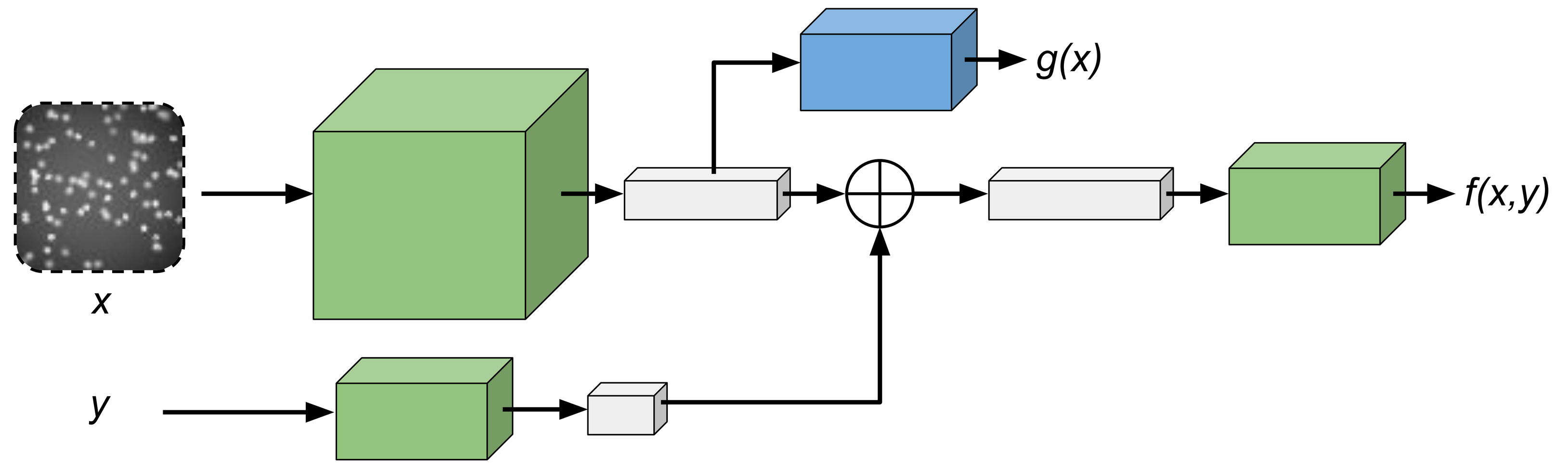
Learning Proposals for Practical Energy-Based Regression

Fredrik K. Gustafsson[1], Martin Danelljan[2], Thomas B. Schön[1]

[1]Department of Information Technology, Uppsala University, Sweden
[2]Computer Vision Lab, ETH Zürich, Switzerland

Overview

- We derive an efficient and convenient objective that can be employed to train a parameterized distribution $q(y|x;\phi)$ by directly minimizing its KL divergence to a conditional energy-based model (EBM) $p(y|x;\theta)$.
- We employ the proposed objective to jointly learn an effective MDN proposal distribution during EBM training, thus addressing the main practical limitations of energy-based regression.



Background: Energy-Based Models

An energy-based model (EBM) specifies a probability distribution $p(x;\theta)$ over $x \in \mathcal{X}$ directly via a parameterized scalar function $f_\theta : \mathcal{X} \to \mathbb{R}$:

$$p(x;\theta) = \frac{e^{f_\theta(x)}}{Z(\theta)}, \quad Z(\theta) = \int e^{f_\theta(\tilde{x})} d\tilde{x}$$

- The EBM $p(x;\theta)$ is thus a highly expressive model that puts minimal restricting assumptions on the true distribution $p(x)$. The normalizing partition function $Z(\theta) = \int e^{f_\theta(\tilde{x})} d\tilde{x}$ is however intractable, which complicates evaluating or sampling from the EBM $p(x;\theta)$.

Background: Energy-Based Regression

Train a neural network $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar value $f_\theta(x,y) \in \mathbb{R}$, then model the distribution $p(y|x)$ with the *conditional* EBM $p(y|x;\theta)$:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

Background: Energy-Based Regression - Prediction

Predict the most likely target under the model given an input $x^\star$ at test-time, i.e. $y^\star = \arg\max_y p(y|x^\star;\theta) = \arg\max_y f_\theta(x^\star, y)$. In practice, $y^\star = \arg\max_y f_\theta(x^\star, y)$ is approximated by refining an initial estimate $\hat{y}$ via T steps of gradient ascent,

$$y \leftarrow y + \lambda \nabla_y f_\theta(x^\star, y).$$

Background: Energy-Based Regression - Training

The neural network $f_\theta(x,y)$ can be trained using various methods for fitting a distribution $p(y|x;\theta)$ to observed data $\{(x_i, y_i)\}_{i=1}^N$.

The most straightforward training method is probably to approximate the negative log-likelihood $\mathcal{L}(\theta) = -\sum_{i=1}^N \log p(y_i|x_i;\theta)$ using importance sampling:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^N \log\left(\frac{1}{M}\sum_{m=1}^M \frac{e^{f_\theta(x_i, y_i^{(m)})}}{q(y_i^{(m)})}\right) - f_\theta(x_i, y_i), \quad (1)$$

$$\{y_i^{(m)}\}_{m=1}^M \sim q(y) \text{ (proposal distribution).}$$

Previous work has also employed noise contrastive estimation (NCE):

$$J_{\text{NCE}}(\theta) = -\frac{1}{N}\sum_{i=1}^N J_{\text{NCE}}^{(i)}(\theta), \quad J_{\text{NCE}}^{(i)}(\theta) = \log \frac{\exp\{f_\theta(x_i, y_i^{(0)}) - \log q(y_i^{(0)})\}}{\sum_{m=0}^M \exp\{f_\theta(x_i, y_i^{(m)}) - \log q(y_i^{(m)})\}},$$

$$y_i^{(0)} \triangleq y_i, \quad \{y_i^{(m)}\}_{m=1}^M \sim q(y) \text{ (noise distribution).}$$

- Effectively, $J_{\text{NCE}}(\theta)$ is the softmax cross-entropy loss for a classification problem with $M+1$ classes (which of the $M+1$ values $\{y_i^{(m)}\}_{m=0}^M$ is the true target y_i?).

Practical Limitations of Energy-Based Regression

In previous work, the proposal/noise distribution $q(y)$ was set to a mixture of K Gaussian components centered at the true target y_i, $q(y) = \frac{1}{K}\sum_{k=1}^K \mathcal{N}(y; y_i, \sigma_k^2 I)$.

- $q(y)$ contains task-dependent hyperparameters K and $\{\sigma_k^2\}_{k=1}^K$.
- $q(y)$ depends on the true target y_i and can thus only be utilized during training.

We address both these limitations by jointly learning a parameterized proposal/noise distribution $q(y|x;\phi)$ during EBM training. We derive an efficient and convenient objective that can be employed to train $q(y|x;\phi)$ by directly minimizing its KL divergence to the EBM $p(y|x;\theta)$.

Learning the Proposal

- We want the proposal/noise distribution $q(y|x;\phi)$ to be a close approximation of the EBM $p(y|x;\theta)$. Specifically, we want to find ϕ that minimizes the KL divergence between $q(y|x;\phi)$ and $p(y|x;\theta)$.
- Therefore, we seek to compute $\nabla_\phi D_{\text{KL}}(p(y|x;\theta) \parallel q(y|x;\phi))$. The gradient $\nabla_\phi D_{\text{KL}}$ is generally intractable, but can be conveniently approximated.

Learning the Proposal

Result 1: For a conditional EBM $p(y|x;\theta) = e^{f_\theta(x,y)} / \int e^{f_\theta(x,\tilde{y})} d\tilde{y}$ and distribution $q(y|x;\phi)$,

$$\nabla_\phi D_{\text{KL}}(p \parallel q) \approx \nabla_\phi \log\left(\frac{1}{M}\sum_{m=1}^M \frac{e^{f_\theta(x, y^{(m)})}}{q(y^{(m)}|x;\phi)}\right),$$

where $\{y^{(m)}\}_{m=1}^M$ are M independent samples drawn from $q(y|x;\phi)$.

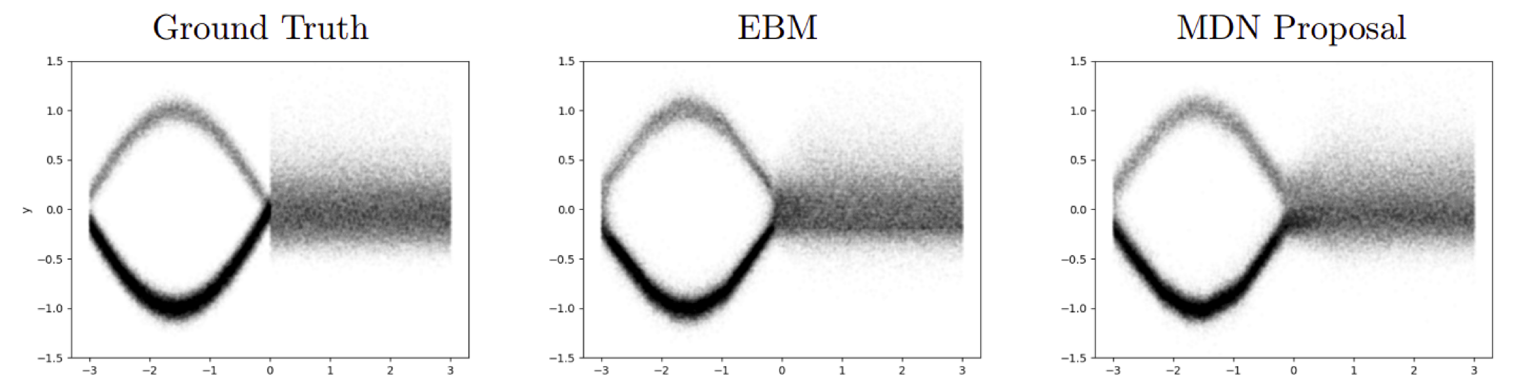
Given data $\{x_i\}_{i=1}^N$, Result 1 implies that $q(y|x;\phi)$ can be trained to approximate the EBM $p(y|x;\theta)$ by minimizing the loss,

$$J_{\text{KL}}(\phi) = \frac{1}{N}\sum_{i=1}^N \log\left(\frac{1}{M}\sum_{m=1}^M \frac{e^{f_\theta(x_i, y_i^{(m)})}}{q(y_i^{(m)}|x_i;\phi)}\right),$$

$$\{y_i^{(m)}\}_{m=1}^M \sim q(y|x_i;\phi).$$

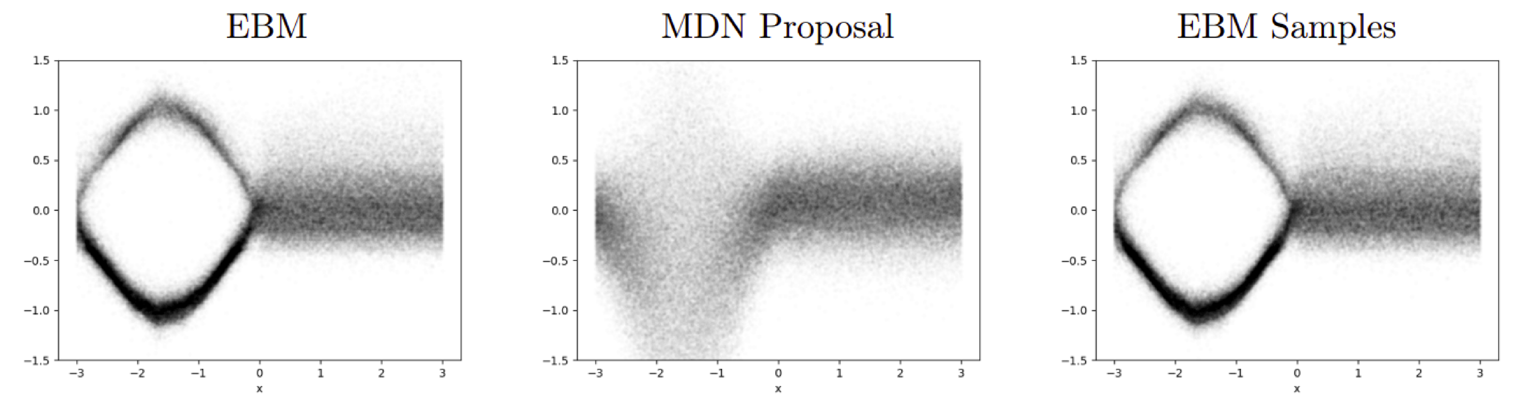
Joint Training Method

- Since $J_{\text{KL}}(\phi)$ is identical to the first term of the EBM loss $J(\theta)$ in (1), the EBM $p(y|x;\theta)$ and proposal $q(y|x;\phi)$ can be trained by jointly minimizing (1) w.r.t. both θ and ϕ.
- The EBM $p(y|x;\theta)$ and proposal/noise distribution $q(y|x;\phi)$ can also be jointly trained by updating ϕ via $J_{\text{KL}}(\phi)$, and updating θ via $J_{\text{NCE}}(\theta)$.



Utilizing the Proposal

As $q(y|x;\phi)$ has been trained to approximate the EBM $p(y|x;\theta)$, it can be utilized with self-normalized importance sampling to e.g. compute the EBM mean at test-time, thus producing a stand-alone prediction $y^\star$. It can also be used to draw approximate samples from the EBM:



fredrik.gustafsson@it.uu.se, martin.danelljan@vision.ee.ethz.ch, thomas.schon@it.uu.se