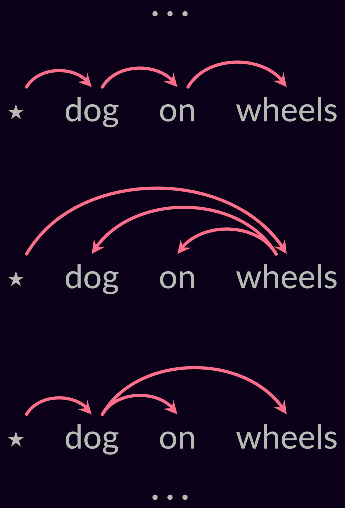
Learning with Sparse Latent Structure

Vlad Niculae
Instituto de Telecomunicações

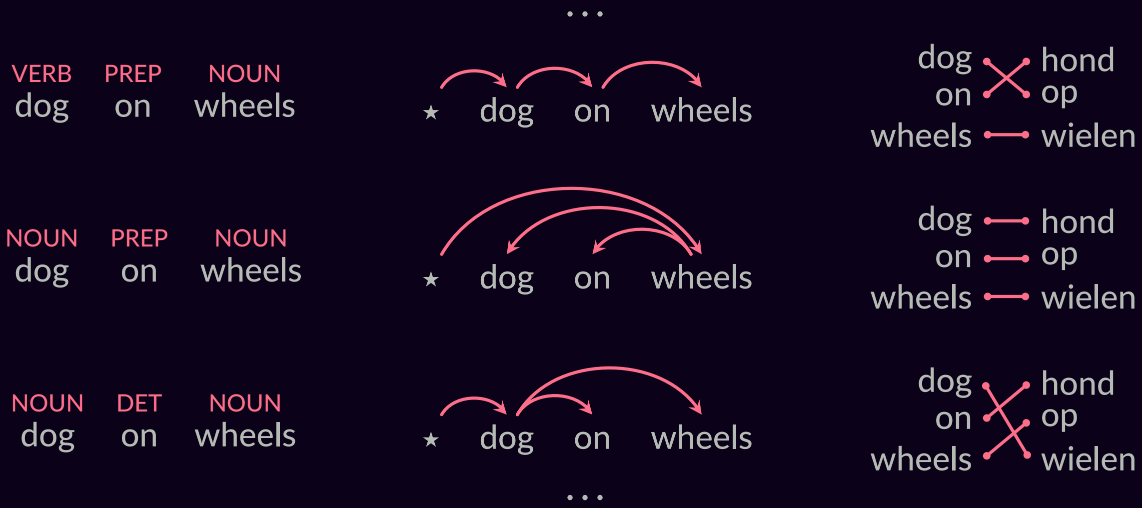
Work with: André Martins, Claire Cardie, Mathieu Blondel

github.com/vene/sparsemap @vnfrombucharest

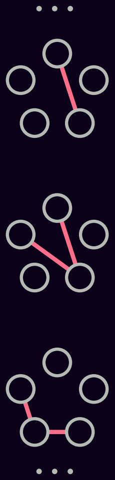
Structured Prediction



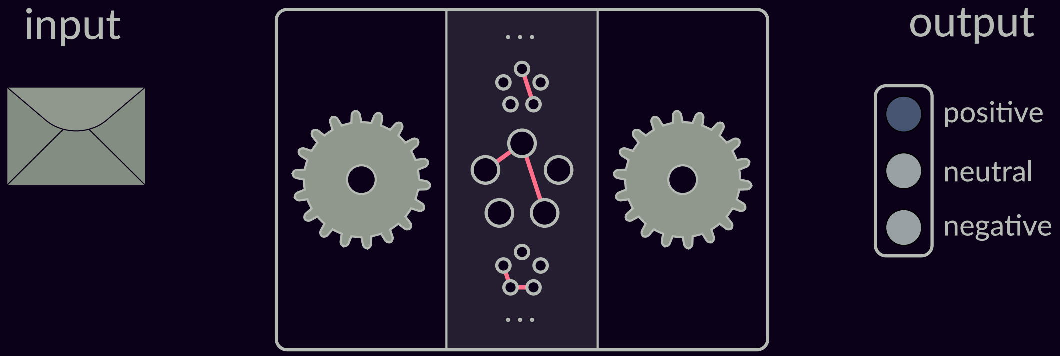
Structured Prediction



Structured Prediction



Latent Structure Models

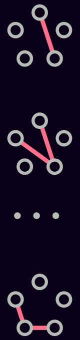


record scratch

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?

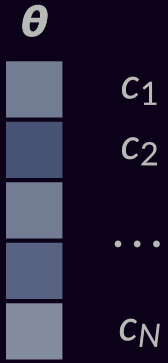
$$c_1$$

$$c_2$$

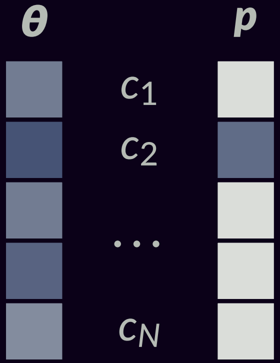
$$\cdots$$

$$c_N$$

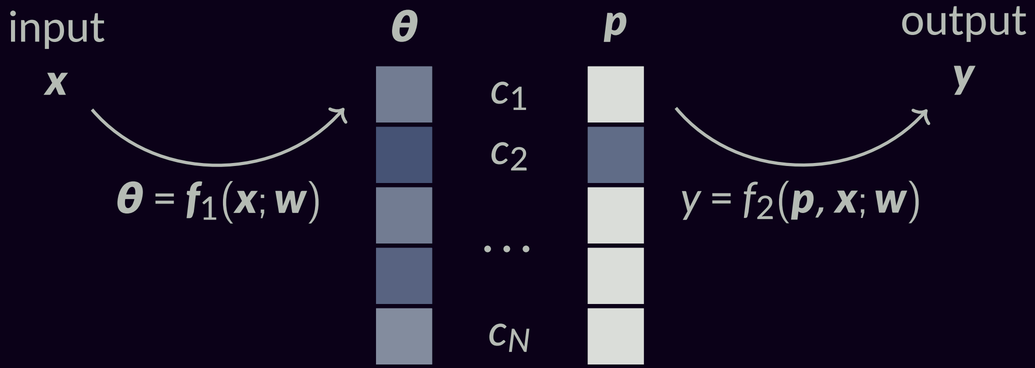
How to select an item from a set?



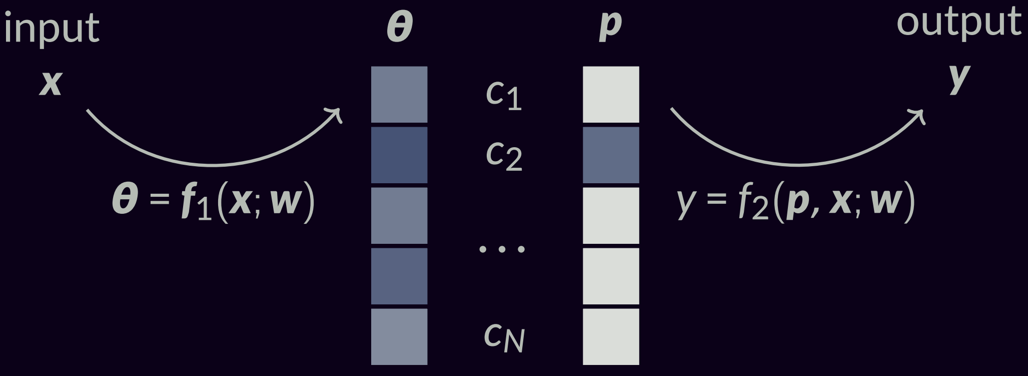
How to select an item from a set?



How to select an item from a set?

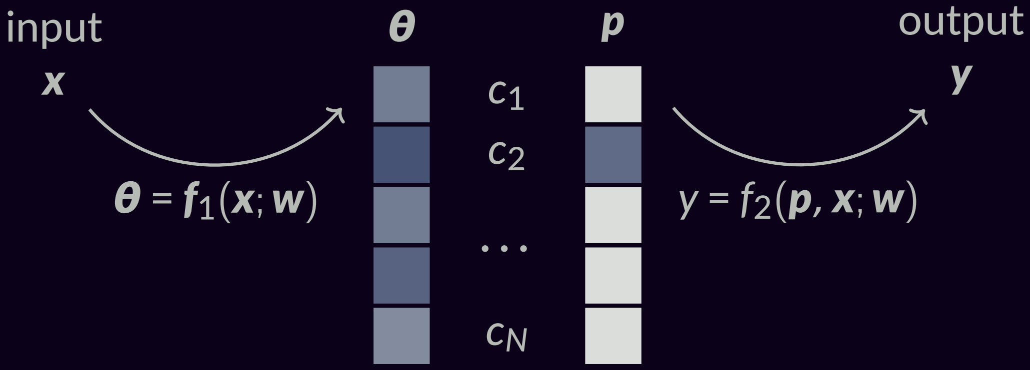


How to select an item from a set?



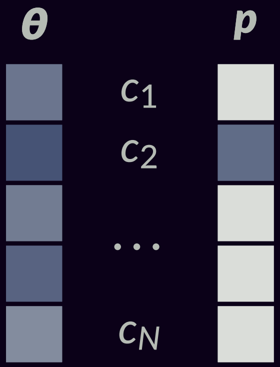
$$\frac{\partial y}{\partial \mathbf{w}} = ?$$

How to select an item from a set?



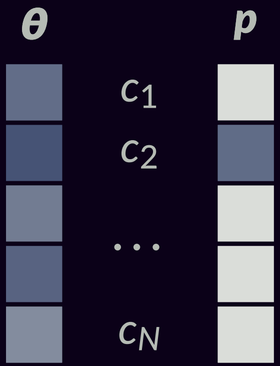
$\frac{\partial y}{\partial \boldsymbol{w}} = ?$ or, essentially, $\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = ?$

Argmax



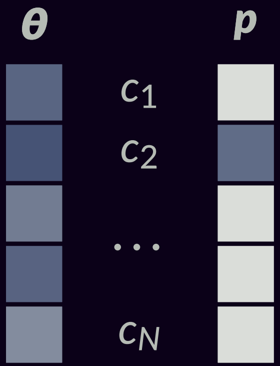
$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = ?$$

Argmax



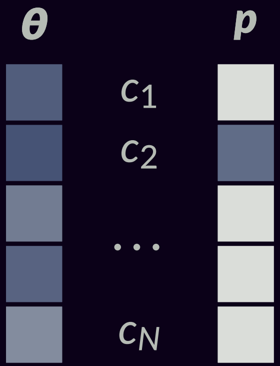
$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = ?$$

Argmax



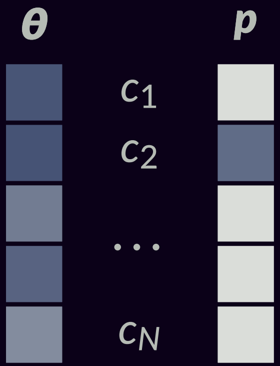
$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = ?$$

Argmax



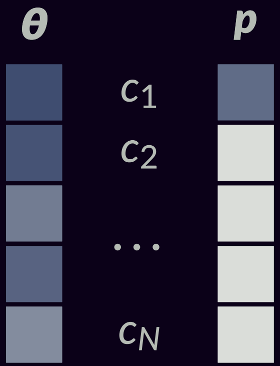
$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = ?$$

Argmax



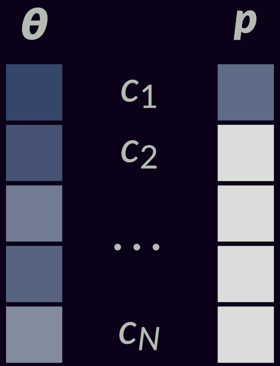
$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = ?$$

Argmax



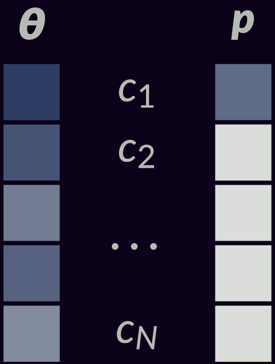
$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = ?$$

Argmax



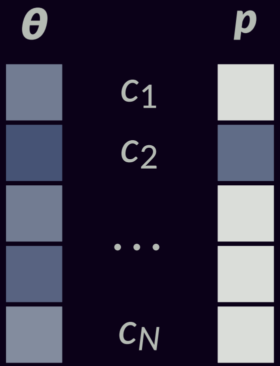
$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = ?$$

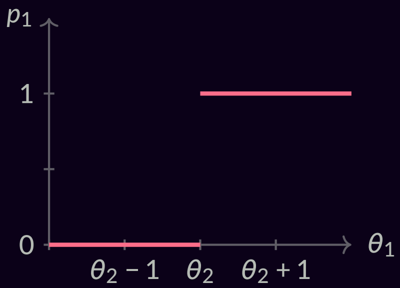
Argmax



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

Argmax

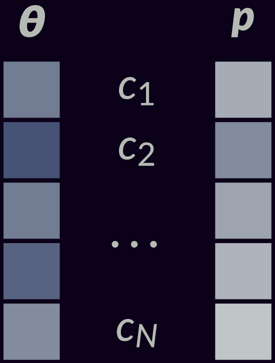


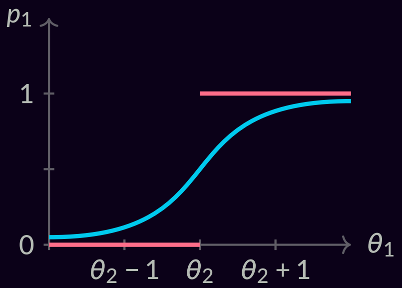


$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = \mathbf{0}$$

Argmax vs. Softmax

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





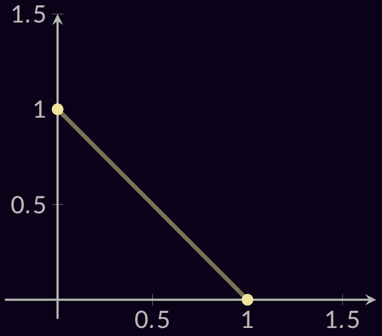
$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = \operatorname{diag}(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^\top$$

Variational Form of Argmax

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

Variational Form of Argmax

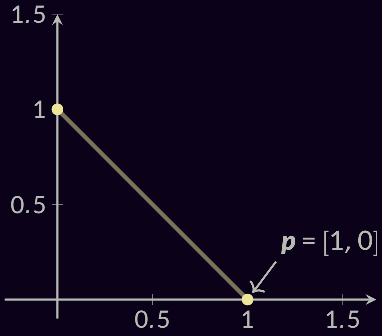
$$\triangle = \{\boldsymbol{p} \in \mathbb{R}^N : \boldsymbol{p} \geq \boldsymbol{0},\ \boldsymbol{1}^\top \boldsymbol{p} = 1\}$$



$N = 2$

Variational Form of Argmax

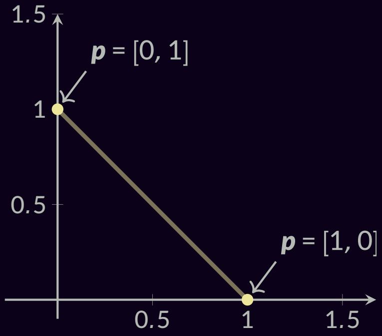
$$\Delta = \{\boldsymbol{p} \in \mathbb{R}^N : \boldsymbol{p} \geq \boldsymbol{0},\ \boldsymbol{1}^\top \boldsymbol{p} = 1\}$$



$N = 2$

Variational Form of Argmax

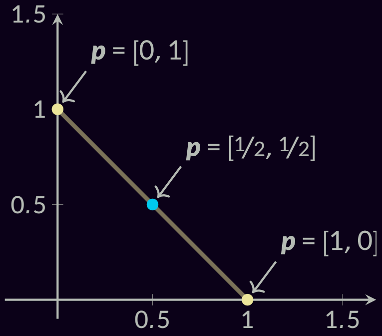
$$\Delta = \{\boldsymbol{p} \in \mathbb{R}^N : \boldsymbol{p} \geq \boldsymbol{0},\ \boldsymbol{1}^\top \boldsymbol{p} = 1\}$$



$N = 2$

Variational Form of Argmax

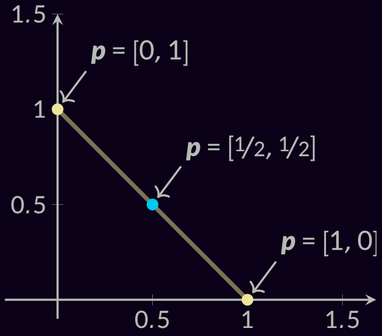
$$\triangle = \{\boldsymbol{p} \in \mathbb{R}^N : \boldsymbol{p} \geq \mathbf{0},\ \mathbf{1}^\top \boldsymbol{p} = 1\}$$



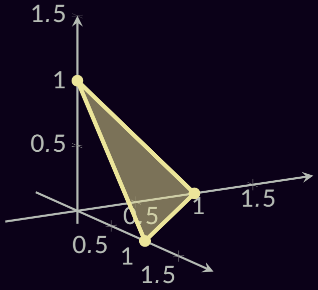
$N = 2$

Variational Form of Argmax

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



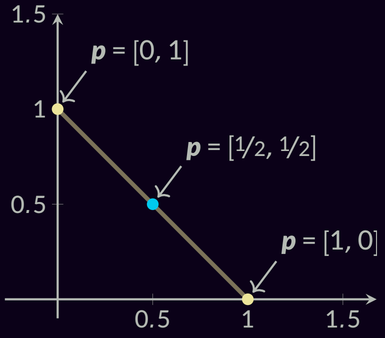
$N = 2$



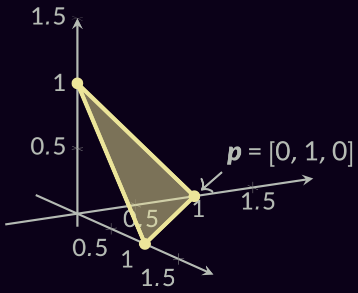
$N = 3$

Variational Form of Argmax

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



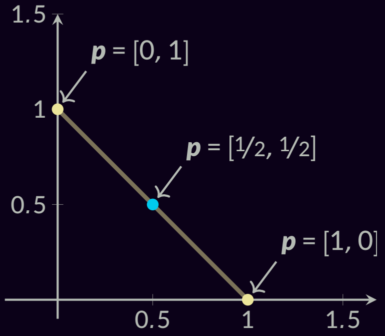
$N = 2$



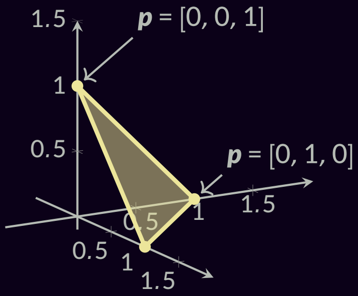
$N = 3$

Variational Form of Argmax

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



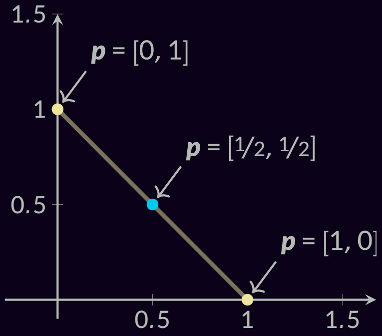
$N = 2$



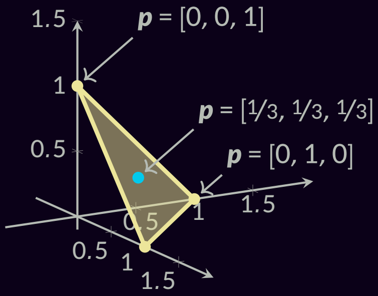
$N = 3$

Variational Form of Argmax

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



$N = 2$

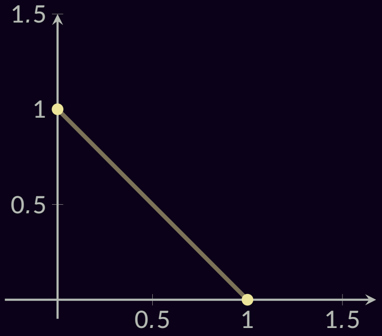


$N = 3$

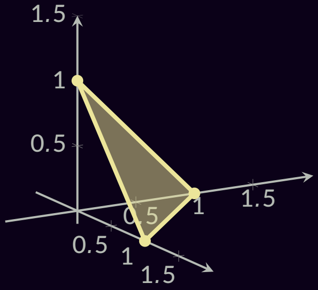
Variational Form of Argmax

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

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



$N = 2$

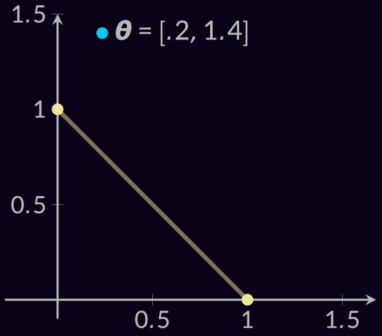


$N = 3$

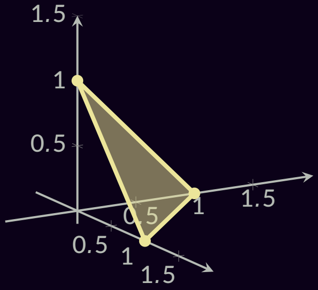
Variational Form of Argmax

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

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



$N = 2$

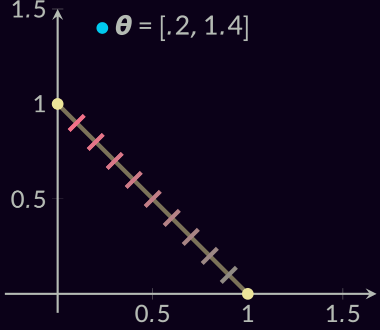


$N = 3$

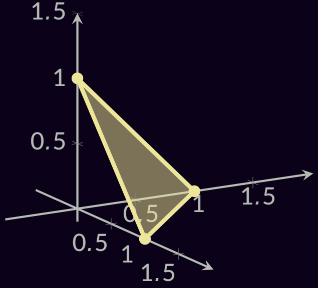
Variational Form of Argmax

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

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



$N = 2$

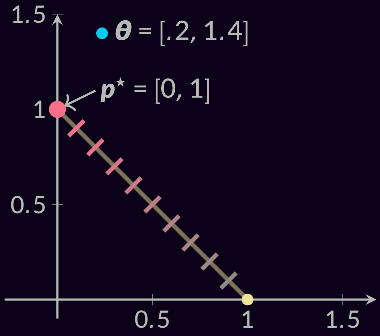


$N = 3$

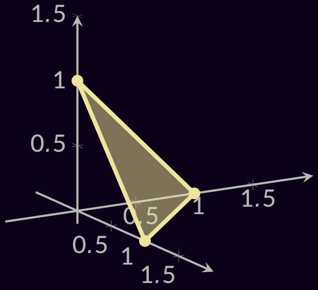
Variational Form of Argmax

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

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



$N = 2$

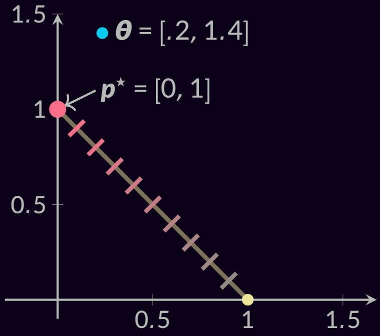


$N = 3$

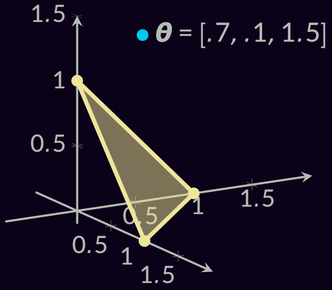
Variational Form of Argmax

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

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



$N = 2$

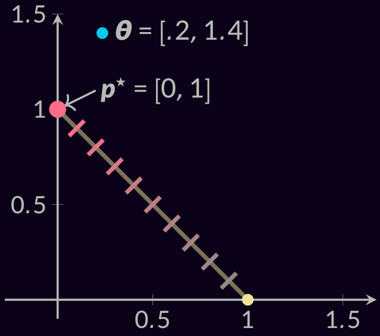


$N = 3$

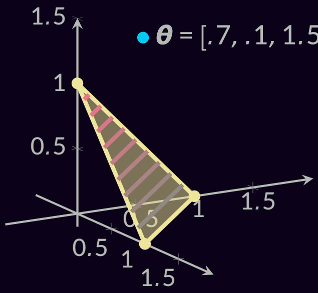
Variational Form of Argmax

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

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



$N = 2$

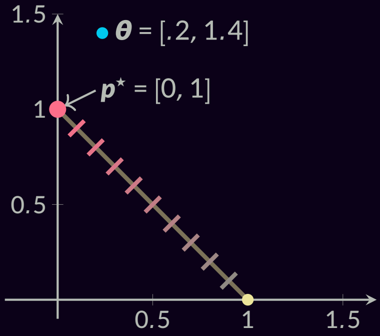


$N = 3$

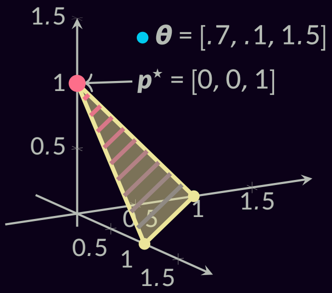
Variational Form of Argmax

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

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



$N = 2$

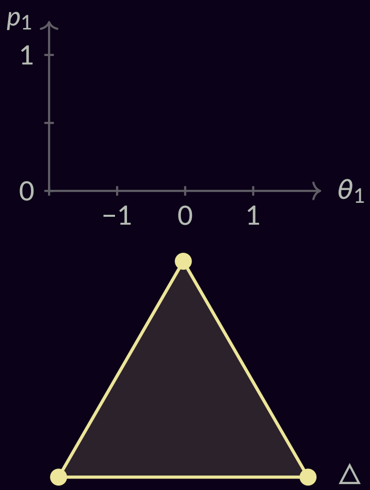


$N = 3$

(Niculae and Blondel, 2017)

Smoothed Max Operators

$$\pi_\Omega(\theta) = \arg\max_{p \in \triangle} p^\top \theta - \Omega(p)$$

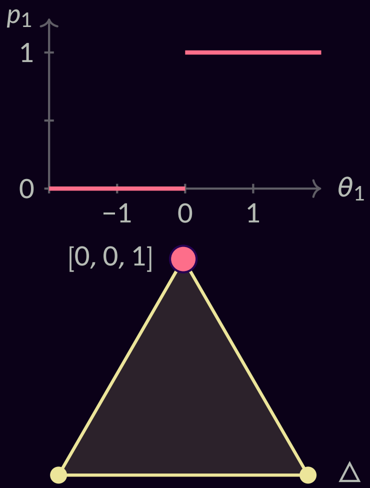


(Niculae and Blondel, 2017)

Smoothed Max Operators

$$\boldsymbol{\pi}_\Omega(\boldsymbol{\theta}) = \underset{\boldsymbol{p}\in\triangle}{\arg\max}\, \boldsymbol{p}^\top \boldsymbol{\theta} - \Omega(\boldsymbol{p})$$

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

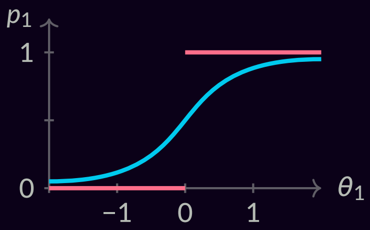


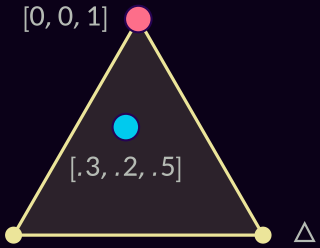
(Niculae and Blondel, 2017)

Smoothed Max Operators

$$\pi_\Omega(\theta) = \arg\max_{p \in \triangle} p^\top \theta - \Omega(p)$$

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



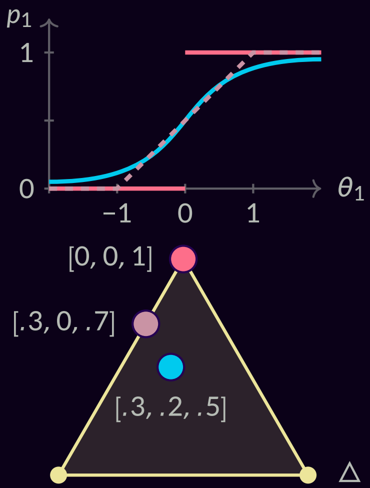


(Niculae and Blondel, 2017)

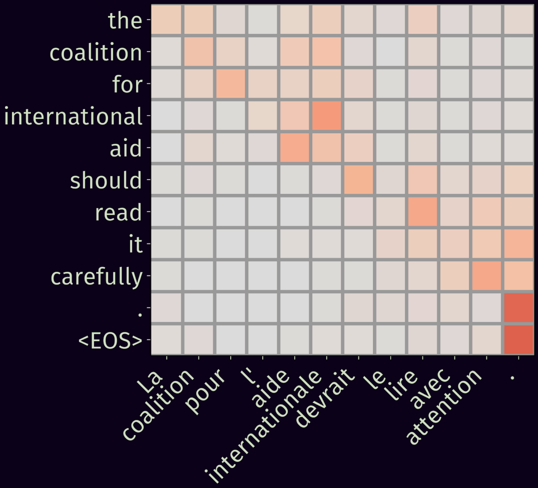
Smoothed Max Operators

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

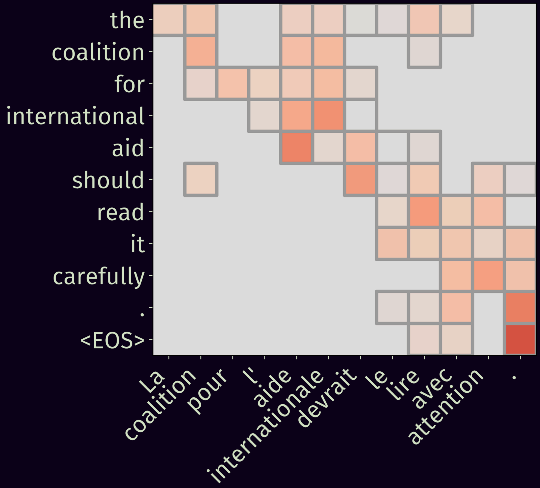
- argmax: $\Omega(\boldsymbol{p}) = 0$
- softmax: $\Omega(\boldsymbol{p}) = \sum_j p_j \log p_j$
- sparsemax: $\Omega(\boldsymbol{p}) = 1/2\|\boldsymbol{p}\|_2^2$



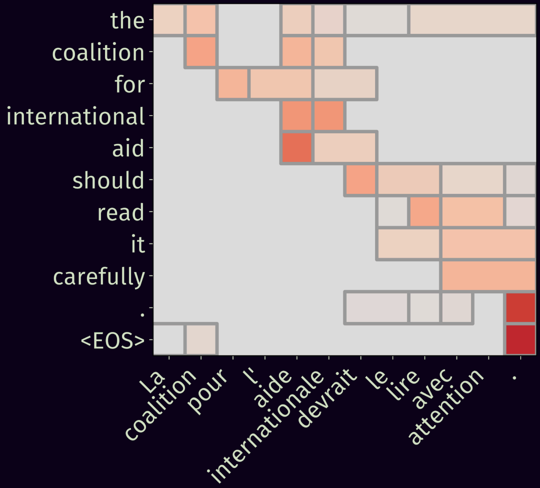
(Martins and Astudillo, 2016)


the
coalition
for
international
aid
should
read
it
carefully
.
<EOS>
La
coalition
pour
l'
aide
internationale
devrait
le
lire
avec
attention
.

softmax


the
coalition
for
international
aid
should
read
it
carefully
.
<EOS>
La
coalition
pour
l'
aide
internationale
devrait
le
lire
avec
attention
.

sparsemax


the
coalition
for
international
aid
should
read
it
carefully
.
<EOS>
La
coalition
pour
l'
aide
internationale
devrait
le
lire
avec
attention
.

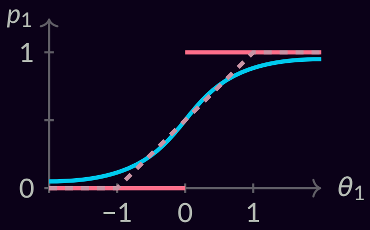
fusedmax ?!

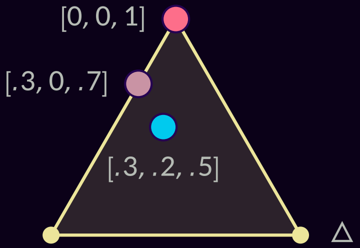
(Niculae and Blondel, 2017)

Smoothed Max Operators

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

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





(Niculae and Blondel, 2017)

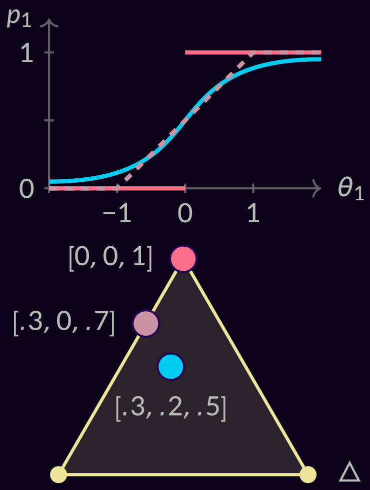
Smoothed Max Operators

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

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

fusedmax: $\Omega(\boldsymbol{p}) = 1/2\|\boldsymbol{p}\|_2^2 + \sum_j |p_j - p_{j-1}|$

csparsemax: $\Omega(\boldsymbol{p}) = 1/2\|\boldsymbol{p}\|_2^2 + \iota(\boldsymbol{a} \le \boldsymbol{p} \le \boldsymbol{b})$



Sparsemax

$$\text{sparsemax}(\boldsymbol{\theta}) = \underset{\boldsymbol{p}\in\triangle}{\arg\max}\, \boldsymbol{p}^\top\boldsymbol{\theta} - 1/2\|\boldsymbol{p}\|_2^2$$

$$= \underset{\boldsymbol{p}\in\triangle}{\arg\min}\, \|\boldsymbol{p} - \boldsymbol{\theta}\|_2^2$$

Sparsemax

$$\text{sparsemax}(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top\boldsymbol{\theta} - 1/2\|\boldsymbol{p}\|_2^2$$

$$= \arg\min_{\boldsymbol{p}\in\triangle} \|\boldsymbol{p} - \boldsymbol{\theta}\|_2^2$$

Computation:

$$\boldsymbol{p}^\star = [\boldsymbol{\theta} - \tau\mathbf{1}]_+$$

$$\theta_i > \theta_j \Rightarrow p_i \geq p_j$$

$\mathcal{O}(d)$ via partial sort

(Held et al., 1974; Brucker, 1984; Condat, 2016)

Sparsemax

$$\operatorname{sparsemax}(\boldsymbol{\theta}) = \underset{\boldsymbol{p}\in\triangle}{\arg\max}\, \boldsymbol{p}^\top \boldsymbol{\theta} - 1/2\|\boldsymbol{p}\|_2^2$$

$$= \underset{\boldsymbol{p}\in\triangle}{\arg\min}\, \|\boldsymbol{p} - \boldsymbol{\theta}\|_2^2$$

Computation:

$$\boldsymbol{p}^\star = [\boldsymbol{\theta} - \tau\mathbf{1}]_+$$

$$\theta_i > \theta_j \Rightarrow p_i \geq p_j$$

$\mathcal{O}(d)$ via partial sort

(Held et al., 1974; Brucker, 1984; Condat, 2016)

Backward pass:

$$\boldsymbol{J}_{\text{sparsemax}} = \operatorname{diag}(\boldsymbol{s}) - \frac{1}{|\mathcal{S}|}\boldsymbol{s}\boldsymbol{s}^\top$$

where $\mathcal{S} = \{j : p_j^\star > 0\}$,

$$s_j = [\![j \in \mathcal{S}]\!]$$

(Martins and Astudillo, 2016)

Sparsemax

$$\text{sparsemax}(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{p} \in \triangle} \boldsymbol{p}^\top \boldsymbol{\theta} - 1/2\|\boldsymbol{p}\|_2^2$$

$$= \arg\min_{\boldsymbol{p} \in \triangle} \|\boldsymbol{p} - \boldsymbol{\theta}\|_2^2$$

Computation:

$\boldsymbol{p}^\star = [($

$\theta_i > \theta_j$

$\mathcal{O}(d)$ via .

(Held et al., 1974; Brucker, 1984; Condat, 2016)

argmin differentiation

(Gould et al., 2016; Amos and Kolter, 2017)

Backward pass:

$\text{g}(\boldsymbol{s}) - \frac{1}{|\mathcal{S}|}\boldsymbol{s}\boldsymbol{s}^\top$

$: p_j^\star > 0\}$,

$\in \mathcal{S}]\!]$

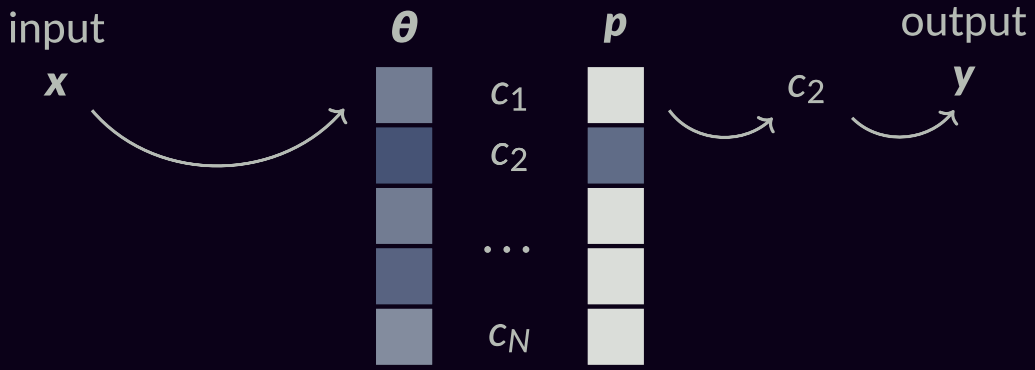
(Martins and Astudillo, 2016)

Structured Prediction

finally

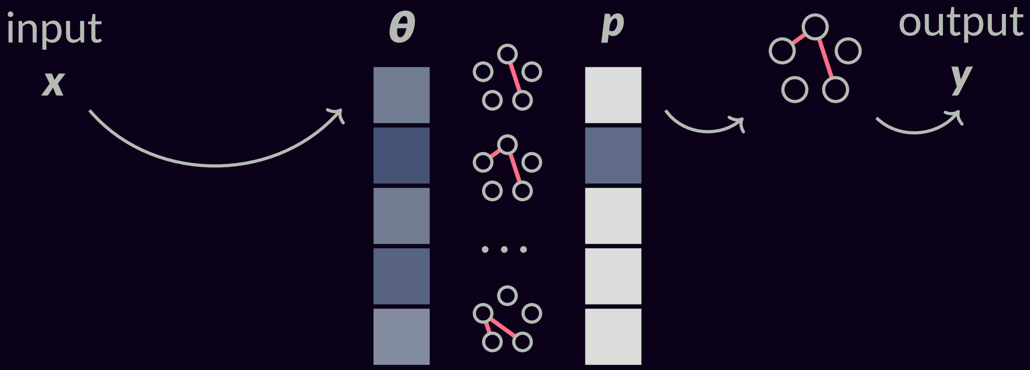
Structured Prediction

is essentially a (very high-dimensional) argmax



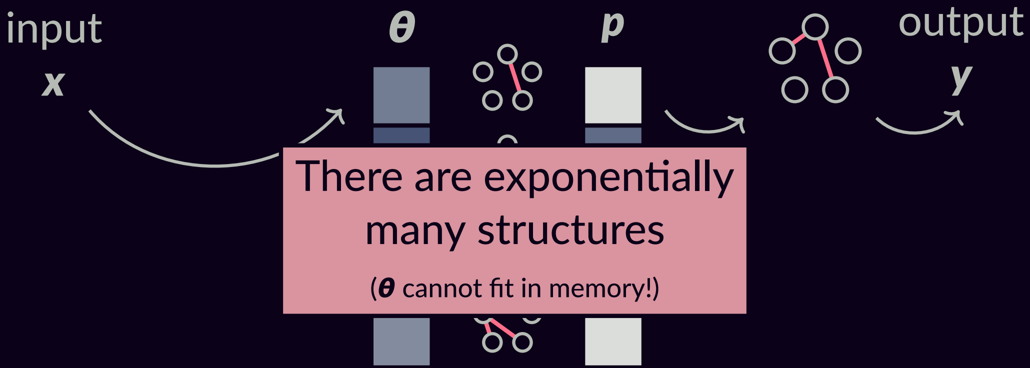
Structured Prediction

is essentially a (very high-dimensional) argmax



Structured Prediction

is essentially a (very high-dimensional) argmax



There are exponentially many structures

($\boldsymbol{\theta}$ cannot fit in memory!)

Factorization Into Parts

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

Factorization Into Parts

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

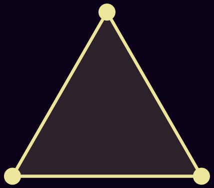
$$
\boldsymbol{A} = \begin{array}{r} \star\!\to\!\text{dog} \\ \text{on}\!\to\!\text{dog} \\ \text{wheels}\!\to\!\text{dog} \\ \hline \star\!\to\!\text{on} \\ \text{dog}\!\to\!\text{on} \\ \text{wheels}\!\to\!\text{on} \\ \hline \star\!\to\!\text{wheels} \\ \text{dog}\!\to\!\text{wheels} \\ \text{on}\!\to\!\text{wheels} \end{array}
\left[\begin{array}{ccccc} 1 & & 0 & 0 & \\ 0 & & 1 & 1 & \\ 0 & & 0 & 0 & \\ \hline 0 & & 1 & 1 & \\ 1 & \dots & 0 & 0 & \dots \\ 0 & & 0 & 0 & \\ \hline 0 & & 0 & 0 & \\ 0 & & 1 & 0 & \\ 1 & & 0 & 1 & \end{array}\right]
\quad
\boldsymbol{\eta} = \left[\begin{array}{r} .1 \\ .2 \\ -.1 \\ \hline .3 \\ .8 \\ .1 \\ \hline -.3 \\ .2 \\ -.1 \end{array}\right]
$$

Factorization Into Parts

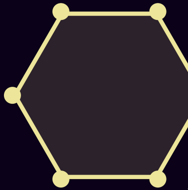
$$\boldsymbol{\theta} = A^\top \boldsymbol{\eta}$$

$$A = \begin{array}{r|cccccc|} \star\to\text{dog} & 1 & & 0 & 0 & \\ \text{on}\to\text{dog} & 0 & & 1 & 1 & \\ \text{wheels}\to\text{dog} & 0 & & 0 & 0 & \\ \hline \star\to\text{on} & 0 & & 1 & 1 & \\ \text{dog}\to\text{on} & 1 & \dots & 0 & 0 & \dots \\ \text{wheels}\to\text{on} & 0 & & 0 & 0 & \\ \hline \star\to\text{wheels} & 0 & & 0 & 0 & \\ \text{dog}\to\text{wheels} & 0 & & 1 & 0 & \\ \text{on}\to\text{wheels} & 1 & & 0 & 1 & \end{array} \quad \boldsymbol{\eta} = \begin{bmatrix} .1 \\ .2 \\ -.1 \\ \hline .3 \\ .8 \\ .1 \\ \hline -.3 \\ .2 \\ -.1 \end{bmatrix}$$

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

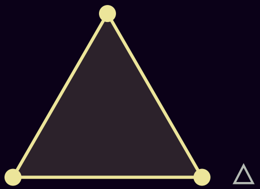


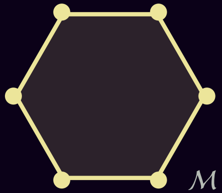
$\triangle$



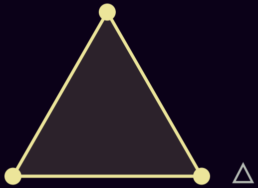
$\mathcal{M}$

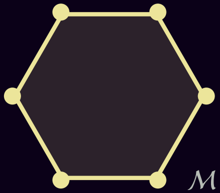
$$\mathcal{M} := \operatorname{conv}\left\{ \boldsymbol{a}_y : y \in \mathcal{Y} \right\}$$



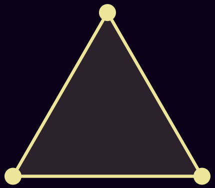


$$\mathcal{M} := \operatorname{conv}\left\{\boldsymbol{a}_y : y \in \mathcal{Y}\right\}$$
$$= \left\{\boldsymbol{A}\boldsymbol{p} : \boldsymbol{p} \in \triangle\right\}$$

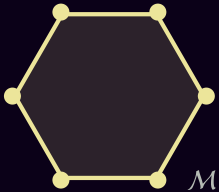




$$\mathcal{M} := \operatorname{conv}\left\{\boldsymbol{a}_y : y \in \mathcal{Y}\right\}$$
$$= \left\{\boldsymbol{A}\boldsymbol{p} : \boldsymbol{p} \in \triangle\right\}$$
$$= \left\{\mathbb{E}_{Y \sim \boldsymbol{p}}\, \boldsymbol{a}_Y : \boldsymbol{p} \in \triangle\right\}$$

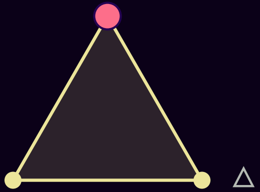


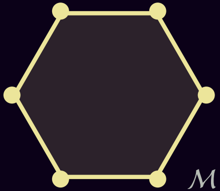
$\triangle$



$\mathcal{M}$

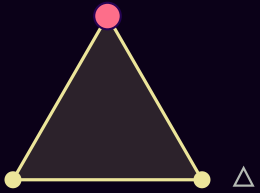
argmax $\arg\max_{\boldsymbol{p} \in \triangle} \boldsymbol{p}^\top \boldsymbol{\theta}$

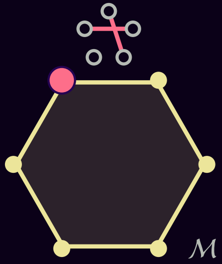




argmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top\boldsymbol{\theta}$

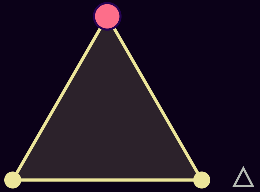
MAP $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top\boldsymbol{\eta}$

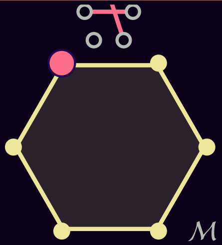




- **argmax** $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{\theta}$
- **MAP** $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta}$

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

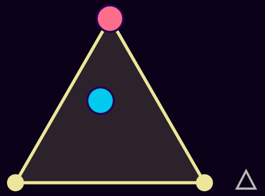


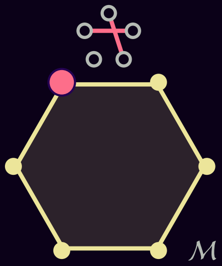


argmax $\arg\max_{\boldsymbol{p} \in \triangle} \boldsymbol{p}^\top \boldsymbol{\theta}$

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

softmax $\arg\max_{\boldsymbol{p} \in \triangle} \boldsymbol{p}^\top \boldsymbol{\theta} + \mathrm{H}(\boldsymbol{p})$



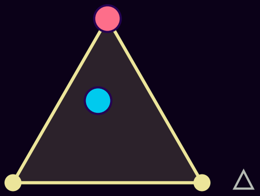


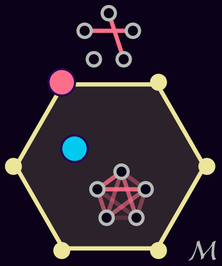
argmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{\theta}$

softmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{\theta} + \mathrm{H}(\boldsymbol{p})$

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

marginals $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} + \widetilde{\mathrm{H}}(\boldsymbol{\mu})$





argmax $\arg\max_{\boldsymbol{p} \in \triangle} \boldsymbol{p}^\top \boldsymbol{\theta}$

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

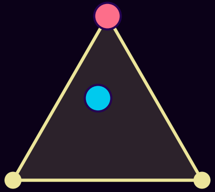
softmax $\arg\max_{\boldsymbol{p} \in \triangle} \boldsymbol{p}^\top \boldsymbol{\theta} + \mathrm{H}(\boldsymbol{p})$

marginals $\arg\max_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} + \widetilde{\mathrm{H}}(\boldsymbol{\mu})$

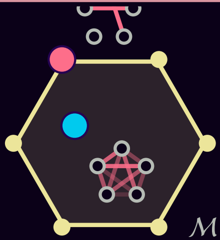
e.g. sequence labelling → forward-backward

(Rabiner, 1989)

As attention: (Kim et al., 2017)



$\triangle$



$\mathcal{M}$

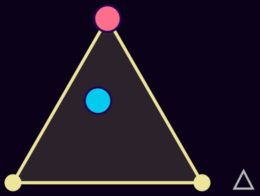
- **argmax** $\arg\max_{\boldsymbol{p} \in \triangle} \boldsymbol{p}^\top \boldsymbol{\theta}$
- **softmax** $\arg\max_{\boldsymbol{p} \in \triangle} \boldsymbol{p}^\top \boldsymbol{\theta} + \mathrm{H}(\boldsymbol{p})$

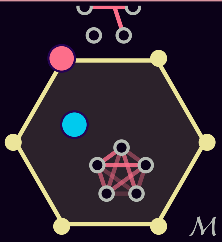
- **MAP** $\arg\max_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta}$
- **marginals** $\arg\max_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} + \widetilde{\mathrm{H}}(\boldsymbol{\mu})$

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 $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{\theta}$

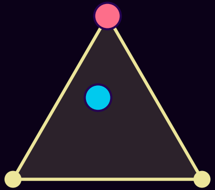
softmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{\theta} + \mathrm{H}(\boldsymbol{p})$

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

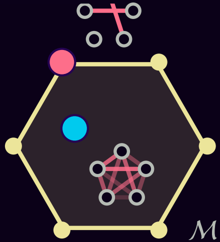
marginals $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} + \widetilde{\mathrm{H}}(\boldsymbol{\mu})$

e.g. matchings → **#P-complete!**

(Taskar, 2004; Valiant, 1979)



$\triangle$



$\mathcal{M}$

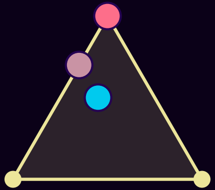
argmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{\theta}$

softmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{\theta} + \mathrm{H}(\boldsymbol{p})$

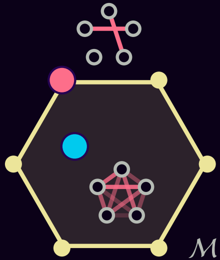
sparsemax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{\theta} - 1/2\|\boldsymbol{p}\|^2$

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

marginals $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} + \widetilde{\mathrm{H}}(\boldsymbol{\mu})$



$\triangle$



$\mathcal{M}$

(Niculae, Martins, Blondel, and Cardie, 2018)

argmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top\boldsymbol{\theta}$

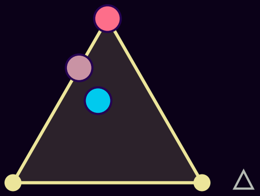
softmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top\boldsymbol{\theta} + \mathrm{H}(\boldsymbol{p})$

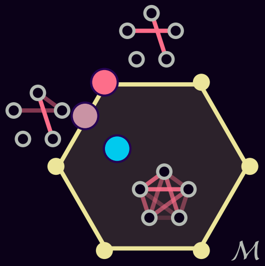
sparsemax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top\boldsymbol{\theta} - 1/2\|\boldsymbol{p}\|^2$

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

marginals $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top\boldsymbol{\eta} + \widetilde{\mathrm{H}}(\boldsymbol{\mu})$

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





SparseMAP Solution

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

=  = .6  + .4 

$= \boldsymbol{A}\boldsymbol{p}^\star$ with very sparse $\boldsymbol{p}^\star \in \triangle^N$

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

$$\boldsymbol{\mu}^\star = \underset{\boldsymbol{\mu}\in\mathcal{M}}{\arg\max}\, \boldsymbol{\mu}^\top \boldsymbol{\eta} - \tfrac{1}{2}\|\boldsymbol{\mu}\|^2$$

linear constraints
(alas, exponentially many!)

quadratic objective

Algorithms for SparseMAP

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

linear constraints
(alas, exponentially many!)

quadratic objective

Conditional Gradient

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

Algorithms for SparseMAP

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

linear constraints
(alas, exponentially many!)

quadratic objective

Conditional Gradient

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

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

Algorithms for SparseMAP

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

linear constraints
(alas, exponentially many!)

quadratic objective

Conditional Gradient

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

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

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

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!)

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

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!)

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

quadratic objective

Conditional Gradient

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

- select a new corne
- update the (sparse)
  - Update rules: van
  - Quadratic objective: **Active Set**
    (Nocedal and Wright, 1999, Ch. 16.4 & 16.5)
    (Wolfe, 1976; Vinyes and Obozinski, 2017)

Active Set achieves
finite & **linear** convergence!

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
    - 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

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
  - 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 $\left(\frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\eta}}\right)^\top \boldsymbol{dy}$

takes $\mathcal{O}(\dim(\boldsymbol{\mu})\,\mathrm{nnz}(\boldsymbol{p}^\star))$

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

Conditi

(Frank and Wolfe, 1956

Completely modular: just add MAP

pass

- select a new c
- 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)

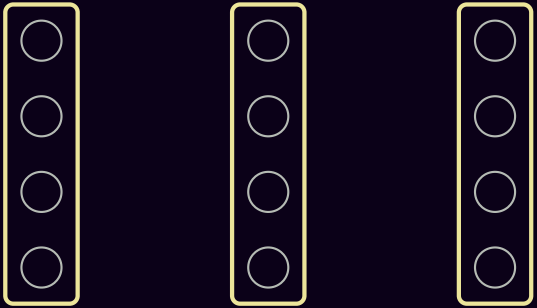
    (Wolfe, 1976; Vinyes and Obozinski, 2017)

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

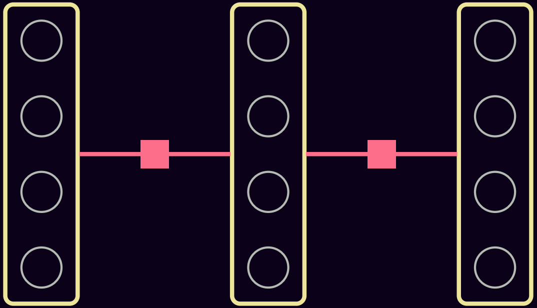
computing $\left(\frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\eta}}\right)^\top \boldsymbol{dy}$

takes $\mathcal{O}(\dim(\boldsymbol{\mu})\, \mathrm{nnz}(\boldsymbol{p}^\star))$

Structured Attention & Graphical Models



Structured Attention & Graphical Models



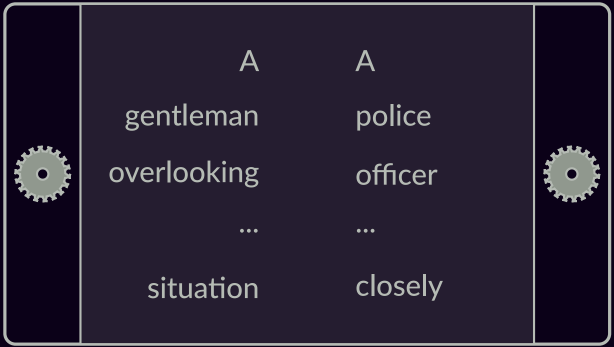
Structured Attention for Alignments

NLI

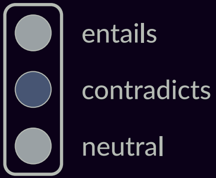
premise: A gentleman overlooking a neighborhood situation.
hypothesis: A police officer watches a situation closely.

input

(P, H)



output



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

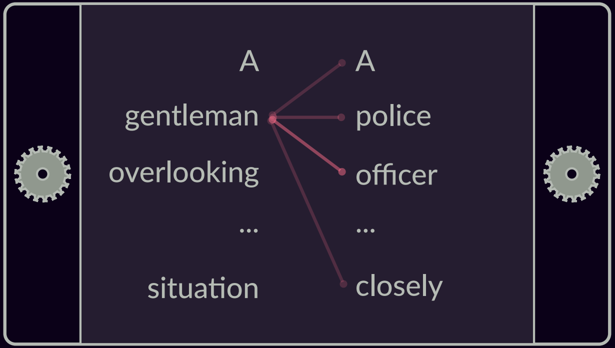
Structured Attention for Alignments

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))

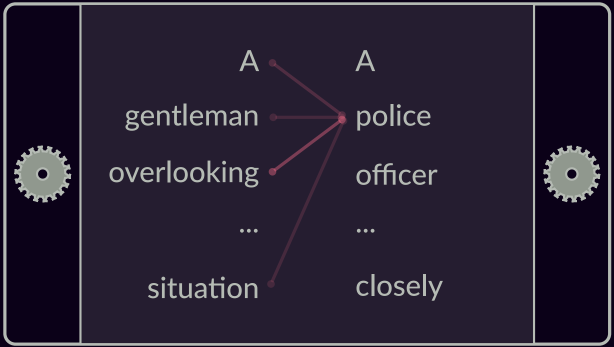
Structured Attention for Alignments

NLI

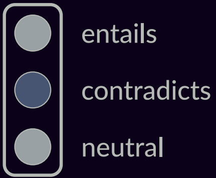
premise: A gentleman overlooking a neighborhood situation.
hypothesis: A police officer watches a situation closely.

input

(P, H)



output

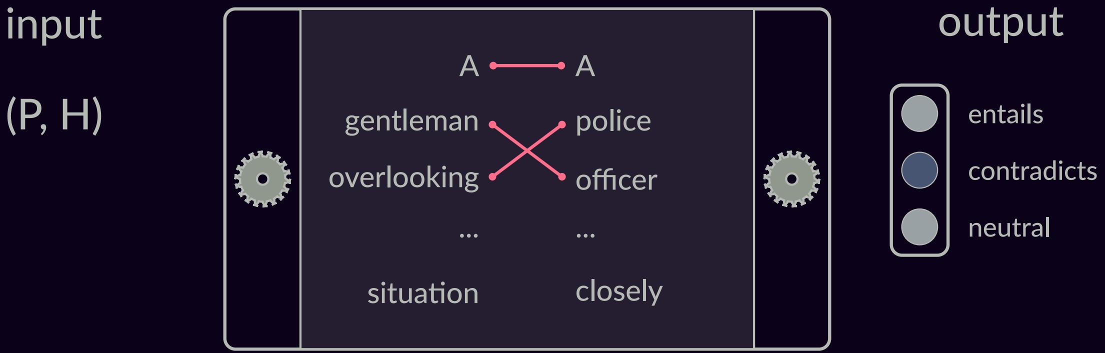


(Model: ESIM (Chen et al., 2017))

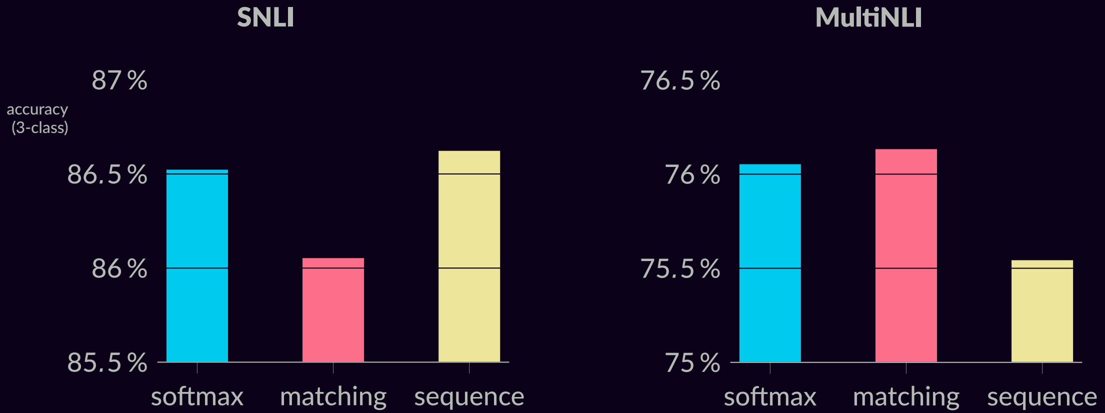
Structured Attention for Alignments

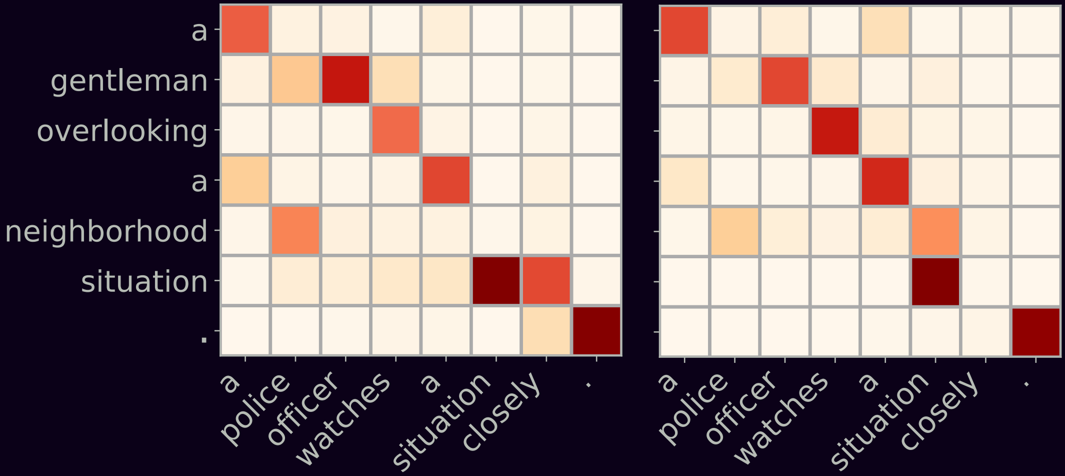
NLI

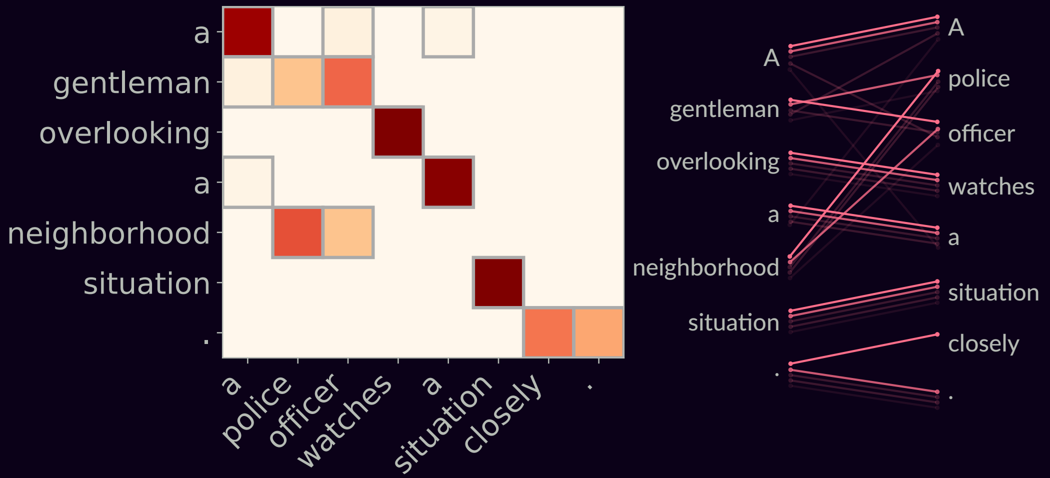
premise: A gentleman overlooking a neighborhood situation.
hypothesis: A police officer watches a situation closely.



(Proposed model: global matching)


SNLI
MultiNLI
accuracy
(3-class)
87 %
86.5 %
86 %
85.5 %
76.5 %
76 %
75.5 %
75 %
softmax
matching
sequence
softmax
matching
sequence


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
.
a
police
officer
watches
a
situation
closely
.
A
gentleman
overlooking
a
neighborhood
situation
.
A
police
officer
watches
a
situation
closely
.

Dynamically inferring the computation graph

Dependency TreeLSTM

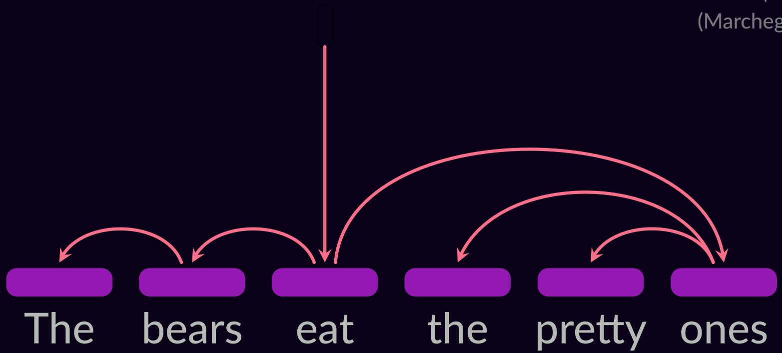
(Tai et al., 2015)

closely related to GCNs, e.g.
(Kipf and Welling, 2017)
(Marcheggiani and Titov, 2017)

Dependency TreeLSTM

(Tai et al., 2015)

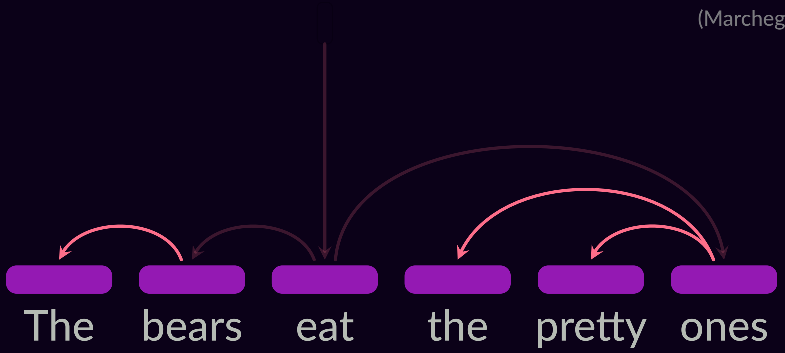
closely related to GCNs, e.g.
(Kipf and Welling, 2017)
(Marcheggiani and Titov, 2017)



Dependency TreeLSTM

(Tai et al., 2015)

closely related to GCNs, e.g.
(Kipf and Welling, 2017)
(Marcheggiani and Titov, 2017)



Dependency TreeLSTM

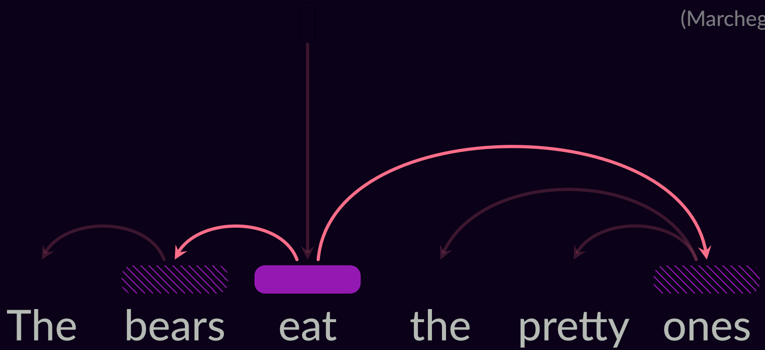
(Tai et al., 2015)

closely related to GCNs, e.g.
(Kipf and Welling, 2017)
(Marcheggiani and Titov, 2017)

Dependency TreeLSTM

(Tai et al., 2015)

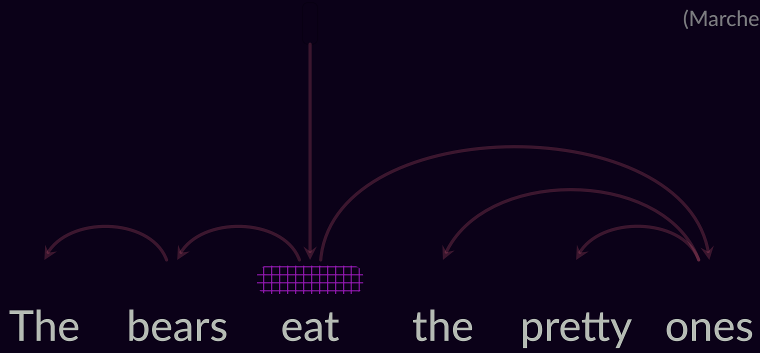
closely related to GCNs, e.g.
(Kipf and Welling, 2017)
(Marcheggiani and Titov, 2017)



Dependency TreeLSTM

(Tai et al., 2015)

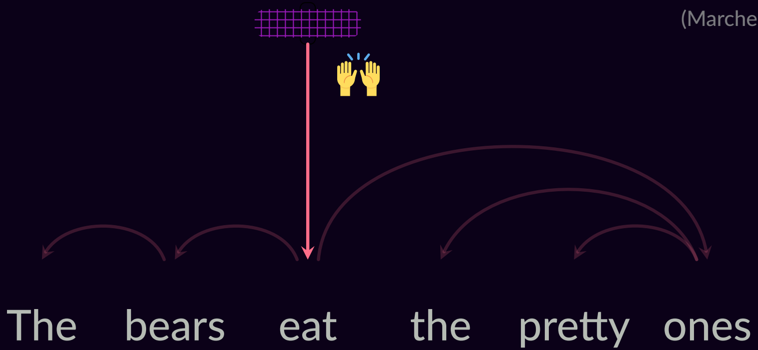
closely related to GCNs, e.g.
(Kipf and Welling, 2017)
(Marcheggiani and Titov, 2017)



Dependency TreeLSTM

(Tai et al., 2015)

closely related to GCNs, e.g.
(Kipf and Welling, 2017)
(Marcheggiani and Titov, 2017)

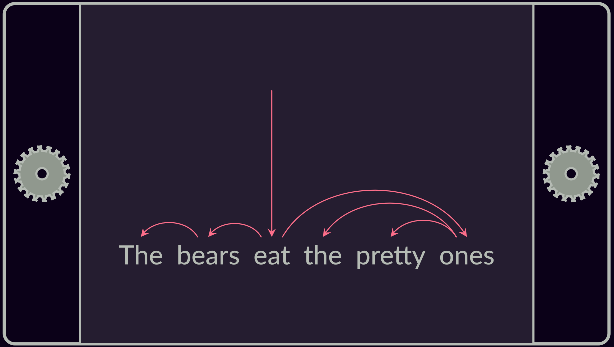


Latent Dependency TreeLSTM

(Niculae, Martins, and Cardie, 2018)

input

x



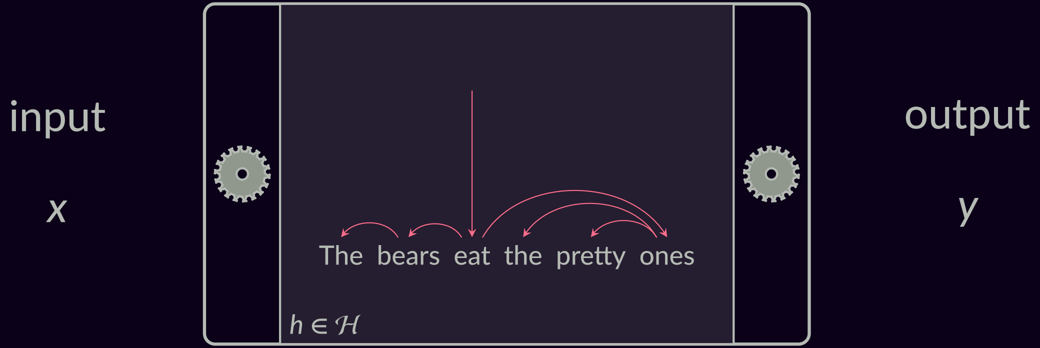
output

y

Latent Dependency TreeLSTM

(Niculae, Martins, and Cardie, 2018)

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



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

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

Structured Latent Variable Models

e.g., a TreeLSTM defined by h

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

Structured Latent Variable Models

e.g., a TreeLSTM defined by h

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

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

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

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

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

Exponentially large sum!

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

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

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

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

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

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

How to define $p_{\boldsymbol{\pi}}$?

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

idea 1

idea 2

idea 3

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

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

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

How to define $p_{\boldsymbol{\pi}}$?

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

idea 1

idea 2

idea 3

Structured Latent Variable Models

sum over all possible trees

e.g., a TreeLSTM defined by h

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

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

How to define $p_{\boldsymbol{\pi}}$?

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

idea 1 $p_{\boldsymbol{\pi}}(h \mid x) = 1$ if $h = h^{\star}$ else 0 argmax

idea 2

idea 3

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

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

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

How to define $p_{\boldsymbol{\pi}}$?

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

idea 1 $p_{\boldsymbol{\pi}}(h \mid x) = 1$ if $h = h^{\star}$ else 0 argmax 😊

idea 2

idea 3

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

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

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

How to define $p_{\boldsymbol{\pi}}$?

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

idea 1 $p_{\boldsymbol{\pi}}(h \mid x) = 1$ if $h = h^\star$ else 0 argmax 😊 😟

idea 2

idea 3

Structured Latent Variable Models

sum over all possible trees

e.g., a TreeLSTM defined by h

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

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

How to define p_π?

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

idea 1 $p_{\pi}(h \mid x) = 1$ if $h = h^\star$ else 0 argmax 😊 😟

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

idea 3

Structured Latent Variable Models

sum over
all possible trees

e.g., a TreeLSTM defined by h

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

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

How to define $p_{\boldsymbol{\pi}}$?

			$\sum_{h \in \mathcal{H}}$	$\frac{\partial p(y \mid x)}{\partial \boldsymbol{\pi}}$
idea 1	$p_{\boldsymbol{\pi}}(h \mid x) = 1$ if $h = h^\star$ else 0	argmax	😊	😟
idea 2	$p_{\boldsymbol{\pi}}(h \mid x) \propto \exp\big(\text{score}_{\boldsymbol{\pi}}(h; x)\big)$	softmax		😊
idea 3				

Structured Latent Variable Models

sum over all possible trees →

e.g., a TreeLSTM defined by h →

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

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

How to define $p_{\boldsymbol{\pi}}$?

			$\sum_{h \in \mathcal{H}}$	$\frac{\partial p(y \mid x)}{\partial \boldsymbol{\pi}}$
idea 1	$p_{\boldsymbol{\pi}}(h \mid x) = 1 \text{ if } h = h^\star \text{ else } 0$	argmax	😊	😟
idea 2	$p_{\boldsymbol{\pi}}(h \mid x) \propto \exp\big(\text{score}_{\boldsymbol{\pi}}(h; x)\big)$	softmax	😱	😊
idea 3				

Structured Latent Variable Models

sum over all possible trees

e.g., a TreeLSTM defined by h

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

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

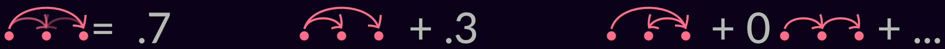
How to define $p_{\boldsymbol{\pi}}$?

			$\sum_{h \in \mathcal{H}}$	$\frac{\partial p(y \mid x)}{\partial \boldsymbol{\pi}}$
idea 1	$p_{\boldsymbol{\pi}}(h \mid x) = 1$ if $h = h^\star$ else 0	argmax	😊	😟
idea 2	$p_{\boldsymbol{\pi}}(h \mid x) \propto \exp\left(\text{score}_{\boldsymbol{\pi}}(h; x)\right)$	softmax	😱	😊
idea 3		SparseMAP	😊	😊

SparseMAP

 = .7  + .3 

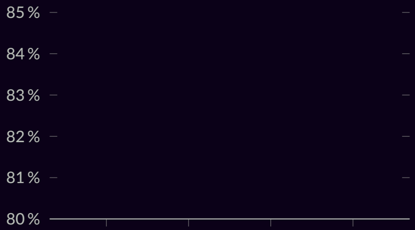
SparseMAP

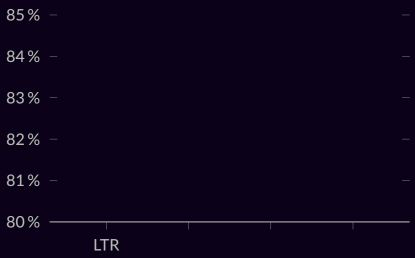


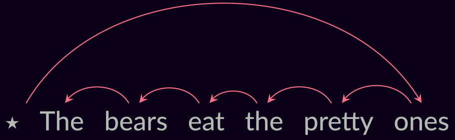
SparseMAP

 $= \ .7$  $+ \ .3$  $+ \ 0$  $+ \ \dots$

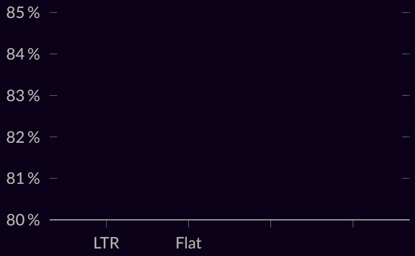
$p(y \mid x) = \ .7 \, p_{\phi}(y \mid$  $) + .3 \, p_{\phi}(y \mid$  $)$

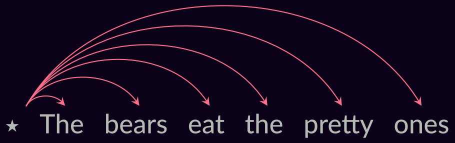

85 %
84 %
83 %
82 %
81 %
80 %



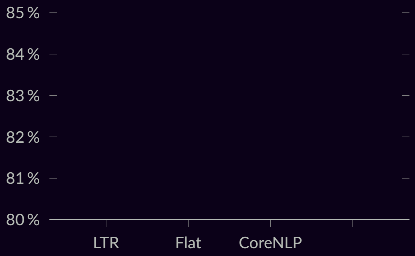


Left-to-right: regular LSTM

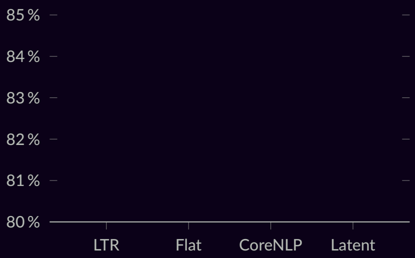


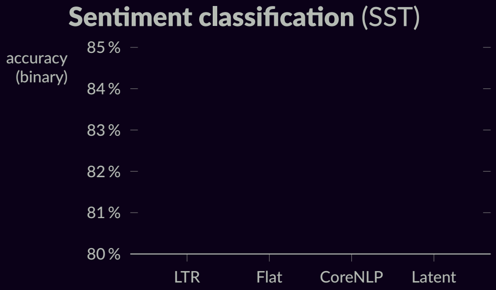


Flat: bag-of-words–like

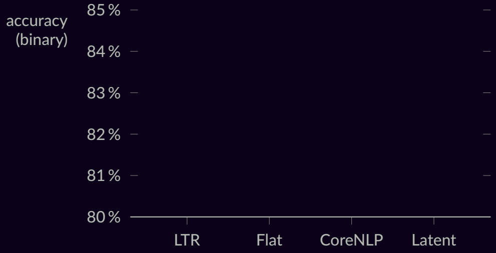


CoreNLP: off-line parser

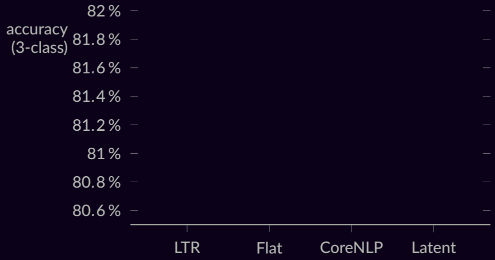

85 %
84 %
83 %
82 %
81 %
80 %
LTR
Flat
CoreNLP
Latent


Sentiment classification (SST)
accuracy (binary)
85 %
84 %
83 %
82 %
81 %
80 %
LTR
Flat
CoreNLP
Latent

Sentiment classification (SST)



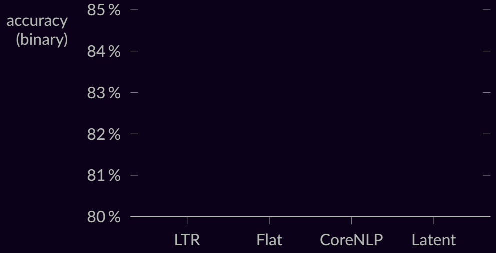
Natural Language Inference (SNLI)



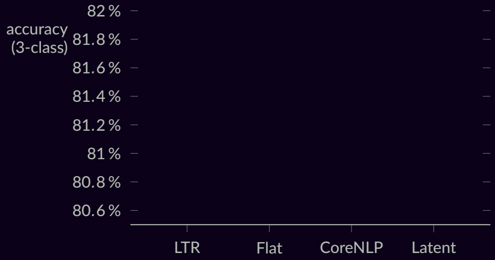
Sentence pair classification (P, H)

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

Sentiment classification (SST)



Natural Language Inference (SNLI)

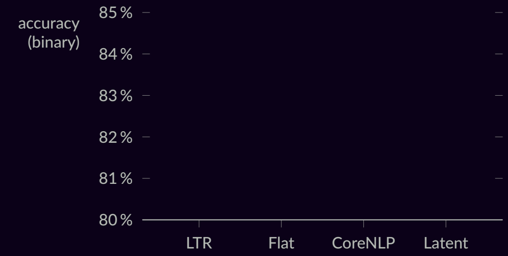


Reverse dictionary lookup

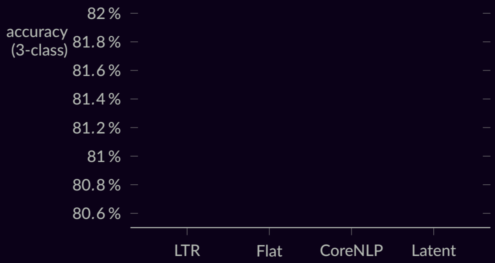
given word description, predict word embedding (Hill et al., 2016)

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

Sentiment classification (SST)

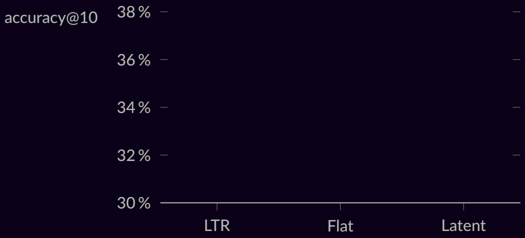


Natural Language Inference (SNLI)

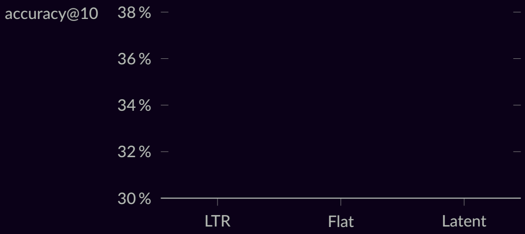


Reverse dictionary lookup

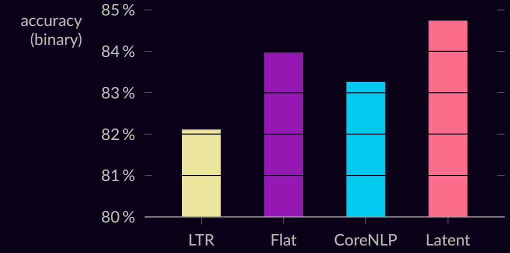
(definitions)



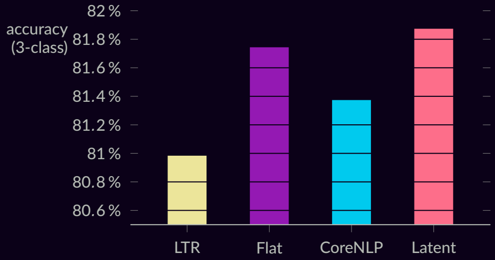
(concepts)



Sentiment classification (SST)

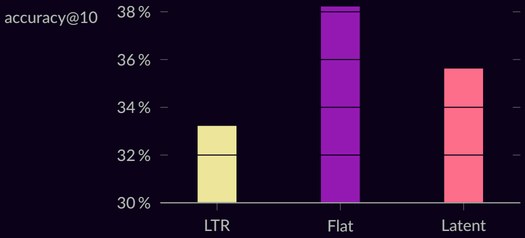


Natural Language Inference (SNLI)

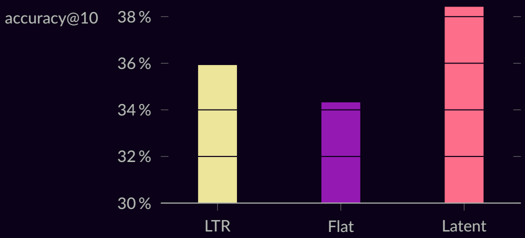


Reverse dictionary lookup

(definitions)



(concepts)



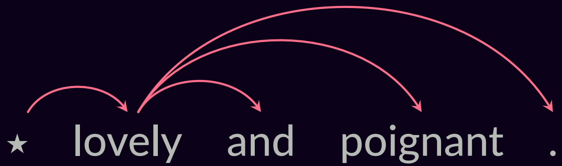
Syntax vs. Composition Order

CoreNLP parse, $p = 21.4\%$

Syntax vs. Composition Order

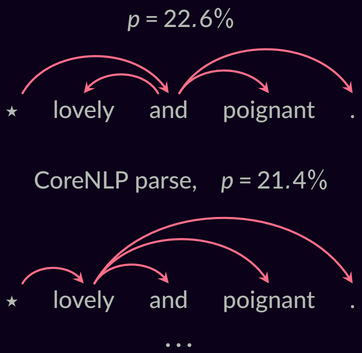
$p = 22.6\%$

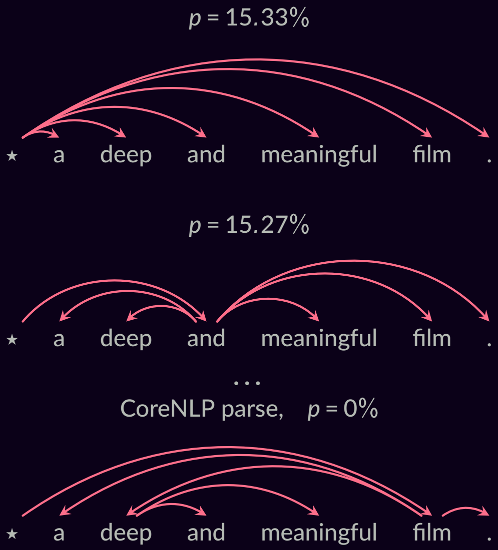
CoreNLP parse, $p = 21.4\%$



. . .

Syntax vs. Composition Order





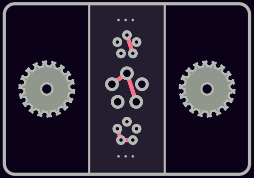
Conclusions

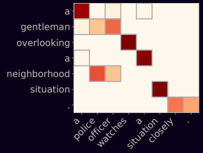
Differentiable & sparse structured inference

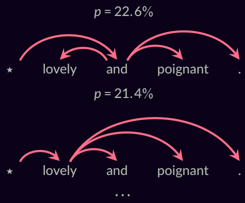
Generic, extensible algorithms

Interpretable structured attention

Dynamically-inferred computation graphs







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, 1966; Prop. B.25 in Bertsekas, 1999)

Danskin's Theorem

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

$$\partial \max_{\boldsymbol{z} \in \mathcal{Z}} \phi(\boldsymbol{x}, \boldsymbol{z}) = \operatorname{conv}\left\{ \nabla_{\boldsymbol{x}} \phi(\boldsymbol{x}, \boldsymbol{z}^\star) \mid \boldsymbol{z}^\star \in \arg\max_{\boldsymbol{z} \in \mathcal{Z}} \phi(\boldsymbol{x}, \boldsymbol{z}) \right\}.$$

Example: maximum of a vector

(Danskin, 1966; Prop. B.25 in Bertsekas, 1999)

Danskin's Theorem

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

$$\partial \max_{\boldsymbol{z} \in \mathcal{Z}} \phi(\boldsymbol{x}, \boldsymbol{z}) = \operatorname{conv}\{\nabla_{\boldsymbol{x}} \phi(\boldsymbol{x}, \boldsymbol{z}^\star) \mid \boldsymbol{z}^\star \in \arg\max_{\boldsymbol{z} \in \mathcal{Z}} \phi(\boldsymbol{x}, \boldsymbol{z})\}.$$

Example: maximum of a vector

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

(Danskin, 1966; Prop. B.25 in Bertsekas, 1999)

Danskin's Theorem

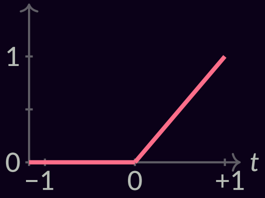
Let $\phi : \mathbb{R}^d \times \mathcal{Z} \to \mathbb{R}$, $\mathcal{Z} \subset \mathbb{R}^d$ compact.

$$\partial \max_{\boldsymbol{z} \in \mathcal{Z}} \phi(\boldsymbol{x}, \boldsymbol{z}) = \operatorname{conv}\left\{ \nabla_{\boldsymbol{x}} \phi(\boldsymbol{x}, \boldsymbol{z}^\star) \mid \boldsymbol{z}^\star \in \arg\max_{\boldsymbol{z} \in \mathcal{Z}} \phi(\boldsymbol{x}, \boldsymbol{z}) \right\}.$$

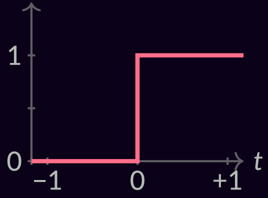
Example: maximum of a vector

$$\begin{aligned} \partial \max_{j \in [d]} \theta_j &= \partial \max_{\boldsymbol{p} \in \triangle} \boldsymbol{p}^\top \boldsymbol{\theta} \\ &= \partial \max_{\boldsymbol{p} \in \triangle} \phi(\boldsymbol{p}, \boldsymbol{\theta}) \\ &= \operatorname{conv}\left\{ \nabla_{\boldsymbol{\theta}} \phi(\boldsymbol{p}^\star, \boldsymbol{\theta}) \right\} \\ &= \operatorname{conv}\left\{ \boldsymbol{p}^\star \right\} \end{aligned}$$

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



$\max_j \theta_j$



$\{g_1 \mid \boldsymbol{g} \in \partial \max_j \theta_j\}$

Fusedmax

$$\text{fusedmax}(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top\boldsymbol{\theta} - 1/2\|\boldsymbol{p}\|_2^2 - \sum_{2\le j\le d} |p_j - p_{j-1}|$$

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

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

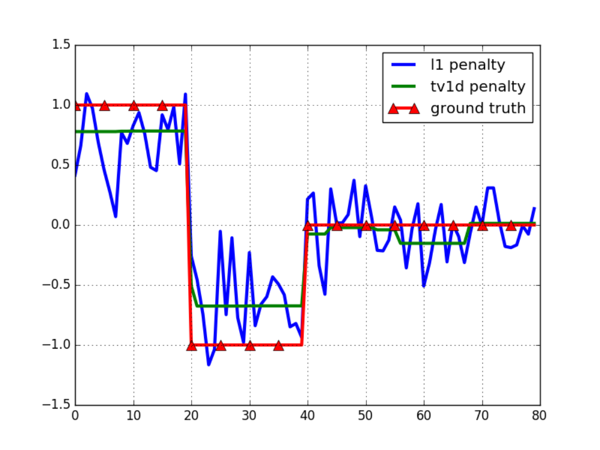
Proposition: $\text{fusedmax}(\boldsymbol{\theta}) = \text{sparsemax}\left(\text{prox}_{\text{fused}}(\boldsymbol{\theta})\right)$

(Niculae and Blondel, 2017)

fusedmax(... $|p_j - p_{j-1}|$

... $_{j-1}|$

$\text{prox}_{\text{fused}}($ ... $_{j-1}|$



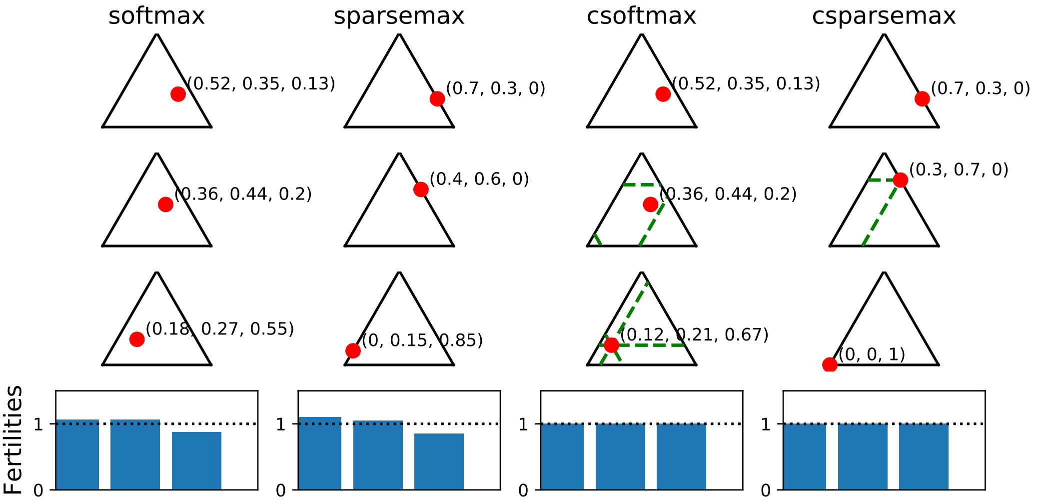
"Fused Lasso" a.k.a. 1-d Total Variation

(Tibshirani et al., 2005)

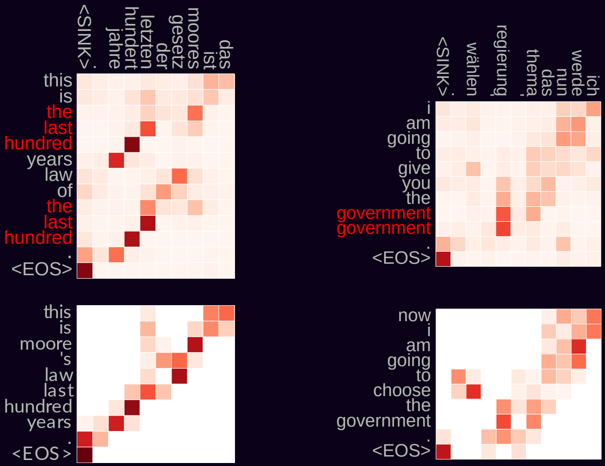
Proposi ... $_{\text{fused}}(\boldsymbol{\theta}))$

(Niculae and Blondel, 2017)

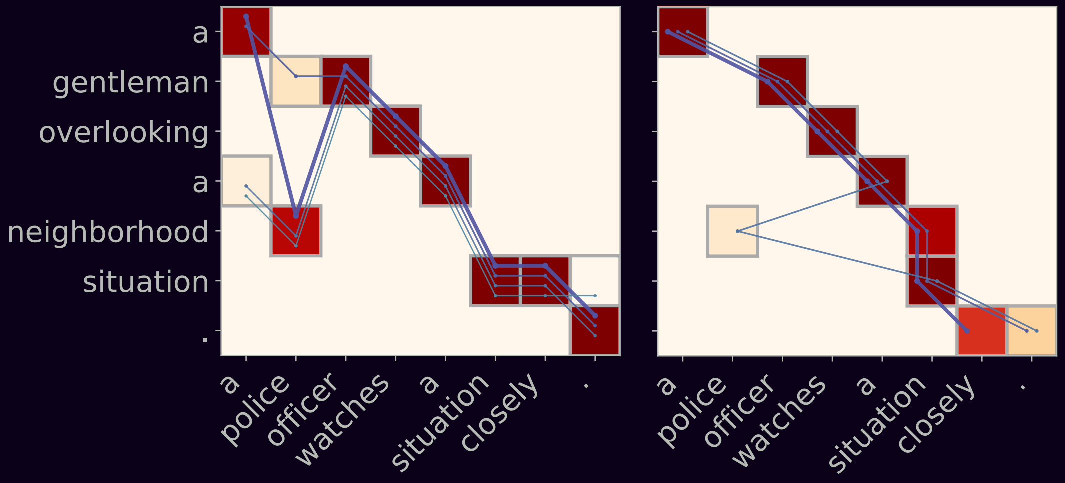
Example: Source Sentence with Three Words



e.g., fertility constraints for NMT



constrained softmax: (Martins and Kreutzer, 2017) **constrained sparsemax:** (Malaviya et al., 2018)


a
gentleman
overlooking
a
neighborhood
situation
.
a
police
officer
watches
a
situation
closely
.
a
police
officer
watches
a
situation
closely
.

Structured Output Prediction

SparseMAP

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

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

Structured Output Prediction

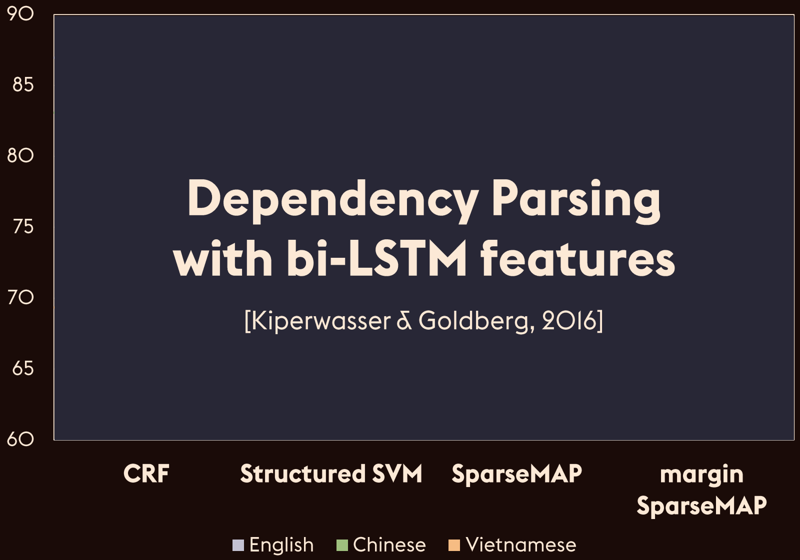
SparseMAP

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

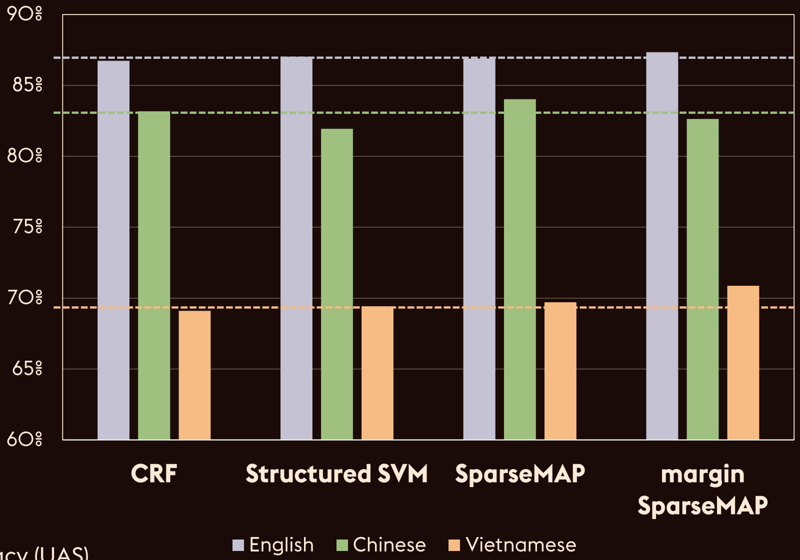
cost-SparseMAP

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

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



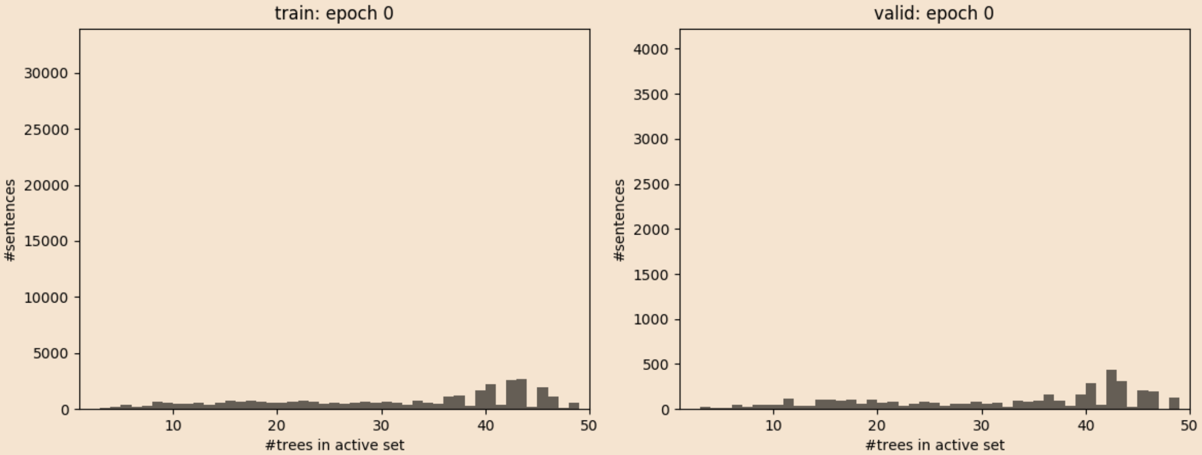
Dependency Parsing with bi-LSTM features
[Kiperwasser & Goldberg, 2016]
90
85
80
75
70
65
60
CRF
Structured SVM
SparseMAP
margin SparseMAP
English
Chinese
Vietnamese



Unlabeled Accuracy (UAS)
Universal Dependencies dataset

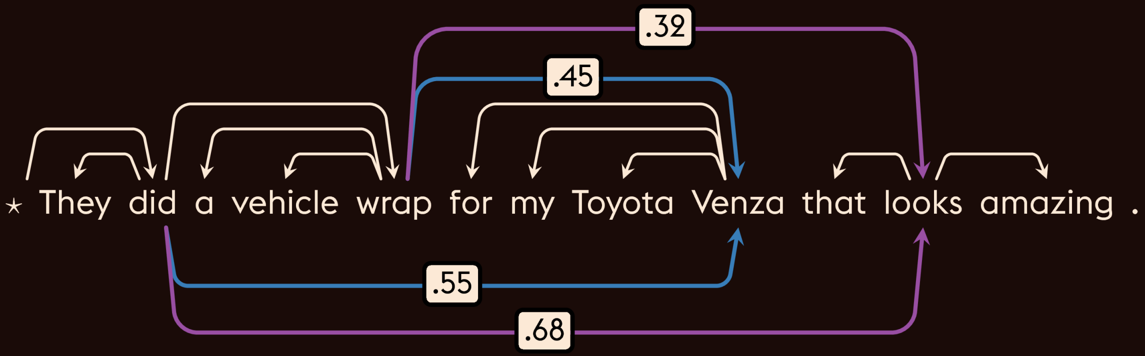
Sparse Structured Output Prediction

As models train, inference gets sparser!



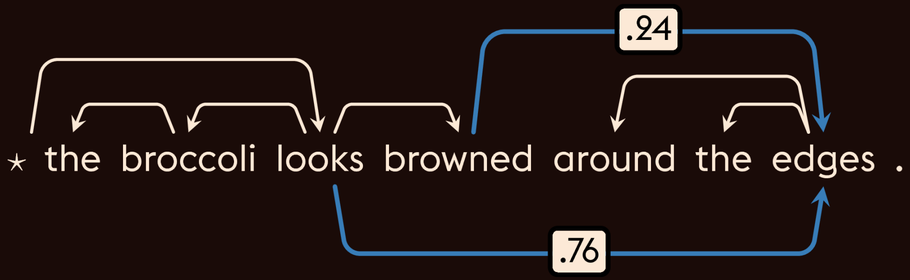
Sparse Structured Output Prediction

Inference captures linguistic ambiguity!



Sparse Structured Output Prediction

Inference captures linguistic ambiguity!



References I

Amos, Brandon and J. Zico Kolter (2017). "OptNet: Differentiable optimization as a layer in neural networks". In: *Proc. of ICML.*

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.*

Brucker, Peter (1984). "An $O(n)$ algorithm for quadratic knapsack problems". In: *Operations Research Letters* 3.3, pp. 163–166.

Chen, Qian et al. (2017). "Enhanced LSTM for natural language inference". In: *Proc. of ACL.*

Condat, Laurent (2016). "Fast projection onto the simplex and the ℓ_1 ball". In: *Mathematical Programming* 158.1-2, pp. 575–585.

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.

References II

Frank, Marguerite and Philip Wolfe (1956). “An algorithm for quadratic programming”. In: *Nav. Res. Log.* 3.1-2, pp. 95–110.

Gould, Stephen et al. (2016). “On differentiating parameterized argmin and argmax problems with application to bi-level optimization”. In: *preprint arXiv:1607.05447*.

Held, Michael, Philip Wolfe, and Harlan P Crowder (1974). “Validation of subgradient optimization”. In: *Mathematical Programming* 6.1, pp. 62–88.

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*.

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*.

References III

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*.

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*.

References IV

Niculae, Vlad and Mathieu Blondel (2017). "A regularized framework for sparse and structured neural attention". In: *Proc. of NeurIPS*.

Niculae, Vlad, André FT Martins, Mathieu Blondel, et al. (2018). "SparseMAP: Differentiable sparse structured inference". In: *Proc. of ICML*.

Niculae, 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*.

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*.

References V

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.