CPSC 550: Machine Learning II **2008/9 Term 2**

Lecture 16 — Mar 19, 2009

Lecturer: Nando de Freitas *Scribe: Nimalan Mahendran*

16.1 Review: Stochastic Approximation in Robbins-Monroe Form

Stochastic approximation is an iterative optimization method that finds optima of functions that can only be observed partially or in the presence of noise [**?**]. Lecture 15 gives the mapping between the noisy or partially observed function to be optimized and the corresponding stochastic approximation update equations.

The general form of the stochastic approximation algorithm is

$$x^{t+1} = x^t + \gamma^{(t)} g(x^{(t)}, \tilde{y}^{(t)})$$

where

- $g(.)$ is the gradient of the function we wish to maximize
- x are the model parameters
- y are the latent (unobserved) variables
- $\tilde{y}$ is a sample from the distribution $P(Y)$
- γ is the learning rate

This differs from the standard gradient descent algorithm in that we must sample $\tilde{y}$. This approximation introduces noise into our gradient term, and it is not immediately obvious when this algorithm will converge.

To better analyze the algorithm, we can re-write the above equation as follows:

$$= x^t + \gamma^{(t)} g(x^{(t)}) + \gamma^{(t)} w^{(t)}$$

where

$$g(x^{(t)}) = \int g(x^{(t)}, y)P(y)dy$$
$$w^{(t)} = g(x^{(t)}, \tilde{y}^{(t)}) - g(x^{(t)})$$

In practice, calculating $g(x^{(t)})$ is intractable, so it is only introduced here to illustrate that the stochastic gradient $g(x^{(t)}, \tilde{y}^{(t)})$ can be broken into two parts: The true gradient $g(x^{(t)})$, and the noise term $w^{(t)}$.

We will soon show that by choosing an appropriate step size $\gamma^{(t)}$, we can prove the convergence of this algorithm.

16.2 Stochastic Approximation for RBMs

The MLE for restricted Boltzmann Machines (RBMs) was derived in lecture 12. The MLE estimator for RBMs can not be solved analytically, but stochastic approximation can be used to find an good approximation.

The MLE estimator for RBMs sets the weights w_{ij} between nodes v_i and h_j such that the gradient of the log likelihood, $g(w, y)$, where $y = \{v, h\}$, is zero.

The gradient of the log likelihood is given by

$$g(w, y) = \frac{1}{N} \sum_{n=1}^{N} \left\{ d_{in} P_{x(t)}(h_{in} = 1 | d_{in}) - \tilde{v}_{in}^{(t+1)} P_{x^{(t)}}(h_{in} = 1 | \tilde{v}_n) \right\}$$

At each iteration, stochastic approximation for parameter training in RBMs will first sample $\tilde{y} = \{\tilde{v}, \tilde{h}\}$ from the folloing Rao-Blackwellized Gibbs Markov chain

$$\tilde{v}^{(t+1)} \sim P_X(v | \tilde{h}^{(t)})$$
$$\tilde{h}^{(t+1)} \sim P_X(h | \tilde{v}^{(t+1)})$$

and then perform the update below.

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \gamma^{(t)} \Big[\frac{1}{N} \sum_{n=1}^{N} \left\{ d_{in} P_{x(t)}(h_{in} = 1 | d_{in}) - \tilde{v}_{in}^{(t+1)} P_{x^{(t)}}(h_{in} = 1 | \tilde{v}_n) \right\} \Big]$$

These sampling-updating iterations will eventually converge on the MLE estimator for the RBM weights.

16.3 Selecting the Learning Rate

It is important to set the learning rate of the stochastic approximation algorithm correctly. If the learning rate is too low, the stochastic approximation algorithm will fall short of the maximum, as shown in Figure **??**. If the learning rate is too low, stochastic approximation algorithm will not converge, as shown in Figure **??**.

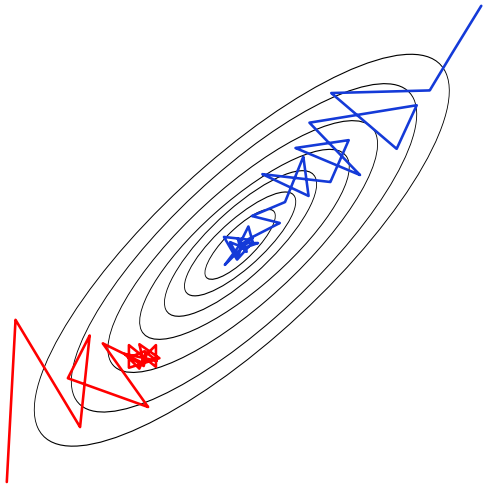

Figure 16.1. The red line depicts the trajectory of the stochastic approximation algorithm when the learning rate is too low, causing its progression towards the maximum to stop prematurely. The blue line depicts the trajectory of the stochastic approximation algorithm when the learning rate is set correctly.

16.3.1 Intuitions for Selecting the Learning Rate

Intuition for selecting $\gamma^{(t)}$ can be gained by unrolling the recursive formula for the stochastic approxmation $x^{(T)}$.

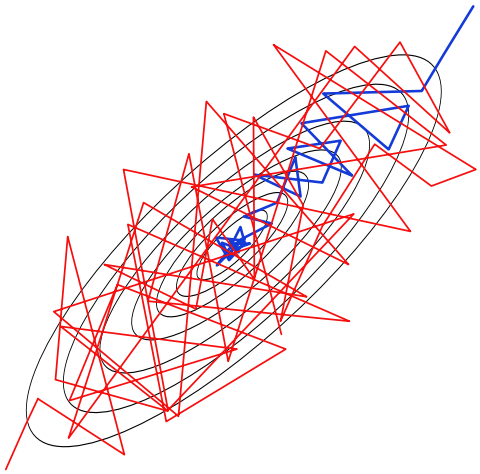

Figure 16.2. The red line depicts the trajectory of the stochastic approximation algorithm when the learning rate is too high, causing it to never converge. The blue line depicts the trajectory of the stochastic approximation algorithm when the learning rate is set correctly.

$$\sum_{t=0}^{T} x^{(t+1)} = \sum_{t=0}^{T} x^{(t)} + \sum_{t=0}^{T} \gamma^{(t)} g(x^{(t)}) + \sum_{t=0}^{T} \gamma^{(t)} w^{(t)}$$

$$\sum_{t=0}^{T} (x^{(t+1)} - x^{(t)}) = \sum_{t=0}^{T} \gamma^{(t)} g(x^{(t)}) + \sum_{t=0}^{T} \gamma^{(t)} w^{(t)}$$

$$x^{(T)} - x^{(0)} = \sum_{t=0}^{T} \gamma^{(t)} g(x^{(t)}) + \sum_{t=0}^{T} \gamma^{(t)} w^{(t)}$$

$$x^{(T)} = x^{(0)} + \sum_{t=0}^{T} \gamma^{(t)} g(x^{(t)}) + \sum_{t=0}^{T} \gamma^{(t)} w^{(t)}$$

$x^{(0)}$ is the starting point of the stochastic approximation. $\sum_{t=0}^{T} \gamma^{(t)} g(x^{(t)})$ is the portion of the stochastic approximation that follows the gradient. The random samples $\tilde{y}^{(t)}$ only occur in the $\sum_{t=0}^{T} \gamma^{(t)} w^{(t)}$ portion of the update. Therefore, $x^{(0)} + \sum_{t=0}^{T} \gamma^{(t)} g(x^{(t)})$ makes up the deterministic portion of the stochastic approximation and $\sum_{t=0}^{T} \gamma^{(t)} w^{(t)}$ makes up the stochastic portion.

Assume that the stochastic terms, $w^{(t)}$, have zero mean and finite variance.

$$\mathbb{E}(w^{(t)}) = 0$$
$$\mathbb{E}((w^{(t)})^2) = (\sigma^{(t)})^2 < \infty, \forall t$$

The following propetries for the entire stochastic portion of $x^{(T)}$ follow,

$$\mathbb{E}(\sum_{t=0}^{T} \gamma^{(t)} w^{(t)}) = 0$$
$$\mathbb{V}(\sum_{t=0}^{T} \gamma^{(t)} w^{(t)}) = \sum_{t=0}^{T} (\gamma^{(t)})^2 (\sigma^{(t)})^2 = k \sum_{t=0}^{T} (\gamma^{(t)})^2$$

$\mathbb{V}(\sum_{t=0}^{T} \gamma^{(t)} w^{(t)})$ can be made finite by setting $\gamma^{(t)} = \frac{1}{t}$. Finite variance ($\mathbb{V}(\sum_{t=0}^{T} \gamma^{(t)} w^{(t)}) < \infty$) together with zero mean ($\mathbb{E}(\sum_{t=0}^{T} \gamma^{(t)} w^{(t)}) = 0$) will guarantee convergence of this estimator.

Example: Strong Law of Large Numbers

The strong law of large numbers states that

$$\frac{1}{T} \sum_{t=1}^{T} x_t \xrightarrow{\text{a.s.}} \mathbb{E}(x)$$

This can be approximated stochastically,

$$\begin{aligned} S_T &= \frac{1}{T} \sum_{t=1}^{T} x_t \\ &= \frac{1}{T} x_T + \frac{1}{T} \frac{T-1}{T-1} \sum_{t=1}^{T-1} x_t \\ S_T &= \frac{1}{T} x_T + (1 - \frac{1}{T}) S_{T-1} \\ S_T &= S_{T-1} + \frac{1}{T}(x_T - S_{T-1}) \end{aligned}$$

16.3.2 Optimal Learning Rate Analysis from [?]

Let Γ be a matrix of learning rates $\gamma_{ij}^{(t)}$ for some x_{ij}.
Let $M_t = g(x^{(t)}, \tilde{y}^{(t)}) - g(x^{(t)})$.

Then the stochastic approximation update equations are as follows

$$\begin{aligned} x^{(t+1)} &= x^{(t)} - \frac{1}{t}\Gamma g(x^{(t)}, \tilde{y}^{(t)}) \\ &= x^{(t)} - \frac{1}{t}\Gamma g(x^{(t)}) - \frac{1}{t}\Gamma(g(x^{(t)}, \tilde{y}^{(t)}) - g(x^{(t)})) \end{aligned}$$

[?] give the following derivation of the optimal learning rate, where the optimal learning rate is the one that results in the least variance in the estimator.

Assume that $\mathbb{V}(M_t) < \infty$. Then $x^{(t)}$ satisfies the following CLT.

$$\sqrt{t}(x^{(t)} - x^*) \sim \mathcal{N}(0, \Sigma)$$

where Σ satisifies

$$-(\frac{I}{2} + \Gamma H)\Sigma - \Sigma(\frac{I}{2} + \Gamma H)^T = \Gamma R \Gamma^T$$

and

$$\begin{aligned} H &= \bigtriangledown g(x^*) \\ R &= \sum_{-\infty}^{\infty} \mathbb{E}(M_t', M_0) \end{aligned}$$

where H is the Hessian.

Let Γ^* be the minimizer with respect to Γ of Σ, the covariance matrix. It can be shown that $\Gamma^* = H^{-1}$ and $\Sigma^* = H^{-1}RH'$.

Hence, the optimal choice Γ^* is such that

$$x^{(t+1)} = x^{(t)} - \frac{1}{t}H^{-1}z^{(t)}$$

This update equation corresponds to Newton's Method.

Two variants of stochastic approximation that involve averaging are presented below. These variants also attain the same CLT with $\Sigma^* = H^{-1}RH'$.

16.4 Stochastic Averaging

Stochastic averaging [**?**] is a variant of stochastic approximation that accumulates an average of the sequential values generated by stochastic approximation. Larger or even fixed learning rates can be used with stochastic averaging because the effect of new iterations diminishes in the accumulated average.

$$x^{(t+1)} = x^{(t)} + \gamma^{(t)} g(x^{(t)}, \tilde{y}^{(t)})$$
$$\bar{x}^{(t+1)} = \frac{1}{t+1} x^{(t+1)} + (1 - \frac{1}{t+1})\bar{x}^{(t)}$$

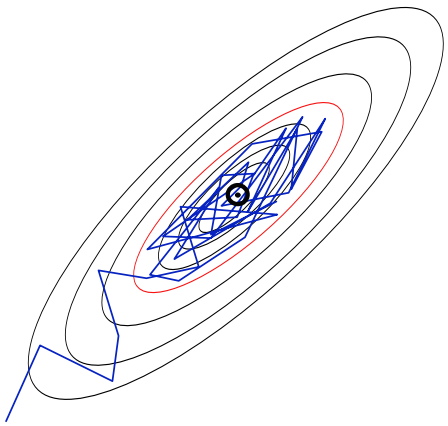

Figure 16.3. The blue line depicts the trajectory of the sequential values in the stochastic averaging algorithm algorithm. This trajectory moves throughout the red ellipse and accumulates in the average $\bar{x}^{(t)}$, represented by the black dot in the centre of the ellipses.

16.5 Gradient Averaging (Momentum)

Gradient averaging or the momentum algorithm is a variant of stochastic approximation that adds a momentum term that allows the effects of gradients from previous time steps to decay exponentially over time. The momentum term smooths out the trajectory of stochastic approximation.

The decay is $\lambda \in [0, 1)$ and momentum term is $x^{(t)} - x^{(t-1)}$.

The update equation for gradient averaging is given by

$$x^{(t+1)} = x^{(t)} + \gamma^{(t)} g(x^{(t)}, \tilde{y}^{(t)}) + \lambda(x^{(t)} - x^{(t-1)})$$

The momentum term can be unrolled to give a recurrence relation that makes the exponential decay of gradients from previous time steps more explicit.

$$\begin{aligned} x^{(t)} - x^{(t-1)} &= x^{(t-1)} + \gamma^{(t-1)} g(x^{(t-1)}, \tilde{y}^{(t-1)}) + \lambda(x^{(t-1)} - x^{(t-2)}) - x^{(t-1)} \\ &= \gamma^{(t-1)} g(x^{(t-1)}, \tilde{y}^{(t-1)}) + \lambda(x^{(t-1)} - x^{(t-2)}) \end{aligned}$$

The unrolled momentum term can be substituted back into the update equation for gradient averaging to show its relation to stochastic approximation.

$$\begin{aligned} x^{(t+1)} &= x^{(t)} + \sum_{k=0}^{t} \gamma^{(k)} \lambda^{t-k} g(x^{(k)}, \tilde{y}^{(k)}) \\ &= x^{(t)} + \gamma^{(t)} \lambda^{t-t} g(x^{(t)}, \tilde{y}^{(t)}) + \sum_{k=0}^{t-1} \gamma^{(k)} \lambda^{t-k} g(x^{(k)}, \tilde{y}^{(k)}) \\ &= x^{(t)} + \gamma^{(t)} g(x^{(t)}, \tilde{y}^{(t)}) + \sum_{k=0}^{t-1} \gamma^{(k)} \lambda^{t-k} g(x^{(k)}, \tilde{y}^{(k)}) \end{aligned}$$

If the decay term is set to $\lambda = 0$, gradient averaging is equivalent to stochastic approximation.

The smoothing effect of the momentum term can be seen in Figure **??**, where the trajectory of the gradient averaging algorithm is presented alongside the stochastic approximation algorithm.

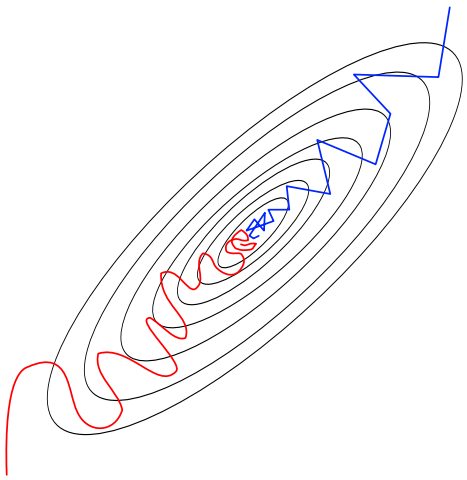

Figure 16.4. The red line depicts a trajectory of the gradient averaging algorithm. The gradient averaging algorithm trajectory is smoothed by the momentum term. The blue line depicts a trajectory of the stochastic approximation algorithm.