
Information-Based Gain Adaptation and Gradient Descent

Stephen Ingram

sfingram@cs.ubc.ca

Abstract

Stochastic Meta Descent is a powerful gradient descent algorithm that minimizes gradient oscillation in an adaptive way. This report examines the lineage of SMD and derives new gain adaptation and stochastic gradient descent algorithms based on the Information Filter. Preliminary experiments show a potential for future work with these algorithms.

1 Introduction and Previous Work

State estimation and function optimization are interrelated problems whose relationship is made evident by comparing their solutions. For state estimation, *Kalman Filtering* provides the optimal estimation of the state of a linear stochastic process. For function optimization, *gradient descent* seeks a function's stationary point by stepping along its gradient. The connection between the two is that sequentially updating the learning rates in gradient descent (also called gain adaptation) and updating process noise covariances in Kalman Filtering are equivalent operations [1].

Stochastic Meta Descent or SMD [3] is a stochastic gradient descent algorithm whose derivation takes advantage of this relationship between state estimation and function optimization. In particular, it derives its gain adaptation strategy ultimately from Kalman Filtering itself. The intermediary steps are the *K1* [2] and *ELKI* [4] algorithms. The precise details of this derivation is the subject of the following section of the report.

Information Filtering is an equivalent reformulation of the Kalman Filter. Instead of using the moment parameterization of the Gaussian distribution, the filter is defined in terms of the canonical parameterization. This allows the filter to rely on the Information Matrix. Perhaps, just as SMD is derived from Kalman Filtering, we can derive an alternative gradient descent algorithm from the Information Filter. Section 3 contains the derivation of two such algorithms along with some interesting intermediary results.

This report makes two contributions. First, it traces the lineage of SMD from the Kalman Filter to Gain Adaptation to Stochastic Gradient Descent. Second, it contributes new algorithms for Gain Adaptation and Gradient Descent based on the Information Filter.

2 The Genealogy of SMD

In the following sections, we review the algorithms that influenced SMD. This illustrates the theoretical foundations of SMD and exposes the strategy its designer took when authoring the algorithm.

2.1 Kalman Filtering

We begin by examining a typical state estimation problem (from [2]). Consider a linear stochastic process

$$y(t) = \theta(t)^T \phi(t) + \eta(t) \quad (1)$$

where y is a scalar observation, ϕ is a vector observation, η is zero-mean noise with covariance R , and θ is the hidden state vector that slowly varies over time, drifting according to a random walk with covariance Q . Then the following equation

$$\theta(t+1) = \theta(t) + K(t)(y(t) - \theta(t)^T \phi(t)) \quad (2)$$

is the Kalman Filtering algorithm for estimating the state θ . K is the Kalman gain defined to be

$$K(t) = \frac{P(t)\phi(t)}{R + \phi(t)^T P(t)\phi(t)} \quad (3)$$

and P is the process covariance matrix that is computed recursively as follows

$$P(t+1) = P(t) - \frac{P(t)\phi(t)\phi(t)^T P(t)}{R + \phi(t)^T P(t)\phi(t)} + Q \quad (4)$$

Kalman filtering is shown to be optimal in the least squares sense, but there are at least two important caveats. First, optimality is guaranteed only if close to exact values are given for R and Q . There are many cases where their acquisition is impossible. If these values differ, then the performance of the Kalman filter can degrade significantly. Second, updating the process covariance matrix is $O(n^2)$ where n is the dimension of θ . For large state spaces, this becomes intractable.

2.2 K1 Gain Adaptation

In response to these problems, Sutton designed the *K1* algorithm [2]. Applied to the process in (1) we retain the estimation step (2). However the gain update (3) is changed to

$$K(t) = \frac{\hat{P}(t)\phi(t)}{\hat{R} + \phi(t)^T \hat{P}(t)\phi(t)} \quad (5)$$

where the process covariance P is approximated by a diagonal matrix \hat{P} . Also, instead of using the exact noise covariance R , we use an estimation \hat{R} . The biggest change from Kalman filtering is the update for the approximate process covariance where K1 adapts the diagonals of \hat{P} by gradient descent.

Rather than update the diagonals \hat{p}_{ii} of \hat{P} directly, we descend on a set of parameters β_i where $\hat{p}_{ii}(t) = e^{\beta_i(t)}$ by the following:

$$\beta_i(t+1) = \beta_i(t) - \frac{1}{2}\mu \frac{\partial \delta^2(t)}{\partial \beta_i} \quad (6)$$

where $\delta(t) = y(t) - \theta(t)^T \phi(t)$ is the error from (2). K1 approximates $\frac{1}{2}\mu \frac{\partial \delta^2(t)}{\partial \beta_i}$ by $-\mu \delta(t) \phi_i(t) h_i(t)$. Here $h_i(t) \approx \frac{\partial \theta(t)}{\partial \beta_i}$ and is updated by

$$h_i(t+1) = (h_i(t) + k_i(t)\delta(t))(1 - k_i(t)\phi_i(t))^+ \quad (7)$$

It is important to note that in equation (7) the partial derivative $h_i(t)$ depends on the previous partial $h_i(t-1)$. This is frequently referred to as ‘‘carrying the partials forward in time’’ [5]. It is also important to note that K1 does not depend on parameter Q from Kalman Filtering and is $O(n)$. Its results indicate it is an accurate, efficient alternative to Kalman Filtering when an exact model of the process is not available.

2.3 Extended Linearized K1

The *Extended Linearized K1* or ELK1 algorithm by Schraudolph [4] illustrates how to apply the K1 strategy to gradient descent when updating the weights of a multi-layer perceptron. Here we try to minimize some objective function $F(y)$ where $y = \theta^T \phi$. Instead of the state estimation in (2) we perform

$$\theta(t+1) = \theta(t) + K(t)g(t) \quad (8)$$

where $g(t) = -\frac{\partial F}{\partial y(t)}$ (Schraudolph's derivation includes squashing functions, but we omit these here for clarity). The gain is still updated using equation (5), however update for the diagonalization P has changed. The diagonals p_{ii} are still parameterized by β_i but they are now updated as follows

$$\begin{aligned}\beta_i(t+1) &= \beta_i(t) - \mu \frac{\partial E}{\partial \beta_i} \\ \beta_i(t+1) &\approx \beta_i(t) + \mu g_i(t) \phi_i(t) h_i(t)\end{aligned}$$

Again, $h_i(t) \approx \frac{\partial \theta_i(t)}{\beta_i}$ however its definition has changed from (7)

$$h_i(t+1) = (h_i(t) + k_i(t)g_i(t))(1 - k_i(t)\phi_i(t))^+$$

In the derivation of this last step, Schraudolph assumes that $\frac{\partial g}{\partial y} = \frac{\partial^2 F}{\partial y^2} \approx 1$. This is likely to break down in many cases. To remedy this, we must incorporate a diagonal of the Hessian of F into our definition of $h_i(t)$.

2.4 SMD

Finally, we are ready to derive SMD from ELK1 [3]. As in ELK1 we have a scalar objective function F we seek to optimize with respect to a parameter θ and a sequence of observations ϕ . However there are three important distinctions between SMD and ELK1

1. *We no longer assume $F(y)$ where $y = \theta^T \phi$.* We only assume that F depends on θ and ϕ and we now express the objective as $F(\theta, \phi)$. We still express the update to $\theta(t)$ as (8). However dropping the linear assumption between our state and observations causes Schraudolph to drop the Kalman-style update in (3) that we used in K1 and ELK1, which means tossing out the diagonalization of \hat{P} .
2. *We now descend on the parameters of K directly.* This is a direct consequence of the previous change. To retain the benefits of the exponentiated parameterization of P , we descend K in log space.
3. *SMD applies a decay to the partials.* This is in contrast to simply ‘‘carrying the partials forward’’ as in (7). This yields an exponential weighting of previous step-sizes [5].

The following SMD update to the gain K incorporates the above changes. Below we consider k to be a vector composed of the diagonals of K :

$$\begin{aligned}\ln k(t+1) &= \ln k(t) - \mu \sum_{i=0}^t \lambda^i \frac{\partial F(\theta(t+1))}{\partial \ln k(t-i)} \\ \ln k(t+1) &= \ln k(t) - \mu g(t+1) \cdot v(t+1) \\ k(t+1) &= k(t) e^{\mu g(t+1) \cdot v(t+1)}\end{aligned}$$

Here $v(t)$ functions like $h(t)$ in ELK1, as an approximation of the decayed sum of partial derivatives of $\theta(t)$ w.r.t $\ln k$. This yields the following update for v .

$$\begin{aligned}v(t+1) &= \sum_{i=0}^t \lambda^i \frac{\partial \theta(t+1)}{\partial \ln k(t-i)} = \sum_{i=0}^t \lambda^i \frac{\partial \theta(t)}{\partial \ln k(t-i)} + \sum_{i=0}^t \lambda^i \frac{\partial(k(t) \cdot g(t))}{\partial \ln k(t-i)} \\ &\approx \lambda v(t) + k(t) \cdot (g(t) + \lambda H(t)v(t))\end{aligned}$$

$H(t)$ is the instantaneous Hessian of F w.r.t. $\theta(t)$.

3 Information-Based Gain Adaptation and Gradient Descent

Just as we have gone from Kalman Filtering to other, less expensive state estimation and gradient descent algorithms, this section performs similar derivations based on the Information Filter. Because Information Filtering recursively updates the information matrix S , perhaps the algorithms we derive will be more stable than SMD in cases where SMD breaks down.

3.1 Information Filtering

We again examine the state estimation problem (2) of the stochastic linear process (1). Now K is the Information gain defined to be

$$K(t) = \frac{\phi(t)R^{-1}}{S(t) + \phi(t)^T R^{-1} \phi(t)} \quad (9)$$

and S is the information matrix. We can use the knowledge that $S^{-1} = P$ and update recursively P according to (4) and invert it (I omit better conditioned methods in the interest of space).

Information Filtering is mathematically equivalent to Kalman Filtering, but differs numerically. The conditioning of each filter is the reciprocal of the other. Unfortunately we still require exact modeling of the stochastic process and the information update is still $O(n^2)$.

3.2 I1 Gain Adaptation

Following K1, we set $R = 1$ and diagonalize the Information Matrix. Thus our gain update becomes

$$K(t) = \frac{\phi(t)}{\hat{S}(t) + \phi(t)^T \phi(t)} \quad (10)$$

We parameterize the elements of \hat{S} , $\hat{s}_{ii}(t) = e^{\beta_i(t+1)}$ and perform gradient descent on β_i as in (6). Because we are information filtering, the derivative $h(t)$ differs from K1. Fortunately, the new derivation is straightforward. First, we need to make some definitions

$$\begin{aligned} D(t) &\equiv \hat{S}(t) + \phi(t)^T \hat{R}^{-1} \phi(t) = \hat{S}(t) + \phi^2(t) \\ k_i(t) &\equiv \phi_i(t) D^{-1}(t) \\ \frac{\partial D(t)}{\partial \beta_i} &= \hat{S}_i(t) \\ \frac{\partial D^{-1}(t)}{\partial \beta_i} &= -D^{-2}(t) \frac{\partial D(t)}{\partial \beta_i} = \frac{\hat{S}_i(t)}{(\hat{S}_i(t) + \phi_i^2(t))^2} \\ \frac{\partial k_i(t)}{\partial \beta_i} &= \frac{\partial}{\partial \beta_i} (D^{-1}(t) \phi_i(t)) = \frac{\partial D^{-1}(t)}{\partial \beta_i} \phi_i(t) \\ &= \frac{\hat{S}_i(t) \phi_i(t)}{(\hat{S}_i(t) + \phi_i^2(t))^2} \end{aligned}$$

Given $\frac{\partial k_i(t)}{\partial \beta_i}$ we can now derive the following update rule for $h_i(t+1)$.

$$\begin{aligned} h_i(t+1) &\approx \frac{\partial \theta_i(t+1)}{\partial \beta_i} \\ &= \frac{\partial}{\partial \beta_i} (\theta_i(t) + k_i(t) \delta(t)) \\ &\approx h_i(t) + k_i(t) \frac{\partial \delta(t)}{\partial \beta_i} + \delta(t) \frac{\partial k_i(t)}{\partial \beta_i} \\ &\approx h_i(t) - k_i(t) h_i(t) \phi_i(t) + \frac{\delta(t) \hat{S}_i(t) \phi_i(t)}{(\hat{S}_i(t) + \phi_i^2(t))^2} \\ &= h_i(t) (1 - k_i(t) \phi_i(t)) + \frac{\delta(t) \hat{S}_i(t) \phi_i(t)}{(\hat{S}_i(t) + \phi_i^2(t))^2} \end{aligned}$$

This yields a new gain adaptation algorithm called I1. Its behavior is equivalent to K1 as we can see when comparing its performance in a simple state estimation task (see Figure 1).

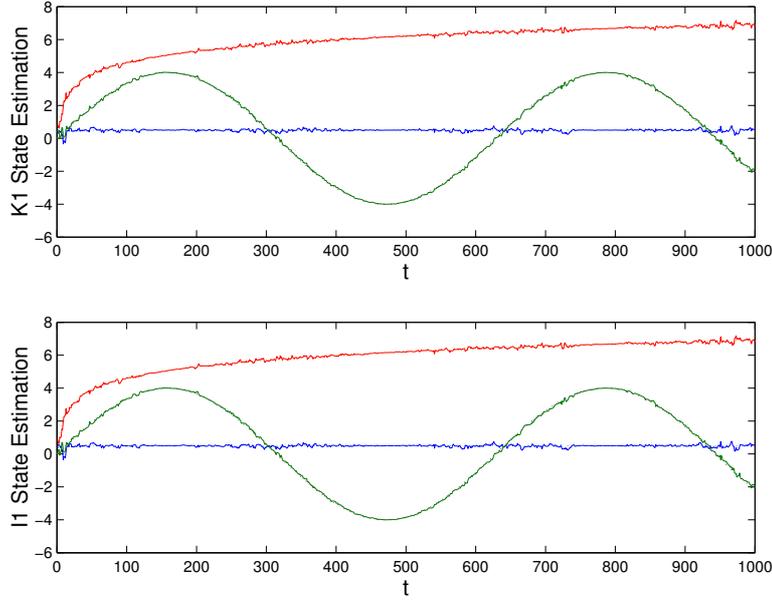


Figure 1: Application of K1 and I1 algorithms to a simple 3-state estimation problem. Here the weights θ drift according to different functions, some nonlinear. The resulting estimations of both algorithms are very nearly identical.

3.3 Extended I1 Descent

We now consider the problem of applying I1 to gradient descent. There are several options:

1. *Create an Extended Linearized I1.* It is straightforward to extend I1 to ELI1, just as it was to go from K1 to ELK1. We opt not to do so for the interests of space.
2. *Create an SMD-type algorithm based on I1.* Unfortunately, this doesn't make any sense. SMD dropped the Kalman-style gain update because the relationship between $\theta(t)$ and $\phi(t)$ is assumed to be nonlinear. Thus, the Information filter approach would be dropped as well and the algorithm would be the same as SMD.
3. *Create a gradient descent algorithm based on the Extended Information Filter (EIF).* This is the approach I take in this report.

The EIF constitutes a minimum variance estimator of some nonlinear function f . To understand how we can use I1 in this context consider the Extended Information Filter equations

$$\begin{aligned} K(t) &= J(t)(S(t) + J(t)^T R(t) J(t))^{-1} \\ \theta(t+1) &= \theta(t) + K(t)(y(t) - f(\theta(t))) \end{aligned}$$

Here J is the Jacobian of f and y is the observed value of the function. In gradient descent, we traverse the gradient until we find a stationary spot, where the gradient equals 0. To place this in the filtering framework we consider the gradient of our objective function $F(\theta, \phi)$ to be the function f we are trying to estimate and the observed value y of the function to be 0. We can now rewrite the EIF equations as

$$\begin{aligned} K(t) &= H(t)(S(t) + H(t)^T R(t) H(t))^{-1} \\ \theta(t+1) &= \theta(t) - K(t)g(t) \end{aligned}$$

where g is the gradient and H is the Hessian. Now we apply the same changes we made in going from information filtering to I1. First we diagonalize the information matrix S to get \hat{S} and the

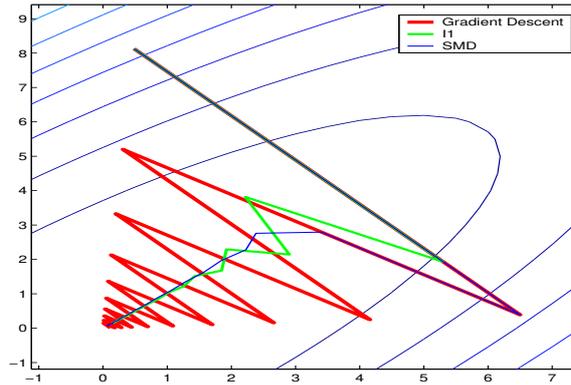


Figure 2: Three gradient descent algorithms applied to a simple quadratic bowl. Both I1 and SMD attempt to minimize oscillation of the gradient.

Hessian H into \hat{H} , then we get the following:

$$\begin{aligned} K(t) &= \hat{H}(t)(\hat{S}(t) + \hat{H}(t)^T(t)\hat{H}(t))^{-1} \\ \theta(t+1) &= \theta(t) + K(t)g(t) \end{aligned}$$

We update the diagonal $\hat{s}_{ii}(t)$ of $\hat{S}(t)$ by gradient descent similar to I1, but omit the derivation in the interests of space.

4 Conclusion and Future Work

SMD is not always stable, in many cases the gain matrix values can tend toward infinity. I1 descent is very stable, however suffers from a potentially serious drawback: it requires diagonal elements of the instantaneous Hessian. On one hand and these are not always available or too expensive to compute. On the other hand these help to properly scale the gradient instead of wasting iterations searching for the proper step-length. It remains to be seen if this form of descent can outperform SMD and stochastic gradient descent. The Linear Chain CRF software the author attempted to work with was woefully devoid of Hessian approximation (in fact deriving these values is the subject of another student's report who had no results at the moment).

A strong direction for future work is to consider a tridiagonalization of the Information matrix. The algorithm would update the parameterized elements of the tridiagonalization by gradient descent. The partials however would consist of short recurrences instead of single values. It seems likely that the computation would remain tractable as long as computing individual elements of the Hessian is relatively inexpensive.

References

- [1] J.F.G. DeFreitas, M. Niranjana, & A.H. Gee. Hierarchical Bayesian-Kalman Models for Regularisation and ARD in Sequential Learning. *Technical report*, Department of Engineering, Cambridge University, 1998.
- [2] R. S. Sutton, "Gain adaptation beats least squares?". *Proc. 7th Yale Workshop on Adaptive and Learning Systems*, 1992, pp. 161–166.
- [3] N. N. Schraudolph, Local gain adaptation in stochastic gradient descent. *Proceedings of the 9th International Conference on Artificial Neural Networks*, 1999, pp. 569–574.
- [4] N. N. Schraudolph, Online local gain adaptation for multi-layer perceptrons, *Tech. Rep. IDSIA-09-98*, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, 1998.
- [5] N. N. Schraudolph, Jin Yu, & Douglas Aberdeen. Fast Online Policy Gradient Learning with SMD Gain Vector Adaptation. *Advances in Neural Information Processing Systems*, to appear 2006.