1. 背景

根据大数定理，存在随机变量 X，当我们拥有一个独立同分布的样例序列$\{x_i\}_{i=1}^N$时，则有：

$$E(X) \approx \bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\bar{x} \to E(X) \ when \ N \to \infty$$

但这种计算方式存在一个问题，即我们需要收集所有采样后才能计算平均值，会使得效率变低；如果我们能够每得到一个样例x_k，就对平均值进行更新，采取这样一种增量式的迭代算法，可以大大加快我们的效率。假设：

$$w_{k+1} = \frac{1}{k}\sum_{i=1}^{k} x_i$$

则有：

$$w_k = \frac{1}{k-1}\sum_{i=1}^{k-1} x_i$$

而w_{k+1}可以用w_k来进行表示：

$$w_{k+1} = \frac{1}{k}\sum_{i=1}^{k} x_i = \frac{1}{k}\left(\sum_{i=1}^{k-1} x_i + x_k\right) = \frac{1}{k}\big((k-1)w_k + x_k\big) = w_k - \frac{1}{k}(w_k - x_k)$$

$$w_{k+1} = w_k - \frac{1}{k}(w_k - x_k)$$

$$\begin{aligned}
w_1 &= x_1, \\
w_2 &= w_1 - \frac{1}{1}(w_1 - x_1) = x_1, \\
w_3 &= w_2 - \frac{1}{2}(w_2 - x_2) = x_1 - \frac{1}{2}(x_1 - x_2) = \frac{1}{2}(x_1 + x_2), \\
w_4 &= w_3 - \frac{1}{3}(w_3 - x_3) = \frac{1}{3}(x_1 + x_2 + x_3), \\
&\vdots \\
w_{k+1} &= \frac{1}{k}\sum_{i=1}^{k} x_i.
\end{aligned}$$

该算法可以进一步推广，形式为：

$$w_{k+1} = w_k - \alpha_k(w_k - x_k), \alpha_k > 0$$

当α_k满足一定条件时，仍能满足$w_k \to E(X)\ when\ k \to \infty$

2. Robbins-Monro 算法

Stochastic approximation(SA)：随机近似，它代表了一大类的是随机的并且迭代的算法，用于方程的求解或优化问题。它的优点在于它不需要知道目标方程或导数的表达式，Robbins-Monro 算法是 SA 领域的一个非常具有开创性的工作

我们需要求解的问题通常可以表示为：

$$g(w) = 0$$

- 优化问题：若我们想找到函数$J(w)$的最大值或最小值，我们可以将问题转换为求解该目标函数的梯度为 0，它是取得最大值或最小值的必要条件，该问题可以表示为：

$$g(w) = \nabla_w J(w) = 0$$

- 若我们想求解$h(w) = c$，可以将常数c移到左边，重新定义一个函数$g(w)$，仍能将其化为$g(w) = 0$ 的形式

RM 算法表示为：

$$w_{k+1} = w_k - a_k \tilde{g}(w_k, \eta_k)$$

- 我们需要求解的问题是$g(w) = 0$，最优解为w^*
- w_k表示对解的第 k 次估计
- $\tilde{g}(w_k, \eta_k) = g(w_k) + \eta_k$是第 k 次加上噪音的观测值
- 我们可以不需要知道$g(w)$的表达式，该算法依赖于输入序列$\{w_k\}$和带噪音的输出序列$\{\tilde{g}(w_k, \eta_k)\}$

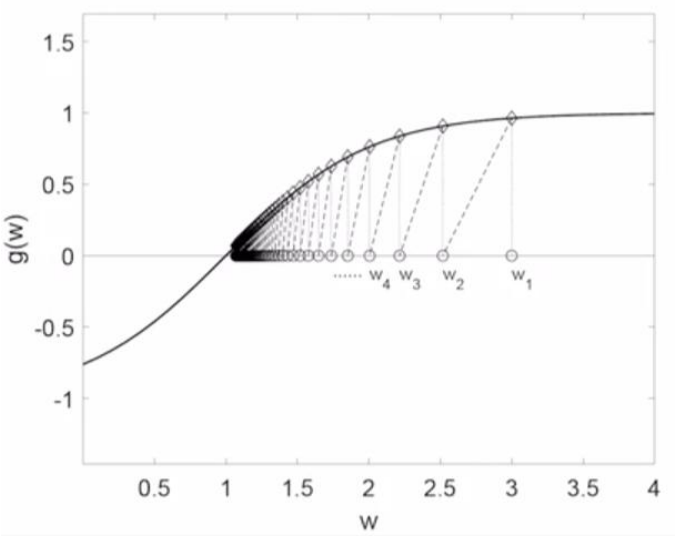
例子：

- 要求解的问题为$g(w) = tanh(w-1) = 0$
- 该方程真正的解为$w^* = 1$
- 初始参数为$w_1 = 3, a_k = \frac{1}{k}, \eta_k = 0$，即在本例中不设噪声

使用 RM 算法求解该方程的过程为：

$$w_{k+1} = w_k - a_k g(w_k)$$

最后的结果如图所示：



为什么一定w_{k+1}优于w_k？

- 当$w_k > w^*$时，$g(w_k) > 0$，则$w_{k+1} = w_k - a_k g(w_k) < w_k$，$w_{k+1}$比$w_k$更接近$w^*$
- 当$w_k < w^*$时，$g(w_k) < 0$，则$w_{k+1} = w_k - a_k g(w_k) > w_k$，$w_{k+1}$比$w_k$更接近$w^*$

3. Stochastic Gradient Descent(SGD)：随机梯度下降

SGD 是一种特殊的 RM 算法，它要解决的问题可以表示为：

$$\min_w J(w) = E[f(w, X)]$$

- w是待优化的参数
- X 是一个随机变量，与期望相关

求解方法：

（1） gradient descend(GD)：梯度下降

$$w_{k+1} = w_k - \alpha_k \nabla_w J(w_k) = w_k - \alpha_k \nabla_w E[f(w_k, X)] = w_k - \alpha_k E[\nabla_w f(w_k, X)]$$

因为我们的目标是最小化函数，所以采取梯度下降法（若要最大化函数，则采取梯度上升法），沿着梯度方向对参数进行更新，更新速度会较快。但我们如果在没有模型的情况下，如何求得期望值？根据数据来进行求解，这就引出了下一种方法：batch gradient descend(BGD)

（2） batch gradient descend(BGD)：批量梯度下降

它的思想是根据大数定理，采样多次求平均值来求得期望：

$$E[\nabla_w f(w_k, X)] \approx \frac{1}{n}\sum_{i=1}^{n} \nabla_w f(w_k, x_i)$$

$$w_{k+1} = w_k - \alpha_k \frac{1}{n}\sum_{i=1}^{n} \nabla_w f(w_k, x_i)$$

但是采用这种方法在每次迭代过程中需要大量采样

（3） stochastic gradient descend(SGD)：随机梯度下降

采用 stochastic gradient（随机梯度）替换 true gradient（真实梯度）$E[\nabla_w f(w_k, X)]$：

$$w_{k+1} = w_k - \alpha_k \nabla_w f(w_k, x_k)$$

即每采样一次，就对参数进行更新一次。在 BGD 中，若 n=1，则算法更新为 SGD

例子：

求解问题为：

$$\min_w J(w) = E[f(w, X)] = E\left[\frac{1}{2}\|w - X\|^2\right]$$

则有：

$$f(w, X) = \frac{1}{2}\|w - X\|^2 \qquad \nabla_w f(w, X) = w - X$$

- 证明该问题的最优解为$w^* = E(X)$

要使$J(w)$取得最小值，则其导数等于 0，而$\nabla_w J(w) = \nabla_w E[f(w, X)] = E[\nabla_w f(w, X)] = E(w - X) = w - E(X) = 0$，则得到$w = E(X)$

- 使用 GD 算法进行求解:

$$\begin{aligned} w_{k+1} &= w_k - \alpha_k \nabla_w J(w_k) \\ &= w_k - \alpha_k E[\nabla_w f(w_k, X)] \\ &= w_k - \alpha_k E[w_k - X] \end{aligned}$$

- 使用 SGD 算法进行求解:

$$\begin{aligned} w_{k+1} &= w_k - \alpha_k \nabla_w f(w_k, x_k) \\ &= w_k - \alpha_k (w_k - x_k) \end{aligned}$$

SGD 的收敛过程是否是随机的?

用δ_k来表示 stochastic gradient 与 true gradient 之间的相对误差, 它的计算方法为:

$$\delta_k = \frac{|\nabla_w f(w_k, x_k) - E[\nabla_w f(w_k, X)]|}{|E[\nabla_w f(w_k, X)]|}$$

又有$E[\nabla_k f(w^*, X)] = 0$，则:

$$\delta_k = \frac{|\nabla_w f(w_k, x_k) - E[\nabla_w f(w_k, X)]|}{|E[\nabla_w f(w_k, X)] - E[\nabla_w f(w^*, X)]|}$$

采用拉格朗日中值定理，存在$\widetilde{w_k} \in [w_k, w^*]$，有:

$$E[\nabla_w^2 f(\widetilde{w_k}, X)(w_k - w^*)] = E[\nabla_w f(w_k, X)] - E[\nabla_w f(w^*, X)]$$

则有:

$$\delta_k = \frac{|\nabla_w f(w_k, x_k) - E[\nabla_w f(w_k, X)]|}{\left|E[\nabla_w^2 f(\widetilde{w_k}, X)(w_k - w^*)]\right|}$$

假设f为严格凸函数，则有:

$$\nabla_w^2 f \geq c > 0$$

则有:

$$\left|E[\nabla_w^2 f(\widetilde{w_k}, X)(w_k - w^*)]\right| = \left|E[\nabla_w^2 f(\widetilde{w_k}, X)](w_k - w^*)\right|$$

$$= \left|E[\nabla_w^2 f(\widetilde{w_k}, X)]\right||(w_k - w^*)| \geq c|w_k - w^*|$$

则有：

$$\delta_k \leq \frac{|\nabla_w f(w_k, x_k) - E[\nabla_w f(w_k, X)]|}{c|w_k - w^*|}$$

其中分子表示 stochastic gradient 与 true gradient 的绝对误差，分母表示到最优解 w^*的距离，则有：

- 当$|w_k - w^*|$较大时，表示当前解与最优解相差较大，此时相对误差较小，SGD 的表现与 GD 相差不大
- 当$|w_k - w^*|$较小时，表示当前解与最优解相差较小，此时相对误差较大，随机性较强，但这是一种好的表现

SGD 算法一定是用来求解包含随机变量和期望的问题的，考虑以下一个问题：

要求解的问题为：

$$\min_w J(w) = \frac{1}{n}\sum_{i=1}^{n} f(w, x_i)$$

w是待优化的参数，$\{x_i\}_{i=1}^n$是一组实数，并不是某随机变量的采样。采用梯度下降法对该问题求解，描述为：

$$w_{k+1} = w_k - \alpha_k \nabla_w J(w_k) = w_k - \alpha_k \frac{1}{n}\sum_{i=1}^{n} \nabla_w f(w_k, x_i)$$

假设我们每次只能从集合$\{x_i\}_{i=1}^n$中拿到一个实数x_k，则该求解步骤描述为：

$$w_{k+1} = w_k - \alpha_k \nabla_w f(w_k, x_k)$$

它的求解步骤与 SGD 非常类似，这个算法是 SGD 算法吗？

我们在集合$\{x_i\}_{i=1}^n$上定义一个随机变量 X，它是服从均匀分布的，即：

$$p(X = x_i) = \frac{1}{n}$$

则原问题转换为：

$$\min_w J(w) = \frac{1}{n}\sum_{i=1}^{n} f(w, x_i) = E[f(w, X)]$$

又由于 X 服从均匀分布，每次从集合中取数都是随机的，即对 X 采样随机。故当引入随机变量 X 后，它是一个 SGD 算法

4. Mini-Batch Gradient Descend：小批量梯度下降

假设我们的问题还是：

$$\min_w J(w) = E[f(w, X)]$$

并且我们有一组对随机变量 X 独立同分布的采样$\{x_i\}_{i=1}^n$，则 BGD、MBGD 和 SGD 的求解分别表示为：

$$w_{k+1} = w_k - \alpha_k \frac{1}{n}\sum_{i=1}^{n} \nabla_w f(w_k, x_i), \qquad BGD$$

$$w_{k+1} = w_k - \alpha_k \frac{1}{m}\sum_{j\in B_k} \nabla_w f(w_k, x_j), \qquad MBGD$$

$$w_{k+1} = w_k - \alpha_k \nabla_w f(w_k, x_k), \qquad SGD$$

在 BGD 算法中，在每一步迭代时都用到了所有样例。当 n 足够大时，即样例足够多时，$\frac{1}{n}\sum_{i=1}^{n} \nabla_w f(w_k, x_i)$非常接近 true gradient

在 MBGD 算法中，B_k是集合$\{1, 2, \ldots, n\}$的一个大小为$|B_k|$的子集，每次迭代用到了部分样例

在 SGD 算法中，每次随机从样例集合中采样一个数据

- 相比较 SGD 算法来说，MBGD 的随机性弱于 SGD，因为它每次迭代采用了不止一个的样例
- 相比较 BGD 算法来说，MBGD 不需要在每次迭代都使用所有样例，更近高效

灵活

- 当 m=1 时，MBGD 算法就变成了 SGD 算法
- 当 m=n 时，MBGD 算法严格意义上来说没有变成 BGD 算法，因为 MBGD 算法是随机取样 n 个样例，它不能保证每个样例都被取到且只被取一次，而 BGD 采用了所有样例

例子：

有随机变量 X，以及对 X 的一组随机采样$\{x_i\}_{i=1}^n$，我们要求解的问题为：

$$\min_w J(w) = \frac{1}{2n}\sum_{i=1}^{n} \|w - x_i\|^2$$

采用 BGD 求解步骤表示如下：

$$w_{k+1} = w_k - \alpha_k \frac{1}{n}\sum_{i=1}^{n} (w_k - x_i) = w_k - \alpha_k (w_k - \bar{x})$$

采用 MBGD 求解步骤表示如下：

$$w_{k+1} = w_k - \alpha_k \frac{1}{n}\sum_{j\in B_k} \left(w_k - x_j\right) = w_k - \alpha_k \left(w_k - \bar{x}^{(m)}\right)$$

$$\bar{x}^{(m)} = \frac{1}{m}\sum_{j\in B_k} x_j$$

采用 SGD 求解步骤表示如下：

$$w_{k+1} = w_k - \alpha_k (w_k - x_k)$$

例 2：给定一个 20*20 大小的空间，中心为(0, 0)，随机生成 100 个点，求坐标的期望值。分别采用不同 batch 大小的算法进行求解，结果如图所示：