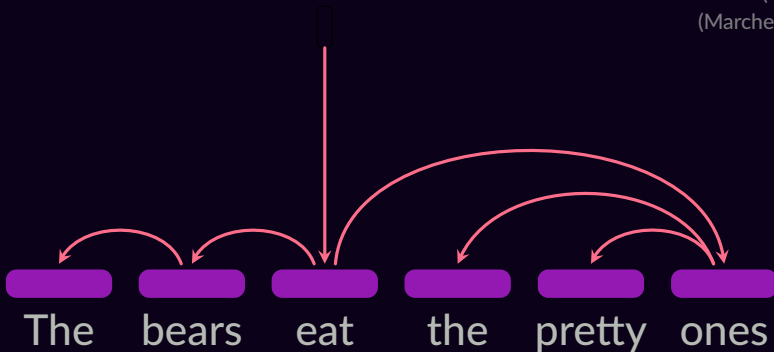
Dependency TreeLSTM

(Tai et al., 2015)

closely related to GCNs, e.g.

(Kipf and Welling, 2017)

(Marcheggiani and Titov, 2017)



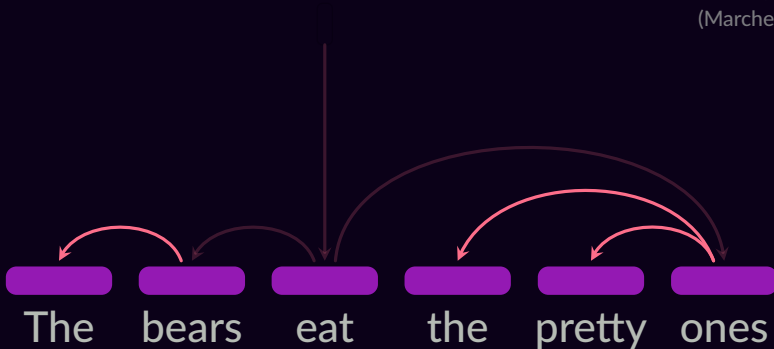
Dependency TreeLSTM

(Tai et al., 2015)

closely related to GCNs, e.g.

(Kipf and Welling, 2017)

(Marcheggiani and Titov, 2017)



Dependency TreeLSTM

(Tai et al., 2015)

closely related to GCNs, e.g.

(Kipf and Welling, 2017)

(Marcheggiani and Titov, 2017)



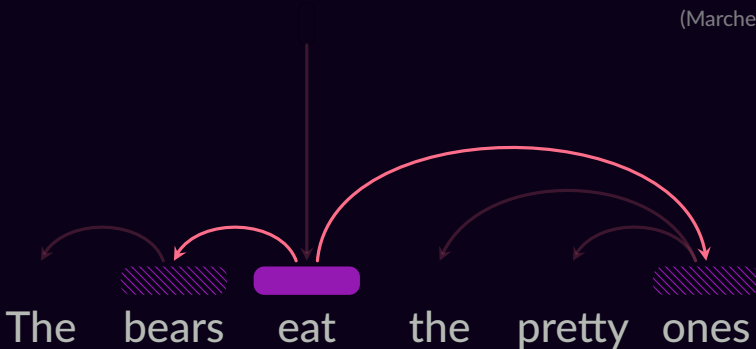
Dependency TreeLSTM

(Tai et al., 2015)

closely related to GCNs, e.g.

(Kipf and Welling, 2017)

(Marcheggiani and Titov, 2017)



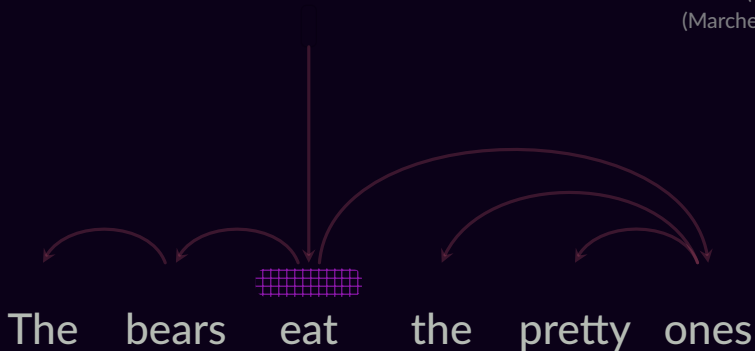
Dependency TreeLSTM

(Tai et al., 2015)

closely related to GCNs, e.g.

(Kipf and Welling, 2017)

(Marcheggiani and Titov, 2017)



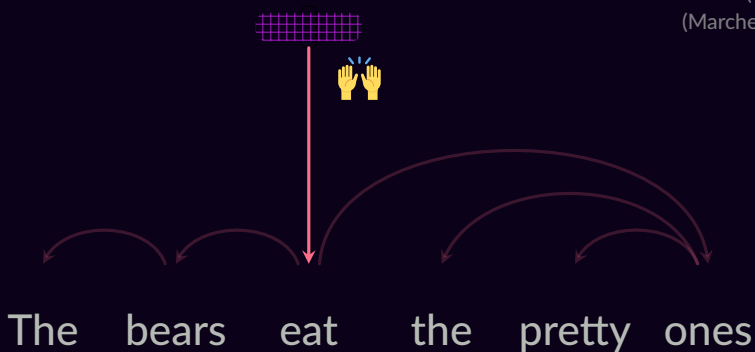
Dependency TreeLSTM

(Tai et al., 2015)

closely related to GCNs, e.g.

(Kipf and Welling, 2017)

(Marcheggiani and Titov, 2017)

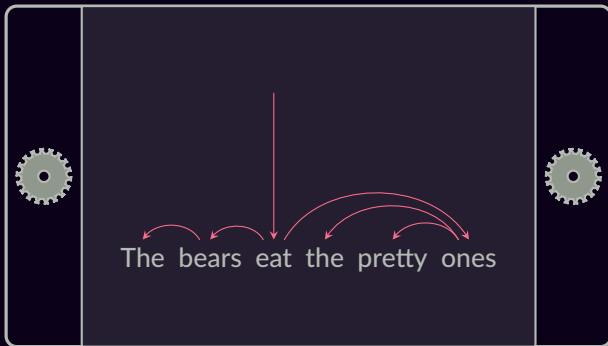


Latent Dependency TreeLSTM

(Niculae, Martins, and Cardie, 2018)

input

x



output

y

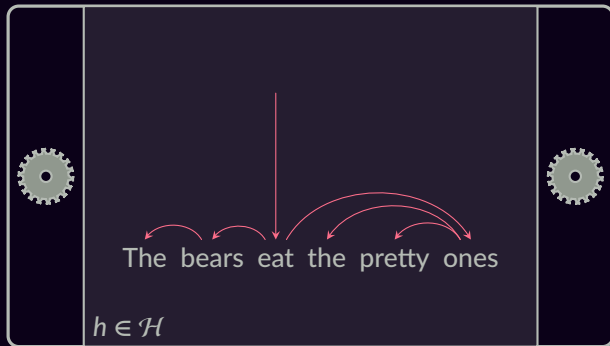
Latent Dependency TreeLSTM

(Niculae, Martins, and Cardie, 2018)

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

input

x



output

y

Structured Latent Variable Models

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


Structured Latent Variable Models

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

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

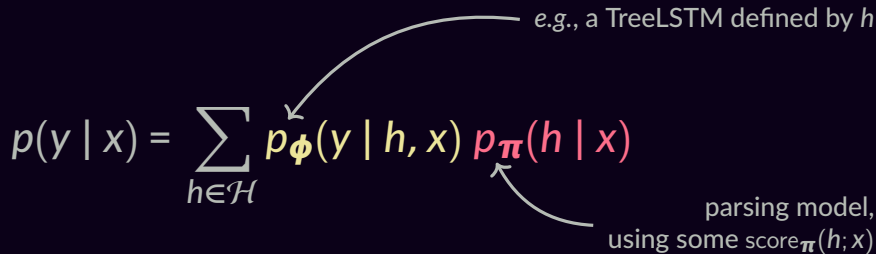


Structured Latent Variable Models

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

e.g., a TreeLSTM defined by h

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



Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

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

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

Exponentially large sum!

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

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

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

How to define p_{π} ?

idea 1

idea 2

idea 3

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

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

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

How to define p_{π} ?

$$\sum_{h \in \mathcal{H}}$$

idea 1

idea 2

idea 3

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

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

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

How to define p_{π} ?

$$\sum_{h \in \mathcal{H}} \frac{\partial p(y | x)}{\partial \pi}$$

idea 1

idea 2

idea 3

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

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

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

How to define p_{π} ?

$$\sum_{h \in \mathcal{H}} \frac{\partial p(y | x)}{\partial \pi}$$

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

argmax

idea 2

idea 3

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

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

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

How to define p_{π} ?

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

argmax

$$\sum_{h \in \mathcal{H}} \frac{\partial p(y | x)}{\partial \pi}$$



idea 2

idea 3

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

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

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

How to define p_{π} ?

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

argmax

idea 2

idea 3

$$\sum_{h \in \mathcal{H}} \frac{\partial p(y | x)}{\partial \pi}$$



Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

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

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

How to define p_{π} ?

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

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

idea 3

argmax

softmax

$$\sum_{h \in \mathcal{H}} \frac{\partial p(y | x)}{\partial \pi}$$



Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

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

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

How to define p_{π} ?

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

$$\sum_{h \in \mathcal{H}} \frac{\partial p(y | x)}{\partial \pi}$$



Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

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

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

How to define p_{π} ?

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

$$\sum_{h \in \mathcal{H}} \frac{\partial p(y | x)}{\partial \pi}$$



Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

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

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

How to define p_{π} ?

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

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

idea 3

argmax

softmax

SparseMAP

$$\sum_{h \in \mathcal{H}} \frac{\partial p(y | x)}{\partial \pi}$$



SparseMAP

 $= .7$

 $+ .3$



SparseMAP

 $= .7$

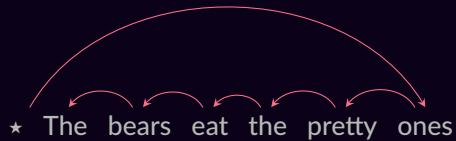
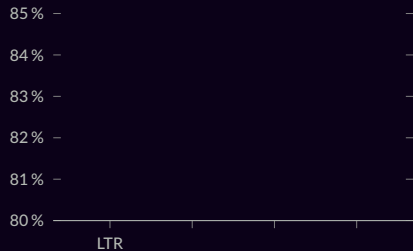
 $+ .3$

 $+ 0$  $+ \dots$

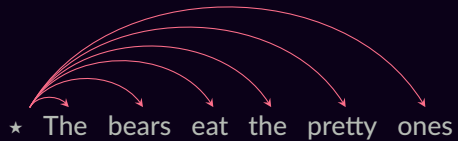
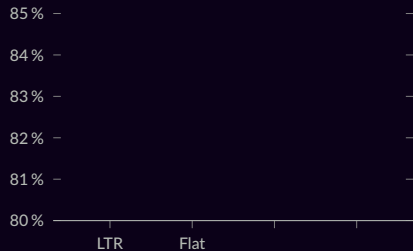
SparseMAP

$$\begin{aligned}
 & \text{Diagram 1} = .7 \quad \text{Diagram 2} + .3 \quad \text{Diagram 3} + 0 \text{Diagram 4} + \dots \\
 p(y \mid x) = & .7 p_{\phi}(y \mid \text{Diagram 1}) + .3 p_{\phi}(y \mid \text{Diagram 2})
 \end{aligned}$$

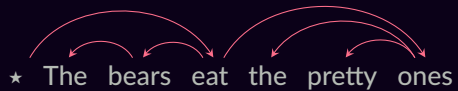
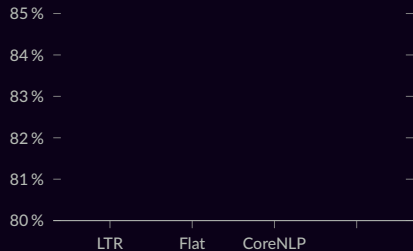
The diagrams consist of three red dots in a horizontal row. In the first diagram, a red curved arrow points from the first dot to the second, and another red curved arrow points from the second dot to the third. In the second diagram, a red curved arrow points from the first dot to the third, and another red curved arrow points from the second dot to the third. In the third diagram, a red curved arrow points from the first dot to the second, and another red curved arrow points from the third dot to the second. In the fourth diagram, a red curved arrow points from the first dot to the second, and another red curved arrow points from the second dot to the third.



Left-to-right: regular LSTM

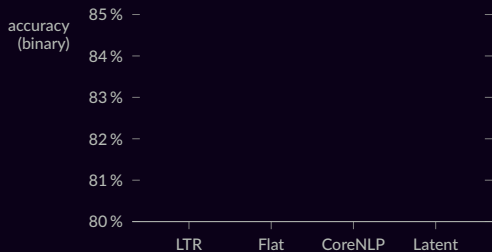


Flat: bag-of-words-like

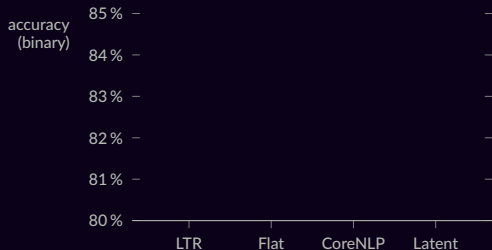


CoreNLP: off-line parser

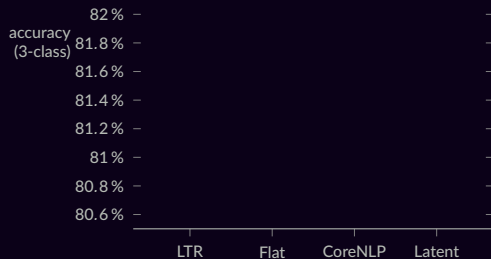
Sentiment classification (SST)



Sentiment classification (SST)



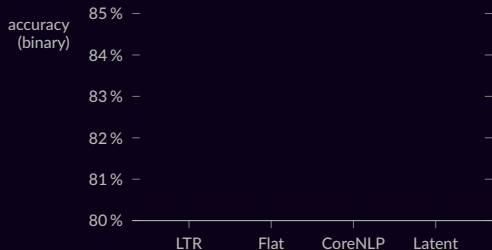
Natural Language Inference (SNLI)



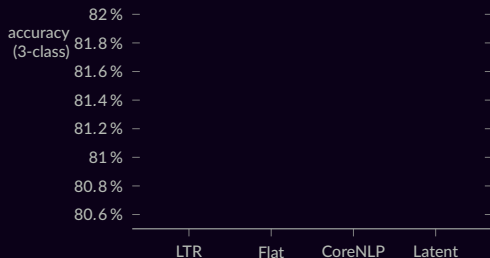
Sentence pair classification (P, H)

$$p(y \mid P, H) = \sum_{h_P \in \mathcal{H}(P)} \sum_{h_H \in \mathcal{H}(H)} p_{\Phi}(y \mid h_P, h_H) p_{\pi}(h_P \mid P) p_{\pi}(h_H \mid H)$$

Sentiment classification (SST)



Natural Language Inference (SNLI)

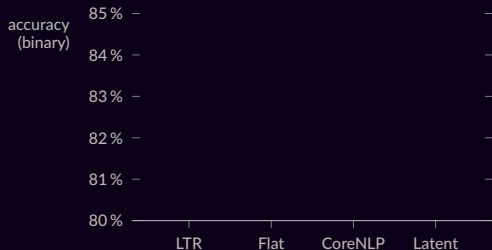


Reverse dictionary lookup

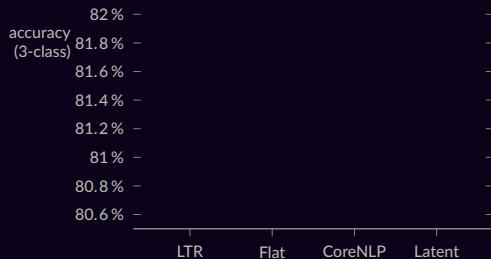
given word description, predict word embedding (Hill et al., 2016)

instead of $p(y | x)$, we model $\mathbb{E}_{p_{\pi}} \mathbf{g}(x) = \sum_{h \in \mathcal{H}} \mathbf{g}(x; h) p_{\pi}(h | x)$

Sentiment classification (SST)

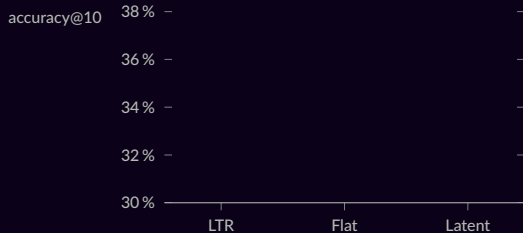


Natural Language Inference (SNLI)

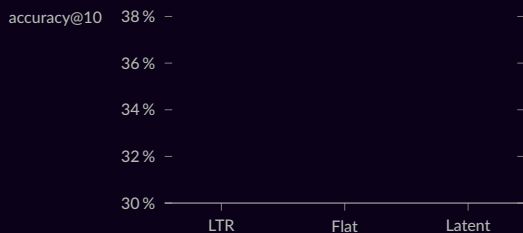


Reverse dictionary lookup

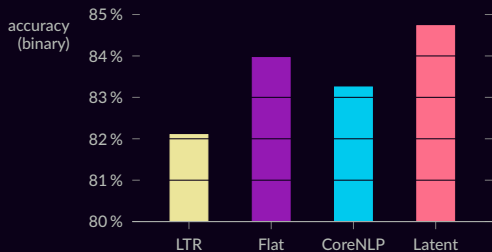
(definitions)



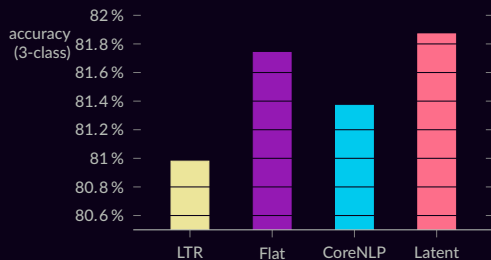
(concepts)



Sentiment classification (SST)

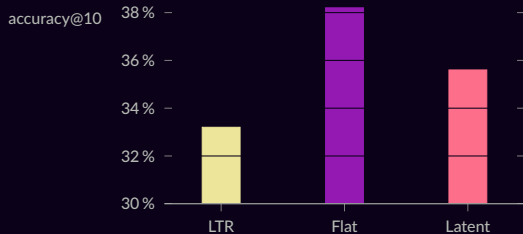


Natural Language Inference (SNLI)

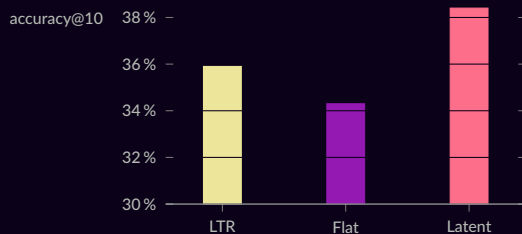


Reverse dictionary lookup

(definitions)

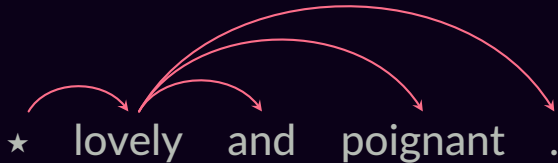


(concepts)



Syntax vs. Composition Order

CoreNLP parse, $p = 21.4\%$

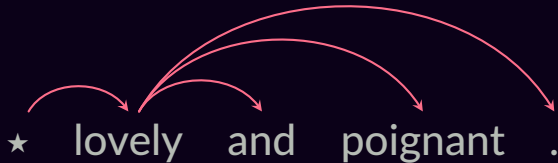


Syntax vs. Composition Order

$p = 22.6\%$

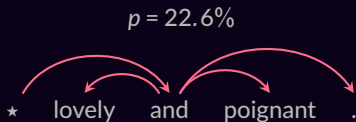


CoreNLP parse, $p = 21.4\%$

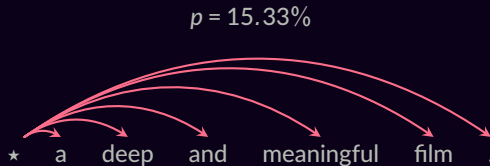


...

Syntax vs. Composition Order



CoreNLP parse, $p = 21.4\%$



$p = 15.27\%$



...
CoreNLP parse, $p = 0\%$



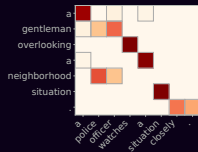
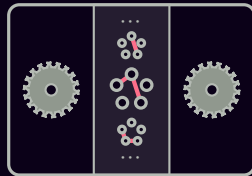
Conclusions

Differentiable & sparse
structured inference

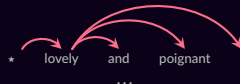
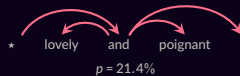
Generic, extensible algorithms

Interpretable structured attention

Dynamically-inferred
computation graphs



$p = 22.6\%$



✉ vlad@vene.ro

🏠 <https://vene.ro>



github.com/vene/sparsemap



@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.

Danskin's Theorem

Let $\phi : \mathbb{R}^d \times \mathcal{Z} \rightarrow \mathbb{R}$, $\mathcal{Z} \subset \mathbb{R}^d$ compact.

$$\partial \max_{\mathbf{z} \in \mathcal{Z}} \phi(\mathbf{x}, \mathbf{z}) = \text{conv} \{ \nabla_{\mathbf{x}} \phi(\mathbf{x}, \mathbf{z}^*) \mid \mathbf{z}^* \in \arg \max_{\mathbf{z} \in \mathcal{Z}} \phi(\mathbf{x}, \mathbf{z}) \}.$$

Example: maximum of a vector

Danskin's Theorem

Let $\phi : \mathbb{R}^d \times \mathcal{Z} \rightarrow \mathbb{R}$, $\mathcal{Z} \subset \mathbb{R}^d$ compact.

$$\partial \max_{\mathbf{z} \in \mathcal{Z}} \phi(\mathbf{x}, \mathbf{z}) = \text{conv} \{ \nabla_{\mathbf{x}} \phi(\mathbf{x}, \mathbf{z}^*) \mid \mathbf{z}^* \in \arg \max_{\mathbf{z} \in \mathcal{Z}} \phi(\mathbf{x}, \mathbf{z}) \}.$$

Example: maximum of a vector

$$\begin{aligned} \partial \max_{j \in [d]} \theta_j &= \partial \max_{\mathbf{p} \in \Delta} \mathbf{p}^\top \boldsymbol{\theta} \\ &= \partial \max_{\mathbf{p} \in \Delta} \phi(\mathbf{p}, \boldsymbol{\theta}) \\ &= \text{conv} \{ \nabla_{\boldsymbol{\theta}} \phi(\mathbf{p}^*, \boldsymbol{\theta}) \} \\ &= \text{conv} \{ \mathbf{p}^* \} \end{aligned}$$

Danskin's Theorem

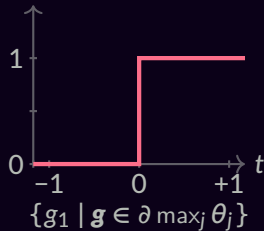
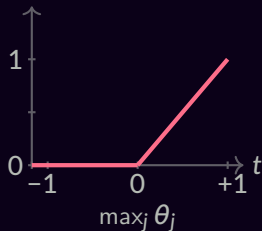
Let $\phi : \mathbb{R}^d \times \mathcal{Z} \rightarrow \mathbb{R}$, $\mathcal{Z} \subset \mathbb{R}^d$ compact.

$$\partial \max_{\mathbf{z} \in \mathcal{Z}} \phi(\mathbf{x}, \mathbf{z}) = \text{conv} \{ \nabla_{\mathbf{x}} \phi(\mathbf{x}, \mathbf{z}^*) \mid \mathbf{z}^* \in \arg \max_{\mathbf{z} \in \mathcal{Z}} \phi(\mathbf{x}, \mathbf{z}) \}.$$

Example: maximum of a vector

$$\begin{aligned} \partial \max_{j \in [d]} \theta_j &= \partial \max_{\mathbf{p} \in \Delta} \mathbf{p}^\top \boldsymbol{\theta} \\ &= \partial \max_{\mathbf{p} \in \Delta} \phi(\mathbf{p}, \boldsymbol{\theta}) \\ &= \text{conv} \{ \nabla_{\boldsymbol{\theta}} \phi(\mathbf{p}^*, \boldsymbol{\theta}) \} \\ &= \text{conv} \{ \mathbf{p}^* \} \end{aligned}$$

$$\boldsymbol{\theta} = [t, 0]$$



Fusedmax

$$\text{fusedmax}(\boldsymbol{\theta}) = \arg \max_{\mathbf{p} \in \Delta} \mathbf{p}^\top \boldsymbol{\theta} - \frac{1}{2} \|\mathbf{p}\|_2^2 - \sum_{2 \leq j \leq d} |p_j - p_{j-1}|$$

$$= \arg \min_{\mathbf{p} \in \Delta} \|\mathbf{p} - \boldsymbol{\theta}\|_2^2 + \sum_{2 \leq j \leq d} |p_j - p_{j-1}|$$

$$\text{prox}_{\text{fused}}(\boldsymbol{\theta}) = \arg \min_{\mathbf{p} \in \mathbb{R}^d} \|\mathbf{p} - \boldsymbol{\theta}\|_2^2 + \sum_{2 \leq j \leq d} |p_j - p_{j-1}|$$

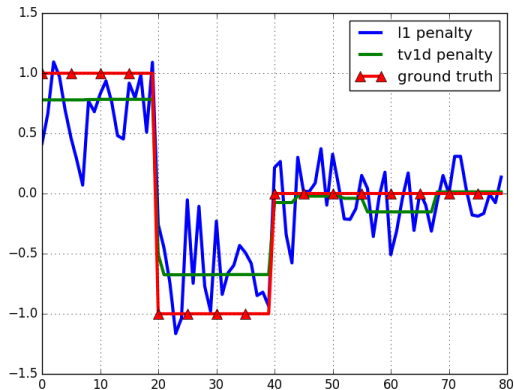
Proposition: $\text{fusedmax}(\boldsymbol{\theta}) = \text{sparsemax}(\text{prox}_{\text{fused}}(\boldsymbol{\theta}))$

(Niculae and Blondel, 2017)

fusedmax(

prox_{fused}(

Proposi



“Fused Lasso” a.k.a. 1-d Total Variation

(Tibshirani et al., 2005)

(Niculae and Blondel, 2017)

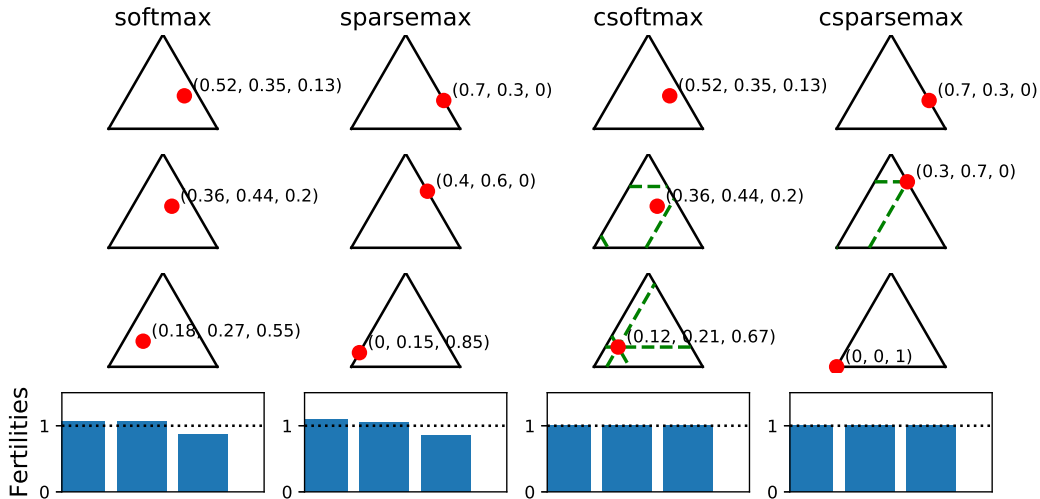
$|p_j - p_{j-1}|$

$|p_{j-1}|$

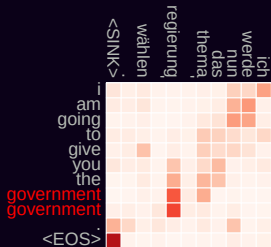
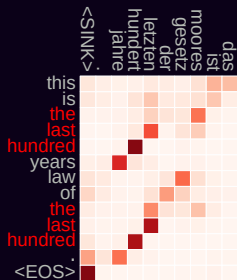
$|p_{j-1}|$

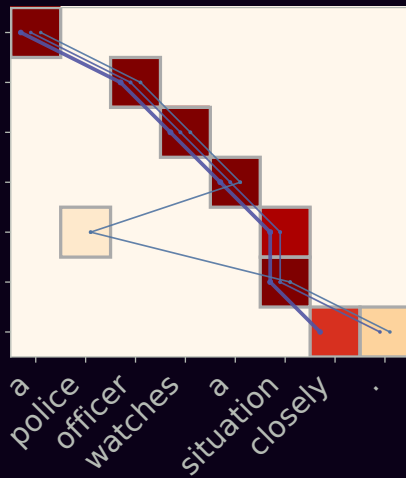
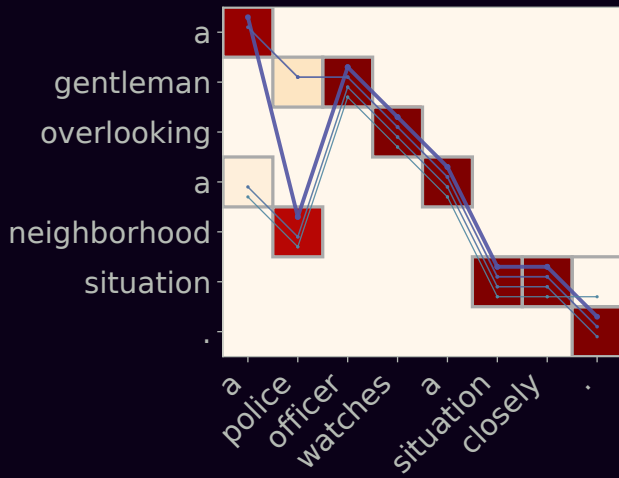
fused(θ)

Example: Source Sentence with Three Words



e.g., fertility constraints for NMT





Structured Output Prediction

SparseMAP

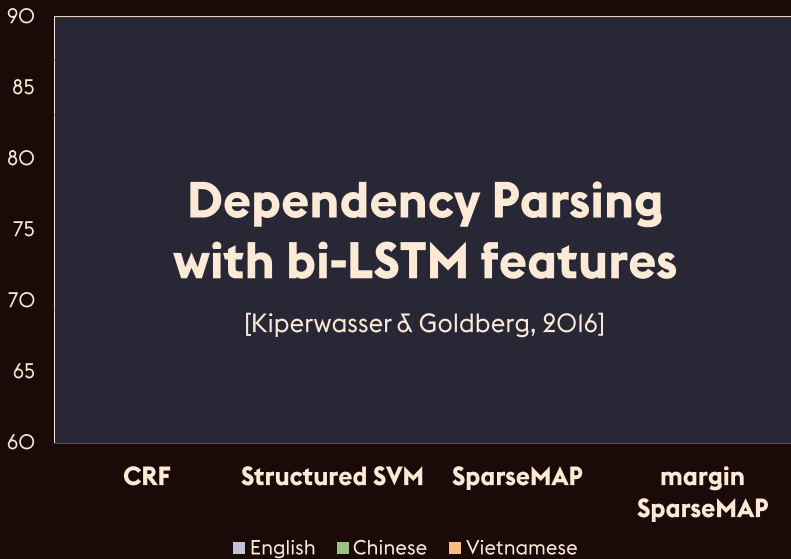
$$L_A(\boldsymbol{\eta}, \bar{\boldsymbol{\mu}}) = \max_{\boldsymbol{\mu} \in \mathcal{M}} \left\{ \boldsymbol{\eta}^\top \boldsymbol{\mu} - 1/2 \|\boldsymbol{\mu}\|^2 \right\} \\ - \boldsymbol{\eta}^\top \bar{\boldsymbol{\mu}} + 1/2 \|\bar{\boldsymbol{\mu}}\|^2$$

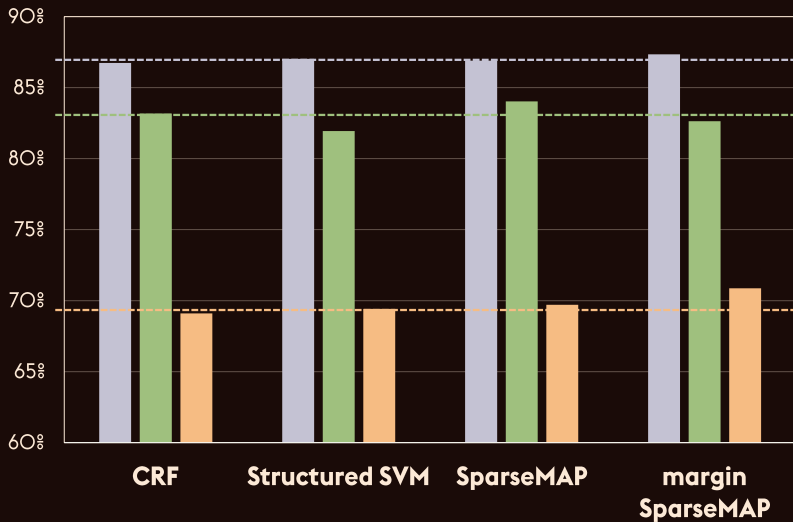
Instance of a structured Fenchel-Young loss, like CRF, SVM, etc. (Blondel, Martins, and Niculae, 2019)

Structured Output Prediction

$$\begin{aligned} \text{SparseMAP} \quad L_A(\boldsymbol{\eta}, \bar{\boldsymbol{\mu}}) &= \max_{\boldsymbol{\mu} \in \mathcal{M}} \left\{ \boldsymbol{\eta}^\top \boldsymbol{\mu} - 1/2 \|\boldsymbol{\mu}\|^2 \right\} \\ &\quad - \boldsymbol{\eta}^\top \bar{\boldsymbol{\mu}} + 1/2 \|\bar{\boldsymbol{\mu}}\|^2 \\ \text{cost-SparseMAP} \quad L_A^\rho(\boldsymbol{\eta}, \bar{\boldsymbol{\mu}}) &= \max_{\boldsymbol{\mu} \in \mathcal{M}} \left\{ \boldsymbol{\eta}^\top \boldsymbol{\mu} - 1/2 \|\boldsymbol{\mu}\|^2 + \rho(\boldsymbol{\mu}, \bar{\boldsymbol{\mu}}) \right\} \\ &\quad - \boldsymbol{\eta}^\top \bar{\boldsymbol{\mu}} + 1/2 \|\bar{\boldsymbol{\mu}}\|^2 \end{aligned}$$

Instance of a structured Fenchel-Young loss, like CRF, SVM, etc. (Blondel, Martins, and Niculae, 2019)



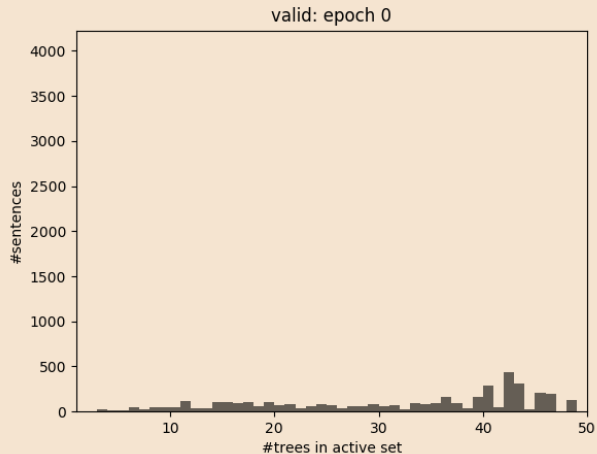
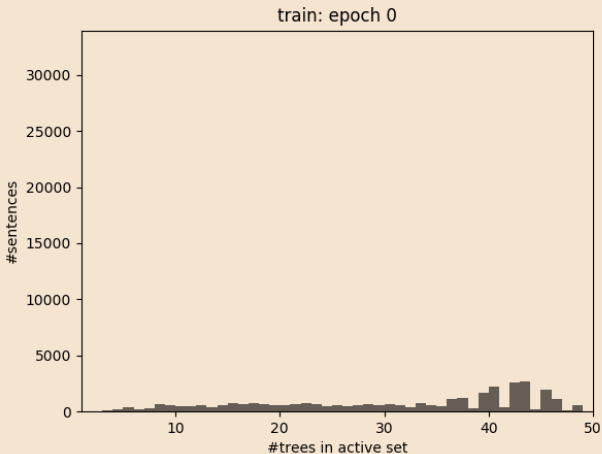


Unlabeled Accuracy (UAS)
Universal Dependencies dataset

■ English ■ Chinese ■ Vietnamese

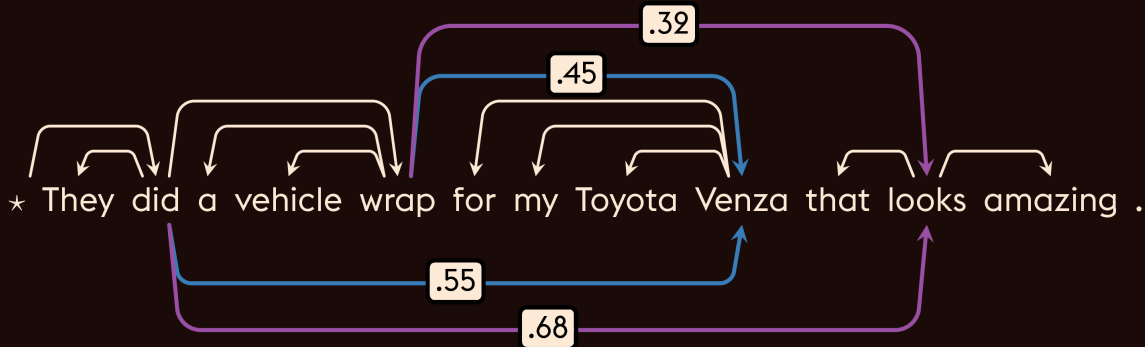
Sparse Structured Output Prediction

As models train, inference gets sparser!



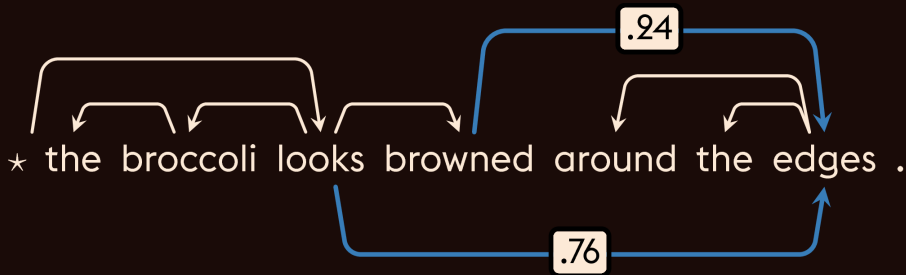
Sparse Structured Output Prediction

Inference captures linguistic ambiguity!



Sparse Structured Output Prediction

Inference captures linguistic ambiguity!



References I

- Bertsekas, Dimitri P (1999). *Nonlinear Programming*. Athena Scientific Belmont.
- Blondel, Mathieu, André FT Martins, and Vlad Niculae (2019). “Learning with Fenchel-Young Losses”. In: *preprint arXiv:1901.02324*.
- Chen, Qian et al. (2017). “Enhanced LSTM for natural language inference”. In: *Proc. of ACL*.
- Danskin, John M (1966). “The theory of max-min, with applications”. In: *SIAM Journal on Applied Mathematics* 14.4, pp. 641–664.
- Dantzig, George B, Alex Orden, and Philip Wolfe (1955). “The generalized simplex method for minimizing a linear form under linear inequality restraints”. In: *Pacific Journal of Mathematics* 5.2, pp. 183–195.
- Frank, Marguerite and Philip Wolfe (1956). “An algorithm for quadratic programming”. In: *Nav. Res. Log.* 3.1-2, pp. 95–110.
- Hill, Felix et al. (2016). “Learning to understand phrases by embedding the dictionary”. In: *TACL* 4.1, pp. 17–30.
- Kim, Yoon et al. (2017). “Structured attention networks”. In: *Proc. of ICLR*.

References II

- Kipf, Thomas N. and Max Welling (2017). "Semi-supervised classification with graph convolutional networks". In: *Proc. of ICLR*.
- Koo, Terry et al. (2007). "Structured prediction models via the matrix-tree theorem". In: *Proc. of EMNLP*.
- Lacoste-Julien, Simon and Martin Jaggi (2015). "On the global linear convergence of Frank-Wolfe optimization variants". In: *Proc. of NeurIPS*.
- Liu, Yang and Mirella Lapata (2018). "Learning structured text representations". In: *TACL 6*, pp. 63–75.
- Malaviya, Chaitanya, Pedro Ferreira, and André F. T. Martins (2018). "Sparse and constrained attention for neural machine translation". In: *Proc. of ACL*.
- Marcheggiani, Diego and Ivan Titov (2017). "Encoding sentences with graph convolutional networks for semantic role labeling". In: *Proc. of EMNLP*.
- Martins, André FT and Ramón Fernandez Astudillo (2016). "From softmax to sparsemax: A sparse model of attention and multi-label classification". In: *Proc. of ICML*.

References III

- Martins, André FT and Julia Kreutzer (2017). “Learning What’s Easy: Fully Differentiable Neural Easy-First Taggers”. In: *Proc. of EMNLP*, pp. 349–362.
- McDonald, Ryan T and Giorgio Satta (2007). “On the complexity of non-projective data-driven dependency parsing”. In: *Proc. of ICPT*.
- Niculescu, Vlad and Mathieu Blondel (2017). “A regularized framework for sparse and structured neural attention”. In: *Proc. of NeurIPS*.
- Niculescu, Vlad, André FT Martins, Mathieu Blondel, et al. (2018). “SparseMAP: Differentiable sparse structured inference”. In: *Proc. of ICML*.
- Niculescu, Vlad, André FT Martins, and Claire Cardie (2018). “Towards dynamic computation graphs via sparse latent structure”. In: *Proc. of EMNLP*.
- Nocedal, Jorge and Stephen Wright (1999). *Numerical Optimization*. Springer New York.
- Rabiner, Lawrence R. (1989). “A tutorial on Hidden Markov Models and selected applications in speech recognition”. In: *P. IEEE* 77.2, pp. 257–286.
- Smith, David A and Noah A Smith (2007). “Probabilistic models of nonprojective dependency trees”. In: *Proc. of EMNLP*.

References IV

- Tai, Kai Sheng, Richard Socher, and Christopher D Manning (2015). “Improved semantic representations from tree-structured Long Short-Term Memory networks”. In: *Proc. of ACL-IJCNLP*.
- Taskar, Ben (2004). “Learning structured prediction models: A large margin approach”. PhD thesis. Stanford University.
- Tibshirani, Robert et al. (2005). “Sparsity and smoothness via the fused lasso”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.1, pp. 91–108.
- Valiant, Leslie G (1979). “The complexity of computing the permanent”. In: *Theor. Comput. Sci.* 8.2, pp. 189–201.
- Vinyes, Marina and Guillaume Obozinski (2017). “Fast column generation for atomic norm regularization”. In: *Proc. of AISTATS*.
- Wolfe, Philip (1976). “Finding the nearest point in a polytope”. In: *Mathematical Programming* 11.1, pp. 128–149.