

# **Efficient Marginalization of Discrete and Structured Latent Variables via Sparsity**

**Gonçalo Correia** Instituto de Telecomunicações, Lisbon

**Vlad Niculae** Ivl, University of Amsterdam

**Wilker Aziz** ILLC, University of Amsterdam

**André Martins** Instituto de Telecomunicações & LUMLIS & Unbabel

# Latent Variable Models

Latent variable  $z$  can be

# Latent Variable Models

Latent variable  $z$  can be continuous



Source: Bouges et al., 2013

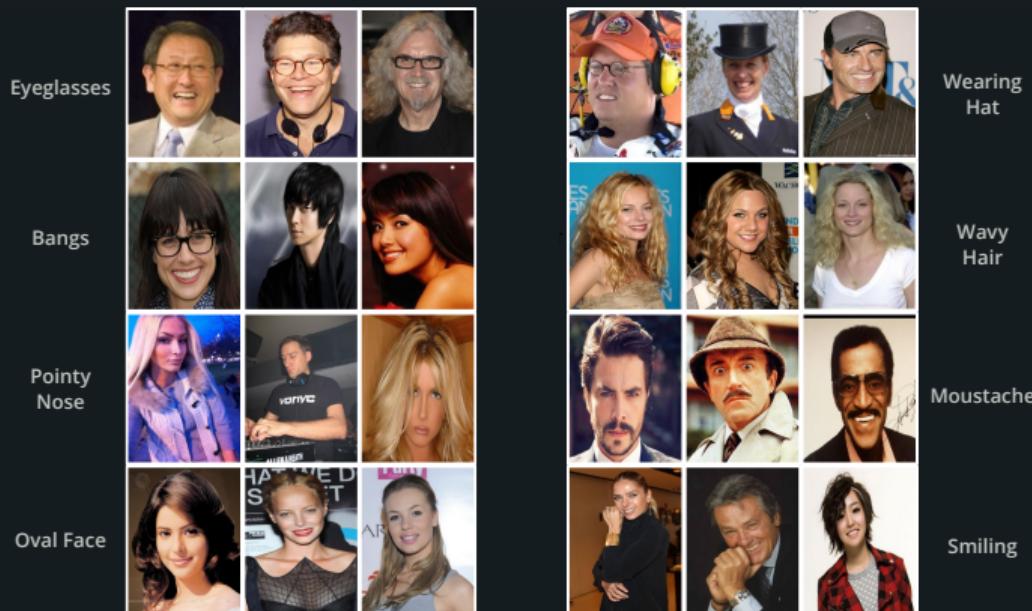
# Latent Variable Models

Latent variable  $z$  can be **continuous**, **discrete**



# Latent Variable Models

Latent variable  $z$  can be continuous, discrete, or structured



Source: Liu et al., 2015

# Training Discrete or Structured Latent Variable Models

Latent variable  $z$  can be

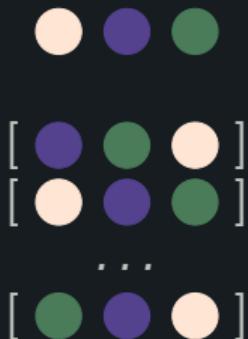
# Training Discrete or Structured Latent Variable Models

Latent variable  $z$  can be discrete



# Training Discrete or Structured Latent Variable Models

Latent variable  $z$  can be **discrete** or **structured**



# Training Discrete or Structured Latent Variable Models

Latent variable  $z$  can be **discrete** or **structured**

$\pi(z|x, \theta)$ : distribution over possible  $z$



# Training Discrete or Structured Latent Variable Models

Latent variable  $z$  can be **discrete** or **structured**

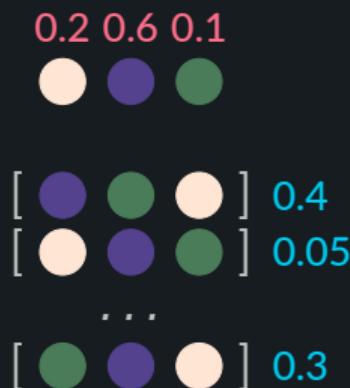
$\pi(z|x, \theta)$ : distribution over possible  $z$



# Training Discrete or Structured Latent Variable Models

Latent variable  $z$  can be **discrete** or **structured**

$\pi(z|x, \theta)$ : distribution over possible  $z$

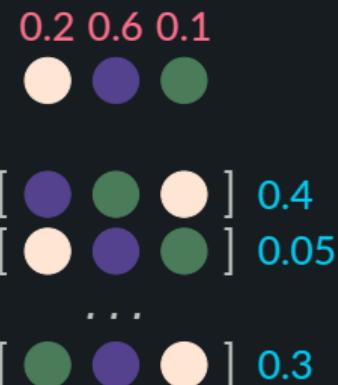


# Training Discrete or Structured Latent Variable Models

Latent variable  $z$  can be **discrete** or **structured**

$\pi(z|x, \theta)$ : distribution over possible  $z$

$\ell(x, z; \theta)$ : downstream loss: ELBO, Log-Likelihood, (...)



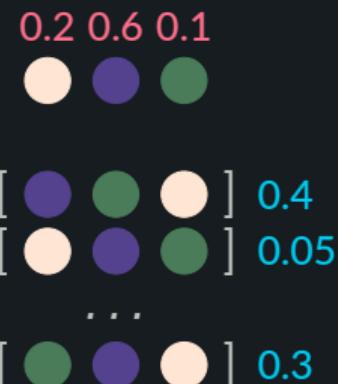
# Training Discrete or Structured Latent Variable Models

Latent variable  $z$  can be **discrete** or **structured**

$\pi(z|x, \theta)$ : distribution over possible  $z$

$\ell(x, z; \theta)$ : downstream loss: ELBO, Log-Likelihood, (...)

To train, we need to compute the following expectation:



# Training Discrete or Structured Latent Variable Models

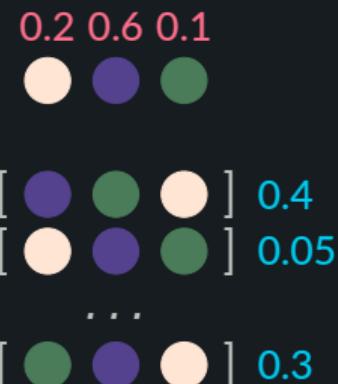
Latent variable  $z$  can be **discrete** or **structured**

$\pi(z|x, \theta)$ : distribution over possible  $z$

$\ell(x, z; \theta)$ : downstream loss: ELBO, Log-Likelihood, (...)

To train, we need to compute the following expectation:

$$\mathcal{L}_x(\theta) = \sum_{z \in \mathcal{Z}} \pi(z|x, \theta) \ell(x, z; \theta)$$

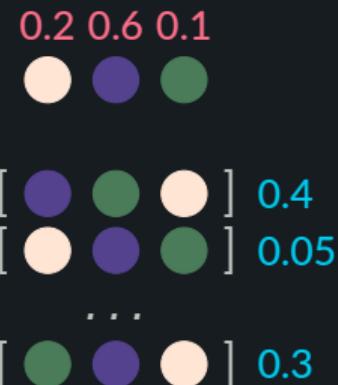


# Training Discrete or Structured Latent Variable Models

Latent variable  $z$  can be **discrete** or **structured**

$\pi(z|x, \theta)$ : distribution over possible  $z$

$\ell(x, z; \theta)$ : downstream loss: ELBO, Log-Likelihood, (...)



To train, we need to compute the following expectation:

$$\mathcal{L}_x(\theta) = \sum_{z \in \mathcal{Z}} \pi(z|x, \theta) \ell(x, z; \theta)$$

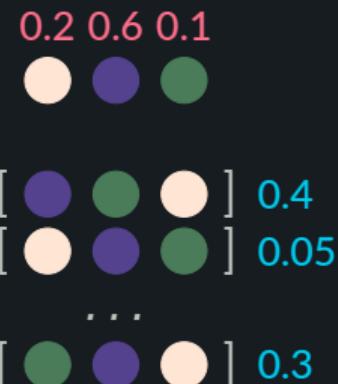
If  $\mathcal{Z}$  is **large**, this sum can get very expensive due to  $\ell(x, z; \theta)$ ! 🍻

# Training Discrete or Structured Latent Variable Models

Latent variable  $z$  can be **discrete** or **structured**

$\pi(z|x, \theta)$ : distribution over possible  $z$

$\ell(x, z; \theta)$ : downstream loss: ELBO, Log-Likelihood, (...)



To train, we need to compute the following expectation:

$$\mathcal{L}_x(\theta) = \sum_{z \in \mathcal{Z}} \pi(z|x, \theta) \ell(x, z; \theta)$$

If  $\mathcal{Z}$  is **combinatorial**, this can be intractable to compute!



# Current Solutions

If  $\mathcal{Z}$  is large, exact gradient computation is prohibitive

# Current Solutions

If  $\mathcal{Z}$  is large, exact gradient computation is prohibitive

One option: SFE (aka REINFORCE)—unbiased but high variance

# Current Solutions

If  $\mathcal{Z}$  is large, exact gradient computation is prohibitive

One option: SFE (aka REINFORCE)—unbiased but high variance

Another option: Gumbel-Softmax—continuous relaxation, biased estimation

# Current Solutions

If  $\mathcal{Z}$  is large, exact gradient computation is prohibitive

One option: SFE (aka REINFORCE)—unbiased but high variance

Another option: Gumbel-Softmax—continuous relaxation, biased estimation

New option: use sparsity! 

# Current Solutions

If  $\mathcal{Z}$  is large, exact gradient computation is prohibitive

One option: SFE (aka REINFORCE)—unbiased but high variance

Another option: Gumbel-Softmax—continuous relaxation, biased estimation

New option: use sparsity! 

no need for sampling -> no variance

# Current Solutions

If  $\mathcal{Z}$  is large, exact gradient computation is prohibitive

One option: SFE (aka REINFORCE)—unbiased but high variance

Another option: Gumbel-Softmax—continuous relaxation, biased estimation

New option: use sparsity! 

no need for sampling -> no variance

no relaxation into the continuous space

# Taking a step back...

Does the expectation over possible  $z$  need to be expensive?

# Taking a step back...

Does the expectation over possible  $z$  need to be expensive?

$$\begin{aligned}\mathcal{L}_x(\theta) &= \sum_{z \in \mathcal{Z}} \pi(z|x, \theta) \ell(x, z; \theta) \\ &= \pi(z_1|x, \theta) \ell(x, z_1; \theta) + \pi(z_2|x, \theta) \ell(x, z_2; \theta) + \dots \\ &\quad + \pi(z_i|x, \theta) \ell(x, z_i; \theta) + \dots + \pi(z_N|x, \theta) \ell(x, z_N; \theta)\end{aligned}$$

# Taking a step back...

Does the expectation over possible  $z$  need to be expensive?

$$\begin{aligned}\mathcal{L}_x(\theta) &= \sum_{z \in \mathcal{Z}} \pi(z|x, \theta) \ell(x, z; \theta) \\ &= \pi(z_1|x, \theta) \ell(x, z_1; \theta) + \pi(z_2|x, \theta) \ell(x, z_2; \theta) + \dots \\ &\quad + \pi(z_i|x, \theta) \ell(x, z_i; \theta) + \dots + \pi(z_N|x, \theta) \ell(x, z_N; \theta)\end{aligned}$$

Usually we normalize  $\pi$  with softmax  $\propto \exp(\pi) \Rightarrow \pi(z_i|x, \theta) > 0$

# Sparse normalizers

We use `sparsemax`, `top-k sparsemax` and `SparseMAP` to allow efficient marginalization

# Sparse normalizers

We use `sparsemax`, `top-k sparsemax` and `SparseMAP` to allow efficient marginalization

These functions are able to assign **probabilities of exactly zero!**

# Sparse normalizers

We use **sparsemax**, **top-k sparsemax** and **SparseMAP** to allow efficient marginalization

These functions are able to assign **probabilities of exactly zero!**

$$\begin{aligned}\mathcal{L}_x(\theta) &= \sum_{z \in \mathcal{Z}} \pi(z|x, \theta) \ell(x, z; \theta) \\ &= \pi(z_1|x, \theta) \ell(x, z_1; \theta) + \pi(z_2|x, \theta) \ell(x, z_2; \theta) + \dots \\ &\quad + \pi(z_i|x, \theta) \ell(x, z_i; \theta) + \dots + \pi(z_N|x, \theta) \ell(x, z_N; \theta)\end{aligned}$$

# Sparse normalizers

We use **sparsemax**, **top-k sparsemax** and **SparseMAP** to allow efficient marginalization

These functions are able to assign **probabilities of exactly zero!**

$$\begin{aligned}\mathcal{L}_x(\theta) &= \sum_{z \in \mathcal{Z}} \pi(z|x, \theta) \ell(x, z; \theta) \\ &= \pi(z_1|x, \theta) \ell(x, z_1; \theta) + \cancel{\pi(z_2|x, \theta)} \cancel{\ell(x, z_2; \theta)} + \dots \\ &\quad + \pi(z_i|x, \theta) \ell(x, z_i; \theta) + \dots + \cancel{\pi(z_N|x, \theta)} \cancel{\ell(x, z_N; \theta)}\end{aligned}$$

No need for computing  $\ell(x, z; \theta)$  for all  $z \in \mathcal{Z}!$

# Results

We test our methods for models with discrete latent variables,

# Results

We test our methods for models with discrete latent variables,

- Semi-Supervised VAE

# Results

We test our methods for models with discrete latent variables,

- Semi-Supervised VAE
- Emergent communication

# Results

We test our methods for models with discrete latent variables,

- Semi-Supervised VAE
- Emergent communication

but also in models with an exponentially large set of  $\mathcal{Z}$ ,

# Results

We test our methods for models with discrete latent variables,

- Semi-Supervised VAE
- Emergent communication

but also in models with an exponentially large set of  $\mathcal{Z}$ ,

- Bit-vector VAE

# Results

We test our methods for models with discrete latent variables,

- Semi-Supervised VAE
- Emergent communication

but also in models with an exponentially large set of  $\mathcal{Z}$ ,

- Bit-vector VAE

Our methods are top-performers and efficient!

# **Efficient Marginalization of Discrete and Structured Latent Variables via Sparsity**

**Gonçalo Correia** Instituto de Telecomunicações, Lisbon

Vlad Niculae Ivl, University of Amsterdam

Wilker Aziz ILLC, University of Amsterdam

André Martins Instituto de Telecomunicações & LUMLIS & Unbabel

# References |

-  Bouges, Pierre, Thierry Chateau, Christophe Blanc, and Gaëlle Loosli (Dec. 2013). "Handling missing weak classifiers in boosted cascade: application to multiview and occluded face detection". In: *EURASIP Journal on Image and Video Processing* 2013, p. 55. DOI: [10.1186/1687-5281-2013-55](https://doi.org/10.1186/1687-5281-2013-55).
-  Liu, Ziwei, Ping Luo, Xiaogang Wang, and Xiaoou Tang (Dec. 2015). "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)*.
-  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*.
-  Niculae, Vlad, André FT Martins, Mathieu Blondel, and Claire Cardie (2018). "SparseMAP: Differentiable sparse structured inference". In: *Proc. of ICML*.