Probabilistic Inference and Learning \textit{under Constraints}

Zhe Zeng

\textit{University of California, Los Angeles}
Where are the constraints from?

Properties
- Architecture
- Regularization
- Explainability

Domain Knowledge
- Physical Laws
- Molecular Structure
- Gene Expression
Where to integrate *constraints*?

Input  Latent Space  Output  Loss
Challenges in *constraint integration*

Non-differentiability
Discrete nature

Intractability
#P-hard
How to integrate *diverse constraints*?

- **Non-differentiability**
  - Discrete nature

- **Intractability**
  - \#P-hard
How to integrate *diverse constraints*?

Outline

• Differentiable learning under constraints

• Constrained probabilistic inference
How to integrate *diverse constraints*?

Outline

- Differentiable learning under constraints
- Constrained probabilistic inference
Why \( k\)-subset constraint?

Discrete VAE

Data \rightarrow \text{Encoder} \rightarrow \text{Decoder} \rightarrow \text{Reconstructed Data}

Exactly \( k \) one's

Discrete Latent Space
Differentiable learning under \textit{k-subset} constraints

Input
Learn to Explain

Latent Space
DiscreteVAE

Output
AI for Science

Loss
Weakly Supervised
Differentiable learning under \textit{k-subset} constraints

Input
Learn to Explain

Latent Space
DiscreteVAE

Output
AI for Science

Loss
Weakly Supervised
SIMPLE: Gradient Estimator for $k$-Subset Sampling [1]
SIMPLE: Gradient Estimator for k-Subset Sampling [1]
**SIMPLE**: Gradient Estimator for $k$-Subset Sampling [1]

Encoder

Discrete Latent Space

Thm. Exact sampling is easy!

$$z \sim p_\theta(z \mid \sum_i z_i = k)$$
SIMPLE: Gradient Estimator for k-Subset Sampling [1]
**SIMPLE**: Gradient Estimator for $k$-Subset Sampling [1]

![Diagram of Encoder, Discrete Latent Space, and Decoder]

- **Encoder** $h_v$:
  - $x \rightarrow h_v \rightarrow \theta$

- **Discrete Latent Space** $z$:
  - $z \sim p_\theta(z | \sum_i z_i = k)$
  - $\nabla_\theta L(x, y; \omega) = \partial_\theta z \nabla_z \ell(f_u(z, x), y)$

- **Decoder** $f_u$:
  - $z \rightarrow f_u \rightarrow \ell(f_u(z, x), y)$
SIMPLE: Gradient Estimator for $k$-Subset Sampling [1]
**SIMPLE: Gradient Estimator for k-Subset Sampling** [1]

**Intuition:** update $\theta$ such that

<table>
<thead>
<tr>
<th>$z_1$</th>
<th>$z_2$</th>
<th>$z_3$</th>
<th>$p(z_1 \mid \sum_i z_i = k)$</th>
<th>$p(z_2 \mid \sum_i z_i = k)$</th>
<th>$p(z_3 \mid \sum_i z_i = k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.7</td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.7</td>
<td>0.3</td>
<td>0.75</td>
</tr>
</tbody>
</table>
In machine learning, we're often interested in approximate sampling and marginals. Gradient estimators, such as discrete VAEs, can be obtained by Gumbel-Softmax, leading to continuous relaxations.

Prop. conditional marginals can be obtained by

\[
\frac{\partial}{\partial \theta_i} \log p(\sum_j z_j = k) = p(z_i | \sum_j z_j = k) = \mu(\theta)
\]
**SIMPLE: Gradient Estimator for k-Subset Sampling** [1]

We propose a gradient estimator for distributions over $k$-subset sampling. We use the Gumbel-Softmax to generalize $k$-subset sampling to the continuous domain. This estimator is comparable to I-MLE and SFE in terms of bias and variance, while it has a lower lower bias and variance.

We achieve lower bias and variance by exact, discrete samples and exact derivative of conditional marginals.
Ablation Study
Why constraint probability helps?

Perturb-and-map (PAM)
- PAM Sampling
- PAM Marginal

Exact computation by SIMPLE
- Exact Sampling  \( z \sim p_\theta(z \mid \sum_i z_i = k) \)
- Exact Marginal  \( \mu(\theta) = p_\theta(z_i \mid \sum_j z_j = k) \)

Baseline
Exact sampling helps reduce bias!
Exact marginal helps reduce variance!
Both are reduced!
Experiment

Metric: exact ELBO
Differentiable learning under $k$-subset constraints

**Input**
Learn to Explain

**Latent Space**
DiscreteVAE

**Output**
AI for Science

**Loss**
Weakly Supervised
**Learn to Explain (L2X)**[1]

<table>
<thead>
<tr>
<th>Input: Key words ((k = 10))</th>
<th>Output: Taste Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>a lite bodied beer with a pleasant taste. was like a reddish color. a little like wood and caramel with a hop finish. has a sort of fruity flavor like grapes or cherry that is sort of buried in there. mouth feel was lite, sort of bubbly. not hard to down, though a bit harder then one would expect given the taste.</td>
<td>0.7</td>
</tr>
</tbody>
</table>
Learn to Explain (L2X)

Results for three aspects with $k = 10$

<table>
<thead>
<tr>
<th>Method</th>
<th>Appearance</th>
<th>Palate</th>
<th>Taste</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Test MSE</td>
<td>Precision</td>
<td>Test MSE</td>
</tr>
<tr>
<td>SIMPLE (Ours)</td>
<td>2.35 ± 0.28</td>
<td>66.81 ± 7.56</td>
<td>2.68 ± 0.06</td>
</tr>
<tr>
<td>L2X ($t = 0.1$)</td>
<td>10.70 ± 4.82</td>
<td>30.02 ± 15.82</td>
<td>6.70 ± 0.63</td>
</tr>
<tr>
<td>SoftSub ($t = 0.5$)</td>
<td>2.48 ± 0.10</td>
<td>52.86 ± 7.08</td>
<td>2.94 ± 0.08</td>
</tr>
<tr>
<td>I-MLE ($r = 30$)</td>
<td>2.51 ± 0.05</td>
<td>65.47 ± 4.95</td>
<td>2.96 ± 0.04</td>
</tr>
</tbody>
</table>

Results for aspect Aroma, for $k$ in {5, 10, 15}

<table>
<thead>
<tr>
<th>Method</th>
<th>$k = 5$</th>
<th>$k = 10$</th>
<th>$k = 15$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Test MSE</td>
<td>Precision</td>
<td>Test MSE</td>
</tr>
<tr>
<td>SIMPLE (Ours)</td>
<td>2.27 ± 0.05</td>
<td>57.30 ± 3.04</td>
<td>2.23 ± 0.03</td>
</tr>
<tr>
<td>L2X ($t = 0.1$)</td>
<td>5.75 ± 0.30</td>
<td>33.63 ± 6.91</td>
<td>6.68 ± 1.08</td>
</tr>
<tr>
<td>SoftSub ($t = 0.5$)</td>
<td>2.57 ± 0.12</td>
<td>54.06 ± 6.29</td>
<td>2.67 ± 0.14</td>
</tr>
<tr>
<td>I-MLE ($r = 30$)</td>
<td>2.62 ± 0.05</td>
<td>54.76 ± 2.50</td>
<td>2.71 ± 0.10</td>
</tr>
</tbody>
</table>

Input:
Key words ($k = 10$)

a lite bodied beer with a pleasant taste. was like a reddish color. a little like wood and caramel with a hop finish. has a sort of fruity flavor like grapes or cherry that is sort of buried in there. mouth feel was lite, sort of bubbly. not hard to down, though a bit harder then one would expect given the taste.

Output: Taste Score

0.7
Differentiable learning under \( k\)-subset constraints

Input
Learn to Explain

Latent Space
DiscreteVAE

Output
AI for Science

Loss
Weakly Supervised
A Unified Approach to Count-Based Weakly-Supervised Learning [2]

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>${x_i}_{i=1}^k$</th>
<th>$\bar{y} = \sum y_i/k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1/3</td>
</tr>
<tr>
<td>1</td>
<td>3/5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>${x_i}_{i=1}^k$</th>
<th>$\bar{y} = \max{y_i}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x$</th>
<th>$\bar{y}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Classical**

Learning from Label Proportions

Multiple Instance Learning

Learning from Positive & Unlabeled

**Objective:** To maximize the probability of weak supervisions, i.e., constraints on label counts
Differentiable learning under \textit{k-subset} constraints

\begin{itemize}
  \item Input: Learn to Explain
  \item Latent Space: DiscreteVAE
  \item Output: AI for Science
  \item Loss: Weakly Supervised
\end{itemize}
Partial Charge Assignment to Metal–Organic Frameworks

Application in Computational Chemistry

Neutral Charge Constraint
\[ \sum_i q_i = 0 \]
Differentiable learning under *k-subset* constraints

**Input**
Learn to Explain

**Latent Space**
DiscreteVAE

**Output**
AI for Science

**Loss**
Weakly Supervised

**Key:** *constraint probability!*
How to integrate *diverse constraints*?

Outline

- Differentiable learning under constraints
  - Key: constraint probability!
- Constrained probabilistic inference
How to integrate *diverse constraints*?

Outline

- Differentiable learning under constraints
  - Key: constraint probability!
- Constrained probabilistic inference
Collapsed inference for Bayesian deep learning

Constrained probabilistic inference
Motivation

Bad Uncertainty Estimation

Confidence by a ReLU neural network [6]

➡ Bayesian Deep Learning for robust and reliable predictions

Risky Point Estimation

Loss surface [7]
Bayesian Model Average (BMA)

Key idea

Point Estimate

Posterior

\[ p(y \mid x, w) \]

\[ p(y \mid x) = \int p(y \mid x, w)p(w) \, dw \]
Motivation

• **Goal:** Bayesian model average

  Predictive posterior \[ p(y \mid x) = \int p(y \mid x, w)p(w) \, dw \]

  Expected prediction \[ E[y] = \int y \, p(y \mid x) \, dy \]

• **Challenge:** DNNs are too big!
  
  ➪ *Costly* to maintain too many samples
  
  ➪ *Low sample efficiency* given the complex integrand

*How complex? 😮*
How complex is the integrand?

Expected prediction

$$\mathbb{E}[y] = \int y \, p(w \mid D) \, p(y \mid f(x), w) \, dw \, dy$$

- Weight posterior
  - 1d Uniform
- Predictive
  - 1d Gaussian
- NN model
  - $$f(x) = \text{ReLU}(wx)$$

Non-convex, multi-modal, no closed form 😳
Motivation

• **Goal:** Bayesian model average

Predictive posterior \[ p(y \mid x) = \int p(y \mid x, w)p(w) \, dw \]

Expected prediction \[ \mathbb{E}[y] = \int y \, p(y \mid x) \, dy \]

Is there a *better* way to estimate the integral than *sampling*?

Yes! 😃
Idea
A reduction from BMA to WMI

• Weighted Model Integration (WMI)$^4$
  • A class of weighted volume computation problems
  • Definition:
    • Region: $\exists$ SMT formula (a logical combination of arithmetic constraints)
    • Weight function $\phi : \exists \rightarrow \mathbb{R}$
  • Existing WMI solvers are able to give exact marginalization results
    • for (piecewise) polynomial weights
Idea

A reduction from BMA to WMI

BMA \quad \xrightarrow{\text{Sampling}} \quad \text{BMA integral}

\quad \text{Approximated by} \quad \text{WMI solvers}

\quad \xrightarrow{\text{WMI integral}} \quad \text{Approximate}

Ground truth BMA

BMA by sampling

BMA by WMI
Accurate approximation!

... but scalability?
Limitations

<table>
<thead>
<tr>
<th></th>
<th>Sampling</th>
<th>BMA via WMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Flexibility</td>
<td>✓</td>
<td>✗*</td>
</tr>
<tr>
<td>Scalability</td>
<td>✓</td>
<td>✗**</td>
</tr>
</tbody>
</table>

* Limited to fully connected layers
** Integration over polytopes in arbitrarily high dimensions is #P-hard

How to combine good from both worlds? 😐
# Limitations

<table>
<thead>
<tr>
<th></th>
<th>Sampling</th>
<th>BMA via WMI</th>
<th>Collapsed Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accuracy</strong></td>
<td>❌</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Flexibility</strong></td>
<td>✔️</td>
<td>❌*</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Scalability</strong></td>
<td>✔️</td>
<td>❌**</td>
<td>✔️</td>
</tr>
</tbody>
</table>

* Limited to fully connected layers
** Integration over polytopes in arbitrarily high dimensions is #P-hard

How to combine good from both worlds? 🤔

➡️ **Collapsed inference scheme!** 💪
Collapsed Inference\textsuperscript{[5]}

Expected prediction in BMA

\[ E[y] = \frac{1}{n} \sum w_s \text{ WMI}(\text{Collapsed set}) \]

**Accuracy + Flexibility, Scalability! 😃**
### Experiment: UCI Regression

<table>
<thead>
<tr>
<th></th>
<th>Boston</th>
<th>Concrete</th>
<th>Yacht</th>
<th>Naval</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CIBER (SECOND)</strong></td>
<td><strong>-2.471 ± 0.140</strong></td>
<td><strong>-2.975 ± 0.102</strong></td>
<td><strong>-0.678 ± 0.301</strong></td>
<td><strong>7.276 ± 0.532</strong></td>
<td><strong>-0.716 ± 0.211</strong></td>
</tr>
<tr>
<td><strong>CIBER (LAST)</strong></td>
<td><strong>-2.471 ± 0.140</strong></td>
<td><strong>-2.959 ± 0.109</strong></td>
<td><strong>-0.687 ± 0.301</strong></td>
<td><strong>7.482 ± 0.188</strong></td>
<td><strong>-0.716 ± 0.211</strong></td>
</tr>
<tr>
<td><strong>SWAG</strong></td>
<td><strong>-2.761 ± 0.132</strong></td>
<td><strong>-3.013 ± 0.086</strong></td>
<td><strong>-0.404 ± 0.418</strong></td>
<td><strong>6.708 ± 0.105</strong></td>
<td><strong>-1.679 ± 1.488</strong></td>
</tr>
<tr>
<td><strong>PCA+ESS (SI)</strong></td>
<td><strong>-2.719 ± 0.132</strong></td>
<td><strong>-3.007 ± 0.086</strong></td>
<td><strong>-0.225 ± 0.400</strong></td>
<td><strong>6.541 ± 0.095</strong></td>
<td><strong>-1.563 ± 1.243</strong></td>
</tr>
<tr>
<td><strong>PCA+VI (SI)</strong></td>
<td><strong>-2.716 ± 0.133</strong></td>
<td><strong>-2.994 ± 0.095</strong></td>
<td><strong>-0.396 ± 0.419</strong></td>
<td><strong>6.708 ± 0.105</strong></td>
<td><strong>-1.715 ± 1.588</strong></td>
</tr>
<tr>
<td><strong>SGD</strong></td>
<td><strong>-2.752 ± 0.132</strong></td>
<td><strong>-3.178 ± 0.198</strong></td>
<td><strong>-0.418 ± 0.426</strong></td>
<td><strong>6.567 ± 0.185</strong></td>
<td><strong>-1.736 ± 1.613</strong></td>
</tr>
<tr>
<td><strong>DVI</strong></td>
<td><strong>-2.410 ± 0.020</strong></td>
<td><strong>-3.060 ± 0.010</strong></td>
<td><strong>-0.470 ± 0.030</strong></td>
<td><strong>6.290 ± 0.040</strong></td>
<td><strong>-1.010 ± 0.060</strong></td>
</tr>
<tr>
<td><strong>DGP</strong></td>
<td><strong>-2.330 ± 0.060</strong></td>
<td><strong>-3.130 ± 0.030</strong></td>
<td><strong>-1.390 ± 0.140</strong></td>
<td><strong>3.600 ± 0.330</strong></td>
<td><strong>-1.320 ± 0.030</strong></td>
</tr>
<tr>
<td><strong>VI</strong></td>
<td><strong>-2.430 ± 0.030</strong></td>
<td><strong>-3.040 ± 0.020</strong></td>
<td><strong>-1.680 ± 0.040</strong></td>
<td><strong>5.870 ± 0.290</strong></td>
<td><strong>-2.380 ± 0.020</strong></td>
</tr>
<tr>
<td><strong>MCD</strong></td>
<td><strong>-2.400 ± 0.040</strong></td>
<td><strong>-2.970 ± 0.020</strong></td>
<td><strong>-1.380 ± 0.010</strong></td>
<td><strong>4.760 ± 0.010</strong></td>
<td><strong>-1.720 ± 0.010</strong></td>
</tr>
<tr>
<td><strong>VSD</strong></td>
<td><strong>-2.350 ± 0.050</strong></td>
<td><strong>-2.970 ± 0.020</strong></td>
<td><strong>-1.140 ± 0.020</strong></td>
<td><strong>4.830 ± 0.010</strong></td>
<td><strong>-1.060 ± 0.010</strong></td>
</tr>
</tbody>
</table>

**CIBER Wins on 7/11!**

<table>
<thead>
<tr>
<th>Elevators</th>
<th>KEGG</th>
<th>KEGGU</th>
<th>Protein</th>
<th>Skillcraft</th>
<th>Pol</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIBER (SECOND)</td>
<td><strong>1.245 ± 0.090</strong></td>
<td><strong>1.125 ± 0.269</strong></td>
<td><strong>-0.720 ± 0.036</strong></td>
<td><strong>-1.003 ± 0.035</strong></td>
<td><strong>2.555 ± 0.115</strong></td>
</tr>
<tr>
<td>CIBER (LAST)</td>
<td><strong>1.178 ± 0.088</strong></td>
<td><strong>0.964 ± 0.231</strong></td>
<td><strong>-0.720 ± 0.036</strong></td>
<td><strong>-1.001 ± 0.032</strong></td>
<td><strong>2.506 ± 0.150</strong></td>
</tr>
<tr>
<td>SWAG</td>
<td><strong>1.080 ± 0.035</strong></td>
<td><strong>0.749 ± 0.029</strong></td>
<td><strong>-0.700 ± 0.051</strong></td>
<td><strong>-1.180 ± 0.033</strong></td>
<td><strong>1.533 ± 1.084</strong></td>
</tr>
<tr>
<td>PCA+ESS (SI)</td>
<td><strong>1.074 ± 0.034</strong></td>
<td><strong>0.752 ± 0.025</strong></td>
<td><strong>-0.734 ± 0.063</strong></td>
<td><strong>-1.181 ± 0.033</strong></td>
<td><strong>-0.185 ± 2.779</strong></td>
</tr>
<tr>
<td>PCA+VI (SI)</td>
<td><strong>1.085 ± 0.031</strong></td>
<td><strong>0.757 ± 0.028</strong></td>
<td><strong>-0.712 ± 0.057</strong></td>
<td><strong>-1.179 ± 0.033</strong></td>
<td><strong>1.764 ± 0.271</strong></td>
</tr>
<tr>
<td>SGD</td>
<td><strong>1.012 ± 0.154</strong></td>
<td><strong>0.602 ± 0.224</strong></td>
<td><strong>-0.854 ± 0.085</strong></td>
<td><strong>-1.162 ± 0.032</strong></td>
<td><strong>1.073 ± 0.858</strong></td>
</tr>
<tr>
<td>OrthoVP</td>
<td><strong>1.022</strong></td>
<td><strong>0.701</strong></td>
<td><strong>-0.914</strong></td>
<td>—</td>
<td><strong>0.159</strong></td>
</tr>
<tr>
<td>NL</td>
<td><strong>0.698 ± 0.039</strong></td>
<td><strong>0.935 ± 0.265</strong></td>
<td><strong>0.670 ± 0.038</strong></td>
<td><strong>-0.884 ± 0.025</strong></td>
<td><strong>-1.002 ± 0.050</strong></td>
</tr>
</tbody>
</table>

44
Experiment: Image Classification

<table>
<thead>
<tr>
<th>Metric</th>
<th>NLL</th>
<th>ACC</th>
<th>ECE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td>CIFAR-10</td>
<td>CIFAR-100</td>
<td>CIFAR-10</td>
</tr>
<tr>
<td>CIBER</td>
<td>0.1927 ± 0.0029</td>
<td>0.9193 ± 0.0027</td>
<td>93.64 ± 0.09</td>
</tr>
<tr>
<td>SWAG</td>
<td>0.2503 ± 0.0081</td>
<td>1.2785 ± 0.0031</td>
<td>93.59 ± 0.14</td>
</tr>
<tr>
<td>SGD</td>
<td>0.3285 ± 0.0139</td>
<td>1.7308 ± 0.0137</td>
<td>93.17 ± 0.14</td>
</tr>
<tr>
<td>SWA</td>
<td>0.2621 ± 0.0104</td>
<td>1.2780 ± 0.0051</td>
<td>93.61 ± 0.11</td>
</tr>
<tr>
<td>SGLD</td>
<td>0.2001 ± 0.0059</td>
<td>0.9699 ± 0.0057</td>
<td>93.55 ± 0.15</td>
</tr>
<tr>
<td>KFAC</td>
<td>0.2252 ± 0.0032</td>
<td>1.1915 ± 0.0199</td>
<td>92.65 ± 0.20</td>
</tr>
</tbody>
</table>

- achieves accurate estimation of uncertainty
- applicable to large NNs
- boosts predictive performance
How to integrate diverse constraints?

Outline

• Differentiable learning under constraints
  • Key: constraint probability!

• Constrained probabilistic inference
  • Key: constraint solvers + statistical ML!
Future Work
How to integrate *diverse constraints*?

- Differentiable learning under constraints
  - Key: constraint probability!
- Constrained probabilistic inference
  - Key: constraint solvers + statistical ML!

*What more constraints are tractable?*

*How to deal with intractable ones? …*

*What inference amenable to the reduction?*

*How to deliver reliable & scalable inference? …*

Thanks!
References


