TTIC 31230, Fundamentals of Deep Learning

David McAllester, Winter 2018

Deep Graphical Models II

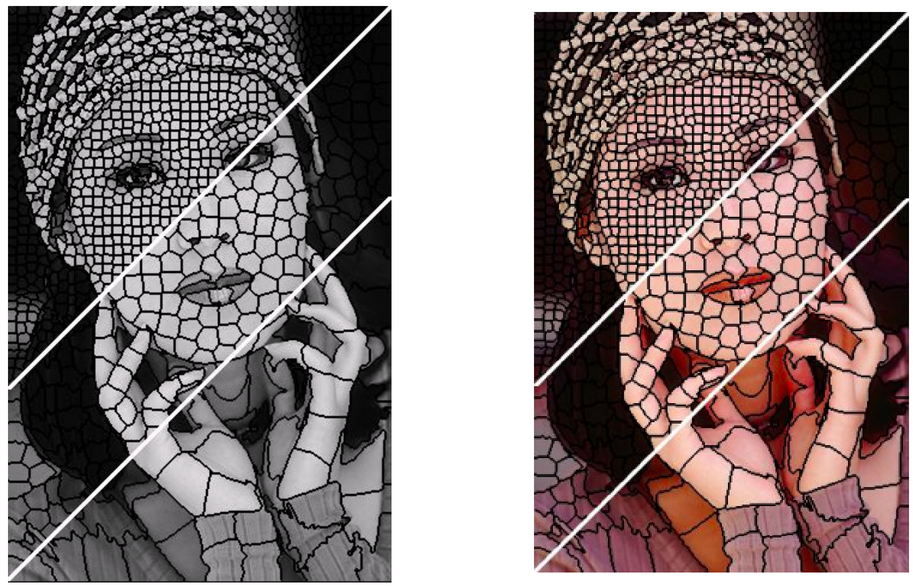
Algorithms for Approximate SGD

MCMC Sampling

Pseudo-Likelihood

Contrastive Divergence

Superpixel Colorization



SLIC superpixels, Achanta et al.

x is a black and white image.

y is a color image drawn from $\mathrm{Pop}(y|x)$.

$\hat{y}$ is an arbitrary color image.

$P_\Phi(\hat{y}|x)$ is the probability that model Φ assigns to the color image y given black and white image x.

Exponential Softmax

The tensor $s_e[\tilde{y}]$ is computed from x and Φ.

$$\begin{aligned} P_s(\hat{y}) &= \operatorname*{softmax}_{\hat{y}} \; s(\hat{y}) \\ s(\hat{y}) &= \sum_{e \in \mathrm{HyperEdges}} s_e[\hat{y}[e]] \end{aligned}$$

Backpropagation

The input is the image x and the parameter package Φ

$$\begin{aligned} &\vdots \\ s_e[\hat{y}] &= \ldots \\ \mathcal{L} &= -\ln\ P(y \mid s_{\mathcal{E}}[\mathcal{Y}]) \end{aligned}$$

We abbreviate $P(\hat{y} \mid s_{\mathcal{E}}[\mathcal{Y}])$ as $P_s(\hat{y})$ — the distribution on $\hat{y}$ defined by the tensor s.

We need to compute $\nabla_s - \ln P_s(y)$, or equivalently, $s_e.\mathrm{grad}[\tilde{y}]$.

$$s_e.\mathrm{grad}[\tilde{y}] = P_e(\tilde{y}) - \mathbb{1}[\tilde{y} = y[e]]$$

Sampling

The quantities $P_e(\tilde{e})$ are **hyperedge marginals**.

We can estimate the hyperedge marginals by sampling $\hat{y}$ from $P_s(\hat{y})$.

Monte Carlo Markov Chain (MCMC) Sampling

Metropolis Algorithm

Pick an initial graph label $\hat{y}$ and then repeat:

1. Pick a "neighbor" $\hat{y}'$ of $\hat{y}$ uniformly at random. The neighbor relation must be symmetric. Perhaps Hamming distance one.

2. If $s(\hat{y}') > s(\hat{y})$ update $\hat{y} = \hat{y}'$

3. If $s(\hat{y}') \leq s(\hat{y})$ then update $\hat{y} = \hat{y}'$ with probability $e^{-(s(\hat{y})-s(\hat{y}'))}$

Markov Processes and Stationary Distributions

A Markov process is a process defined by a fixed state transition probability $P(\hat{y}'|\hat{y}) = M_{\hat{y}',\hat{y}}$.

Let P^t the probability distribution for time t.

$$P^{t+1} = MP^t$$

If every state can be reached form every state (ergodic process) then P^t converges to a unique **stationary distribution** P^∞

$$P^\infty = MP^\infty$$

Metropolis Correctness

To verify that the Metropolis process has the correct stationary distribution we simply verify that $MP = P$ where P is the desired distribution.

This can be done by checking that under the desired distribution the flow from $\hat{y}$ to $\hat{y}'$ equals the flow from $\hat{y}'$ to $\hat{y}$ (**detailed balance**).

Metropolis Correctness

For $s(\hat{y}) \geq s(\hat{y}')$

$$\begin{aligned} \text{flow}(\hat{y}' \to \hat{y}) &= \frac{1}{Z} e^{s(\hat{y}')} \frac{1}{N} \\ \text{flow}(\hat{y} \to \hat{y}') &= \frac{1}{Z} e^{s(\hat{y})} \frac{1}{N} e^{-\Delta f} = \frac{1}{Z} e^{s(\hat{y}')} \frac{1}{N} \end{aligned}$$

But detailed balance is not required in general (see Hamiltonian MCMC).

Gibbs Sampling

The Metropolis algorithm wastes time by rejecting proposed moves.

Gibbs sampling avoids this move rejection.

In Gibbs sampling we select a node i at random and change that node by drawing a new node value conditioned on the current values of the other nodes.

Gibbs Sampling

$$P_s(i = \tilde{y} \mid \hat{y}) \doteq P_s(\hat{y}[i] = \tilde{y} \mid \hat{y}[1], \ldots, \hat{y}[i-1], \hat{y}[i+1], \ldots, \hat{y}[I])$$

Markov Blanket Property:

$$P_s(i = \tilde{y} \mid \hat{y}) = P_s(i = \tilde{y} \mid \hat{y}[N(i)])$$

Gibbs Sampling, Repeat:

- Select i at random
- draw $\tilde{y}$ from $P_s(i = \tilde{y} \mid \hat{y})$
- $\hat{y}[i] = \tilde{y}$

Gibbs Sampling

Let $\hat{y}[i = \tilde{y}]$ be the assignment $\hat{y}'$ equal to $\hat{y}$ except $\hat{y}'[i] = \tilde{y}$.

$$P_s(i = \tilde{y} \mid \hat{y}) \;=\; \frac{P_s(\hat{y}[i] = \tilde{y})}{\sum_{\tilde{y}} P_s(\hat{y}[i] = \tilde{y})}$$

$$=\; \frac{e^{s(\hat{y}[i=\tilde{y}])}}{\sum_{\tilde{y}} e^{s(\hat{y}[i=\tilde{y}])}}$$

Correctness Proof

$P_s(\hat{y})$ is a stationary distribution of Gibbs Sampling.

- Select i at random
- draw $\tilde{y}$ from $P_s(i = \tilde{y} \mid \hat{y})$
- $\hat{y}[i] = \tilde{y}$

The distribution before the update equals the distribution after the update.

Pseudolikelihood

In Pseudolikelihood we replace the objective $-\log P_s(\hat{y})$ with the objective $-\log \tilde{Q}_s(\hat{y})$ where

$$\tilde{Q}_s(\hat{y}) \doteq \prod_i P_s(i = \hat{y}[i] \mid \hat{y})$$

$$\mathrm{loss}(f) \doteq -\log \tilde{Q}(y)$$

$$s.\mathrm{grad}[e, \tilde{y}] = \sum_i -\partial \log P_s[i = \hat{y}[i] \mid \hat{y}]/\partial s[e, \tilde{y}]$$

Pseudolikelihood Consistency

$$\underset{Q}{\operatorname{argmin}}\ E_{y\sim\mathrm{Pop}}\ -\log\tilde{Q}(y)=\mathrm{Pop}$$

Proof of Consistency I

We have

$$\min_{Q} \; E_{y \sim \mathrm{Pop}} \; -\log \tilde{Q}(y) \;\; \leq \;\; E_{y \sim \mathrm{Pop}} \; -\log \widetilde{\mathrm{Pop}}(y)$$

If we can show

$$\min_{Q} \; E_{y \sim \mathrm{Pop}} \; -\log \tilde{Q}(y) \;\; \geq \;\; E_{y \sim \mathrm{Pop}} \; -\log \widetilde{\mathrm{Pop}}(y)$$

Then the minimizer (the argmin) is Pop as desired.

Proof of Consistency II

We will prove the case of two nodes.

$$\min_{Q} \; E_{y\sim\mathrm{Pop}} - \log Q(y[1]|y[2]) \; Q(y[2]|y[1])$$

$$\geq \min_{P_1,P_2} E_{y\sim\mathrm{Pop}} - \log P_1(y[1]|y[2]) \; P_2(y[2]|y[1])$$

$$= \min_{P_1} E_{y\sim\mathrm{Pop}} - \log P_1(y[1]|y[2]) + \min_{P_2} E_{y\sim\mathrm{Pop}} - \log P_2(y[2]|y[1])$$

$$= E_{y\sim\mathrm{Pop}} - \log \mathrm{Pop}(y[1]|y[2]) + E_{y\sim\mathrm{Pop}} - \log \mathrm{Pop}(y[2]|y[1])$$

$$= E_{y\sim\mathrm{Pop}} - \log \widetilde{\mathrm{Pop}}(y|x)$$

Contrastive Divergence

Algorithm (CDk): Run k steps of MCMC for $P_s(\hat{y})$ **starting from** y to get $\hat{y}$.

Then set

$$s.\mathrm{grad}[e, \tilde{y}] = \mathbb{1}[\hat{y}[e] = \tilde{y}] - \mathbb{1}[y[e] = \tilde{y}]$$

Theorem: If $P_s(\hat{y}) = \mathrm{Pop}$ then

$$E_{y \sim \mathrm{Pop}}\ \mathbb{1}[\hat{y}[e] = \tilde{y}] - \mathbb{1}[y[e] = \tilde{y}] = 0$$

Here we can take $k = 1$ — **no mixing time required**.

END