Nijkamp et al. 2019: On the Anatomy of MCMC-Based Maximum Likelihood Learning of Energy-Based Models

Minqi Pan

March 5, 2020

On the Anatomy of MCMC-Based Maximum Likelihood Learning of Energy-Based Models

- AAAI 2020 "ML: Probabilistic Methods II", Feb 12nd, 2020
- Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, Ying Nian Wu
- UCLA Department of Statistics

Outline

1. Learning Energy-Based Models
   - Maximum Likelihood Estimation
   - MCMC Sampling with Langevin Dynamics
   - MCMC Initialization
2. Two Axes of ML Learning
   - First Axis: Expansion or Contraction
   - Second Axis: MCMC Convergence or Non-Convergence
   - Learning Algorithm
3. Experiments
   - Low-Dimensional Toy Experiments
   - Synthesis from Noise with Non-Convergent ML Learning
   - Convergent ML Learning
   - Mapping the Image Space

Outline

1. Learning Energy-Based Models
   - Maximum Likelihood Estimation
   - MCMC Sampling with Langevin Dynamics
   - MCMC Initialization
2. Two Axes of ML Learning
   - First Axis: Expansion or Contraction
   - Second Axis: MCMC Convergence or Non-Convergence
   - Learning Algorithm
3. Experiments
   - Low-Dimensional Toy Experiments
   - Synthesis from Noise with Non-Convergent ML Learning
   - Convergent ML Learning
   - Mapping the Image Space

Gibbs-Boltzmann Density

- $p_i \propto \exp\{-\frac{\varepsilon_i}{kT}\}$
  - E.g. softmax $\sigma(z_1, \dots, z_K) = (\dots, \frac{\exp\{z_i\}}{\sum_j \exp z_j}, \dots)$
- $p_\theta(x) = \frac{1}{Z(\theta)} \exp\{-U(x;\theta)\}$
  - $x \in \mathcal{X} \subset \mathbb{R}^N$
  - $U(x;\theta) \subset \mathcal{U} = \{U(\,\cdot\,;\theta) : \theta \in \Theta\}$
  - $Z(\theta) = \int_{\mathcal{X}} \exp\{-U(x;\theta)\} dx$
- $U(x;\theta) = F(x;\theta)$
  - F is a ConvNet: $\mathbb{R}^N \to \mathbb{R}$
  - $\theta \in \mathbb{R}^D$

ML Learning via Kullback-Leibler Divergence

- $\arg\min_\theta \mathcal{L}(\theta) = \arg\min_\theta D_{\mathsf{KL}}(q||p_\theta)$
- $\arg\min_\theta \mathcal{L}(\theta) = \arg\min_\theta \int_{-\infty}^{\infty} q(x) \log(\frac{q(x)}{p_\theta(x)}) dx$
- $\arg\min_\theta \mathcal{L}(\theta) = \arg\min_\theta \{\log Z(\theta) + E_q[U(X;\theta)]\}$
- $\frac{d}{d\theta}\mathcal{L}(\theta) = \frac{d}{d\theta}\log Z(\theta) + \frac{d}{d\theta}E_q[U(X;\theta)]$
  - $\frac{d}{d\theta}\log Z(\theta) = -E_{p_\theta}[\frac{\partial}{\partial\theta}U(X;\theta)]$
- $\frac{d}{d\theta}\mathcal{L}(\theta) = \frac{d}{d\theta}E_q[U(X;\theta)] - E_{p_\theta}[\frac{\partial}{\partial\theta}U(X;\theta)]$
- $X^+ \sim q,\ X^- \sim p_\theta$
  - $\frac{d}{d\theta}\mathcal{L}(\theta) \approx \frac{\partial}{\partial\theta}(\frac{1}{n}\sum_{i=1}^{n} U(X_i^+;\theta) - \frac{1}{m}\sum_{i=1}^{m} U(X_i^-;\theta))$

Sampling

- $X_1^+, \ldots, X_n^+ \sim q$, iid
  - $\{X_i^+\}_{i=1}^n$ are a batch of training images
- $X_1^-, \ldots, X_m^- \sim p_\theta$, iid
  - Samping from current learned distribution p_θ is computationally intensive (must be performed for each update of θ)
  - Gibbs of Metropolis–Hastings MCMC updates each dimension (one pixel of the image) sequentially, hence is computationally infeasible when training an energy for standard image sizes

Outline

1. Learning Energy-Based Models
   - Maximum Likelihood Estimation
   - MCMC Sampling with Langevin Dynamics
   - MCMC Initialization
2. Two Axes of ML Learning
   - First Axis: Expansion or Contraction
   - Second Axis: MCMC Convergence or Non-Convergence
   - Learning Algorithm
3. Experiments
   - Low-Dimensional Toy Experiments
   - Synthesis from Noise with Non-Convergent ML Learning
   - Convergent ML Learning
   - Mapping the Image Space

Langevin Dynamics

- Stokes' law: $M\ddot{X} = -6\pi\eta R\dot{X}$
- Langevin equation: $M\ddot{X} = -\nabla U(X) - \gamma\dot{X} + \sqrt{2\gamma k_B T}R(t)$
  - $\langle R(t)\rangle = 0$
  - $\langle R(t)R(t')\rangle = \delta(t - t')$
- Itô diffusion: $\dot{X} = \frac{1}{2}\nabla \log \pi(X) + \dot{W}$
  - $X(t) \sim \rho(t)$
  - $\lim_{t\to\infty} \rho(t) = \pi$
- $X_{l+1} = X_l - \frac{\varepsilon^2}{2}\frac{\partial}{\partial x}U(X_l;\theta) + \varepsilon Z_l$
  - $Z_l \sim N(0, I_N)$
  - $\varepsilon > 0$
  - X has stationary distribution p_θ

Outline

1. Learning Energy-Based Models
   - Maximum Likelihood Estimation
   - MCMC Sampling with Langevin Dynamics
   - MCMC Initialization
2. Two Axes of ML Learning
   - First Axis: Expansion or Contraction
   - Second Axis: MCMC Convergence or Non-Convergence
   - Learning Algorithm
3. Experiments
   - Low-Dimensional Toy Experiments
   - Synthesis from Noise with Non-Convergent ML Learning
   - Convergent ML Learning
   - Mapping the Image Space

Two Branches

- Informative initialization
  - Data-based initialization. E.g. Contrastive Divergence (CD)
  - Persistent initialization. E.g. Persistent Contrastive Divergence (PCD)
- Noninformative initialization
  - Noise initialization. E.g. uniform, Gaussian

Outline

1. Learning Energy-Based Models
   - Maximum Likelihood Estimation
   - MCMC Sampling with Langevin Dynamics
   - MCMC Initialization
2. Two Axes of ML Learning
   - First Axis: Expansion or Contraction
   - Second Axis: MCMC Convergence or Non-Convergence
   - Learning Algorithm
3. Experiments
   - Low-Dimensional Toy Experiments
   - Synthesis from Noise with Non-Convergent ML Learning
   - Convergent ML Learning
   - Mapping the Image Space

Inspection of $\frac{d}{d\theta}\mathcal{L}(\theta)$

- cf. $\frac{d}{d\theta}\mathcal{L}(\theta) = \frac{d}{d\theta}E_q[U(X;\theta)] - E_{p_\theta}[\frac{\partial}{\partial\theta}U(X;\theta)]$
- Inspection of the gradient $\frac{d}{d\theta}\mathcal{L}(\theta)$ reveals the central role of the difference of the average energy of negative and positive samples
- Given the finite-step MCMC sampler and initialization used
  - Let s_t denote the distribution of negative samples at training step t: $X^- \sim s_t$
  - Let $d_{s_t}(\theta)$ denote the difference of the average energy of negative and positive samples
  - $d_{s_t}(\theta) \equiv E_q[U(X;\theta)] - E_{s_t}[U(X;\theta)]$

$d_{s_t}(\theta) = E_q[U(X;\theta)] - E_{s_t}[U(X;\theta)]$

- cf. $\frac{d}{d\theta}\mathcal{L}(\theta) \approx \frac{\partial}{\partial\theta}(\frac{1}{n}\sum_{i=1}^{n} U(X_i^+;\theta) - \frac{1}{m}\sum_{i=1}^{m} U(X_i^-;\theta))$
- d_{s_t} measures whether the positive samples from the data distribution q or the negative samples from s_t are more likely under the model p_θ
  - Perfect Learning & Exact MCMC Convergence: $p_\theta = q \wedge p_\theta = s_t \Rightarrow d_{s_t}(\theta) = 0$
  - $|d_{s_t}| > 0 \Rightarrow$ Divergent Learning or Divergent Sampling
- However
  - $d_{s_t}(\theta) = 0 \not\Rightarrow$ Perfect Learning & Exact MCMC Convergence
  - Divergent Learning: $p_\theta \neq q \not\Rightarrow |d_{s_t}| > 0$
  - Divergent Sampling: $p_\theta \neq s_t \not\Rightarrow |d_{s_t}| > 0$

$d_{s_t}(\theta) = E_q[U(X;\theta)] - E_{s_t}[U(X;\theta)]$

- For each update t on the parameter path $\{\theta_t\}_{t=1}^{T+1}$
  - 1st Axis: $\text{sign}(d_{s_t})$
    - "Contraction", "vanishing gradients":
      $d_{s_t}(\theta_t) > 0 \Rightarrow E_q[U(X;\theta)] > E_{s_t}[U(X;\theta)]$
    - "Expansion", "exploding gradients":
      $d_{s_t}(\theta_t) < 0 \Rightarrow E_q[U(X;\theta)] < E_{s_t}[U(X;\theta)]$
  - 2nd Axis: s_t and p_{θ_t}
    - Convergent MCMC: $s_t \approx p_{\theta_t}$
    - Divergent MCMC: $s_t \not\approx p_{\theta_t}$
- cf. $\frac{d}{d\theta}\mathcal{L}(\theta) \approx \frac{\partial}{\partial\theta}(\frac{1}{n}\sum_{i=1}^{n} U(X_i^+;\theta) - \frac{1}{m}\sum_{i=1}^{m} U(X_i^-;\theta))$
- cf. $X_{l+1} = X_l - \frac{\varepsilon^2}{2}\frac{\partial}{\partial x}U(X_l;\theta) + \varepsilon Z_l$

Discoveries

- Only the 1st axis governs the stability and synthesis results
  - Stable ML Learning: Oscillation of expansion and contraction updates
- Behavior along the 2nd axis determines the realism of steady-state samples from the final learned energy
  - Samples from p_{θ_t} is realistics $\Leftrightarrow s_t \approx p_{\theta_t}$
  - We define "convergent ML" $\equiv$ implementations s.t. $s_t \approx p_{\theta_t}$
- All prior ConvNet potentials are learned with non-convergent ML
- Without proper tuning of the sampling phase, the learning heavily gravitates towards non-convergent ML

Average Image Gradient Magnitude v_t

- cf. $d_{s_t}(\theta) \equiv E_q[U(X;\theta)] - E_{s_t}[U(X;\theta)]$
- Suppose Langevin chain $(Y_t^{(0)}, \ldots, Y_t^{(L)}) \sim w_t$ and $Y_t^{(L)} \sim s_t$
  - $v_t \equiv E_{w_t}[\frac{1}{L+1} \sum_{l=0}^{L} \| \frac{\partial}{\partial y} U(Y_t^{(l)}; \theta_t) \|_2]$
- If v_t is very large, gradients will overwhelm the noise, and the resulting dynamics are similar to gradient descent
- If v_t is very small, sampling becomes an isotropic random walk

v_t and d_{s_t}

- Gradient magnitude v_t and computational loss d_{s_t} are highly correlated at the current iteration, and exhibit significant negative correlation at a short-range lag
- v_t and d_{s_t} both have significant negative autocorrelation for short-range lag
- Expansion and contraction updates tend to have opposite effects on v_t
- Expansion updates tend to increase **gradient strength** in the near future and vice-versa
- Expansion updates tend to follow **contraction updates** and vice-versa
- The natural oscillation between expansion and contraction updates underlies the stability of ML

Unstable Learning

- Consecutive updates in the expansion phase will increase v_t so that the gradient can better overcome noise and samples can more quickly reach low-energy regions. But learning can become unstable when U is updated in the expansion phase for many consecutive iterations if $v_t \to \infty, U(X^+) \to -\infty, U(X^-) \to \infty$
- many consecutive contraction updates can cause v_t to shrink to 0, leading to the solution $U(x) = c$ for some constant c
- In proper ML learning, the expansion updates that follow contraction updates prevent the model from collapsing to a flat solution and force U to learn meaningful features of the data

Discoveries

- Network can easily learn to balance the energy of the positive and negative samples so that $d_{s_t}(\theta_t) \approx 0$ after only a few model updates
- ML Learning can adjust v_t so that the gradient is strong enough to balance d_{s_t} and obtain high-quality samples from virtually any initial distribution in a small number of MCMC steps
- The natural oscillation of ML learning is the foundation of the robust synthesis capabilities of ConvNet potentials

Outline

1. Learning Energy-Based Models
   - Maximum Likelihood Estimation
   - MCMC Sampling with Langevin Dynamics
   - MCMC Initialization
2. Two Axes of ML Learning
   - First Axis: Expansion or Contraction
   - Second Axis: MCMC Convergence or Non-Convergence
   - Learning Algorithm
3. Experiments
   - Low-Dimensional Toy Experiments
   - Synthesis from Noise with Non-Convergent ML Learning
   - Convergent ML Learning
   - Mapping the Image Space

Discoveries

- High-quality synthesis is possible, and actually easier to learn, when there is a drastic difference between the finite-step MCMC distribution s_t and true steady-state samples of p_θ
- In prior arts, running the MCMC sampler for significantly longer than the number of training steps results in samples with significantly lower energy and unrealistic appearance
- Oscillation of expansion and contraction updates occurs for both convergent and non-convergent ML learning, but for very different reasons

Average Image Space Displacement r_t

- Define average image space displacement $r_t \equiv \frac{\varepsilon^2}{2} v_t$
- cf. $d_{s_t}(\theta) = E_q[U(X;\theta)] - E_{s_t}[U(X;\theta)]$
- cf. $X_{l+1} = X_l - \frac{\varepsilon^2}{2}\frac{\partial}{\partial x}U(X_l;\theta) + \varepsilon Z_l$
- cf. average image gradient magnitude $v_t \equiv E_{w_t}[\frac{1}{L+1}\sum_{l=0}^{L} \|\frac{\partial}{\partial y}U(Y_t^{(l)};\theta_t)\|_2]$
- In convergent ML, we expect v_t to converge to a constant that is balanced with the noise magnitude ε at a value that reflects temperature of the data density q
- ConvNet can circumvent this desireed behavior by tunning v_t w.r.t. the burn-in energy landscape rather than noise ε

The Case of Noise Initialization w/ Low ε

- Define $R \equiv$ the average distance between an image from the noise initialization distribution and an image from the data distribution
- The model adjusts v_t so that $r_t L \approx R$
- The MCMC paths are nearly linear from the starting point to the ending point
- L increases $\Rightarrow$ r_t shrinks $\Rightarrow$ mixing does not improve
- The model tunes v_t to control how far along the burn-in path the negative samples travel $\Rightarrow$ oscillation of expansion and contraction updates occurs

The Case of Data & Persistent Initialization w/ Low ε

- $U(x) \to c$ as $v_t, r_t \to 0$ because contraction updates dominate the learning dynamics
- Low $\varepsilon \Rightarrow$ sampling reduces to gradient descent $\Rightarrow$ samples initialized from the data will easily have lower energy than the data
- **Data-based initialization**: the energy can easily collapse to a trivial flat solution $\Rightarrow$ No authors trained ConvNet energy with CD
- **Persistent initialization**: the model learns to synthesize meaningful features early in learning and then contracts in gradient strength once it becomes easy to find negative samples with lower energy than the data

Convergence is Possible

- For all three initialization types, convergence becomes possible when ε is large enough
- The MCMC samples complete burn-in and begin to mix for large L, and increasing L will indeed lead to improved MCMC convergence as usual
- For noise initialization:
  - When L is small, it behaves similarly for high and low ε
  - When L is large, high $\varepsilon \Rightarrow$ the model tunes v_t to balance with ε rather than R/L
- For data-based and persistent initialization:
  - v_t adjusts to balance with ε instead of contracting to 0
  - Because the noise added during Langevin sampling forces U to learn meaningful features

Outline

1. Learning Energy-Based Models
   - Maximum Likelihood Estimation
   - MCMC Sampling with Langevin Dynamics
   - MCMC Initialization
2. Two Axes of ML Learning
   - First Axis: Expansion or Contraction
   - Second Axis: MCMC Convergence or Non-Convergence
   - Learning Algorithm
3. Experiments
   - Low-Dimensional Toy Experiments
   - Synthesis from Noise with Non-Convergent ML Learning
   - Convergent ML Learning
   - Mapping the Image Space

Noise and Step Size for Non-Convergent ML

- The tuning of noise τ and stepsize ε have little effect on training stability
- d_{s_t} is controlled by the depth of samples along the burnin path $\Rightarrow$ noise is not needed for oscillation
- Including low noise appears to improve synthesis quality

Noise and Step Size for Convergent ML

- It is essential to include noise with $\tau = 1$ and precisely tune ε so that the network learns true mixing dynamics through the gradient strength
- The step size ε should approximately match the local standard deviation of the data along the most constrained direction
  - An effective ε for 32×32 images with pixel values in $[-1, 1]$ appears to lie around 0.015.

Number of Steps

- When $\tau = 0$ or $\tau = 1$ and ε is very small
  - Learning leads to similar non-convergent ML outcomes for any $L \geq 100$
- When $\tau = 1$ and ε is correctly tuned
  - Sufficiently high values of L lead to convergent ML
  - Lower values of L lead to non-convergent ML

Informative Initialization

- For non-convergent ML even with as few as $L = 100$ Langevin updates
  - Informative MCMC initialization is NOT needed
  - The model can naturally learn fast pathways to realistic negative samples from an arbitrary initial distribution
- For convergent ML
  - Informative initialization can greatly reduce the magnitude of L needed

Network structure

- For the 1st convolutional layer
  - A 3×3 convolution with stride 1 helps to avoid checkerboard patterns or other artifacts
- For convergent ML
  - Use of non-local layers appears to improve synthesis realism

Regularization and Normalization

- NOT NEEDED!
  - Prior distributions (e.g. Gaussian)
  - Weight regularization
  - Batch normalization
  - Layer normalization
  - Spectral normalization

Optimizer and Learning Rate

- For non-convergent ML
  - Adam improves training speed and image quality
- For convergent ML
  - Adam appears to interfere with learning a realistic steady-state
    - When $\tau = 1$ and properly tuned ε and L, higher values of learning rate γ lead to non-convergent ML
    - When $\tau = 1$ and properly tuned ε and L, sufficiently low values of learning rate γ lead to convergent ML

Outline

1. Learning Energy-Based Models
   - Maximum Likelihood Estimation
   - MCMC Sampling with Langevin Dynamics
   - MCMC Initialization
2. Two Axes of ML Learning
   - First Axis: Expansion or Contraction
   - Second Axis: MCMC Convergence or Non-Convergence
   - Learning Algorithm
3. Experiments
   - Low-Dimensional Toy Experiments
   - Synthesis from Noise with Non-Convergent ML Learning
   - Convergent ML Learning
   - Mapping the Image Space

Convergence and Non-convergence

- Both have a standard deviation of 0.15 along the most constrained direction – the ideal step size ε for Langevin dynamics is close to 0.15
- Non-convergence
  - Noise MCMC initialization used
  - $L = 500$
  - $\varepsilon = 0.125$
  - Short-run samples reflect the ground-truth densities
  - Learned densities are sharply concentrated and different from the ground-truths
- Convergence
  - Can be learned with sufficient Langevin noise

Outline

1. Learning Energy-Based Models
   - Maximum Likelihood Estimation
   - MCMC Sampling with Langevin Dynamics
   - MCMC Initialization
2. Two Axes of ML Learning
   - First Axis: Expansion or Contraction
   - Second Axis: MCMC Convergence or Non-Convergence
   - Learning Algorithm
3. Experiments
   - Low-Dimensional Toy Experiments
   - Synthesis from Noise with Non-Convergent ML Learning
   - Convergent ML Learning
   - Mapping the Image Space

Sampling from Scratch

- Informative MCMC intialization is NOT NEEDED for successful synthesis
- High-fidelity and diverse images generated FROM NOISE for MNIST, Oxford Flowers 102, CelebA, CIFAR-10
  - Langevin starts from uniform noise for each update of θ
  - Langevin steps $L = 100$
  - $\tau = 0$
  - $\varepsilon = 1$
  - Adam used
  - Learning rate $\gamma = 0.0001$

Outline

1. Learning Energy-Based Models
   - Maximum Likelihood Estimation
   - MCMC Sampling with Langevin Dynamics
   - MCMC Initialization
2. Two Axes of ML Learning
   - First Axis: Expansion or Contraction
   - Second Axis: MCMC Convergence or Non-Convergence
   - Learning Algorithm
3. Experiments
   - Low-Dimensional Toy Experiments
   - Synthesis from Noise with Non-Convergent ML Learning
   - Convergent ML Learning
   - Mapping the Image Space

Convergence w/ Correct Langevin Noise

- Noise initialization
  - $L \approx 20000$
- Persistent initialization
  - SGD,$\gamma = 0.0005, \tau = 1, \varepsilon = 0.015$
  - For each batch, initialize $10,000$ persistent images from noise and update 100 images
  - L reduces to 500
- MCMC samples mix in the steady-state energy spectrum throughout training
- MCMC samples approximately converge for each parameter update t (beyond burn-in)
- The model eventually learns a realistic steady-state

Outline

1. Learning Energy-Based Models
   - Maximum Likelihood Estimation
   - MCMC Sampling with Langevin Dynamics
   - MCMC Initialization
2. Two Axes of ML Learning
   - First Axis: Expansion or Contraction
   - Second Axis: MCMC Convergence or Non-Convergence
   - Learning Algorithm
3. Experiments
   - Low-Dimensional Toy Experiments
   - Synthesis from Noise with Non-Convergent ML Learning
   - Convergent ML Learning
   - Mapping the Image Space

The Structure of a Convergent Energy

- A well-formed energy function partitions the image space into meaningful Hopfield basins of attraction.
- First identify many metastable MCMC samples
- Then sort the metastable samples from lowest energy to highest energy and sequentially group images if travel between samples is possible in a magnetized energy landscape
- Continue until all minima have been clustered
- Basin structure of learned $U(x)$ for the Oxford Flowers 102 dataset visualized