Beyond Softmax: Sparsity, Constraints, Latent Structure

... all end-to-end differentiable!!

André Martins

FOTRAN Workshop, Helsinki, 28/9/18
The **softmax** transformation is prevalent in language generation:

1. Softmax over the vocabulary to obtain a distribution over words
2. Attention mechanisms to condition of some property of the input (Bahdanau et al., 2015; Sukhbaatar et al., 2015)

This talk: **new transformations that capture sparsity, constraints, and structure**

- Sparsemax, Constrained Softmax/Sparsemax, SparseMAP
- All differentiable (efficient forward and backward propagation)
- Can be used at hidden or output layers.
Transformations from the Euclidean space $\mathbb{R}^K$ to the simplex.

Joint work with Ramon Astudillo, Julia Kreutzer, Chaitanya Malaviya, Pedro Ferreira, Vlad Niculue, Mathieu Blondel, and Claire Cardie.
Outline

1. Sparsity
2. Constraints
3. Latent Structure
4. Conclusions
Sparse Attention with Sparsemax

André F. T. Martins and Ramon Astudillo.

“From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification.”

ICML 2016.
Recap: Softmax

- The transformation $\text{softmax} : \mathbb{R}^K \rightarrow \Delta^{K-1}$ is defined as:

$$\text{softmax}_i(z) = \frac{\exp(z_i)}{\sum_{k=1}^{K} \exp(z_k)}$$

- Resulting distribution has full support: $\text{softmax}(z) > 0, \forall z$
- A disadvantage if a sparse probability distribution is desired
- Common workaround: threshold and truncate
We propose as an alternative:

$$\text{sparsemax}(z) := \arg\min_{p \in \Delta^{K-1}} \|p - z\|^2.$$  

In words: Euclidean projection of $z$ onto the probability simplex

Likely to hit the boundary of the simplex, in which case \text{sparsemax}(z) becomes sparse (hence the name)

We’ll see that \text{sparsemax} retains many of the properties of softmax, having in addition the ability of producing sparse distributions!
Sparsemax in Closed Form

- Projecting onto the simplex amounts to a soft-thresholding operation:
  \[ \text{sparsemax}_i(z) = \max\{0, z_i - \tau\} \]
  where \( \tau \) is a normalizing constant such that \( \sum_j \max\{0, z_j - \tau\} = 1 \)
- To evaluate the sparsemax, all we need is to compute \( \tau \)
- Runtime is \( O(K \log K) \) with a naive sort; \( O(K) \) using linear-time selection (Pardalos and Kovoor, 1990; Duchi et al., 2008)
- Evaluating softmax costs \( O(K) \) too
What about Backprop?

- Sparsemax is differentiable almost everywhere
- Backprop is more efficient than softmax: runtime linear in the number of nonzeros
- See Martins and Astudillo (2016) for details
Two Dimensions

■ Parametrize $z = (t, 0)$
■ The 2D softmax is the logistic (sigmoid) function:

$$\text{softmax}_1(z) = (1 + \exp(-t))^{-1}$$

■ The 2D sparsemax is the “hard” version of the sigmoid:
Three Dimensions

- Parameterize $z = (t_1, t_2, 0)$ and plot $\text{softmax}_1(z)$ and $\text{sparsemax}_1(z)$ as a function of $t_1$ and $t_2$

- sparsemax is piecewise linear, but asymptotically similar to softmax
This gives us all the ingredients to use sparsemax inside a neural network (e.g. a “sparse” attention mechanism)
Neural Networks with Attention Mechanisms

- SNLI corpus (Bowman et al., 2015): 570K sentence pairs (a premise and an hypothesis), labeled as **entailment**, **contradiction**, or **neutral**
- We used an attention-based architecture as Rocktäschel et al. (2015)

---

**Diagram:**

```
  u
     |________|
     |        |
  |        |        |
  |        |        |
  |________|

  h1 ---- h2 ---- h3 ---- h4 ---- h5
     |        |        |        |
     |        |        |        |
     |________|

  x1 ---- x2 ---- x3 ---- x4 ---- x5
     |        |        |        |
     |        |        |        |
     |________|

  h6 ---- h7 ---- h8 ---- h9
     |        |        |        |
     |        |        |        |
     |________|

A wedding party taking pictures  ||  Someone got married
(premise)                        (hypothesis)
```
Experimental Results

Four neural attention strategies:

- **NoAttention**, a RNN-based system without attention
- **LogisticAttention**, which uses independent logistic activations
- **SoftAttention**, using a softmax attention-based system
- **SparseAttention**, using a sparsemax attention-based system

<table>
<thead>
<tr>
<th></th>
<th>Dev Acc.</th>
<th>Test Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoAttention</td>
<td>81.84</td>
<td>80.99</td>
</tr>
<tr>
<td>LogisticAttention</td>
<td>82.11</td>
<td>80.84</td>
</tr>
<tr>
<td>SoftAttention</td>
<td>82.86</td>
<td>82.08</td>
</tr>
<tr>
<td>SparseAttention</td>
<td>82.52</td>
<td><strong>82.20</strong></td>
</tr>
</tbody>
</table>

- Soft and sparse-activated attention systems perform similarly
- Both outperform the **NoAttention** and **LogisticAttention** systems
Some Examples

- *In blue*, the premise words selected by **SparseAttention**
- *In red*, the hypothesis
- Only a few words are selected, which are key for the system’s decision
- The sparsemax activation yields a compact and more interpretable selection, which can be particularly useful in long sentences

A boy *rides on a camel* in a crowded area while talking on his cellphone.
—— A boy is riding an animal. [entailment]

A young girl wearing *a pink coat* plays with a *yellow* toy golf club.
—— A girl is wearing a blue jacket. [contradiction]

Two black dogs are *frolicking* around the *grass together*.
—— Two dogs swim in the lake. [contradiction]

A man wearing a yellow striped shirt *laughs* while *seated next* to another *man* who is wearing a light blue shirt and *clasping* his hands together.
—— Two mimes sit in complete silence. [contradiction]
More: Sparsemax as a Loss Function

- Sparsemax can also be used in the **output layer**, replacing logistic/cross-entropy loss
- There is a continuous family of transformations that includes both softmax and sparsemax
- The corresponding loss functions are called **Fenchel-Young losses**

Mathieu Blondel, André F. T. Martins, and Vlad Niculae.

“Learning Classifiers with Fenchel-Young Losses: Generalized Entropies, Margins, and Algorithms.”

Arxiv preprint 2018.
Outline

1 Sparsity

2 Constraints

3 Latent Structure

4 Conclusions
Sparse and Constrained Attention


Constrained Softmax

Constrained softmax resembles softmax, but it allows imposing hard constraints on the maximal probability assigned to each word.

- Given scores $z \in \mathbb{R}^K$ and upper bounds $u \in \mathbb{R}^K$:

$$\text{csoftmax}(z; u) = \arg\min_{p \in \Delta^{K-1}} \text{KL}(p \parallel \text{softmax}(z)) \quad \text{s.t.} \quad p \leq u$$

- Related to posterior regularization (Ganchev et al., 2010)

Particular cases:

- If $u \geq 1$, all constraints are loose and this reduces to softmax
- If $u \in \Delta^{K-1}$, they are tight and we must have $p = u$
How to Evaluate?

Forward computation takes $O(K \log K)$ time (Martins and Kreutzer, 2017):

- Let $\mathcal{A} = \{ i \in [K] \mid p_i^* < u_i \}$ be the constraints that are met strictly
- Then by writing the KKT conditions we can express the solution as:

$$p_i^* = \min \left\{ \frac{\exp(z_i)}{Z}, u_i \right\} \quad \forall i \in [K], \quad \text{where} \quad Z = \frac{\sum_{i \in \mathcal{A}} \exp(z_i)}{1 - \sum_{i \notin \mathcal{A}} u_i}.$$

- Identifying the set $\mathcal{A}$ can be done in $O(K \log K)$ time with a sort
How to Backpropagate?

We need to compute gradients with respect to both \( z \) and \( u \)

**Can be done in \( O(K) \) time** (Martins and Kreutzer, 2017):

- Let \( L(\theta) \) be a loss function, \( dp = \nabla_\alpha L(\theta) \) be the output gradient, and \( dz = \nabla_z L(\theta) \) and \( du = \nabla_u L(\theta) \) be the input gradients.

- Then, the input gradients are given as:

\[
    dz_i = 1(i \in A)p_i(dp_i - m),
\]

\[
    du_i = 1(i \notin A)(dp_i - m),
\]

where \( m = (\sum_{i \in A} p_i dp_i)/(1 - \sum_{i \notin A} u_i) \).
This opens the door for using constrained softmax attention in a neural network, backpropagating through the scores and the upper bounds...

Constrained Sparsemax (Malaviya et al., 2018)

Similar idea, but replacing softmax by sparsemax:

- Given scores $z \in \mathbb{R}^K$ and upper bounds $u \in \mathbb{R}^K$:

$$\text{csparsemax}(z; u) = \arg\min_{p \in \Delta^{K-1}} \|p - z\|^2$$

subject to $p \leq u$

- Both sparse and upper bounded
- If $u \geq 1$, all constraints are loose and this reduces to sparsemax
- If $u \in \Delta^{K-1}$, they are tight and we must have $p = u$
How to Evaluate?

Forward computation can be done with a sort in $O(K\log K)$ time

Can be reduced to $O(K)$ (Malaviya et al., 2018; Pardalos and Kovoor, 1990):

- Let $\mathcal{A} = \{i \in [K] \mid 0 < p_i^\ast < u_i\}$ be the constraints that are met strictly
- Let $\mathcal{A}_R = \{i \in [K] \mid p_i^\ast = u_i\}$
- Then by writing the KKT conditions we can express the solution as:
  
  $$p_i^\ast = \max\{0, \min\{u_i, z_i - \tau\}\} \quad \forall i \in [K],$$

  where $\tau$ is a constant.

- Identifying the sets $\mathcal{A}$ and $\mathcal{A}_R$ can be done in $O(K\log K)$ time with a sort
How to Backpropagate?

We need to compute gradients with respect to both $z$ and $u$

**Can be done in sublinear time** $O(|\mathcal{A}| + |\mathcal{A}_R|)$ (Malaviya et al., 2018):

- Let $L(\theta)$ be a loss function, $d\mathbf{p} = \nabla_\alpha L(\theta)$ be the output gradient, and $d\mathbf{z} = \nabla_z L(\theta)$ and $d\mathbf{u} = \nabla_u L(\theta)$ be the input gradients.
- Then, the input gradients are given as:

\[
\begin{align*}
    d\mathbf{z}_i &= 1(i \in \mathcal{A})(dp_i - m) \\
    d\mathbf{u}_i &= 1(i \in \mathcal{A}_R)(dp_i - m),
\end{align*}
\]

where $m = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} dp_i$. 
Next, we show how to use these constrained attentions in neural machine translation decoders, inspired by the idea of *fertility* (IBM Model 2)…
Modeling Fertility in NMT

We do the following procedure:

1. Align the training data with fast_align
2. Train a separate BILSTM to predict fertility $f_i$ for each word
3. At each decoder step, use upper bound $u_i = f_i - \beta_i$ for each word, where $\beta_i$ is the cumulative attention

See Malaviya et al. (2018) for more details.
Example: Source Sentence with Three Words

Assume each word is given fertility 1:

softmax

sparsemax

csoftmax

csparsemax
BLEU Scores
Baselines are softmax and two other coverage models (Wu et al., 2016; Tu et al., 2016)

BLEU Scores

<table>
<thead>
<tr>
<th></th>
<th>De-En</th>
<th>Ja-En</th>
<th>Ro-En</th>
</tr>
</thead>
<tbody>
<tr>
<td>softmax</td>
<td>29.51</td>
<td>20.36</td>
<td>29.67</td>
</tr>
<tr>
<td>softmax + CovPenalty</td>
<td>29.69</td>
<td>20.7</td>
<td>29.81</td>
</tr>
<tr>
<td>softmax + CovVector</td>
<td>29.63</td>
<td>21.53</td>
<td>30.08</td>
</tr>
<tr>
<td>csparsemax</td>
<td>29.85</td>
<td>21.31</td>
<td>29.77</td>
</tr>
</tbody>
</table>

METEOR Scores

<table>
<thead>
<tr>
<th></th>
<th>De-En</th>
<th>Ja-En</th>
<th>Ro-En</th>
</tr>
</thead>
<tbody>
<tr>
<td>softmax</td>
<td>31.43</td>
<td>23.83</td>
<td>32.05</td>
</tr>
<tr>
<td>softmax + CovPenalty</td>
<td>31.43</td>
<td>24.12</td>
<td>32.15</td>
</tr>
<tr>
<td>softmax + CovVector</td>
<td>31.54</td>
<td>24.5</td>
<td>32.22</td>
</tr>
<tr>
<td>csparsemax</td>
<td>31.76</td>
<td>24.51</td>
<td>32.1</td>
</tr>
</tbody>
</table>
Coverage Scores

Account for **repetitions** and **dropped** source words (lower is better):

**REP Scores**

<table>
<thead>
<tr>
<th></th>
<th>De-En</th>
<th>Ja-En</th>
<th>Ro-En</th>
</tr>
</thead>
<tbody>
<tr>
<td>softmax</td>
<td>3.37</td>
<td>13.48</td>
<td>2.45</td>
</tr>
<tr>
<td>softmax+ CovPenalty</td>
<td>2.93</td>
<td>11.07</td>
<td>2.48</td>
</tr>
<tr>
<td>softmax+ CovVector</td>
<td>3.47</td>
<td>11.4</td>
<td>2.42</td>
</tr>
</tbody>
</table>

**DROP Scores**

<table>
<thead>
<tr>
<th></th>
<th>De-En</th>
<th>Ja-En</th>
<th>Ro-En</th>
</tr>
</thead>
<tbody>
<tr>
<td>softmax</td>
<td>5.89</td>
<td>23.3</td>
<td>5.59</td>
</tr>
<tr>
<td>softmax+ CovPenalty</td>
<td>5.74</td>
<td>22.79</td>
<td>5.49</td>
</tr>
<tr>
<td>softmax+ CovVector</td>
<td>5.65</td>
<td>22.18</td>
<td>5.47</td>
</tr>
</tbody>
</table>

Helsinki, 28/9/18
Softmax (left) vs Constrained Sparsemax (right) for De-En:
### Sentence Examples

<table>
<thead>
<tr>
<th>input</th>
<th>so ungefähr, sie wissen schon.</th>
</tr>
</thead>
<tbody>
<tr>
<td>reference</td>
<td>like that, you know.</td>
</tr>
<tr>
<td>softmax</td>
<td>so, you know, you know.</td>
</tr>
<tr>
<td>sparsemax</td>
<td>so, you know, you know.</td>
</tr>
<tr>
<td>csoftmax</td>
<td>so, you know, you know.</td>
</tr>
<tr>
<td>csparsemax</td>
<td>like that, you know.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>input</th>
<th>und wir benutzen dieses wort mit solcher verachtung.</th>
</tr>
</thead>
<tbody>
<tr>
<td>reference</td>
<td>and we say that word with such contempt.</td>
</tr>
<tr>
<td>softmax</td>
<td>and we use this word with such contempt.</td>
</tr>
<tr>
<td>sparsemax</td>
<td>and we use this word with such contempt.</td>
</tr>
<tr>
<td>csoftmax</td>
<td>and we use this word with like this.</td>
</tr>
<tr>
<td>csparsemax</td>
<td>and we use this word with such contempt.</td>
</tr>
</tbody>
</table>
Code (Pytorch + OpenNMT):

www.github.com/Unbabel/sparse_constrained_attention
Outline

1. Sparsity
2. Constraints
3. Latent Structure
4. Conclusions
SparseMAP

Vlad Niculae, André F. T. Martins, Mathieu Blondel, and Claire Cardie.
“SparseMAP: Differentiable Sparse Structured Inference.”
ICML 2018.
SparseMAP

- Generalizes sparsemax to **sparse structured prediction**
- Works both as output layer and hidden layer
- With latent models, similar to structured attention networks (Kim et al., 2017), but **sparse**
- Efficient forward/backprop requiring only an argmax (MAP) oracle!
Two Scenarios:

- Structured output prediction
- Latent structured inference
Structured Output Prediction

- Many NLP tasks require predicting linguistic structure as output
- Examples: sequence tagging, dependency parsing, alignments
Latent Structured Inference

- Sometimes it’s convenient to induce linguistic structure as a latent variable for some downstream task
- Examples: latent syntax for MT; latent alignments for NLI
Marginal Polytope

- Vertices are codewords of combinatorial structures
- Points correspond to marginal distributions over those structures
Structured Inference

<table>
<thead>
<tr>
<th>Unstructured</th>
<th>Structured</th>
</tr>
</thead>
<tbody>
<tr>
<td>argmax</td>
<td>MAP inference</td>
</tr>
<tr>
<td>softmax</td>
<td>Marginal inference</td>
</tr>
<tr>
<td>sparsemax</td>
<td>?</td>
</tr>
<tr>
<td>Unstructured</td>
<td>Structured</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>argmax</td>
<td>MAP inference</td>
</tr>
<tr>
<td>softmax</td>
<td>Marginal inference</td>
</tr>
<tr>
<td>sparsemax</td>
<td>SparseMAP</td>
</tr>
</tbody>
</table>
Sparse Structured Prediction

SparseMAP yields a **sparse combination of vertices**, hence it selects only a small number of structures (out of exponentially many)
Efficiently Computing SparseMAP

Boils down to **projecting onto the marginal polytope**

**Key Result:** can be solved as a (small) sequence of argmax (MAP) calls

Gradient backprop comes for free once we have done forward!
Example: Latent Structured Alignments in SNLI

(a) softmax

(b) sequence

(c) matching
Example: Dependency Parsing

![Bar chart showing performance of different models in various languages.

- ssvm
- crf
- smap
- m-smap

Languages: en, zh, vi, ro, ja]
Example: Dependency Parsing

- Suitable for capturing ambiguity in natural language!
Learning to be Sparse

![Graphs showing the decrease in norm for both training and validation sets over epochs.](image)

- Training and Validation norms over epochs.
- \( \|p\|_0 \) and \( \|\mu\|_0/n \) norms decreasing.

Author: André Martins (Unbabel/IT)

Helsinki, 28/9/18
Related Work

- Structured attention networks (Kim et al., 2017): not sparse
- SPIGOT (Peng et al., 2018): different framework, same building blocks (our active set algo for polytope projection applies there too)
- ... but SPIGOT gradients are *inexact* while ours are exact
- Fusedmax (and other structured sparse) attention (Niculae and Blondel, 2017):
Outline

1 Sparsity

2 Constraints

3 Latent Structure

4 Conclusions
Conclusions

- Transformations from real numbers to distributions are ubiquitous
- We introduced new transformations that handle sparsity, constraints, and structure
- All are differentiable and their gradients are efficient to compute
- Can be used as hidden layers or as output layers
- Various experiments in NMT and sentence pair tasks, with improved interpretability
- Recent work: dynamically determining the computation graph based on the SparseMAP selected structures
Vlad Niculae, André F. T. Martins and Claire Cardie

“Towards Dynamic Computation Graphs via Sparse Latent Structure”

EMNLP 2018
ERC project **DeepSPIN** (Deep Structured Prediction in NLP)

- ERC starting grant, started in 2018
- Post-doc positions may open next year
- Topics: deep learning, structured prediction, NLP, machine translation
- Involving Unbabel and the University of Lisbon
- More details: [https://deep-spin.github.io](https://deep-spin.github.io)
We’re Hiring at Unbabel!

Excited about MT, NLP, and Lisbon? ⇒ jobs@unbabel.com.

Open positions: ML/NLP Software Engineer, Sr Research Scientist
### Lisbon 5 Day Weather

**4:20 pm WEST**

<table>
<thead>
<tr>
<th>DAY</th>
<th>DESCRIPTION</th>
<th>HIGH / LOW</th>
<th>PRECIP</th>
<th>WIND</th>
<th>HUMIDITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>TONIGHT</td>
<td>Clear</td>
<td>-/-21°</td>
<td>0%</td>
<td>NNW 23 km/h</td>
<td>57%</td>
</tr>
<tr>
<td>SEP 27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FRI</td>
<td>Mostly Sunny</td>
<td>31°/18°</td>
<td>0%</td>
<td>NNE 22 km/h</td>
<td>54%</td>
</tr>
<tr>
<td>SEP 28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAT</td>
<td>Sunny</td>
<td>30°/17°</td>
<td>0%</td>
<td>N 13 km/h</td>
<td>55%</td>
</tr>
<tr>
<td>SEP 29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUN</td>
<td>Sunny</td>
<td>30°/18°</td>
<td>10%</td>
<td>NNW 16 km/h</td>
<td>59%</td>
</tr>
<tr>
<td>SEP 30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MON</td>
<td>Partly Cloudy</td>
<td>30°/18°</td>
<td>0%</td>
<td>NNW 19 km/h</td>
<td>49%</td>
</tr>
<tr>
<td>OCT 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TUE</td>
<td>Sunny</td>
<td>31°/17°</td>
<td>0%</td>
<td>ENE 18 km/h</td>
<td>33%</td>
</tr>
<tr>
<td>OCT 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


References II


