Stochastic Gradient Descend Algorithms

Danilo Comminiello

Adaptive Algorithms and Machine Learning 2018/2019

March 27, 2019

Sapienza
Università di Roma

Lecture content highlights

- In this lecture, online learning techniques are introduced for estimating the unknown parameter vector.
- Adaptive learning algorithms are required when the solution changes in time and needs to be *updated*.
- Gradient descent method and its iterative scheme to derive the optimal solution are presented.
- A stochastic approximation of the gradient descent is then introduced.

Table of contents

1. **Introduction to Online Learning**
   - Main problems in optimal linear estimation
   - Online Learning Methods and Their Main Advantages
2. **Gradient Descent Methods**
   - Gradient Descent
   - Steepest Descent Algorithm
   - Application to the MSE Loss Function
3. **Stochastic Approximation Methods**
   - Stochastic Approximation Methods
   - The Robbins-Monro Algorithm
   - Application of the MSE Linear Estimation

1 **Introduction to Online Learning**

- Main problems in optimal linear estimation
- Online Learning Methods and Their Main Advantages

Main problems in optimal linear estimation

We introduced the notion of mean-square error (MSE) optimal linear estimation and stated the normal equations for computing the coefficients of the optimal estimator/filter.

A prerequisite for the normal equations is the knowledge of the second order statistics of the involved processes/variables in order to obtain the correlation matrix of the input and the input-output cross-correlation vector.

However, most often in practice, the correlation matrix and the cross-correlation vector have to be estimated somehow, although only a limited set of training points is available.

More important, in a number of practical applications, the underlying statistics may be time varying.

Online learning methods

In order to partially overcome the practical problems of the optimal estimation, online learning techniques are introduced.

Online learning methods are based on time iterations, in each of which a measurement set (input-output pair of observations) is available and it is used to update the current estimate.

In contrast to the so-called batch processing methods, which process the whole block of data as a single entity, online methods operate on a single data point at a time.

Therefore, online schemes do not require the training data to be known and stored in advance.

Adaptivity of online learning

Online algorithmic schemes learn the underlying statistics from the data in a time iterative fashion.

Hence, no further statistical information needs to be provided.

Working in a time iterative mode gives these methods the agility to *learn* and *track* slow time variations of the statistics of the involved processes/variables.

This is the reason why these algorithms are also known as *time-adaptive* or simply **adaptive**, since they can adapt to the needs of a changing environment.

Computational complexity advantages

Another favorable characteristic of online learning is its computational simplicity.

The required complexity for updating the estimate of the unknown parameter vector is linear with respect to the number of the unknown parameters.

Conversely, the complexity of block processing techniques can amount to prohibitive levels, for today's technology.

This is one of the major reasons that have made online learning schemes very popular in a number of practical applications.

Widespread application of online learning algorithms

Online/time-adaptive algorithms have been used *extensively* since the early 1960s in a wide range of applications including signal processing, control, and communications, among others.

More recently, the philosophy behind such schemes is gaining in popularity in the context of big data applications with massive number of data points that reside in large data bases, possibly distributed in various sites.

For such tasks, storing all the data points in the memory may not be possible, and they have to be considered one at a time.

2 Gradient Descent Methods

- Gradient Descent
- Steepest Descent Algorithm
- Application to the MSE Loss Function

Gradient descent and its Iterative Solution

The method of the **gradient descent** is one of the most widely used methods for iterative minimization of a differentiable cost function $J(\mathbf{w})$, with $\mathbf{w} \in \mathbb{R}^M$.

The iterative solution of the CF minimization evolves along the negative direction of the gradient of the CF.

Therefore, starting from an initial condition (i.c.) of the parameter vector $\mathbf{w}_{-1} = \mathbf{0}$, the optimal solution $\mathbf{w}_*$ can be obtained after a number of iterations:

$$\mathbf{w}_{-1} \rightarrow \mathbf{w}_0 \rightarrow \mathbf{w}_1 \rightarrow \ldots \rightarrow \mathbf{w}_k \rightarrow \mathbf{w}_*.$$

Iterative solution of the gradient descent

With reference to the unconstrained optimization methods of nonlinear programming [Unc17], the optimal estimator assumes this form:

$$\mathbf{w}_k = \mathbf{w}_{k-1} + \mu_k \mathbf{d}_k$$

where k is the *iteration index*, μ_k is the *step size* (that can be constant or can change at each iteration) and $\mathbf{d}_k$ is the *update direction* (or *search direction*).

The choice of the update direction is done to guarantee that

$$J\left(\mathbf{w}_k\right) < J\left(\mathbf{w}_{k-1}\right),$$

except at a minimizer $\mathbf{w}_*$.

Representation of the iterative optimization process

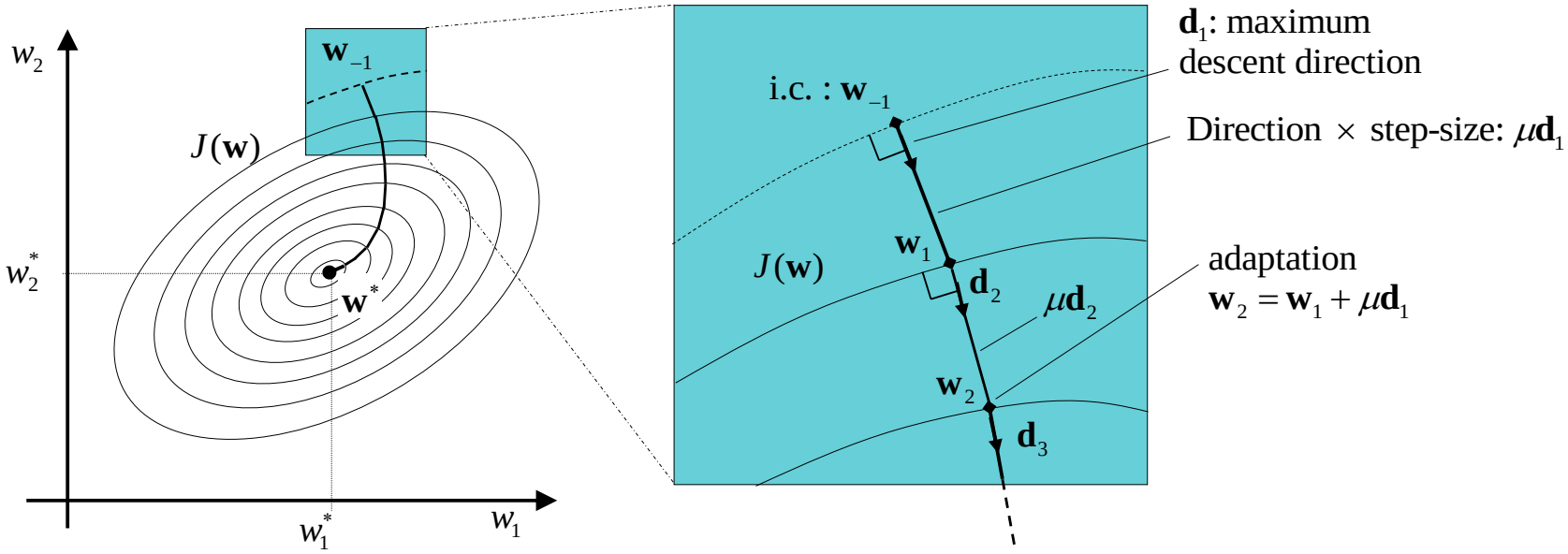

Figure 1: Qualitative evolution of the trajectory of the weights $\mathbf{w_k}$ during the optimization process towards the optimal solution $\mathbf{w}*$ for a generic two-dimensional objective function [Unc17].

First and second order algorithms

Online learning algorithms can be of first or second order:

- First-order algorithms make use of gradient information only,
- while second-order algorithms also employ the Hessian matrix, or a suitable approximation.

First-order algorithms are less computationally demanding than second-order ones, but are slower in reaching convergence.

Another possible categorization is given by the order in which the update parameters are chosen:

- Line search methods initially choose a direction, then compute the optimal update step.
- Trust region methods find an optimal update in a local approximation of the CF.

Steepest descent algorithm

The simplest iterative methods to achieve the optimal solutions are the *search methods* of the first order.

Among this class of methods, the most popular algorithm is the **steepest descent algorithm** (SDA):

$$\mathbf{w}_k = \mathbf{w}_{k-1} - \mu_k \nabla J\left(\mathbf{w}_{k-1}\right) \tag{1}$$

in which the update direction is equal to the gradient in the descent direction (or *steepest descent direction*), i.e.:

$$\mathbf{d}_k = -\nabla J\left(\mathbf{w}_{k-1}\right).$$

Properties of gradient descent methods

The gradient descent method exhibits approximately linear convergence; that is, the error between $\mathbf{w}_k$ and the true minimum *converges to zero asymptotically* in the form of a geometric series.

However, the convergence rate depends heavily on the condition number of the Hessian matrix of $J(\mathbf{w}_k)$.

For very large values of the condition number, e.g., 1000, the rate of convergence can become extremely slow.

The great advantage of the method lies in its *low* computational requirements.

Application to the MSE loss function I

Let us consider the gradient descent scheme to derive an iterative algorithm to minimize the MSE CF:

$$J(\mathbf{w}_k) = \mathsf{E}\left\{|e|^2\right\} = \mathsf{E}\left\{\left(y - \mathbf{x}^{\mathsf{T}}\mathbf{w}_{k-1}\right)^2\right\}$$
$$= \sigma_y^2 - 2\mathbf{g}^{\mathsf{T}}\mathbf{w}_{k-1} + \mathbf{w}_{k-1}^{\mathsf{T}}\mathbf{R}_x\mathbf{w}_{k-1}$$

where $\mathbf{R}_x = \mathsf{E}\left\{\mathbf{x}\mathbf{x}^{\mathsf{T}}\right\}$ and $\mathbf{g} = \mathsf{E}\left\{y\mathbf{x}\right\}$ are, respectively, the correlation matrix and the cross-correlation vector, and σ_y^2 is the variance of y.

The gradient of the CF is:

$$\nabla J(\mathbf{w}_k) = 2\mathbf{R}_x\mathbf{w}_{k-1} - 2\mathbf{g}.$$

Application to the MSE loss function II

Employing the above gradient in the update recursion (1) of the SDA and absorbing the factor 2 in the *fixed* step size, we obtain:

$$\begin{aligned} \mathbf{w}_k &= \mathbf{w}_{k-1} - \mu \left(\mathbf{R}_x \mathbf{w}_{k-1} - \mathbf{g} \right) \\ &= \mathbf{w}_{k-1} + \mu \left(\mathbf{g} - \mathbf{R}_x \mathbf{w}_{k-1} \right) \end{aligned} \tag{2}$$

It turns out that the values of the step size that guarantee convergence lie in the interval

$$0 < \mu < \frac{2}{\lambda_{\text{max}}}$$

where λ_{max} denotes the *maximum eigenvalue* of $\mathbf{R}_x$.

3 Stochastic Approximation Methods

- Stochastic Approximation Methods
- The Robbins-Monro Algorithm
- Application of the MSE Linear Estimation

Stochastic approximation methods

The solution of the normal equations, as well as the use of the gradient descent iterative scheme (for the case of the MSE), implies to having access to the second order statistics of the involved processes/variables.

However, in most of the cases, this is not known and it has to be *approximated* using a set of measurements.

To this end, since the Robbins-Monro algorithm (RMA) (1951), many **stochastic approximation methods** were developed in the literature to learn the statistics *iteratively* via the training set.

Dealing with unknown statistics

Let us consider the case of a function which is defined in terms of the expected value of another one, i.e.:

$$f(\mathbf{w}) = \mathsf{E}\{\phi(\mathbf{w}, \boldsymbol{\eta})\}, \quad \mathbf{w} \in \mathbb{R}^M$$

where $\boldsymbol{\eta}$ is a random vector of unknown statistics.

The goal is to compute a root of $f(\mathbf{w})$.

If the statistics were *known*, the expectation could be computed (in principle) and the roots can be easily obtain.

The problem emerges when the statistics is *unknown*, hence the exact form of $f(\mathbf{w})$ is not known, but only sequence of i.i.d. observations $(\eta_0, \eta_1, \ldots)$ is available.

Robbins-Monro theorem

Robbins and Monro proved that the following iterative scheme:

$$\mathbf{w}_n = \mathbf{w}_{n-1} - \mu_n \phi\left(\mathbf{w}_{n-1}, \boldsymbol{\eta}_n\right) \tag{3}$$

starting from an arbitrary initial condition, $\mathbf{w}_{-1}$, converges (in probability) to a root of $f\left(\mathbf{w}\right)$, under some general conditions and provided that

$$\sum_n \mu_n^2 < \infty, \qquad \sum_n \mu_n \to \infty. \tag{4}$$

The previous iterative scheme **does not need to use the expectation operation** for $\phi\left(\cdot, \cdot\right)$, which is computed using the current observations/measurements and the currently available estimate.

Iterative learning of the statistics

Hence, the algorithm learns both the statistics as well as the root at the same time.

Ideally, the step-size value should decrease as iterations progress, but *not* in an aggressive manner.

Thus, the algorithm remains active for (i.e., requires) a sufficient number of iterations in order to learn the solution.

If the step size tends to zero very fast, then updates are practically frozen after a few iterations, without the algorithm having acquired enough information so that to get close to the solution.

Optimization of a general cost function

In the context of optimizing a general differentiable CF of the form,

$$J(\mathbf{w}_n) = \mathsf{E}\{\mathcal{L}(\mathbf{w}_n, y[n], \mathbf{x}_n)\}$$

an iterative scheme can be mobilized to find a root of the respected gradient, i.e,

$$\nabla J(\mathbf{w}_n) = \mathsf{E}\{\nabla \mathcal{L}(\mathbf{w}_n, y[n], \mathbf{x}_n)\}$$

where the expectation is with respect to the pair $(y[n], \mathbf{x}_n)$.

Recall that, such cost functions in the Machine Learning terminology are also known as the expected risk or the expected loss.

Iterative algorithm for a general CF

Given the sequence of observations $(y[n], \mathbf{x}_n)$, $n = 0, 1, \ldots$, the recursion scheme (3) now becomes:

$$\mathbf{w}_n = \mathbf{w}_{n-1} - \mu_n \nabla \mathcal{L}(\mathbf{w}_{n-1}, y[n], \mathbf{x}_n).$$

Let us now assume, for simplicity, that the expected risk has a unique minimum, $\mathbf{w}_*$.

Then, according to Robbins-Monro theorem and using an appropriate sequence μ_n, $\mathbf{w}_n$ will converge to $\mathbf{w}_*$.

However, although this information is important, it is not by itself enough. In practice, one has to seize iterations after a finite number of steps.

Mean and covariance of the estimate

To this end, the mean $\mathrm{E}\{\mathbf{w}_n\}$ and the covariance matrix $\mathbf{C}_{\mathbf{w},n}$ of the estimate at iteration n are of interest.

If $\mu_n = \mathcal{O}(1/n)$ and assuming that iterations have brought the estimate close to the optimal value, then:

$$\mathrm{E}\{\mathbf{w}_n\} = \mathbf{w}_* + \frac{1}{n}\mathbf{c}, \qquad \mathbf{C}_{\mathbf{w},n} = \frac{1}{n}\mathbf{V} + \mathcal{O}\left(\frac{1}{n^2}\right),$$

where $\mathbf{c}$ and $\mathbf{V}$ are constants that depend on the cost function.

Both the mean as well as the standard deviations of the components follow an $\mathcal{O}(1/n)$ pattern.

Fluctuations of the estimate

The equations of $\mathrm{E}\{\mathbf{w}_n\}$ and $\mathbf{C}_{\mathbf{w},n}$ indicate that the parameter vector estimate fluctuates around the optimal value.

Such fluctuations depend on the choice of the sequence μ_n, being smaller for small values of the step-size sequence.

However, μ_n cannot be made to decrease very fast due to the convergence conditions, as discussed before.

This is the price for using the noisy version of the gradient and it is the reason that such schemes suffer from relatively slow convergence rates.

Itarative algorithm for the MSE estimation

Let us apply now the Robbins-Monro algorithm (RMA) (3) to solve for the optimal MSE linear estimator if the correlation matrix and the cross-correlation vector are unknown.

We know that the solution corresponds to the root of the gradient of the cost function, which can be written in the form:

$$\mathbf{R}_x \mathbf{w}_n - \mathbf{g} = \mathsf{E}\left\{\mathbf{x}_n \left(\mathbf{x}_n^{\mathsf{T}} \mathbf{w}_{n-1} - y\left[n\right]\right)\right\} = \mathbf{0}.$$

Given the training sequence of observations, $(y_n, \mathbf{x}_n)$, which are assumed to be i.i.d. drawn from the joint distribution of $f\left(y|\mathbf{x}\right)$, the RMA (3) becomes:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu_n \mathbf{x}_n \left(y\left[n\right] - \mathbf{x}_n^{\mathsf{T}} \mathbf{w}_{n-1}\right) \tag{5}$$

which is also known as a stochastic gradient descent scheme, or simply **stochastic gradient algorithm** (SGA).

Comparison between GDA and SGA

The previous equation converges to the optimal MSE solution provided that the two sufficient conditions in eq. (4) are satisfied.

We can compare the above SGA recursions (5) with the gradient descent one, i.e., eq. (2), thus:

$$\mathbf{w}_k = \mathbf{w}_{k-1} + \mu\left(\mathbf{g} - \mathbf{R}_x\mathbf{w}_{k-1}\right) \qquad \text{GDA}$$

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu_n\mathbf{x}_n\left(y\left[n\right] - \mathbf{x}_n^{\mathsf{T}}\mathbf{w}_{n-1}\right) \qquad \text{SGA}$$

The former equation results from the latter one by dropping out the expectation operations and using an iteration-dependent step size.

Moreover, the iteration index n in eq. (5) coincides with time index; this enables to account for time-varying environments.

Next lecture

- The main stochastic gradient descent method is presented: the least-mean square algorithm.
  - The least-mean square family will be introduced including the most significant variants of the LMS algorithm.

- We will show how to evaluate the goodness of an adaptive algorithm.
  - The main performance measures will be introduces.
  - How to conduct a performance analysis will be shown.

References I

[MIK00] D. G. Manolakis, V. K. Ingle, and S. M. Kogon, *Statistical and adaptive signal processing*, McGraw-Hill, 2000.

[Say03] A. H. Sayed, *Fundamentals of adaptive filtering*, IEEE Wiley - Interscience, 2003.

[The15] S. Theodoridis, *Machine learning. a Bayesian and optimization perspective*, Elsevier, 2015.

[Unc15] A. Uncini, *Fundamentals of adaptive signal processing*, Signals and Communication Technology, Springer International Publishing, 2015.

[Unc17] _____, *Machine learning mathematical elements: Functional analysis, nonlinear programming, stochastic processes and estimation theory*, 2017.

[Unc18] _____, *Introduction to adaptive algorithms and machine learning*, 2018.

References II

[WH60] B. Widrow and M. E. Hoff, *Adaptive switching circuits*, Conv. Rec. IRE WESCON **4** (1960), 96–104.

[Wid66] B. Widrow, *Adaptive filters I: Fundamentals*, Stanford Electron. Labs., Stanford, CA, SEL-66-126 (1966).

[WS85] B. Widrow and S. D. Stearns, *Adaptive signal processing*, Prentice Hall, 1985.

Stochastic Gradient Descend Algorithms

Adaptive Algorithms and Machine Learning

2018/2019

Danilo Comminiello

Dept. Information Engineering, Electronics and Telecommunications (DIET)
Sapienza University of Rome

http://danilocomminiello.site.uniroma1.it

danilo.comminiello@uniroma1.it