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.

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!
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

\[ \text{working} \leftrightarrow (\text{cooling} \land (t^2 < 30)) \lor (t < 5) \]
working $\leftrightarrow (\text{cooling} \land (t^2 < 30)) \lor (t < 5)$

$p(\text{cooling}) = 0.99$

$t \sim N_t(20, 5)$
Compile logic structure into arithmetic circuit.
Arithmetic circuits are related to sum-product-networks.
\[ \int dt \quad N_t(20, 5) \]
\[
WMI(\phi, w|\mathbf{x}, \mathbf{b}) = \int \sum_{b_I \in \mathcal{I}_{b_i b_a}(\phi_a)} \prod_{b_i \in b_I} \alpha_{b_i}(\mathbf{x}) w_{\mathbf{x}}(\mathbf{x}) \, d\mathbf{x} \quad (1)
\]

\[
= \int \Psi(\mathbf{x}) w_{\mathbf{x}}(\mathbf{x}) \, d\mathbf{x} \quad (2)
\]

\[
= \mathbb{E}_{w_{\mathbf{x}}(\mathbf{x})}[\Psi(\mathbf{x})] \quad (3)
\]
Cross-Entropy $H$

Tells you how different two distributions $p$ and $q$ are.

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

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

Minimize $H(p, q)$ by learning the parameters $\theta$. 
Gradient Descent

\[
\theta \leftarrow \theta + \eta \nabla_{\theta} \mathbb{E}_p[- \log q] \tag{5}
\]
Gradient of the Cross-Entropy

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

\[ = \mathbb{E}_p[- \frac{1}{q(\theta)} \nabla_{\theta} \sum_{b_I \in \mathcal{I} \cap \mathcal{b}_{a}(\phi_{a})} \prod_{b_i \in \mathcal{b}_I} \alpha_{b_i}(\theta)] \]

\[ = \mathbb{E}_p[- \frac{1}{\psi(\theta)} \nabla_{\theta} \psi(\theta)] \]
\[
\frac{\partial}{\partial \theta_i} \left[ \begin{array}{c}
\Psi_{\text{broken}} \\
\end{array} \right]
\]

\[
\begin{array}{c}
\left[t(\theta_i) \geq 5\right] \\
\left[t(\theta_i) < 5\right] \\
0.99 \left[t(\theta_i)^2 < 30\right]
\end{array}
\]
Applying the Product Rule

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

\[ = \mathbb{E}_p[- \frac{1}{q(\theta)} \nabla_\theta \sum_{b_I \in \mathcal{I}_{b, b_a}(\phi_a) \ b_i \in b_I} \prod_{\mathcal{I}} \alpha_{b_i}(\theta)] \] (10)

\[ = \mathbb{E}_p[- \frac{1}{\psi(\theta)} \nabla_\theta \psi(\theta)] \] (11)

\[ = \mathbb{E}_p[- \frac{1}{\psi(\theta)} \sum_{b_I \in \mathcal{I}_{b, b_a}(\phi_a) \ b_i \in b_I} \sum \nabla_\theta (\alpha_{b_i}(\theta)) \prod_{b_j \in b_I \setminus \{b_i\}} \alpha_{b_j}(\theta)] \] (12)
A Simple One Dimensional Case

\[ \nabla_{\theta} \alpha(\theta) = \frac{\partial \alpha(\theta)}{\partial \theta} \]

(13)

\[ = \frac{\partial \Theta(x(\theta) - k)}{\partial \theta} \]

(14)
Heaviside Step Function

\[ \Theta(x(\theta) - k) \]
A Simple One Dimensional Case

\[
\nabla_\theta \alpha(\theta) = \frac{\partial \alpha(\theta)}{\partial \theta} \\
= \frac{\partial \Theta(\chi(\theta) - k)}{\partial \theta} \\
= \delta(\chi(\theta) - k) \frac{\partial \chi(\theta)}{\partial \theta}
\]
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.