From Sparse Modeling to Sparse Communication

André Martins

IC Colloquium EPFL, Lausanne, October 3, 2022
Our Amazing Team

SARDINE: Structure AwaRe moDellng for Natural LanguagE
DeepSPIN

- ERC starting grant (2018–23)
- Goal: put together *deep learning* and *structured prediction* for *natural language processing*
- More details: [https://deep-spin.github.io](https://deep-spin.github.io)
From Sparse Modeling ...

- Mostly used with linear models, lots of work in the 2000s
- Main idea: embed a sparse regularizer (e.g. $\ell_1$-norm) in the learning objective
- Irrelevant features get zero weight and can be discarded
- Extensions to structured sparsity (group-lasso, fused-lasso, etc.)

... to Sparse Communication:

- Mostly used with neural networks, most work after 2015
- Main idea: sparse neuron activations (biological plausibility)
- Predictions are triggered by a few neurons only (input-dependent)
- Example: ReLUs, dropout, sparse attention mechanisms
This Talk

An inventory of transformations that capture sparsity and structure:

- All differentiable (efficient forward and backward propagation)
- Can be used at hidden (attention) or output layers (loss)
- Can make a bridge between the continuous and discrete worlds
- Effective in several natural language processing tasks.

Building block:

Sparse transformations from the Euclidean space to the simplex $\triangle$. 

\[
\begin{align*}
\cdot z \\
\text{p}^\bullet
\end{align*}
\]
Machine-Human Communication

Continuous

Input

Hidden

Output

Discrete

A 文
The Bell System Technical Journal

Vol. XXVII
July, 1948
No. 3

A Mathematical Theory of Communication
By C. E. SHANNON

INTRODUCTION

PART I: DISCRETE NOISELESS SYSTEMS

PART II: THE DISCRETE CHANNEL WITH NOISE

PART III: MATHEMATICAL PRELIMINARIES

PART IV: THE CONTINUOUS CHANNEL

PART V: THE RATE FOR A CONTINUOUS SOURCE
PART I: DISCRETE NOISELESS SYSTEMS

PART II: THE DISCRETE CHANNEL WITH NOISE

PART III: MATHEMATICAL PRELIMINARIES

PART IV: THE CONTINUOUS CHANNEL

PART V: THE RATE FOR A CONTINUOUS SOURCE
PART I: DISCRETE NOISELESS SYSTEMS

PART II: THE DISCRETE CHANNEL WITH NOISE

PART III: MATHEMATICAL PRELIMINARIES

PART IV: THE CONTINUOUS CHANNEL

PART V: THE RATE FOR A CONTINUOUS SOURCE
Commonly we have to opt between discrete or continuous models:

- Language is symbolic and discrete
- Neural networks use (and learn) continuous representations

We should look at what happens in-between!

Sparsity might help with this, but...
Commonly we have to opt between **discrete** or **continuous** models:

- Language is symbolic and *discrete*
- Neural networks use (and learn) *continuous* representations

We should look at what happens in-between!

**Sparsity** might help with this, but...

... sparse probabilities are understudied and often excluded from theory:

- Hammersley-Clifford theorem in graphical models
- Pitman-Koopman-Darmois theorem (sufficient statistics and exponential families)
- Log-likelihood is $-\infty$ if estimated probability is 0.
Motivating Example: John’s Life

John splits his day as follows: he works 8h/day, and stays home 16h/day.
Motivating Example: John’s Life

John splits his day as follows: he works 8h/day, and stays home 16h/day.
Motivating Example: John’s Life

John splits his day as follows: he works 8h/day, and stays home 15h/day. He is in transit 1h/day to commute to work and back.
Motivating Example: John’s Life

John splits his day as follows: he works 8h/day, and stays home 15h/day. He is in transit 1h/day to commute to work and back.
Motivating Example: John’s Life

John splits his day as follows: he works 8h/day, and stays home 15h/day. He is in transit 1h/day to commute to work and back.

Is John’s location a discrete or continuous random variable?
Motivating Example: John’s Life

John splits his day as follows: he works 8h/day, and stays home 15h/day. He is in transit 1h/day to commute to work and back.

Is John’s location a discrete or continuous random variable? It’s mixed.
Outline

1. Sparse Transformations
2. Fenchel-Young Losses
3. Mixed Distributions
4. Conclusions
Recap: Softmax and Argmax

Softmax exponentiates and normalizes:

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

- **Fully dense**: $\text{softmax}(z) > 0, \forall z$
- Used both as a loss function (cross-entropy) and for attention.
Recap: Softmax and Argmax

Softmax exponentiates and normalizes:

\[
\text{softmax}(z) = \frac{\exp(z)}{\sum_{k=1}^{K} \exp(z_k)}
\]

- **Fully dense**: \( \text{softmax}(z) > 0, \forall z \)
- Used both as a loss function (cross-entropy) and for attention.

Argmax can be written as:

\[
\text{argmax}(z) := \arg \max_{p \in \triangle} z^\top p
\]

\[
= \lim_{\tau \to 0^+} \text{softmax}(z/\tau) \quad \text{(temperature trick)}
\]

- Retrieves a **one-hot vector** for the highest scored index.
Argmax is an extreme case of sparsity, but it is **discontinuous**.

Is there a **sparse** and **differentiable** alternative?
softmax($z$)  

sparsemax($z$)  

argmax($z$)  

(Same $z = [1.0716, -1.1221, -0.3288, 0.3368, 0.0425]$)

- Argmax is an extreme case of sparsity, but it is **discontinuous**.
- Is there a **sparse** and **differentiable** alternative?
**Sparsemax** (Martins and Astudillo, 2016, ICML)

Euclidean projection of $z$ onto the probability simplex $\Delta$:

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

$$= \arg \max_{p \in \Delta} z^\top p - \frac{1}{2} \|p\|^2.$$

- Likely to hit the boundary of the simplex, in which case $\text{sparsemax}(z)$ becomes sparse (hence the name)
- End-to-end differentiable
- Forward pass: $O(K \log K)$ or $O(K)$, (almost) as fast as softmax
- Backprop: sublinear, **better than softmax**!
Sparsemax in 2D and 3D

(Martins and Astudillo, 2016, ICML)

Sparsemax is piecewise linear, but asymptotically similar to softmax.
For convex $\Omega$, define the $\Omega$-regularized argmax transformation:

$$\text{argmax}_\Omega(z) := \arg\max_{p \in \Delta} z^\top p - \Omega(p)$$

- **Argmax** corresponds to no regularization, $\Omega \equiv 0$
- **Softmax** amounts to entropic regularization, $\Omega(p) = \sum_{i=1}^{K} p_i \log p_i$
- **Sparsemax** amounts to $\ell_2$-regularization, $\Omega(p) = \frac{1}{2} \| p \|^2$

Is there something in-between?
Entmax (Peters et al., 2019, ACL)

Parametrized by $\alpha \geq 0$:

$$
\Omega_{\alpha}(p) := \begin{cases} 
\frac{1}{\alpha(\alpha-1)} \left(1 - \sum_{i=1}^{K} p_i^{\alpha}\right) & \text{if } \alpha \neq 1 \\
\sum_{i=1}^{K} p_i \log p_i & \text{if } \alpha = 1.
\end{cases}
$$

Related to Tsallis generalized entropies (Tsallis, 1988).

- **Argmax** corresponds to $\alpha \to \infty$
- **Softmax** amounts to $\alpha \to 1$
- **Sparsemax** amounts to $\alpha = 2$.

**Key result:** always sparse for $\alpha > 1$, sparsity increases with $\alpha$

- Forward pass for general $\alpha$ can be done with a bisection algorithm
- Backward pass runs in sublinear time.
Entmax in 2D (Peters et al., 2019, ACL)

$\alpha = 1.5$ is a sweet spot!

- Efficient exact algorithm (nearly as fast as softmax), smooth, and good empirical performance.

Pytorch code: https://github.com/deep-spin/entmax
Sparse Transformations (Peters et al., 2019, ACL)

\[ \alpha = 1 \]
softmax(\( z \))

\[ \alpha = 1.5 \]
1.5-entmax(\( z \))

\[ \alpha = 2 \]
sparsemax(\( z \))

\[ \alpha = \infty \]
argmax(\( z \))

(Same \( z = [1.0716, -1.1221, -0.3288, 0.3368, 0.0425] \))
Example: Sparse Attention for Machine Translation

(Peters et al., 2019, ACL)

- **Selects** source words when generating a target word (sparse alignments)
- Better interpretability
- Can also model fertility: **constrained sparsemax**
  (Malaviya et al., 2018, ACL)
- Can also learn $\alpha$ (adaptively sparse transformers):
  (Correia et al., 2019, EMNLP)
Example: Sparse Attention for Explainability

(Treviso and Martins, 2020, BlackboxNLP)

- A classifier makes a prediction
- An “explainer” (embedded or not in the classifier) generates a sparse message that explains the classifier’s decision
- The layperson receives the message and tries to guess the classifier’s prediction (also called simulatability, forward simulation/prediction)
- Communication success rate: how often the two predictions match?
New: Scaffold Maximizing Training (SMaT)

(Fernandes et al., 2022, NeurIPS)

- Similar to above, but uses **bilevel optimization** (a la meta-learning) to \textit{learn} to optimize model explanations for teaching.
- The teacher explainer is a linear combination of the classifier’s attention layers and is used to regularize the student (Pruthi et al., 2022)
- The performance of the student in held-out data (simulability loss) is used as an objective for the teacher explainer.
Other Related Transformations

Constrained softmax (Martins and Kreutzer, 2017, EMNLP),

Constrained sparsemax (Malaviya et al., 2018, ACL):
- Allows placing a **budget** on how much attention a word can receive
- Useful to model **fertility** in machine translation

Fusedmax (Niculae and Blondel, 2017, NeurIPS):
- Can promote **structured sparsity** (contiguous selection)

(LP-)SparseMAP Niculae et al. (2018, ICML), Niculae and Martins (2020, ICML):
- Extends sparsemax to **sparse structured prediction**.
- Can be used as hidden differentiable layer or output layer.
- Works with arbitrary factor graph (e.g. logic constraints).
Sparse and Continuous Attention
(Martins et al., 2020a, NeurIPS)

- So far: attention over a finite set (words, pixel regions, etc.)
- We generalize attention to arbitrary sets, possibly continuous.
- Applications: VQA; long-range $\infty$-former (Martins et al., 2022, ACL)
Outline

1. Sparse Transformations
2. Fenchel-Young Losses
3. Mixed Distributions
4. Conclusions
Entmax can also be used as a loss in the output layer (to replace logistic/cross-entropy loss)

However, not expressed as a log-likelihood (which could lead to log(0) problems due to sparsity)

Instead, we build a entmax loss inspired by Fenchel-Young losses.
Recap: $\Omega$-Regularized Argmax (Niculae and Blondel, 2017, NeurIPS)

For convex $\Omega$, define the $\Omega$-regularized argmax transformation:

$$\text{argmax}_\Omega(z) := \arg\max_{p \in \Delta} z^\top p - \Omega(p)$$

- **Argmax** corresponds to no regularization, $\Omega \equiv 0$
- **Softmax** amounts to entropic regularization, $\Omega(p) = \sum_{i=1}^K p_i \log p_i$
- **Sparsemax** amounts to $\ell_2$-regularization, $\Omega(p) = \frac{1}{2} \| p \|^2$

All these are particular cases of $\alpha$-entmax (Peters et al., 2019, ACL).
Fenchel-Young Losses (Blondel et al., 2020, JMLR)

Assess compatibility between groundtruth \( q \in \Delta \) and scores \( z \in \mathbb{R}^K \)

Convex conjugate \( \Omega^*(z) := \max_{p \in \Delta} z^\top p - \Omega(p) \)

\[
L_{\Omega}(z, q) := \Omega^*(z) + \Omega(q) - z^\top q
\]
Fenchel-Young Losses (Blondel et al., 2020, JMLR)

Assess compatibility between groundtruth $q \in \Delta$ and scores $z \in \mathbb{R}^K$

Convex conjugate $\Omega^*(z) := \max_{p \in \Delta} z^\top p - \Omega(p)$

$$L_\Omega(z, q) := \Omega^*(z) + \Omega(q) - z^\top q$$

Properties:

- $L_\Omega(z, q) \geq 0$ (automatic from Fenchel-Young inequality)
- $L_\Omega(z, q) = 0$ iff $q = \arg\max \Omega(z)$
- $L_\Omega$ is convex and differentiable with $\nabla L_\Omega(z, q) = \arg\max \Omega(z) - q$

Recovers cross-entropy loss, sparsemax loss, and many other known losses

Also called “mixed-type Bregman divergences” (Amari, 2016).
Key result: for all $\alpha > 1$, all transformations are \textbf{sparse} and lead to losses with \textbf{margins}!

The \textbf{margin size} is related to the \textbf{slope} of the entropy in the simplex corners! ($\frac{1}{\alpha - 1}$ for entmax losses.)

See paper for details!

Pytorch code: \url{https://github.com/deep-spin/entmax}
Example: Machine Translation

(Peters et al., 2019, ACL) (Peters and Martins, 2021, NAACL)

- Only a few words get non-zero probability at each time step
- Auto-completion when several words in a row have probability 1
- Useful for predictive translation.

(Source: “Dies ist ein weiterer Blick auf den Baum des Lebens.”)
Use the entmax loss for training language models.
At test time, sample from this sparse distribution.
Better quality with less repetitions than other methods:
Outline

1. Sparse Transformations
2. Fenchel-Young Losses
3. Mixed Distributions
4. Conclusions
We saw how to obtain sparse probability distributions.

How can we use them to bridge the gap between discrete and continuous domains?

We'll see how next.
John splits his day as follows: he works 8h/day, and stays home 15h/day. He is in transit 1h/day to commute to work and back.
Back to John’s Life

John splits his day as follows: he works 8h/day, and stays home 15h/day. He is in transit 1h/day to commute to work and back.

That’s a sad life!
After work, John spends 2h in the pub with friends.
After work, John spends 2h in the pub with friends.
After work, John spends 2h in the pub with friends.
After work, John spends 2h in the pub with friends.
After work, John spends 2h in the pub with friends.

We need a way to represent this probability mass in vertices, edges, face.
Densities over $\triangle_{K-1}$

We denote by $\text{ri}(\triangle_{K-1})$ the relative interior of $\triangle_{K-1}$.

Common densities on the simplex:

- Dirichlet distribution
- Logistic-Normal (a.k.a. Gaussian-Softmax)
- Concrete (a.k.a. Gumbel-Softmax)

None of these place any probability mass on the boundary $\triangle_{K-1} \setminus \text{ri}(\triangle_{K-1})$. 
Dirichlet Distribution

\[ Y \sim \text{Dirichlet}(\alpha) \iff p_Y(y; \alpha) \propto \prod_{k=1}^{K} y_k^{\alpha_k-1}, \quad \alpha > 0. \]
Logistic Normal (a.k.a. Gaussian-Softmax)

(Atchison and Shen, 1980)

Generative story:

\[ Y \sim \text{LogisticNormal}(z, \Sigma) \Leftrightarrow N \sim \mathcal{N}(0, I) \]
\[ Y = \text{softmax}(z + \Sigma^{1/2} N). \]
Concrete (a.k.a. Gumbel-Softmax)
(Maddison et al., 2017; Jang et al., 2017)

Continuous relaxation of a categorical.
Approaches categorical as $\lambda \to 0^+$ (Luce, 1959; Papandreou and Yuille, 2011).

Generative story:

\[ Y \sim \text{Concrete}(z, \lambda) \iff G_k \sim \text{Gumbel}(0, 1) \]
\[ Y = \text{softmax}(\lambda^{-1}(z + G)) \]

\( \lambda = 0 \) \quad \lambda = 1/2 \quad \lambda = 1 \quad \lambda = 2 \)
When $K = 2$, the simplex is isomorphic to unit interval, $\triangle_1 \simeq [0, 1]$. A point in $\triangle_1$ can be represented as $y = [y, 1 - y]$.

**Truncated** densities have been proposed for $K = 2$:

- Binary Hard Concrete
- Rectified Gaussian
Binary Hard Concrete

(Louizos et al., 2018)

- Stretches the Concrete and applies a “hard” sigmoid transformation to place point masses at 0 and 1.
- Similar to spike-and-slab (Mitchell and Beauchamp, 1988; Ishwaran et al., 2005).
Rectified Gaussian

(Hinton and Ghahramani, 1997; Palmer et al., 2017)

- Applies a “hard” sigmoid transformation to a univariate Gaussian.

\[ p_Y(y) = \mathcal{N}(y; z, \sigma^2) + \frac{1 - \text{erf} \left( \frac{z}{\sqrt{2\sigma}} \right)}{2} \delta_0(y) + \frac{1 + \text{erf} \left( \frac{z-1}{\sqrt{2\sigma}} \right)}{2} \delta_1(y). \]

Extending such distributions to the multivariate case \((K > 2)\) is non-trivial:

- Combinatorially many multiple order Diracs would be needed
- Dirac deltas have \(-\infty\) differential entropy.
Our Approach: Face Stratification

How to extend these “truncated densities” to $K > 2$?

Our solution relies on the face lattice of the simplex:

0-faces are vertices, 1-faces are edges, etc.

There is one $(K - 1)$-face: the simplex $\triangle_{K-1}$ itself.
Direct Sum Measure (Farinhas et al., 2022, ICLR)

Let $\mathcal{F}$ denote the set of proper faces of $\triangle_{K-1}$; we have $|\mathcal{F}| = 2^K - 1$.

We define a **direct sum measure** $\mu^\oplus$ on $\triangle_{K-1}$ as a sum of Lebesgue measures on each non-vertex face, and a counting measure on the vertices:

$$\mu^\oplus(A) = \sum_{f \in \mathcal{F}} \mu_f(A \cap \operatorname{ri}(f)), \quad A \subseteq \triangle_{K-1}.$$

We define probability densities w.r.t. this base measure.
Mixed Random Variables (Farinhas et al., 2022, ICLR)

Discrete RVs assign probability only to 0-faces (vertices of $\triangle_{K-1}$).
Continuous RVs assign probability only to the maximal face ($\text{ri}(\triangle_{K-1})$).
Mixed RVs generalize both: can assign probability to all faces of $\triangle_{K-1}$. 
Mixed Random Variables (Farinhas et al., 2022, ICLR)

Discrete RVs assign probability only to 0-faces (vertices of $\triangle_{K-1}$).
Continuous RVs assign probability only to the maximal face ($\text{ri}({\triangle}_{K-1})$).
**Mixed RVs generalize both:** can assign probability to all faces of $\triangle_{K-1}$.
They can be defined via:

- Their face probability mass function $P_F(f) = \Pr\{y \in \text{ri}(f)\}$, $f \in \mathcal{F}$.
- Their face-conditional densities $p_{Y|F}(y \mid f)$, for $f \in \mathcal{F}$, $y \in \text{ri}(f)$.

The probability of a set $A \subseteq \triangle_{K-1}$ is given by:

$$
\Pr\{y \in A\} = \sum_{f \in \mathcal{F}} P_F(f) \int_{A \cap \text{ri}(f)} p_{Y|F}(y \mid f).
$$
Two ways of characterizing mixed RVs:

- **Extrinsic characterization**: start with a distribution over $\mathbb{R}^K$ and then apply a deterministic transformation to project it to $\Delta_{K−1}$

- **Intrinsic characterization**: specify a mixture of distributions directly over the faces of $\Delta_{K−1}$, by specifying $P_F$ and $p_{Y|F}$ for each $f \in \mathcal{F}$
Uses an extrinsic characterization, via “stretch-and-project.”

Generative story:

\[ Y \sim \text{HardConcrete}(z, \lambda, \tau) \iff Y' \sim \text{Concrete}(z, \lambda) \]
\[ Y = \text{sparsemax}(\tau Y'), \quad \text{with} \quad \tau \geq 1. \]

- Recovers the binary Hard Concrete for \( K = 2 \)
- The larger \( \tau \), the higher the tendency to hit a non-maximal face of the simplex and induce sparsity.
Gaussian-Sparsemax (Farinhas et al., 2022, ICLR)

Uses an extrinsic characterization, by sampling from a Gaussian and projecting.

Generative story:

\[ Y \sim \text{GaussianSparsemax}(z, \Sigma) \iff N \sim \mathcal{N}(0, I) \]
\[ Y = \text{sparsemax}(z + \Sigma^{1/2} N). \]

- Sparsemax counterpart of the Logistic-Normal.
- Can assign nonzero probability mass to the boundary of the simplex.
- When \( K = 2 \), we recover the double-sided rectified Gaussian.
- For \( K > 2 \), an intrinsic representation can be expressed via the orthant probability of multivariate Gaussians.
Logistic-Normal (left) assigns zero probability to all faces but $\text{ri}(\Delta_{K-1})$

Gaussian-Sparsemax (right) is a **mixed distribution**: it assigns probability to the full simplex, including its boundary.
Mixed Dirichlet (Farinhas et al., 2022, ICLR)

Uses an intrinsic characterization.

- Uses two parameters: \( \mathbf{w} \in \mathbb{R}^K \) and \( \mathbf{\alpha} \in \mathbb{R}^K_{>0} \)
- First, sample a face \( f \sim P_F(f) \propto \prod_{k \in f} w_k \), where \( \mathbf{w} \in \mathbb{R}^K \)
- Then, sample \( Y|F = f \sim \text{Dir}(\mathbf{\alpha}|_f) \), where \( \mathbf{\alpha}|_f \) “masks out” entries of \( \mathbf{\alpha} \) not supported by \( f \).
- Sampling \( f \) can be done in \( \Theta(K) \) with dynamic programming.
“Direct sum” entropy using $\mu^\oplus$ as the base measure:

$$H^\oplus(Y) := H(F) + H(Y \mid F)$$

$$= -\sum_{f \in \mathcal{F}} P_F(f) \log P_F(f) + \sum_{f \in \mathcal{F}} P_F(f) \left( - \int p_{Y\mid F}(y \mid f) \log p_{Y\mid F}(y \mid f) \right).$$

- Average length of the optimal code where $f$ must be encoded losslessly and where $y\mid f$ has a predefined bit precision $N$
- Max-ent is written as a generalized Laguerre polynomial (see paper)
  - e.g. $\log_2(2 + 2^N)$ for $K = 2$ (vs. $\log_2(2) = 1$ in the purely discrete case)
- KL divergence and mutual information defined similarly.
Experiment: Emergent Communication

The first agent needs to communicate a code to the second agent that represents a given image.

Given the code, the second agent needs to identify the correct image among 16 possibilities. (Random guess is $1/16 = 6.25\%$.)

Success average and standard error over 10 runs:

<table>
<thead>
<tr>
<th>Method</th>
<th>Success (%)</th>
<th>Nonzeros</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gumbel-Softmax</td>
<td>78.84 ± 8.07</td>
<td>256</td>
</tr>
<tr>
<td>Gumbel-Softmax ST</td>
<td>49.96 ± 9.51</td>
<td>1</td>
</tr>
<tr>
<td>$K$-D Hard Concrete</td>
<td>76.07 ± 7.76</td>
<td>21.43 ± 17.56</td>
</tr>
<tr>
<td>Gaussian-Sparsemax</td>
<td>80.88 ± 0.50</td>
<td>1.57 ± 0.02</td>
</tr>
</tbody>
</table>

(See paper for more experiments with VAEs on FashionMNIST and MNIST.)
Outline

1. Sparse Transformations
2. Fenchel-Young Losses
3. Mixed Distributions
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 (Fenchel-Young losses).
- Mixed distributions are in-between the discrete and continuous worlds.
- Examples: Gaussian-Sparsemax, Gumbel-Sparsemax, Mixed Dirichlet.
- Sparse communication potentially useful as a path for explainability.
DeepSPIN (“Deep Structured Prediction in NLP”)

- ERC starting grant, started in 2018
- Topics: deep learning, structured prediction, NLP
- More details: https://deep-spin.github.io


References IV


