Differentiation and Weighted Model Integration

Pedro Zuidberg Dos Martires

Weighted Model Integration:

- Probabilistic inference in the discrete-continuous domain.
- Generalization of weighted model counting.
- Performing counting and integration in this hybrid domain.

Weighted Model Integration:

- Probabilistic inference in the discrete-continuous domain.
- Generalization of weighted model counting.
- Performing counting and integration in this hybrid domain.

Why do we need differentiation?

- Probabilistic inference is hard, many approximation schemes rely on differentiation, e.g. variational inference, Hamilton Monte Carlo
- Taking derivatives allows for gradient based optimization!

Weighted Model Integration:

- Probabilistic inference in the discrete-continuous domain.
- Generalization of weighted model counting.
- Performing counting and integration in this hybrid domain.

Why do we need differentiation?

- Probabilistic inference is hard, many approximation schemes rely on differentiation, e.g. variational inference, Hamilton Monte Carlo
- Taking derivatives allows for gradient based optimization!

This talk:

- Show how differentiation can be done in the weighted model integration context.
- Show difficulties that lie ahead!

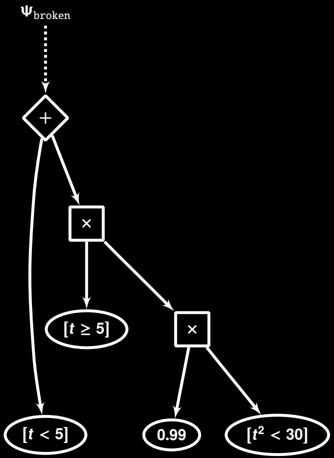
SMT: Satisfiability Modulo Theory

$$\texttt{working} \leftrightarrow (\texttt{cooling} \wedge (\texttt{t}^2 < 30)) \vee (\texttt{t} < 5)$$

WMI: Probability of SMT Formulas

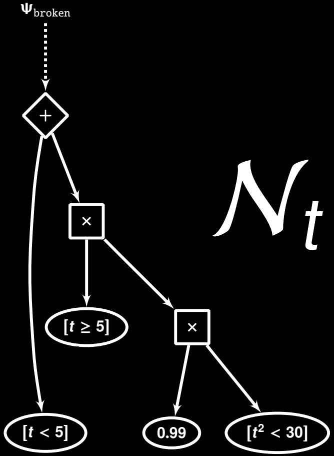
$$\texttt{working} \leftrightarrow (\texttt{cooling} \wedge (\texttt{t}^2 < 30)) \vee (\texttt{t} < 5)$$

$$p(\texttt{cooling}) = 0.99$$
$$\texttt{t} \sim \mathcal{N}_{\texttt{t}}(20, 5)$$



Compile logic structure into arithmetic circuit.

Arithmetic circuits are related to sum-product-networks.

$\int dt$  $\mathcal{N}_t(20, 5)$

$$\mathrm{WMI}(\phi, w|\mathbf{x}, \mathbf{b}) = \int \sum_{\mathbf{b}_{\mathcal{I}} \in \mathcal{I}_{\mathbf{b},\mathbf{b}_a}(\phi_a)} \prod_{b_i \in \mathbf{b}_{\mathcal{I}}} \alpha_{b_i}(\mathbf{x}) w_x(\mathbf{x}) d\mathbf{x} \tag{1}$$

$$= \int \Psi(\mathbf{x}) w_x(\mathbf{x}) d\mathbf{x} \tag{2}$$

$$= \mathbb{E}_{w_x(\mathbf{x})}[\Psi(\mathbf{x})] \tag{3}$$

Cross-Entropy H

Tells you how different two distributions p and q are.

$$H(p, q) = \mathbb{E}_p[-\log q] \tag{4}$$

p is the *true* distribution (observation of the world).
q is our model of the true distribution.
q depends on the parameters $\boldsymbol{\theta}$.

Minimize $H(p, q)$ by learning the parameters $\boldsymbol{\theta}$.

Gradient Descent

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta \nabla_{\boldsymbol{\theta}} \mathbb{E}_p[-\log q] \tag{5}$$

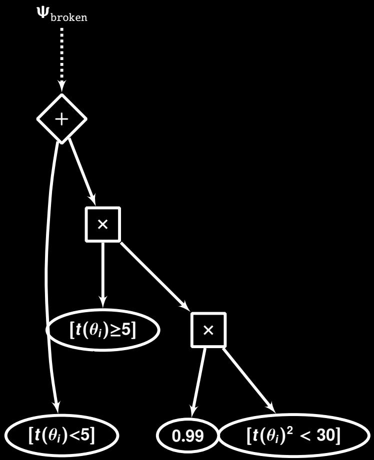
Gradient of the Cross-Entropy

$$\nabla_\theta \mathbb{E}_p[-\log q] \tag{6}$$

$$= \mathbb{E}_p[-\frac{1}{q(\boldsymbol{\theta})} \nabla_\theta \sum_{\mathbf{b}_\mathcal{I} \in \mathcal{I}_{\mathbf{b},\mathbf{b}_a}(\phi_a)} \prod_{b_i \in \mathbf{b}_\mathcal{I}} \alpha_{b_i}(\boldsymbol{\theta})] \tag{7}$$

$$= \mathbb{E}_p[-\frac{1}{\Psi(\boldsymbol{\theta})} \nabla_\theta \Psi(\theta)] \tag{8}$$

$$\frac{\partial}{\partial\theta_i}$$



Applying the Product Rule

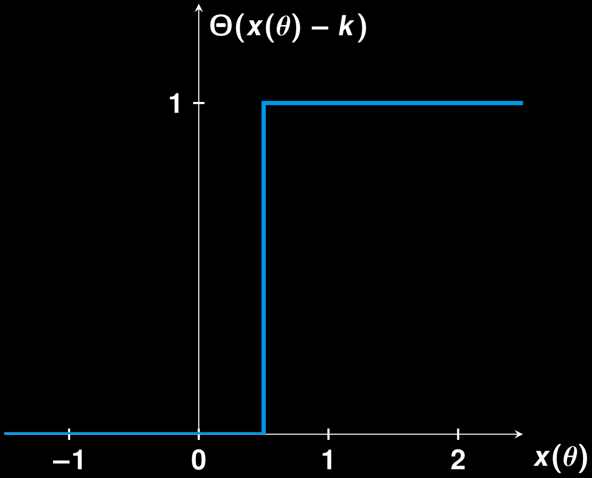
$$
\begin{aligned}
&\nabla_{\boldsymbol{\theta}} \mathbb{E}_p[-\log q] && (9)\\
&= \mathbb{E}_p[-\frac{1}{q(\boldsymbol{\theta})} \nabla_{\boldsymbol{\theta}} \sum_{\mathbf{b}_{\mathcal{I}} \in \mathcal{I}_{\mathbf{b},\mathbf{b}_a}(\phi_a)} \prod_{b_i \in \mathbf{b}_{\mathcal{I}}} \alpha_{b_i}(\boldsymbol{\theta})] && (10)\\
&= \mathbb{E}_p[-\frac{1}{\Psi(\boldsymbol{\theta})} \nabla_{\boldsymbol{\theta}} \Psi(\theta)] && (11)\\
&= \mathbb{E}_p[-\frac{1}{\Psi(\boldsymbol{\theta})} \sum_{\mathbf{b}_{\mathcal{I}} \in \mathcal{I}_{\mathbf{b},\mathbf{b}_a}(\phi_a)} \sum_{b_i \in \mathbf{b}_{\mathcal{I}}} \nabla_{\boldsymbol{\theta}}(\alpha_{b_i}(\boldsymbol{\theta})) \prod_{b_j \in \mathbf{b}_{\mathcal{I}} \setminus \{b_i\}} \alpha_{b_j}(\boldsymbol{\theta})] && (12)
\end{aligned}
$$

A Simple One Dimensional Case

$$\nabla_{\boldsymbol{\theta}} \alpha(\boldsymbol{\theta}) = \frac{\partial \alpha(\theta)}{\partial \theta} \tag{13}$$

$$= \frac{\partial \Theta(x(\theta) - k)}{\partial \theta} \tag{14}$$

Heaviside Step Function



A Simple One Dimensional Case

$$\nabla_{\boldsymbol{\theta}} \alpha(\boldsymbol{\theta}) = \frac{\partial \alpha(\theta)}{\partial \theta} \tag{15}$$

$$= \frac{\partial \Theta(x(\theta) - k)}{\partial \theta} \tag{16}$$

$$= \delta(x(\theta) - k) \frac{\partial x(\theta)}{\partial \theta} \tag{17}$$

In Higher Dimensions

- Gradient is generalization of inward normal derivative.
- Leads to surface integral (boundary of indicator function)

Open questions:

- What is the computational hardness of the surface integral?
- Is it equivalent to the optimization of the 0-1 loss (NP-complete)?
- Can we use convex relaxation for a practical algorithm?
- Is it beneficial to restrict ourselves to a subclass of constraints? (very probably yes)

Where might this be useful?

- Parameter learning in hybrid probabilistic programs.
- Probabilistic inference through (stochastic) variational inference.
- Probabilistic inference through Hamilton Monte Carlo.