第 6 章第 5 节 随机梯度下降法

SMaLL

[1] 中国石油大学（华东）
SMaLL 课题组
small.sem.upc.edu.cn
liangxijunsd@163.com

2023

第 6 章第 5 节 随机梯度下降法

随机梯度 (SG) 方法综述 I

最形式化优化问题陈述

在指定的函数族 $\mathcal{H}$ 中找到决策/预测函数 $h(\cdot)$, 通过用给定的损失函数 $\ell(\cdot)$ 优化 $\mathbb{E}[\ell(h)]$.

总结: 这是一个一般预测函数族 $\mathcal{H}$ 上的变分优化问题.

Solution: 假设 $h(\cdot)$ 具有固定形式，且由 $w \in \mathbb{R}^d$ 参数化.

假设预测函数族为

$$\mathcal{H} := \{h(x; w) : w \in \mathbb{R}^d\},$$

并定义损失函数为

$$\ell : \mathbb{R}^{d_y} \times \mathbb{R}^{d_y} \to \mathbb{R},$$

假设输入输出对为 (x, y), 刻画预测值 $h(x; w)$ 和真实标签 y 的损失 $\ell(h(x; w), y)$.

随机梯度 (SG) 方法综述 II

- 经验风险

$$\mathcal{R}_n(w) = \frac{1}{n}\sum_{i=1}^{n} \ell(h(x_i; w), y_i)$$

- 结构风险

$$\mathcal{R}_n(w) + \lambda\Omega(w)$$

即

$$\frac{1}{n}\sum_{i=1}^{n} \ell(h(x_i; w), y_i) + \frac{\lambda}{2} w^T w$$

简化的优化问题

将单个样本表示为 $\xi = (x, y)$
给定 (w, ξ) 的损失, i.e., $\ell(h(w, x), y)$ as $f(w; \xi)$
样本集 $\{(x_i, y_i)\}_{i=1}^n$ 表示为 $\{\xi_{[i]}\}_{i=1}^n$
参数向量 w 相对于第 i 个样本所引起的损失为 $f_i(w) := f(w; \xi_{[i]})$
通过上述简化的符号，期望风险和经验风险可以公式化为

- 期望风险

$$\mathcal{R}(w) := \mathbb{E}[f(w; \xi)]$$

- 经验风险

$$\mathcal{R}_n(w) := \frac{1}{n} \sum_{i=1}^{n} f_i(w)$$

随机优化方法与批处理方法

$$\min_{w\in\mathbb{R}^d} F(w) := \frac{1}{n}\sum_{i=1}^{n} f_i(w)$$

最速下降法

$$w_{k+1} := w_k - \frac{\alpha_k}{n}\sum_{i=1}^{n}\nabla f_i(w_k)$$

随机梯度法

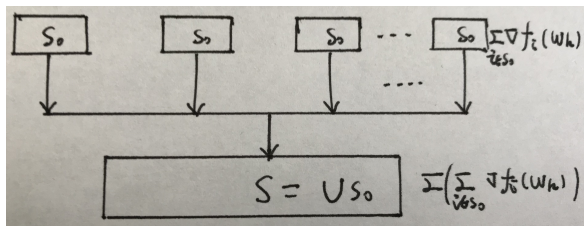
$$w_{k+1} := w_k - \alpha_k \nabla f_{i_k}(w_k)$$

其中 i_k 从 $\{1,\cdots,n\}$ 中随机地选取.

随机方法的动机 I

直观动机: 与批处理方法相比 SG 法利用样本信息的效率更高

考虑: 训练集 $\mathbf{S}$ 由集合 $\mathbf{S}_{\text{sub}}$ 重复 10 次组成

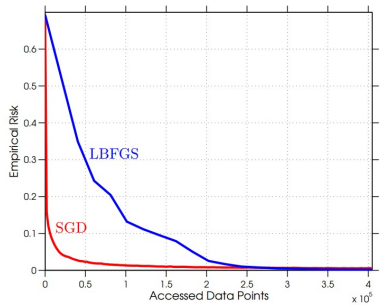


- 批处理方法在数据集 $\mathbf{S}$ 上优化经验风险 $\rightarrow$ 每次迭代的计算代价将比只在一个 $\mathbf{S}_{\text{sub}}$ 上计算经验风险要高 10 倍

随机方法的动机 II

实践动机

对比固定步长的 SG 和 L-BFGS 在二分类问题上的性能, 目标函数使用 Logistic 损失, 使用 RCV1 数据集.



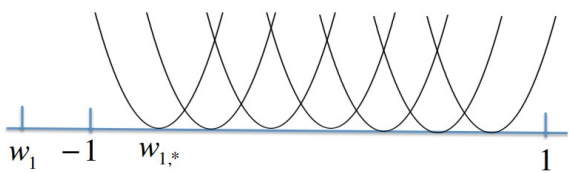
- SG 在迭代的初始阶段可以**快速改进**目标函数值
- 经过一两个轮次的迭代后，目标函数值改进将**非常缓慢**

随机方法的动机 III

实践动机

假设经验风险 $R_n(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{w})$ 中

- 每个 f_i 均是最小值为零的凸二次函数
- 每个 f_i 的最优解 $\mathbf{w}_{i,*}$ 均匀分布在区间 $[0, 1]$ 上



随机方法的动机 IV

理论动机

当经验损失 R_n 强凸时 $\rightarrow$ 存在 $\rho \in [0,1]$, 对 $k \in \mathbf{N}$ 都满足

- 批处理梯度法

$$R_n(\mathbf{w}_k) - R_n^* \leqslant O\left(\rho^k\right) \qquad \text{线性收敛}$$

$\Rightarrow$ 求得 ϵ 最优解所需的计算量与 $n\log\frac{1}{\epsilon}$ 成比例

- 随机梯度法

$$\mathbb{E}[F(w_k) - F^*] = \mathcal{O}(\frac{1}{k}) \qquad \text{次线性收敛}$$

$\Rightarrow$ 为获得 ε 的最优解所需的总计算量与 $\frac{1}{\varepsilon}$ 成正比

备注:

. 对于不太大的 n 和 ϵ 值 $\rightarrow$ $\frac{1}{\epsilon}$ 可能大于 $n\log\frac{1}{\epsilon}$

. 对于 n 很大时 (大数据场景) $\rightarrow$ $n\log(\frac{1}{\varepsilon}) > \frac{1}{\varepsilon}$

$\Rightarrow$ SG 凭借其更高的计算效率而备受青睐

小批量随机梯度下降 mini-batch SG

挑战

- 经多个轮次迭代后批处理方法的性能可能最终超过随机方法
- SG 容易陷入局部最小值点或者鞍点

⇒ 结合批处理和随机算法的最佳算法 (小批量随机梯度下降)

$$w_{k+1} := w_k - \frac{\alpha_k}{|S_k|} \sum_{i \in S_k} \nabla f_i(w_k)$$

1. 降低了参数更新的方差 → 收敛更稳定
2. 在计算小批量梯度时可以利用一定程度的并行化

⇒ 在实践中被广泛使用

第 6 章第 5 节 随机梯度下降法

SG 算法伪代码

随机梯度 (SG) 方法

- 选择初始向量 w_1
- **for** $k = 1, 2, \cdots$ **do**
- 　　生成 ξ_k
- 　　计算 $g(w_k; \xi_k)$
- 　　选择步长 $\alpha_k > 0$
- 　　迭代更新: $w_{k+1} \leftarrow w_k - \alpha_k g(w_k; \xi_k)$
- **end for**

SG 算法的收敛性分析

SG 算法的收敛性主要基于

i) 目标函数的光滑化假设

ii) 随机向量 $g(w_k;\xi_k)$ 的一阶矩、二阶矩假设

假设 1: 目标函数 F 连续可微，且 F 的梯度 Lipschitz 连续.

$$\|\nabla F(w)-\nabla F(\overline{w})\|_2 \le L\|w-\overline{w}\|_2 \ for\ all\ \{w,\overline{w}\}\subset\mathbb{R}^d.$$

$\rightarrow$ 保证 F 的梯度相对于参数向量的变化率有界

$\Rightarrow$

$$F(w)\leqslant F(\overline{w})+\nabla F(\overline{w})^{\mathrm{T}}(w-\overline{w})+\frac{1}{2}L\|w-\overline{w}\|_2^2,\forall w,\overline{w}\subset\mathbb{R}^d$$

SG 算法的收敛性分析

假设 2(一阶矩和二阶矩假设): 目标函数和 SG 算法满足

(a) 迭代序列 $\{w_k\}$ 包含在一个开集中，F 在开集上有下界 F_*.

(b) 对于所有的 k，有 $\mu_G \geq \mu > 0$

$$\nabla F(w_k)^T \mathbb{E}_{\xi_k}[g(w_k;\xi_k)] \geq \mu \|\nabla F(w_k)\|_2^2 \text{ 且}$$

$$\|\mathbb{E}_{\xi_k}[g(w_k;\xi_k)]\|_2 \leq \mu_G \|\nabla F(w_k)\|_2.$$

(c) 存在常数 $M > 0$ 和 $M_V \geq 0$ 使得对于所有的 k

$$\mathbb{V}_{\xi_k}[g(w_k;\xi_k)] \leq M + M_V \|\nabla F(w_k)\|_2^2,$$

其中

$$\mathbb{V}_{\xi_k}[g(w_k;\xi_k)] := \mathbb{E}_{\xi_k}[\|g(w_k;\xi_k)\|_2^2] - \|\mathbb{E}_{\xi_k}[g(w_k;\xi_k)]\|_2^2.$$

$\Rightarrow$ 要求 $g(w_k,\xi_k)$ 的二阶矩满足

$$\mathbb{E}_{\xi_k}[\|g(w_k,\xi_k)\|_2^2] \leq M + M_G \|\nabla F(w_k)\|_2^2$$

其中 $M_G := M_V + \mu_G^2$.

Remarks

- 假设 2(a) 仅要求目标函数在算法探索的区域下方有界.
- 假设 2(b) 指出，在预期中，向量 g 是一个足够下降的方向。如果 n 是 F 的无偏估计，则它与 k 立即成立假设 2(b) 指出，预期向量 $-g(w_k;\xi_k)$ 是一个合适的下降方向. 如果 $g(w_k;\xi_k)$ 是 $\nabla F(w_k)$ 的无偏估计，则它与 $\mu_G = \mu = 1$ 成立.
- 对于假设 2(c), 记作

$$\mathbb{V}_{\xi_k}[g(w_k,\xi_k)] := \mathbb{E}_{\xi_k}[\|g(w_k,\xi_k)\|_2^2] - \|\mathbb{E}_{\xi_k}[g(w_k,\xi_k)]\|_2^2,$$

它认为

$$\mathbb{E}_{\xi_k}[\|g(w_k,\xi_k)\|_2^2] \leq M + M_G\|\nabla F(w_k)\|_2^2$$

其中 $M_G := M_V + \mu_G^2$.

SG 算法的收敛性分析

假设 3(强凸性)

目标函数 $F: \mathbf{R}^d \to \mathbf{R}$ 是强凸的, 即存在常数 $c > 0$ 使

$$F(\overline{\boldsymbol{w}}) \geqslant F(\boldsymbol{w}) + \nabla F(\boldsymbol{w})^{\mathrm{T}}(\overline{\boldsymbol{w}} - \boldsymbol{w}) + \frac{1}{2}c\|\overline{\boldsymbol{w}} - \boldsymbol{w}\|_2^2, \forall(\overline{\boldsymbol{w}}, \boldsymbol{w}) \in \mathbf{R}^d \times \mathbf{R}^d$$

由此, F 存在唯一最优值, 记为 $F_* := F(\boldsymbol{w}_*)$, 其中 $\boldsymbol{w}_* \in \mathbf{R}^d$

$\to$ 用目标函数在该点的梯度 ℓ_2 范数的平方来估计最优间隙的界

$$2c\,(F(\boldsymbol{w}) - F_*) \leqslant \|\nabla F(\boldsymbol{w})\|_2^2, w \in \mathbf{R}^d$$

SG 算法的收敛性分析

定理 1 (强凸目标函数, 固定步长)

在假设 *1-3* 成立的条件下, 记 F_* 为目标函数的最优值, 对于所有 $k \in \mathbf{N}$, 设步长 $\alpha_k = \bar{\alpha}$ 满足

$$0 < \bar{\alpha} \leqslant \frac{\mu}{LM_G} \tag{1}$$

则对所有的 $k \in \mathbf{N}$, 最优间隙的期望满足以下不等式:

$$E[F(\mathbf{w}_k) - F_*] \leqslant \frac{\bar{\alpha}LM}{2c\mu} + (1 - \bar{\alpha}c\mu)^{k-1}\left(F(\mathbf{w}_1) - F_* - \frac{\bar{\alpha}LM}{2c\mu}\right) \overset{k\to\infty}{\longrightarrow} \frac{\bar{\alpha}LM}{2c\mu} \tag{2}$$

- 如果 $g(w_k; \xi_k)$ 是**无偏估计** $\rightarrow \mu = 1$
- 且如果 $g(w_k; \xi_k)$ 中**没有噪声**
  $\rightarrow$ 可假设 $M_G = 1$, 简化为 $\alpha \in (0, \frac{1}{L})$(FG 经典步长要求)

SG 算法的收敛性分析

- 在 Robbins 和 Monro 的开创性作品中，步长采用:

$$\sum_{i=1}^{\infty} \alpha_k = \infty, \sum_{i=1}^{\infty} \alpha_k^2 < \infty.$$

定理 2 (强凸性，步长递减)

在假设 *1-3* 成立的条件下 *SG* 方法的步长满足

$$\alpha_k = \frac{\beta}{\gamma + k} \text{ 对于 } \beta > \frac{1}{c\mu} \text{ 且 } \gamma > 0 \text{ 使其满足 } \alpha_1 \leq \frac{c}{LM_G}.$$

那么，

$$\mathbb{E}[F(w_k) - F_*] \leq \frac{\nu}{\gamma + k}.$$

其中

$$\nu := \max\left\{ \frac{\beta^2 LM}{2(\beta c\mu - 1)}, (\gamma + 1)(F(w_1) - F_*) \right\}.$$

降噪方法

梯度估计中的噪声会阻止目标函数值的收敛

- 在固定步长下的解决方案 $\mathbb{E}[F(w_k) - F_*] \overset{K\to\infty}{\leq} \frac{\bar{\alpha}LM}{2c\mu}$
- 步长递减序列下的次线性收敛速度

降噪方法

开发具有降噪能力的方法

- 动态采样方法：逐渐增加梯度计算中使用的小批量大小
- 梯度聚合方法：通过存储与先前迭代中使用的样本相对应的梯度估计，提高搜索方向的质量
- 迭代平均方法：在优化过程中保持迭代 w_k 的平均值

第 6 章第 5 节 随机梯度下降法

降噪-梯度聚合方法

Q: 是否可以通过重复使用或修改先前计算的信息来降低方差?

- 如果当前迭代没有与以前的迭代相距较大 → 来自以前迭代的随机梯度信息可能仍然有用
- 如果在存储中保持索引梯度估计 → 可以在收集新信息时修改特定的估计

⇒ 梯度聚合的概念，如 SVRG (随机方差约减梯度)、SAGA（随机平均梯度算法）

SVRG (随机方差约减梯度)

SVRG 属于循环法

在每个循环的开始有一个初始值 $\boldsymbol{w}_k$, 并计算出一个批量梯度

$$\nabla R_n\left(\boldsymbol{w}_k\right)=\frac{1}{n}\sum_{i=1}^{n}\nabla f_i\left(\boldsymbol{w}_k\right),$$

并初始化 $\tilde{\boldsymbol{w}}_1 \leftarrow \boldsymbol{w}_k$, 然后执行一组由 j 作为索引的 m 次内部迭代更新 $\tilde{\boldsymbol{w}}_{j+1} \leftarrow \tilde{\boldsymbol{w}}_j-\alpha\tilde{\boldsymbol{g}}_j$, 其中

$$\tilde{\boldsymbol{g}}_j \leftarrow \nabla f_{i_j}\left(\tilde{\boldsymbol{w}}_j\right)-\left(\nabla f_{i_j}\left(\boldsymbol{w}_k\right)-\nabla R_n\left(\boldsymbol{w}_k\right)\right)$$

$i_j \in\{1, \cdots, n\}$ 是随机选择的指标

- 关于 $\tilde{\boldsymbol{g}}_j \leftarrow \nabla f_{i_j}\left(\tilde{\boldsymbol{w}}_j\right)-\left(\nabla f_{i_j}\left(\boldsymbol{w}_k\right)-\nabla R_n\left(\boldsymbol{w}_k\right)\right)$ 的简单解释:
  - $\mathbb{E}_{i_j} \nabla f_{i_j}\left(w_k\right)=\nabla R_n\left(w_k\right)$
  - $\tilde{g}_j$ 代表了 $\nabla R_n\left(\tilde{w}_j\right)$ 的一个无偏估计量，但其方差预期会比简单选择 SG $\left(\tilde{g}_j=\nabla f_{i_j}\left(\tilde{w}_j\right)\right)$ 小很多

最小化 R_n 的 SVRG 方法

初始值 $w_1 \in \mathbb{R}^d$, 步长 $\alpha > 0$, m 次迭代
for $k = 1, 2, \cdots,$ **do**
计算出一个批量梯度 $\nabla R_n(w_k)$
初始化 $\tilde{w}_1 \leftarrow w_k$
for $j = 1, 2, \cdots, m$ **do**
从 $\{1, \cdots, n\}$ 随机选择 i_j
$\tilde{g}_j \leftarrow \nabla f_{i_j}(\tilde{w}_j) - \left(\nabla f_{i_j}(w_k) - \nabla R_n(w_k)\right)$
$\tilde{w}_{j+1} \leftarrow \tilde{w}_j - \alpha \tilde{g}_j$
end for
选项 (a): 取 $w_{k+1} = \tilde{w}_{m+1}$
选项 (b): 取 $w_{k+1} = \frac{1}{m}\sum_{j=1}^{m} \tilde{w}_{j+1}$
选项 (c): 从 $1, \cdots, m$ 选择 j ， 取 $w_{k+1} = \tilde{w}_{j+1}$
end for

SVRG (随机方差约减梯度)

对于选项 (b) 和 (c), 在最小化强凸目标 R_n 时, 算法可以实现**线性收敛速度**。选择合适的步长 α 和内循环次数 m, 使下式

$$\rho := \frac{1}{1-2\alpha L}\left(\frac{1}{mc\alpha} + 2L\alpha\right) < 1,$$

并假设算法的当前迭代点为 $\boldsymbol{w}_k$, 那么, 有如下不等式成立

$$\mathbb{E}\left[R_n\left(\boldsymbol{w}_{k+1}\right) - R_n\left(\boldsymbol{w}_*\right)\right] \leqslant \rho\mathbb{E}\left[R_n\left(\boldsymbol{w}_k\right) - R_n\left(\boldsymbol{w}_*\right)\right]$$

其中, 期望是对内循环中的随机变量取的。

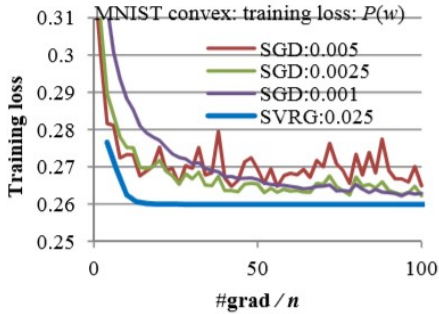
这里得到的线性收敛率适用于外部迭代点列 $\{w_k\}$, 其中

- 从 w_k 到 w_{k+1} 的每一步需要计算 $2m+n$ 个样本梯度

$\Rightarrow$ SVRG 的一次迭代比 SG 的一次迭代的计算量要多得多, 实际上与全梯度方法一次迭代的计算量相当

SVRG (随机方差约减梯度)

与 SG 对比



- 在实践中，如果需要高训练精度，则与 SG 相比，SVRG 在某些应用中似乎非常有效.

SAGA（随机平均梯度算法）

SAGA 方法不涉及周期性操作, 除在初始点以外无需计算全梯度
在每次迭代中, 它计算一个随机向量 $\boldsymbol{g}_k$

- 存储目标函数中各 f_i 在某迭代点的梯度 $\nabla f_i\left(\boldsymbol{w}_{[i]}\right)$
  - 其中 $\boldsymbol{w}_{[i]}$ 表示最近一次计算 f_i 的梯度时使用的迭代点
- 在第 k 个迭代步, 随机选择一个整数 $j \in \{1, \cdots, n\}$, 计算 $\boldsymbol{g}_k$

$$\boldsymbol{g}_k \leftarrow \nabla f_j\left(\boldsymbol{w}_k\right) - \nabla f_j\left(\boldsymbol{w}_{[j]}\right) + \frac{1}{n}\sum_{i=1}^{n} \nabla f_i\left(\boldsymbol{w}_{[i]}\right)$$

若将 $\boldsymbol{g}_k$ 关于 $j \in \{1, \cdots, n\}$ 取期望, 则有 $\mathbb{E}\left[\boldsymbol{g}_k\right] = \nabla R_n\left(\boldsymbol{w}_k\right)$
$\Rightarrow$ SAGA 计算的随机向量是梯度的无偏估计, 但其方差小于基本 SG 算法所产生的方差

最小化 R_n 的 SARA 方法

选择一个初始值 $\boldsymbol{w}_1 \in \mathbf{R}^d$, 步长 $\alpha > 0$
for $i \in \{1, \cdots, n\}$ **do**
计算梯度 $\nabla f_i(\boldsymbol{w}_1)$
储存 $\nabla f_i\left(\boldsymbol{w}_{[i]}\right) \leftarrow \nabla f_i(\boldsymbol{w}_1)$
end for
for $k = 1, 2, \cdots$ **do**
从 $\{1, \cdots, n\}$ 中随机选择一个整数 j
计算梯度 $\nabla f_j(\boldsymbol{w}_k)$
令 $\boldsymbol{g}_k \leftarrow \nabla f_j(\boldsymbol{w}_k) - \nabla f_j\left(\boldsymbol{w}_{[j]}\right) + \frac{1}{n}\sum_{i=1}^{n} \nabla f_i\left(\boldsymbol{w}_{[i]}\right)$
储存 $\nabla f_j\left(\boldsymbol{w}_{[j]}\right) \leftarrow \nabla f_j(\boldsymbol{w}_k)$
令 $\boldsymbol{w}_{k+1} \leftarrow \boldsymbol{w}_k - \alpha \boldsymbol{g}_k$
end for

SAGA（随机平均梯度算法）

当最小化强凸 R_n 时，该方法可以实现线性收敛速度

具体来说，取 $\alpha = 1/(2(cn + L))$, 可得如下估计

$$\mathbb{E}\left[\|w_k - w_*\|_2^2\right] \leq \left(1 - \frac{c}{2(cn+L)}\right)^k \left(\|w_1 - w_*\|_2^2 + \frac{nD}{cn+L}\right)$$

$$\text{其中} D := R_n(w_1) - \nabla R_n(w_*)^T (w_1 - w_*) - R_n(w_*)$$

SG vs 梯度聚合方法

尽管本小节中所介绍的梯度聚合方法比 SG 具有更快的收敛速度，这并不能表明它们明显优于 SG

- SG 的计算时间为

$$T(n, \epsilon) \sim \kappa^2/\epsilon$$

  其中 $\kappa := L/c$

- SVRG 和 SAGA 的计算时间为

$$\mathcal{T}(n, \epsilon) \sim (n + \kappa) \log(1/\epsilon)$$

  随着样本点数量 n 的增加而增加

$\Rightarrow$

* 对于非常大的 n，梯度聚合方法的时间复杂度与批处理相似
* 如果 κ 接近 1 , 那么 SG 显然会更有效
* 当 $\kappa \gg n$ 时, 梯度聚合方法会表现的更好

第 6 章第 5 节 随机梯度下降法

二阶方法

动机

- 一阶方法，例如 SG 和全梯度法，它们不是尺度不变的.
- 一阶方法的每次迭代通过计算目标函数的二阶泰勒近似的极小值点作为后续迭代点.
- 目标函数的高度非线性和病态条件的不利影响.

二阶方法的动机 I

动机.

尺度不变：

* 考虑最小化连续可微函数 $F: R^d \to R$,

$$w_{k+1} \leftarrow w_k - \alpha_k \nabla F(w_k). \tag{3}$$

* 交替迭代:

$$\min_{\bar{w}} \ F(B\bar{w})$$

假设 B 为对称正定矩阵. 全梯度迭代格式:

$$\bar{w}_{k+1} \leftarrow \bar{w}_k - \alpha_k B \nabla F(B\bar{w}_k),$$

两端乘以 B 并由定义 $w_k = B\bar{w}_k$, 可得如下迭代

$$w_{k+1} \leftarrow w_k - \alpha_k B^2 \nabla F(w_k). \tag{4}$$

对比 (3) 和 (4):

- 在这种尺度变化的情况下，算法的表现会有所不同.

二阶方法的动机 II

- 例如, 当 F 是具有唯一极小值 w_* 的强凸二次函数时,
  (3): 通常需要多次迭代才能逼近极小值
  (4): 只需要一步的迭代 $B = (\nabla^2 F(w_k))^{-1/2}$, $\alpha = 1$.
  (牛顿方法的一次迭代).

高斯-牛顿方法 I

介绍

- 非线性最小二乘法的经典方法，即最小化问题，其中目标函数是平方和。
- 这个想法也适用于其他流行的损失函数。
- 优点：它仅使用保证是半正定的一阶信息来构造对 Hessian 的近似，
- 限制: 忽略（一般）损失函数 ℓ 的曲率.

高斯-牛顿方法 II

输入输出对 $\xi := (x, y)$, 由参数向量 w 引起的损失是在 $h(x; w) \in R^d$ 和 $y \in R^d$ 之间测量的:

$$f(w, \xi) = \ell(h(x_\xi; w), y_\xi) = \frac{1}{2}\|h(x_\xi; w) - y_\xi\|_2^2 = \frac{1}{2}\sum_{i=1}^{N}(h(x_{\xi_i}; w) - y_{\xi_i})^2.$$

- 牛顿迭代：$f(w, \xi)$ 在 w_k 附近的二阶泰勒级数模型
- 高斯-牛顿迭代：对二次损失函数内的预测函数 h 进行仿射近似。

高斯-牛顿迭代 I

$J_h(\cdot;\xi)$: $h(x_\xi;\cdot)$ 相对于 w 的雅可比

$$h(x_\xi; w) \approx h(x_\xi; w_k) + J_h(w_k;\xi)(w - w_k),$$

这导致

$$\begin{aligned} f(w;\xi) &\approx \frac{1}{2}\|h(x_\xi; w_k) + J_h(w_k;\xi)(w - w_k) - y_\xi\|_2^2 \\ &= \frac{1}{2}\|h(x_\xi; w_k) - y_\xi\|_2^2 + (h(x_\xi; w_k) - y_\xi)^T J_h(w_k;\xi)(w - w_k) \\ &\qquad + \tfrac{1}{2}(w - w_k) J_h(w_k;\xi)^T J_h(w_k;\xi)(w - w_k). \end{aligned}$$

高斯-牛顿矩阵

用高斯-牛顿矩阵替换次采样的 Hessian 矩阵

$$G_{S_k^H}(w_k;\xi_k^H) = \frac{1}{|S_k^H|}\sum_{i\in S_k^H} J_h(w_k;\xi_{k,i})^T J_h(w_k;\xi_{k,i}). \tag{5}$$

比较高斯-牛顿近似和牛顿近似

高斯-牛顿迭代 II

$$f(w,\xi) = \ell(h(x_\xi; w), y_\xi) = \frac{1}{2}\|h(x_\xi; w) - y_\xi\|_2^2 = \sum_{i=1}^{N}(h(x_{\xi_i}; w) - y_{\xi_i})^2.$$

$f(w,\xi)$ 在 $w = w_k$ 的泰勒展开式:

$$f(w,\xi) \approx f(w_k,\xi) + \langle \nabla_w f(w_k,\xi), w - w_k\rangle + \frac{1}{2}(w - w_k)^T \nabla_w^2 f(w_k,\xi)\cdot(w - w_k),$$

$$\nabla_w f(w_k,\xi) = \nabla_w h(x_\xi, w_k)\cdot(h(x_\xi, w_k) - y_\xi) =$$

$$[\nabla_w h(x_{\xi_1}, w_k), \cdots, \nabla_w h(x_{\xi_N}, w_k)]\cdot\begin{bmatrix} h(x_{\xi_1}, w_k) - y_{\xi_1} \\ \cdots \\ h(x_{\xi_N}, w_k) - y_{\xi_N}\end{bmatrix},$$

$$\nabla_w^2 f(w_k,\xi) = \sum_{i=1}^{N}\nabla_w h(x_{\xi_i}, w_k)\cdot\nabla_w h(x_{\xi_i}, w_k)^T$$

$$+\sum_{i=1}^{N}\nabla_w^2 h(x_{\xi_i}, w_k)\cdot(h(x_{\xi_i}, w_k) - y_{\xi_i}).$$

高斯-牛顿矩阵的奇异问题

- 挑战. 高斯-牛顿矩阵通常是奇异的或近似奇异的.
- 解决. 通过向其添加单位矩阵的正倍数来正则化.
- 应用. 不精确的 Hessian 自由牛顿方法和具有如 (??) 中定义的梯度位移向量的随机拟牛顿方法可以应用（正则化的）高斯-牛顿近似: 保证是正定的.

广义高斯-牛顿法

$$f(w,\xi) = \ell(h(x_\xi; w), y_\xi) = \frac{1}{2}\sum_{i=1}^{N} \ell(h(x_{\xi_i}; w) - y_{\xi_i}).$$

预测函数的仿射逼近 $h(x_\xi; w)$ + 损失函数的二阶泰勒展开式 $l \longrightarrow$ 广义高斯-牛顿矩阵

$$G_{S_k^H}(w_k; \xi_k^H) = \frac{1}{|S_k^H|}\sum_{i\in S_k^H} J_h(w_k; \xi_{k,i})^T H_l(w_k; \xi_{k,i}) J_h(w_k; \xi_{k,i}). \tag{6}$$

其中 $H_l(w_k; \xi) = \frac{\partial^2 l}{\partial h^2}(h(x_\xi; w_k), y_k)$ 捕获损失函数的曲率 l. * 它是 (5) 的推广，其中 $H_l = I$.

广义高斯-牛顿法

训练 DNN 时, 使用形式为 $f(w;\xi) = -\log(h(x_\xi; w))$ 的对数损失广义高斯-牛顿矩阵:

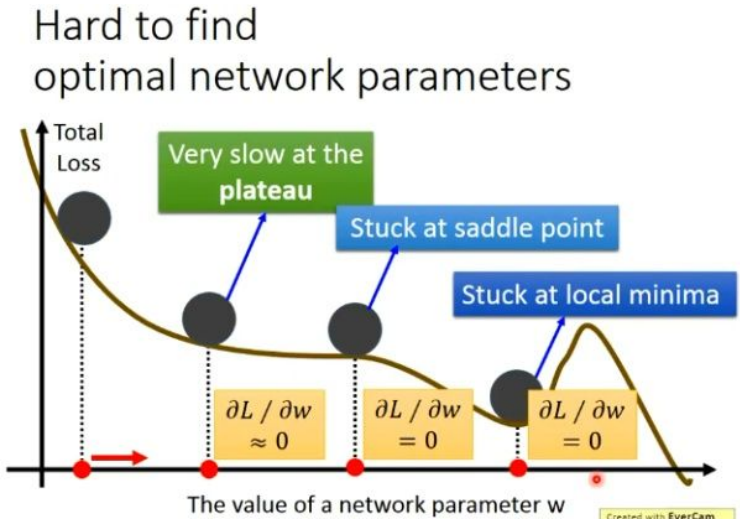
$$G_{S_k^H}(w_k;\xi_k^H) = \frac{1}{|S_k^H|}\sum_{i\in S_k^H} J_h(w_k;\xi_{k,i})^T \frac{1}{h(w;\xi_{k,i})^2} J_h(w_k;\xi_{k,i}) \\ \frac{1}{|S_k^H|}\sum_{i\in S_k^H} \nabla f(w_k;\xi_{k,i})\nabla f(w_k;\xi_{k,i})^T, \tag{7}$$

它不需要雅可比矩阵的显式计算 J_h.

第 6 章第 5 节 随机梯度下降法

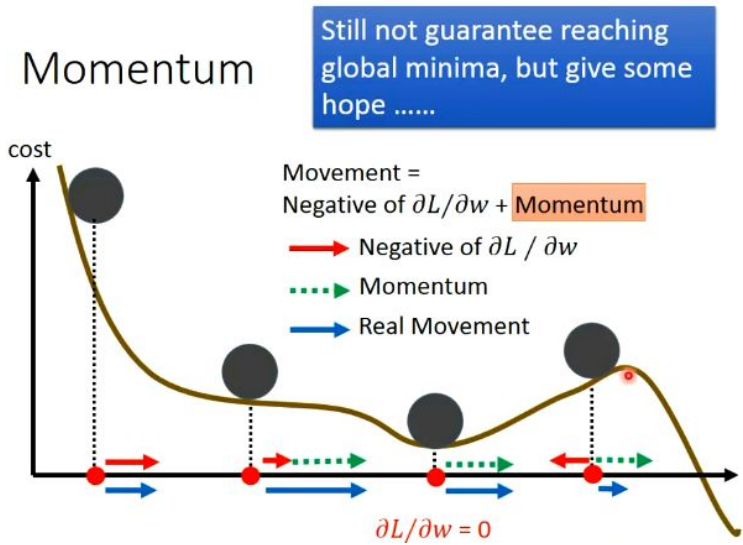
1. 随机梯度 (SG) 方法综述
2. SG 收敛性分析
3. 梯度聚合
4. 二阶方法
5. 其他流行方法

动量梯度法的动机



From: 李宏毅：机器学习

动量梯度法的动机



From: 李宏毅：机器学习

动量梯度法 Gradient Methods with Momentum

动量梯度法是最速下降方向和最近迭代位移法的组合，这些方法的特点是迭代为

$$w_{k+1} \leftarrow w_k - \alpha_k \nabla F(w_k) + \beta_k (w_k - w_{k-1}), \tag{7.1}$$

右边的项被称为动量项.

- 当 $\beta_k = 0$ 时, 对于所有 $k \in N$, 它简化为最速下降法.
- 当 $\alpha_k = \alpha > 0$ 且 $\beta_k = \beta > 0$ 时，对于所有 $k \in N$, 它被称为重球法 (the heavy ball method).

动量梯度法 Gradient Methods with Momentum

重球法的另一种观点是通过扩展 (7.1):

$$w_{k+1} \leftarrow w_k - \alpha_k \nabla F(w_k) + \beta_k (w_k - w_{k-1}) \tag{7-1}$$

$$w_{k+1} \leftarrow w_k - \alpha \sum_{j=1}^{k} \beta^{k-j} \nabla F(w_j)$$

这些步骤倾向于在持续下降的方向上积累贡献，而振荡的方向往往会被抵消，或者至少保持较小。

加速梯度法 Accelerated Gradient Methods

迭代类似于 (7.1) 的方法，但具有自己独特的性质，是涅斯捷罗夫 (Nesterov) 提出的加速梯度法.

$$\tilde{w}_k \leftarrow w_k + \beta_k (w_k - w_{k-1})$$
$$w_{k+1} \leftarrow \tilde{w}_k - \alpha_k \nabla F(\tilde{w}_k),$$

从而形成缩合形式

$$w_{k+1} \leftarrow w_k - \alpha_k \nabla F(w_k + \beta_k (w_k - w_{k-1})) + \beta_k (w_k - w_{k-1}). \tag{8}$$

总结

对比随机和 Batch 梯度方法

$$\min_{w\in\mathbb{R}^d} F(w) := \frac{1}{n}\sum_{i=1}^{n} f_i(w)$$

最速下降算法

$$w_{k+1} := w_k - \frac{\alpha_k}{n}\sum_{i=1}^{n}\nabla f_i(w_k)$$

随机梯度算法

$$w_{k+1} := w_k - \alpha_k \nabla f_{i_k}(w_k)$$

其中 i_k 是从 $\{1,\cdots,n\}$ 随机选取的

mini-batch SG

$$w_{k+1} := w_k - \alpha_k \cdot \frac{1}{|S_k|}\sum_{i\in S_k}\nabla f_i(w_k)$$

总结

SG 方法的收敛性

- 对于 FG, 如果 F 是强凸函数, 那么

$$F(w_k) - F^* \leq \mathcal{O}(\rho^k), \text{线性收敛}$$

  其中 $\rho \in (0,1)$. 求得 ε 最优解所需的计算量与 $n\log(\frac{1}{\varepsilon})$ 成正比.

- 对于 SG,

$$\mathbb{E}[F(w_k) - F^*] = \mathcal{O}(\frac{1}{k}), \text{次线性收敛}$$

  求得 ε 最优解所需的计算量与 $\frac{1}{\varepsilon}$ 成正比.

备注: 在大数据场景下, 对于 n 很大时, 有 $n\log(\frac{1}{\varepsilon}) > \frac{1}{\varepsilon}$.

总结

假设 1: F 的梯度函数是 Lipschitz 连续的,
$\|\nabla F(w) - \nabla F(\overline{w})\|_2 \leq L\|w - \overline{w}\|_2$ *for all* $\{w, \overline{w}\} \subset \mathbb{R}^d$.

假设 2: 一阶矩和二阶矩假设
(a) F 在开集上有下界;
(b) $\exists\, \mu_G \geq \mu > 0, \forall\, k$

$$\nabla F(w_k)^T \mathbb{E}_{\xi_k}[g(w_k; \xi_k)] \geq \mu \|\nabla F(w_k)\|_2^2 \text{ 且}$$

$$\|\mathbb{E}_{\xi_k}[g(w_k; \xi_k)]\|_2 \leq \mu_G \|\nabla F(w_k)\|_2.$$

(c) $\exists\, M > 0, M_V \geq 0, \forall k, \mathbb{V}_{\xi_k}[g(w_k; \xi_k)] \leq M + M_V \|\nabla F(w_k)\|_2^2$,
其中 $\mathbb{V}_{\xi_k}[g(w_k; \xi_k)] := \mathbb{E}_{\xi_k}[\|g(w_k; \xi_k)\|_2^2] - \|\mathbb{E}_{\xi_k}[g(w_k; \xi_k)]\|_2^2$.

总结

定理 3 (强凸, 固定步长)

在假设 *1* 和 *2* 成立的条件下, 假设 F 是强凸的, SG 方法的步长满足 $\alpha_k = \bar{\alpha}$

$$0 < \bar{\alpha} \leq \frac{\mu}{LM_G},$$

那么,

$$\mathbb{E}[F(w_k) - F_*] \leq \frac{\bar{\alpha} LM}{2c\mu} + (1 - \bar{\alpha} c\mu)^{k-1} \left(F(w_1) - F_* - \frac{\bar{\alpha} LM}{2c\mu} \right).$$

总结

梯度聚合算法

是否通过重复使用或修改先前计算的信息来实现较低的方差?

- SVRG (随机方差约减梯度)

$$\tilde{g}_j \leftarrow \nabla f_{i_j}\left(\tilde{w}_j\right) - \left(\nabla f_{i_j}\left(w_k\right) - \nabla R_n\left(w_k\right)\right)$$

- SAGA(随机平均梯度算法)

$$g_k \leftarrow \nabla f_j\left(w_k\right) - \nabla f_j\left(w_{[j]}\right) + \frac{1}{n}\sum_{i=1}^{n} \nabla f_i\left(w_{[i]}\right)$$

梯度聚合方法 vs SG

- SG 的计算时间为 $T(n,\epsilon) \sim \kappa^2/\epsilon$，其中 $\kappa := L/c$
- SVRG 和 SAGA 的计算时间为 $\mathcal{T}(n,\epsilon) \sim (n+\kappa)\log(1/\epsilon)$，随着样本点数量 n 的增加而增加

总结

其他流行优化算法

- Gradient Methods with Momentum 动量梯度法

$$\mathbf{v}_t = \gamma \mathbf{v}_{t-1} + \eta \nabla_\theta F(w)$$
$$\mathbf{w}_t = \mathbf{w}_t - \mathbf{v}_t$$

- Accelerated Gradient Methods 加速梯度法

$$\tilde{w}_k \leftarrow w_k + \beta_k (w_k - w_{k-1})$$
$$w_{k+1} \leftarrow \tilde{w}_k - \alpha_k \nabla F(\tilde{w}_k),$$

作业

1. 编程实现 SVRG 和 SAGA 算法, 分别求解岭回归模型:

$$\min_{\boldsymbol{w}\in\mathbf{R}^{\mathrm{d}}} \frac{1}{N}\sum_{i=1}^{n}\left(\boldsymbol{x}_i^{\top}\boldsymbol{w}-y_i\right)^2+\lambda\cdot\Omega(\boldsymbol{w})$$

其中, $(\boldsymbol{x}_i, y_i)\in\mathbf{R}^d\times\mathbf{R}$ 为观测样本, $i=1,\cdots,n$, 正则化项 $\Omega(\boldsymbol{w})=\boldsymbol{w}^{\top}\boldsymbol{w}, \lambda>0$ 为给定的正则化参数。