AdaCat: Adaptive Categorical Discretization for Autoregressive Models

Qiyang Li, Ajay Jain, Pieter Abbeel

University of California, Berkeley

web

arXiv

Motivations

- Autoregressive generative models can estimate complex data distributions such as language, audio and images.
- Most existing methods either operate on discrete data distribution or discretize continuous data into several bins and use categorical distributions over the bins to approximate the continuous data distribution.
- Such approximation cannot express sharp changes in density without using significantly more bins, making it parameter inefficient.

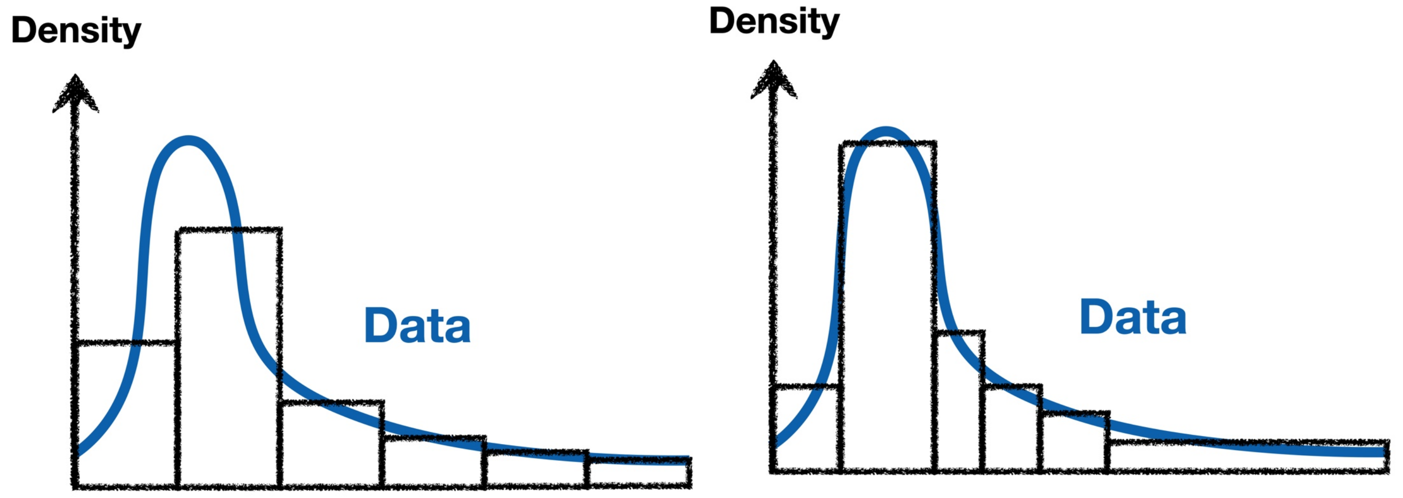
Adaptive Discretization

AdaCat is a mixture of k non-overlapping truncated uniforms $(w, h \in \mathbb{R}^k)$

$$\mathrm{AdaCat}_k(w,h) : f_{w,h,k}(x) = \sum_{i=1}^{k} \left\{ \mathbb{I}_{[c_i \le x < c_i + w_i]} \frac{h_i}{w_i} \right\}, c_i = \sum_{j=1}^{k-1} w_j \quad (1)$$

- If w is fixed: Uniform discretization
- If h is fixed: Quantile regression

1D Illustrative Example $(k = 6)$



Uniform Discretization

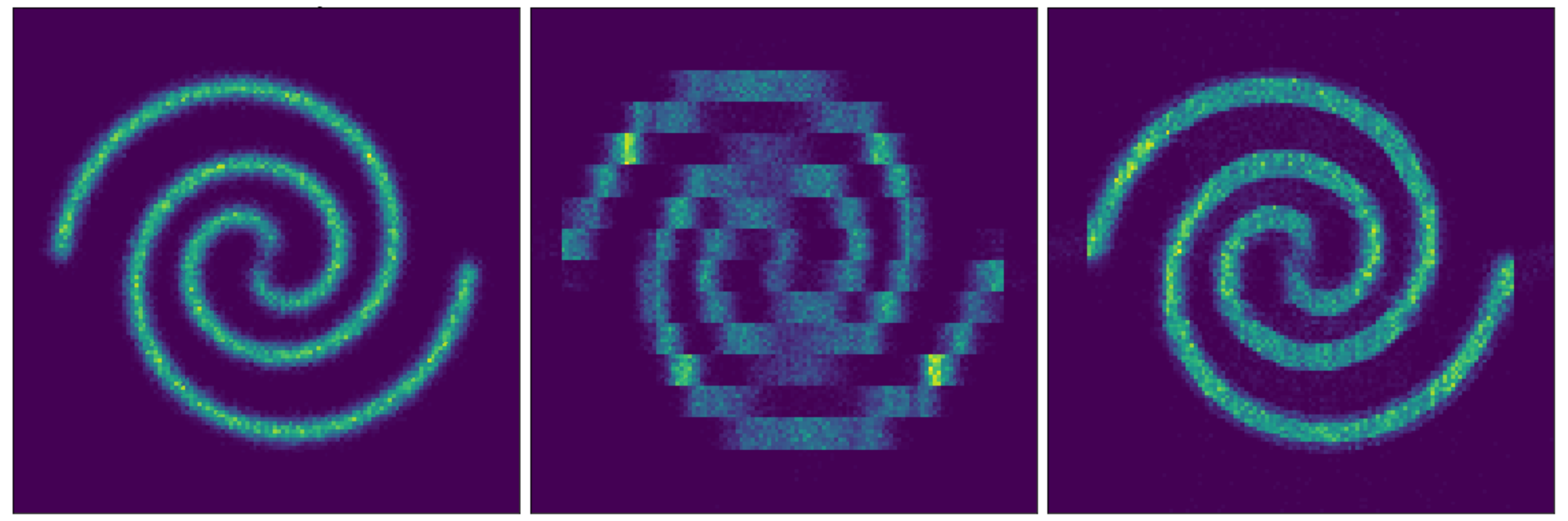
Adaptive Discretization

2D Toy Density Estimation $(k = 12)$

Real data samples

Uniform Categorical Discretization

Adaptive Categorical Discretization (AdaCat)



Overcoming Non-Smoothness

Empirical Negative Log-likelihood Objective

$$\hat{\mathcal{L}}_{\text{orig}}(\theta) = \frac{1}{n}\sum_{d=1}^{n}\sum_{t=1}^{m} \log p_\theta(x_d^t | x_d^{<t}) = \frac{1}{n}\sum_{d=1}^{n}\sum_{t=1}^{m} \log f_{w^t,h^t,k}(x_d^t)$$

where $w^t, h^t = \mathrm{nn}(x_d^{<t})$ are the predicted parameters of the AdaCat distribution.

- The gradient of $\hat{\mathcal{L}}_{\text{orig}}$ with respect to the w^t, h^t is ill-defined at the bin boundaries.
- Therefore, $\nabla_\theta \hat{\mathcal{L}}_{\text{orig}}$ might be biased. Optimizing $\hat{\mathcal{L}}_{\text{orig}}$ can lead to bin collapse (red below).

Smooth Objective

$$\hat{\mathcal{L}}_{\text{smooth}} = \frac{1}{n}\sum_{d=1}^{n}\sum_{t=1}^{m}\left[\int_{\tilde{x}} \zeta(\tilde{x}|x_d^t) \log p_\theta(\tilde{x}|x_d^{<t}) d\tilde{x}\right]$$

where $\zeta(\tilde{x}|x_d^t)$ is a smoothing distribution.

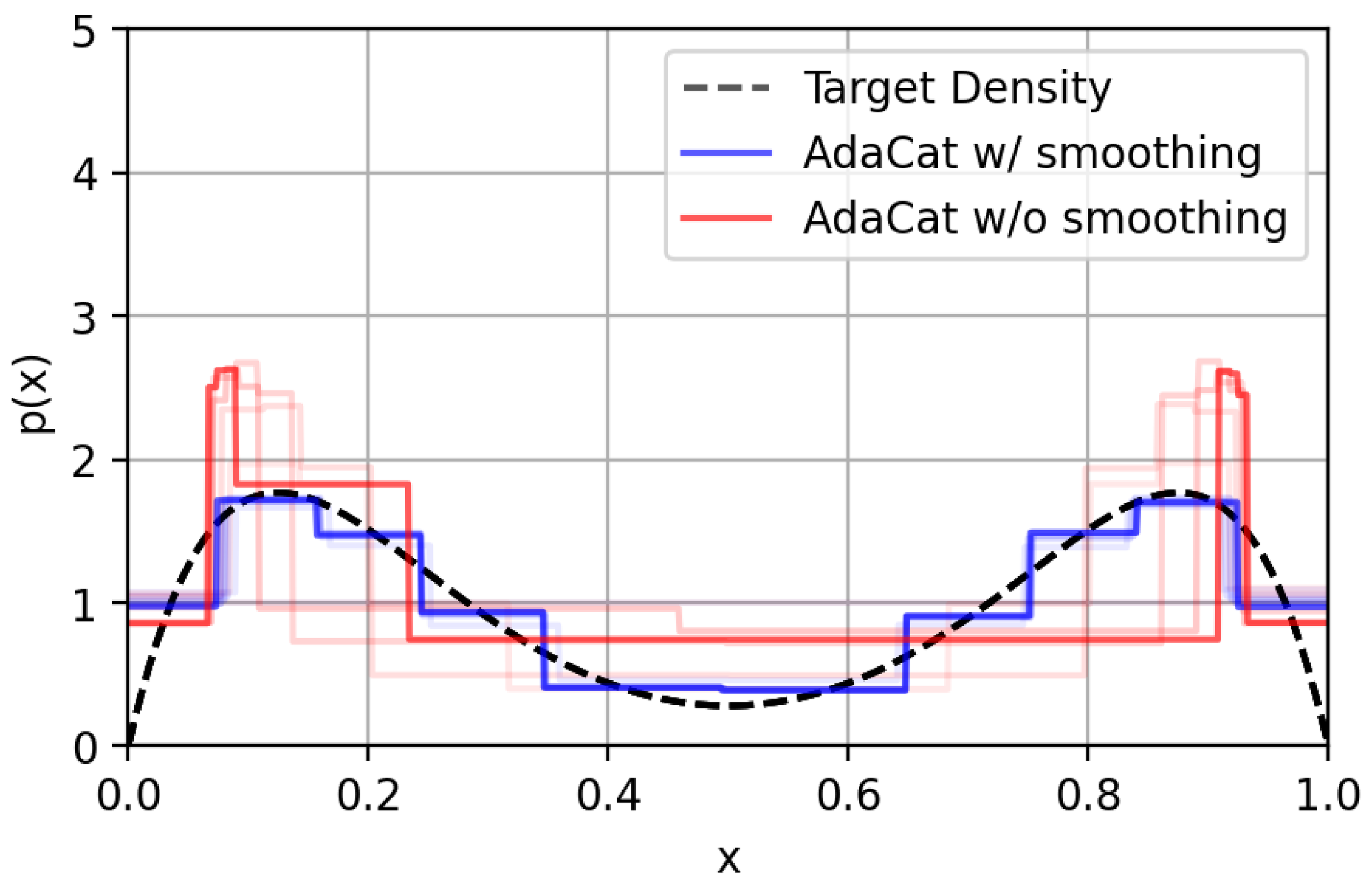
The inner integral can be computed as follows:

$$\int_{\tilde{x}} \zeta(\tilde{x}) \log f_{w,h,k}(\tilde{x}) d\tilde{x} = \sum_{j=1}^{k} [(F(c_j + w_j) - F(c_j))(\log h_j - \log w_j)]$$

where $F(x)$ is the cumulative density function (CDF) of $\xi(x)$.

- The integral can be easily computed for ζ's when their CDFs are simple (uniform and Gaussian).
- The gradient is well-defined everywhere and $\mathcal{L}_{\text{smooth}}$ prevents bin collapse (blue below).

Optimization Dynamics of Original vs. Smooth Objective



Empirical Results

MNIST Density Estimation

Parameters	Uniform	Adaptive Quantile	DMoL [2]	AdaCat
512	N/A	×	0.761	**0.561**
256	**0.561**	×	0.698	0.573
216	0.838	×	0.704	**0.615**
180	1.061	×	0.684	**0.629**
152	1.299	×	0.776	**0.612**
128	1.490	×	0.700	**0.608**
64	2.453	×	0.720	**0.695**
32	3.392	1.276	**0.715**	0.793
Best	**0.561**	1.276	0.715	**0.561**

×: training diverged

Offline RL with Model-based Planning in D4RL (Trajectory Transformer [1])

Dataset	Uniform	Quantile	AdaCat
HalfCheetah-Medium	44.0 ±0.31	46.9 ±0.4	**47.8** ±0.22
Hopper-Medium	67.4 ±2.9	61.1 ±3.6	**69.2** ±4.5
Walker2d-Medium	**81.3** ±2.1	79.0 ±2.8	79.3 ±0.8

See our paper for more results on tabular and audio data: `arxiv.org/abs/2208.02246`

Try it out on PyTorch! – `pip install adacat`

```
from adacat.torch import Adacat
params = torch.nn.Parameter(torch.randn(10 * 2))
optim = torch.optim.Adam([params], lr=0.01)

for _ in range(n_its):
    dist = Adacat(params) # PyTorch distribution
    xs = sample_batch()   # Sample from data distribution
    loss = -dist.log_prob(xs, smooth_coeff=0.001).mean()
    optim.zero_grad()
    loss.backward()       # Optimize the smooth loss
    optim.step()
```

References

[1] M. Janner, Q. Li, and S. Levine. Offline reinforcement learning as one big sequence modeling problem. In *Advances in Neural Information Processing Systems*, 2021.

[2] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.