Training Products of Experts by Minimizing Contrastive Divergence

Geoffrey E. Hinton

presented by Frank Wood

BROWN

Goal

- Learn parameters for probability distribution models of high dimensional data
  - (Images, Population Firing Rates, Securities Data, NLP data, etc)

Mixture Model

$$p\left(\vec{d} \mid \theta_1, \ldots, \theta_n\right) = \sum_m \alpha_m f_m\left(\vec{d} \mid \theta_m\right)$$

Use EM to learn parameters

Product of Experts

$$p\left(\vec{d} \mid \theta_1, \ldots, \theta_n\right) = \frac{\prod_m f_m\left(\vec{d} \mid \theta_m\right)}{\sum_{\vec{c}} \prod_m f_m\left(\vec{c} \mid \theta_m\right)}$$

Use Contrastive Divergence to learn parameters.

BROWN

Take Home

- Contrastive divergence is a general MCMC gradient ascent learning algorithm particularly well suited to learning Product of Experts (PoE) and energy- based (Gibbs distributions, etc.) model parameters.
- The general algorithm:
  - Repeat Until "Convergence"
    - Draw samples from the current model *starting from the training data.*
    - Compute the expected gradient of the log probability w.r.t. all model parameters over both samples and the training data.
    - Update the model parameters according to the gradient.

BROWN

Sampling – Critical to Understanding

- Uniform
  - rand() Linear Congruential Generator
    - x(n) = a * x(n-1) + b mod M

      0.2311 0.6068 0.4860 0.8913 0.7621 0.4565 0.0185
- Normal
  - randn() Box-Mueller
    - x1,x2 ~ U(0,1) -> y1,y2 ~N(0,1)
      - y1 = sqrt(- 2 ln(x1)) cos(2 pi x2)
      - y2 = sqrt(- 2 ln(x1)) sin(2 pi x2)
- Binomial(p)
  - if(rand()<p)
- More Complicated Distributions
  - Mixture Model
    - Sample from a Gaussian
    - Sample from a multinomial (CDF + uniform)
  - Product of Experts
    - Metropolis and/or Gibbs

BROWN

The Flavor of Metropolis Sampling

- Given some distribution $p(\vec{d} \mid \theta)$, a random starting point $\vec{d}_{t-1}$, and a symmetric proposal distribution $J(\vec{d}_t \mid \vec{d}_{t-1})$.
- Calculate the ratio of densities $r = \frac{p(\vec{d}_t \mid \theta)}{p(\vec{d}_{t-1} \mid \theta)}$ where $\vec{d}_t$ is sampled from the proposal distribution.
- With probability $\min(r,1)$ accept $\vec{d}_t$.
- Given sufficiently many iterations

$$\{\vec{d}_n, \vec{d}_{n+1}, \vec{d}_{n+2}, \ldots\} \sim p(\vec{d} \mid \theta)$$

Only need to know the distribution up to a proportionality!

BROWN

Contrastive Divergence (Final Result!)

Model parameters.

Training data (empirical distribution).

Samples from model.

$$\Delta\theta_m \propto \left\langle \frac{\partial \log f_{\theta_m}}{\partial \theta_m} \right\rangle_{P^0} - \left\langle \frac{\partial \log f_{\theta_m}}{\partial \theta_m} \right\rangle_{P_\theta^1}$$

Law of Large Numbers, compute expectations using samples.

$$\Delta\theta_m \propto \frac{1}{N}\sum_{d \in \mathrm{D}} \frac{\partial \log f_{\theta_m}(d)}{\partial \theta_m} - \frac{1}{N}\sum_{c \sim P_\theta^1} \frac{\partial \log f_{\theta_m}(c)}{\partial \theta_m}$$

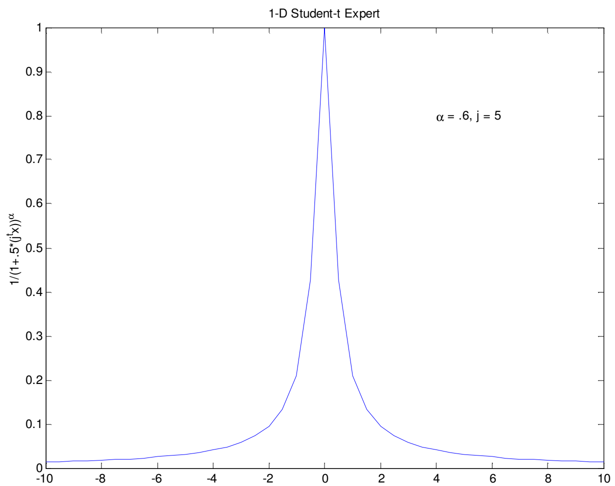
Now you know how to do it, let's see why this works!

BROWN

But First: The last vestige of concreteness.

- Looking towards the future:
  - Take f to be a Student-t.

$$f_{\theta^m}\left(\vec{d}\right) = f_{\alpha_m;\vec{j}_m}\left(\vec{d}\right) = \frac{1}{\left(1+\frac{1}{2}\left(\vec{j}_m^{\mathrm{T}}\vec{d}\right)\right)^{\alpha_m}}$$



Dot product ⇔Projection ⇔1-D Marginal

  - Then (for instance)

$$\frac{\partial \log f_{\alpha_m;\vec{j}_m}\left(\vec{d}\right)}{\partial \alpha_m} = \frac{-\partial \alpha_m \log\left(1+\frac{1}{2}\left(\vec{j}_m^{\mathrm{T}}\vec{d}\right)\right)}{\partial \alpha_m} = -\log\left(1+\frac{1}{2}\left(\vec{j}_m^{\mathrm{T}}\vec{d}\right)\right)$$

BROWN

Maximizing the training data log likelihood

- We want maximizing parameters

$$\arg\max_{\theta_1,\ldots,\theta_n} \log p(\mathrm{D} \mid \theta_1,\ldots,\theta_n) = \arg\max_{\theta_1,\ldots,\theta_n} \log \prod_{\vec{d}\in \mathrm{D}} \left(\frac{\prod_m f_m(\vec{d} \mid \theta_m)}{\sum_{\vec{c}} \prod_m f_m(\vec{c} \mid \theta_m)} \right)$$

Standard PoE form

Over all training data.

Assuming d's drawn independently from p()

- Differentiate w.r.t. to all parameters and perform gradient ascent to find optimal parameters.
- The derivation is somewhat nasty.

BROWN

Maximizing the training data log likelihood

$$\frac{\partial \log p(\mathrm{D} \mid \theta_1,\ldots,\theta_n)}{\partial \theta_m} = \frac{\partial \log \prod_{\vec{d} \in \mathrm{D}} p(\vec{d} \mid \theta_1,\ldots,\theta_n)}{\partial \theta_m}$$

$$= \sum_{\vec{d} \in \mathrm{D}} \frac{\partial}{\partial \theta_m} \log p(\vec{d} \mid \theta_1,\ldots,\theta_n)$$

$$= N \left\langle \frac{\partial \log p(\vec{d} \mid \theta_1,\ldots,\theta_n)}{\partial \theta_m} \right\rangle_{P_\theta^\infty}$$

Remember this equivalence!

BROWN

Maximizing the training data log likelihood

$$\frac{1}{N}\frac{\partial \log p(\mathrm{D} \mid \theta_1,\ldots,\theta_n)}{\partial \theta_m} = \frac{1}{N}\frac{\partial}{\partial \theta_m}\log \prod_{\vec{d}\in \mathrm{D}} \frac{\prod_m f_m(\vec{d} \mid \theta_m)}{\sum_{\vec{c}}\prod_m f_m(\vec{c} \mid \theta_m)}$$

$$= \frac{1}{N}\sum_{\vec{d}\in \mathrm{D}} \frac{\partial \log f_m\left(\vec{d} \mid \theta_m\right)}{\partial \theta_m} - \frac{1}{N}\sum_{\vec{d}\in \mathrm{D}} \frac{\partial \log \sum_{\vec{c}} \prod_m f_m\left(\vec{c} \mid \theta_m\right)}{\partial \theta_m}$$

$$= \frac{1}{N}\sum_{\vec{d}\in \mathrm{D}} \frac{\partial \log f_m\left(\vec{d} \mid \theta_m\right)}{\partial \theta_m} - \frac{\partial \log \sum_{\vec{c}} \prod_m f_m\left(\vec{c} \mid \theta_m\right)}{\partial \theta_m}$$

BROWN

Maximizing the training data log likelihood

$$= \frac{1}{N}\sum_{d \in D} \frac{\partial \log f_m(\vec{d} \mid \theta_m)}{\partial \theta_m} - \frac{\partial \log \sum_{\vec{c}} \prod_m f_m(\vec{c} \mid \theta_m)}{\partial \theta_m}$$

$$= \left\langle \frac{\partial \log f_m(\vec{d} \mid \theta_m)}{\partial \theta_m} \right\rangle_{P^0} - \frac{\partial \log \sum_{\vec{c}} \prod_m f_m(\vec{c} \mid \theta_m)}{\partial \theta_m}$$

log(x)' = x'/x

$$= \left\langle \frac{\partial \log f_m(\vec{d} \mid \theta_m)}{\partial \theta_m} \right\rangle_{P^0} - \frac{1}{\sum_{\vec{c}} \prod_m f_m(\vec{c} \mid \theta_m)} \frac{\partial \sum_{\vec{c}} \prod_m f_m(\vec{c} \mid \theta_m)}{\partial \theta_m}$$

BROWN

Maximizing the training data log likelihood

$$= \left\langle \frac{\partial \log f_m(\vec{d} \mid \theta_m)}{\partial \theta_m} \right\rangle_{P^0} - \frac{1}{\sum_{\vec{c}} \prod_m f_m(\vec{c} \mid \theta_m)} \frac{\partial \sum_{\vec{c}} \prod_m f_m(\vec{c} \mid \theta_m)}{\partial \theta_m}$$

$$= \left\langle \frac{\partial \log f_m(\vec{d} \mid \theta_m)}{\partial \theta_m} \right\rangle_{P^0} - \frac{1}{\sum_{\vec{c}} \prod_m f_m(\vec{c} \mid \theta_m)} \frac{\sum_{\vec{c}} \prod_{j \neq m} f_j(\vec{c} \mid \theta_j) \partial f_m(\vec{c} \mid \theta_m)}{\partial \theta_m}$$

log(x)’ = x’/x

$$= \left\langle \frac{\partial \log f_m(\vec{d} \mid \theta_m)}{\partial \theta_m} \right\rangle_{P^0} - \frac{1}{\sum_{\vec{c}} \prod_m f_m(\vec{c} \mid \theta_m)} \frac{\sum_{\vec{c}} \prod_m f_m(\vec{c} \mid \theta_m) \partial \log f_m(\vec{c} \mid \theta_m)}{\partial \theta_m}$$

BROWN

Maximizing the training data log likelihood

$$= \left\langle \frac{\partial \log f_m\left(\vec{d} \mid \theta_m\right)}{\partial \theta_m} \right\rangle_{P^0} - \frac{1}{\sum_{\vec{c}} \prod_m f_m(\vec{c} \mid \theta_m)} \frac{\sum_{\vec{c}} \prod_m f_m(\vec{c} \mid \theta_m) \partial \log f_m(\vec{c} \mid \theta_m)}{\partial \theta_m}$$

$$= \left\langle \frac{\partial \log f_m\left(\vec{d} \mid \theta_m\right)}{\partial \theta_m} \right\rangle_{P^0} - \sum_{\vec{c}} \left(\frac{\prod_m f_m(\vec{c} \mid \theta_m)}{\sum_{\vec{c}} \prod_m f_m(\vec{c} \mid \theta_m)} \frac{\partial \log f_m(\vec{c} \mid \theta_m)}{\partial \theta_m} \right)$$

$$= \left\langle \frac{\partial \log f_m\left(\vec{d} \mid \theta_m\right)}{\partial \theta_m} \right\rangle_{P^0} - \sum_{\vec{c}} p(\vec{c} \mid \theta_1, \ldots, \theta_n) \frac{\partial \log f_m(\vec{c} \mid \theta_m)}{\partial \theta_m}$$

Maximizing the training data log likelihood

$$= \left\langle \frac{\partial \log f_m(\vec{d} \mid \theta_m)}{\partial \theta_m} \right\rangle_{P^0} - \sum_c p(c \mid \theta_1, \ldots, \theta_n) \frac{\partial \log f_m(\vec{c} \mid \theta_m)}{\partial \theta_m}$$

$$= \left\langle \frac{\partial \log f_m(\vec{d} \mid \theta_m)}{\partial \theta_m} \right\rangle_{P^0} - \left\langle \frac{\partial \log f_m(\vec{c} \mid \theta_m)}{\partial \theta_m} \right\rangle_{P_\theta^\infty}$$

Phew! We're done! So:

$$\frac{\partial \log p(\mathrm{D} \mid \theta_1, \ldots, \theta_n)}{\partial \theta_m} \Leftrightarrow N \left\langle \frac{\partial \log p_\theta^\infty(\vec{d} \mid \theta_m)}{\partial \theta_m} \right\rangle_{P^0}$$

$$\propto \left\langle \frac{\partial \log f_m(\vec{d} \mid \theta_m)}{\partial \theta_m} \right\rangle_{P^0} - \left\langle \frac{\partial \log f_m(\vec{c} \mid \theta_m)}{\partial \theta_m} \right\rangle_{P_\theta^\infty}$$

BROWN

Equilibrium Is Hard to Achieve

- With:

$$\frac{\partial \log p(\mathrm{D} \mid \theta_1, \ldots, \theta_n)}{\partial \theta_m} \propto \left\langle \frac{\partial \log f_m\left(\vec{d} \mid \theta_m\right)}{\partial \theta_m} \right\rangle_{P^0} - \left\langle \frac{\partial \log f_m(\vec{c} \mid \theta_m)}{\partial \theta_m} \right\rangle_{P_\theta^\infty}$$

  we can now train our PoE model.

- But… there's a problem:
  - P_θ^∞ is computationally infeasible to obtain (esp. in an inner gradient ascent loop).
  - Sampling Markov Chain must converge to target distribution. Often this takes a *very* long time!

BROWN

Solution: Contrastive Divergence!

$$\frac{\partial \log p(\mathrm{D} \mid \theta_1,\ldots,\theta_n)}{\partial \theta_m} \propto \left\langle \frac{\partial \log f_m\left(\vec{d} \mid \theta_m\right)}{\partial \theta_m} \right\rangle_{P^0} - \left\langle \frac{\partial \log f_m\left(\vec{c} \mid \theta_m\right)}{\partial \theta_m} \right\rangle_{P_\theta^1}$$

- Now we don't have to run the sampling Markov Chain to convergence, instead we can stop after 1 iteration (or perhaps a few iterations more typically)
- Why does this work?
  - Attempts to minimize the ways that the model distorts the data.

BROWN

Equivalence of argmax log P() and argmax KL()

$$P^0 \| P_\theta^\infty = \sum_{\vec{d}} P^0(\vec{d}) \log \frac{P^0(\vec{d})}{P_\theta^\infty(\vec{d})}$$

$$= \sum_{\vec{d}} P^0(\vec{d}) \log P^0(\vec{d}) - \sum_{\vec{d}} P^0(\vec{d}) \log P_\theta^\infty(\vec{d})$$

$$= H(P^0) - \left\langle \log P_\theta^\infty(\vec{d}) \right\rangle_{P^0}$$

$$\frac{\partial P^0 \| P_\theta^\infty}{\partial \theta_m} = -\left\langle \frac{\partial \log P_\theta^\infty(\vec{d})}{\partial \theta_m} \right\rangle_{P^0}$$

This is what we got out of the nasty derivation!

Contrastive Divergence

- We want to "update the parameters to reduce the tendency of the chain to wander away from the initial distribution on the first step".

$$\frac{\partial}{\partial \theta_m}\left(P^0 \| P_\theta^\infty - P_\theta^1 \| P_\theta^\infty\right) = -\left\langle \frac{\partial \log P_\theta^\infty(\vec{d})}{\partial \theta_m} \right\rangle_{P^0} - \left\langle \frac{\partial \log P_\theta^\infty(\vec{d})}{\partial \theta_m} \right\rangle_{P_\theta^1}$$

$$\propto \left\langle \frac{\partial \log f_m(\vec{d} \mid \theta_m)}{\partial \theta_m} \right\rangle_{P^0} - \left\langle \frac{\partial \log f_m(\vec{c} \mid \theta_m)}{\partial \theta_m} \right\rangle_{P_\theta^\infty} - \left\langle \frac{\partial \log f_m(\vec{d} \mid \theta_m)}{\partial \theta_m} \right\rangle_{P_\theta^1} + \left\langle \frac{\partial \log f_m(\vec{c} \mid \theta_m)}{\partial \theta_m} \right\rangle_{P_\theta^\infty}$$

$$\propto \left\langle \frac{\partial \log f_m(\vec{d} \mid \theta_m)}{\partial \theta_m} \right\rangle_{P^0} - \left\langle \frac{\partial \log f_m(\vec{d} \mid \theta_m)}{\partial \theta_m} \right\rangle_{P_\theta^1}$$

BROWN

Contrastive Divergence (Final Result!)

Model parameters.

Training data (empirical distribution).

Samples from model.

$$\Delta\theta_m \propto \left\langle \frac{\partial \log f_{\theta_m}}{\partial \theta_m} \right\rangle_{P^0} - \left\langle \frac{\partial \log f_{\theta_m}}{\partial \theta_m} \right\rangle_{P_\theta^1}$$

$$\updownarrow$$

Law of Large Numbers, compute expectations using samples.

$$\Delta\theta_m \propto \frac{1}{N}\sum_{d\in D}\frac{\partial \log f_{\theta_m}(d)}{\partial \theta_m} - \frac{1}{N}\sum_{c\sim P_\theta^1}\frac{\partial \log f_{\theta_m}(c)}{\partial \theta_m}$$

Now you know how to do it and why it works!

BROWN