Implicit MLE: Backpropagating Through Discrete Exponential Family Distributions

Authors: Mathias Niepert, Pasquale Minervini, Luca Franceschi
(presented by J. Le Roux)

“”

November 11, 2022

Outline

Introduction

many things in this paper!

We will see:

a method to learn via SGD a model which utilizes a discrete distribution internally based on:

- perturb and MAP
- approximate differentiation

We won't cover:

~~a novel class of noise distribution~~

Definition of the problem

Parameterized Mapping from $\mathcal{X}$ to $\mathcal{Y}$ via latent $\mathcal{Z}$

- from input $\boldsymbol{x} \in \mathcal{X}$ extract features $\boldsymbol{\theta} = h_{\boldsymbol{v}}(\boldsymbol{x}) \in \Theta$
- sample an internal (unobserved) discrete structure $\mathcal{Z} \ni \boldsymbol{z} \sim p(\cdot; \boldsymbol{\theta})$
- compute output structure $f_{\boldsymbol{u}}(\boldsymbol{z}) = \boldsymbol{y} \in \mathcal{Y}$

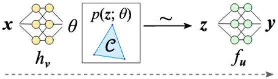

Figure 1: Illustration of the addressed learning problem. $\boldsymbol{z}$ is the discrete (latent) structure.

Mapping Parameters $(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{\omega}$ set from data $\mathcal{D} = \{(\hat{\boldsymbol{x}}_j, \hat{\boldsymbol{y}}_j)\}_{j=1}^{N}$

$$\min_{\boldsymbol{\omega}} \frac{1}{N} \sum_j L(\hat{\boldsymbol{x}}_j, \hat{\boldsymbol{y}}_j, \boldsymbol{\omega})$$

where:

- $L(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \boldsymbol{\omega}) = \mathbb{E}_{\hat{\boldsymbol{z}} \sim p(\cdot; \hat{\boldsymbol{\theta}})} \left[\ell\big(f_{\boldsymbol{u}}(\hat{\boldsymbol{z}}), \boldsymbol{y}\big) \right]$
- $\hat{\boldsymbol{\theta}} = h_{\boldsymbol{v}}(\hat{\boldsymbol{x}})$

Definition of p (1)

$\boldsymbol{z}$ in state space $\mathcal{Z}$ verifying linear constraints $\mathcal{C}$.

$$p(\boldsymbol{z};\theta) = \begin{cases} \frac{\exp \beta(\boldsymbol{z}\cdot\boldsymbol{\theta})}{\sum_{\boldsymbol{z'}\in\mathcal{C}} \exp \beta(\boldsymbol{z'}\cdot\boldsymbol{\theta})} = \exp(\beta(\boldsymbol{z}\cdot\boldsymbol{\theta}) - A(\boldsymbol{\theta})) & \text{if } \boldsymbol{z}\in\mathcal{C}, \\ 0 & \text{otherwise.} \end{cases}$$

where $A(\boldsymbol{\theta}) = \log \sum_{\boldsymbol{z'}\in\mathcal{C}} \exp \beta(\boldsymbol{z'}\cdot\boldsymbol{\theta})$ is the log-partition function

Notations

- marginals $\mu(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{z}\sim p(\cdot;\theta)}[\boldsymbol{z}] = \sum_{\boldsymbol{z}} p(\boldsymbol{z};\boldsymbol{\theta}) \times \boldsymbol{z}$ ($\approx$ average structure)
- $\texttt{MAP}(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{z}\in\mathcal{C}} \boldsymbol{z}\cdot\boldsymbol{\theta}$

Useful tricks:

- sample via perturb and MAP : $\boldsymbol{z}\sim p(\cdot;\boldsymbol{\theta}) \approx \boldsymbol{z} = \texttt{MAP}(\boldsymbol{\theta}+\varepsilon)$ with ε Gumbel noise (or other distribution)
- approximate expectations via sampling:

$$\mathbb{E}_{\boldsymbol{z}\sim p(\cdot;\boldsymbol{\theta})}[f(\boldsymbol{z})] \approx \frac{1}{S}\sum_{i=1}^{S} f(\boldsymbol{z_i}) = \frac{1}{S}\sum_{i=1}^{S} f(\texttt{MAP}(\boldsymbol{\theta}+\varepsilon_i))$$

Definition of p (2)

Fun fact: gradient of log-partion is the marginal vector!!

$$
\begin{aligned}
\nabla_{\boldsymbol{\theta}} A(\boldsymbol{\theta}) &= \nabla_{\boldsymbol{\theta}} \log \sum_{\boldsymbol{z}} \exp(\boldsymbol{z} \cdot \boldsymbol{\theta}) = \frac{\sum_{\boldsymbol{z}} \nabla_{\boldsymbol{\theta}} \exp(\boldsymbol{z} \cdot \boldsymbol{\theta})}{\sum_{\boldsymbol{z'}} \exp(\boldsymbol{z'} \cdot \boldsymbol{\theta})} \\
&= \sum_{\boldsymbol{z}} \frac{\exp(\boldsymbol{z} \cdot \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \boldsymbol{z} \cdot \boldsymbol{\theta}}{\sum_{\boldsymbol{z'}} \exp(\boldsymbol{z'} \cdot \boldsymbol{\theta})} \\
&= \sum_{\boldsymbol{z}} p(\boldsymbol{z}; \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \boldsymbol{z} \cdot \boldsymbol{\theta} = \sum_{\boldsymbol{z}} p(\boldsymbol{z}; \boldsymbol{\theta}) \boldsymbol{z} \\
&= \mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z}; \boldsymbol{\theta})}[\boldsymbol{z}] = \mu(\boldsymbol{\theta})
\end{aligned}
$$

Example

Learning to explain in opinion analysis

- from a text $\boldsymbol{x}$ (describing products) learn to predict a review score $\boldsymbol{y}$
- while providing a *proof* $\boldsymbol{z}$: the best k words which *explain* the assigned score
- Examples are $(\boldsymbol{x}, \boldsymbol{y})$, i.e $\boldsymbol{z}$ is not provided!

This means (high level):

1. retrieve a vector v for each word w (via lookup table, features...) in $\boldsymbol{x}$;
2. select k words $w_1 \dots w_k$ from x from distribution p over k-tuples
3. predict a score, for instance $f_{\boldsymbol{u}} = \sum_{p=1}^{k} u_p^\top w_p$

variants

if input is a single sentence: *proof* $\boldsymbol{z}$ is a syntactic or semantic parse of the input

Learning via Stochastic Gradient Descent

cheapest way to parameterize your system (and sometimes the only one)

$$\boldsymbol{\omega}^{k+1} = \boldsymbol{\omega}^k - \nabla_{\boldsymbol{\omega}} L(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}; \boldsymbol{\omega})$$

How to compute $\nabla_{\boldsymbol{\omega}} L(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}; \boldsymbol{\omega})$?

Remember $\boldsymbol{\omega} = (\boldsymbol{u}, \boldsymbol{v})$, so $\nabla_{\boldsymbol{\omega}} = (\nabla_{\boldsymbol{u}} \ \nabla_{\boldsymbol{v}})$ (as a column vector)

- Compute this gradient in two steps, one for u, one for v since they play a different role
- v is *part of* the expectation
- u is *inside* the expectation

How to compute $\nabla_{\boldsymbol{u}} L(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}; \boldsymbol{\omega})$?

For one example $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$:

$$
\begin{aligned}
\nabla_{\boldsymbol{u}} L(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}; \boldsymbol{\omega}) &= \nabla_{\boldsymbol{u}} \mathbb{E}_{\boldsymbol{z} \sim p(\cdot; \boldsymbol{\theta})} [\ell(f_{\boldsymbol{u}}(\boldsymbol{z}), \hat{\boldsymbol{y}})] && \text{(def. } L\text{)} \\
&= \nabla_{\boldsymbol{u}} \sum_{\boldsymbol{z}} p(\boldsymbol{z}; \boldsymbol{\theta}) \ell(f_{\boldsymbol{u}}(\hat{\boldsymbol{z}}), \hat{\boldsymbol{y}}) && \text{(def. } \mathbb{E}\text{)} \\
&= \sum_{\boldsymbol{z}} p(\boldsymbol{z}; \boldsymbol{\theta}) \nabla_{\boldsymbol{u}} \ell(f_{\boldsymbol{u}}(\boldsymbol{z}), \hat{\boldsymbol{y}}) && \text{(sum} \leftrightarrow \text{gradient)} \\
&= \mathbb{E}_{\boldsymbol{z} \sim p(\cdot; \boldsymbol{\theta})} [\nabla_{\boldsymbol{u}} \ell(f_{\boldsymbol{u}}(\boldsymbol{z}), \hat{\boldsymbol{y}})]
\end{aligned}
$$

And:

$$\nabla_{\boldsymbol{u}} \ell(f_{\boldsymbol{u}}(\boldsymbol{z}), \hat{\boldsymbol{y}}) = (\partial_{\boldsymbol{u}} f_{\boldsymbol{u}}(\boldsymbol{z}))^\top (\nabla_{\boldsymbol{y}} \ell(\boldsymbol{y}, \hat{\boldsymbol{y}})) \text{ where } \boldsymbol{y} = f_{\boldsymbol{u}}(\boldsymbol{z}) \text{ as variables}$$

- easy to compute (manually or via autodiff)
- $\boldsymbol{u}$ is *inside* the expectation $\rightarrow$ approximate expectation with a few samples

How to compute $\nabla_{\boldsymbol{v}} L(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}; \boldsymbol{\omega})$?

- Remember that $\boldsymbol{\theta} = h_{\boldsymbol{v}}(\hat{\boldsymbol{x}})$

For one example $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$

$$
\begin{aligned}
\nabla_{\boldsymbol{v}} L(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}; \boldsymbol{\omega}) &= \nabla_{\boldsymbol{v}} \mathbb{E}_{\boldsymbol{z} \sim p(\cdot;\boldsymbol{\theta})}[\ell(f_{\boldsymbol{u}}(\boldsymbol{z}), \hat{\boldsymbol{y}})] && \text{(def. } L) \\
&= \nabla_{\boldsymbol{v}} \sum_{\boldsymbol{z}} p(\boldsymbol{z}; \boldsymbol{\theta}) \ell(f_{\boldsymbol{u}}(\boldsymbol{z}), \hat{\boldsymbol{y}}) && \text{(def. } \mathbb{E}) \\
&= \nabla_{\boldsymbol{v}} \sum_{\boldsymbol{z}} p(\boldsymbol{z}; h_{\boldsymbol{v}}(\hat{\boldsymbol{x}})) \ell(f_{\boldsymbol{u}}(\boldsymbol{z}), \hat{\boldsymbol{y}}) && \text{(def. } \boldsymbol{\theta}) \\
&= (\partial_{\boldsymbol{v}} h_{\boldsymbol{v}}(\hat{\boldsymbol{x}}))^\top \nabla_{\boldsymbol{\theta}} \sum_{\boldsymbol{z}} p(\boldsymbol{z}; \boldsymbol{\theta}) \ell(f_{\boldsymbol{u}}(\boldsymbol{z}), \hat{\boldsymbol{y}}) && \text{(composition)} \\
&= (\partial_{\boldsymbol{v}} h_{\boldsymbol{v}}(\hat{\boldsymbol{x}}))^\top \sum_{\boldsymbol{z}} \nabla_{\boldsymbol{\theta}} p(\boldsymbol{z}; \boldsymbol{\theta}) \ell(f_{\boldsymbol{u}}(\boldsymbol{z}), \hat{\boldsymbol{y}}) && \text{(not an expectation)}
\end{aligned}
$$

- difficult to compute (manually or via autodiff) $\rightarrow$ need to enumerate through all valid $\boldsymbol{z}$ (or use *score function estimator*)
- $\boldsymbol{\theta}$ *defines* the expectation

Target Distribution and (Implicit) MLE (1)

target distribution q with the same form as p:

$$\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z};\boldsymbol{\theta}')}[\ell(f_{\boldsymbol{u}}(\boldsymbol{z}), \hat{\boldsymbol{y}})] \leq \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z};\boldsymbol{\theta})}[\ell(f_{\boldsymbol{u}}(\boldsymbol{z}), \hat{\boldsymbol{y}})]$$

- Idea: if we *push* p closer to q, loss is lower
- This the idea behind minimizing cross-entropy, behind minimizing:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}') = -\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z};\boldsymbol{\theta}')}[\log p(\boldsymbol{z};\boldsymbol{\theta}')] = \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z};\boldsymbol{\theta}')}[A(\boldsymbol{\theta}) - \boldsymbol{z}\cdot\boldsymbol{\theta}]$$

- New idea: replace $\nabla_{\boldsymbol{\theta}} L$ by (an approximation of) $\nabla_{\boldsymbol{\theta}}\mathcal{L}$

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}') &= \nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z};\boldsymbol{\theta}')}[A(\boldsymbol{\theta}) - \boldsymbol{z}\cdot\boldsymbol{\theta}] \\
&= \nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z};\boldsymbol{\theta}')}[A(\boldsymbol{\theta})] - \nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z};\boldsymbol{\theta}')}[\boldsymbol{z}\cdot\boldsymbol{\theta}] \\
&= \nabla_{\boldsymbol{\theta}}A(\boldsymbol{\theta}) - \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z};\boldsymbol{\theta}')}[\nabla_{\boldsymbol{\theta}}\boldsymbol{z}\cdot\boldsymbol{\theta}] \\
&= \mu(\boldsymbol{\theta}) - \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z};\boldsymbol{\theta}')}[\boldsymbol{z}] \\
&= \mu(\boldsymbol{\theta}) - \mu(\boldsymbol{\theta}')
\end{aligned}$$

Target Distribution and (Implicit) MLE (2)

Now approximate log-partitions via *perturb-and-MAP*

$$\hat{\nabla}_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}') = \frac{1}{S}(\texttt{MAP}(\boldsymbol{\theta} + \varepsilon_i) - \texttt{MAP}(\boldsymbol{\theta}' + \varepsilon_i))$$

- with ε_i a noise sample for $i = 1, \ldots, S$
- use Gumbel distribution or the one we won't cover:sum of gamma

Question: what is $\boldsymbol{\theta}'$???

What is a good Target Distribution?

go back to the paper and enjoy 3.1 ;)

What is the Target Distribution (1)?

Let us modify L to take only the $f_{\boldsymbol{u}}$ of the average:

- old $L(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \boldsymbol{\omega}) = \mathbb{E}_{\hat{\mathbf{z}} \sim p(\cdot;\hat{\boldsymbol{\theta}})} \Big[\ell\big(f_{\boldsymbol{u}}(\hat{\mathbf{z}}), \mathbf{y}\big) \Big]$
- new $L(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \boldsymbol{\omega}) = \ell\big(f_{\boldsymbol{u}}(\mu(\boldsymbol{\theta})), \mathbf{y}\big)$

Domke(2010) showed that in this case:

$$\nabla_{\boldsymbol{\theta}} L(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}; \boldsymbol{\omega}) = \lim_{\lambda \to 0} \left\{ \frac{1}{\lambda} \left[\boldsymbol{\mu}(\boldsymbol{\theta}) - \boldsymbol{\mu}\left(\boldsymbol{\theta} - \lambda \nabla_{\boldsymbol{\mu}} L(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}; \boldsymbol{\omega})\right) \right] \right\},$$

with:

$$\nabla_{\boldsymbol{\mu}} L = \partial_{\boldsymbol{\mu}} f_{\boldsymbol{u}}(\boldsymbol{\mu})^{\intercal} \nabla_{\boldsymbol{y}} \ell(\boldsymbol{y}, \hat{\boldsymbol{y}}).$$

which is simplified further here (straight through gradient estimator):

$$, \nabla_{\boldsymbol{\mu}} \hat{L} = \partial_{\boldsymbol{\mu}} \boldsymbol{z}^{\intercal} \nabla_{\boldsymbol{z}} L \approx \nabla_{\boldsymbol{z}} \hat{L}$$

(assuming z is a function of μ)

What is the Target Distribution (2)?

Adapating previous gradient we have:

$$\nabla_{\boldsymbol{\theta}} L(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}; \boldsymbol{\omega}) \approx \frac{1}{\lambda}\left[\boldsymbol{\mu}(\boldsymbol{\theta}) - \boldsymbol{\mu}\left(\boldsymbol{\theta} - \lambda \nabla_{\boldsymbol{z}} L(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}; \boldsymbol{\omega})\right)\right] = \frac{1}{\lambda} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta} - \lambda \nabla_{\boldsymbol{z}} L(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}; \boldsymbol{\omega})),$$

which finally gives:

$$q(\boldsymbol{z}; \boldsymbol{\theta}') = p(\boldsymbol{z}; \boldsymbol{\theta} - \lambda \nabla_{\boldsymbol{z}} \ell(f_{\boldsymbol{u}}(\overline{\boldsymbol{z}}), \hat{\boldsymbol{y}})) \text{ with } \overline{\boldsymbol{z}} = \texttt{MAP}(\boldsymbol{\theta} + \boldsymbol{\epsilon}) \text{ and } \boldsymbol{\epsilon} \sim \rho(\boldsymbol{\epsilon}),$$