2019 年 12 月 23 日
慶應義塾大学理工学部物理学科
岡崎健人

勾配降下法 自分用教科書

m 個の訓練データ $\left(\boldsymbol{x}^{(i)}, y^{(i)}\right),\ i = 1\ldots, m$ が与えられたとする。$\boldsymbol{x}^{(i)} = \left[x_1^{(i)}, \ldots, x_n^{(i)}\right]^\mathsf{T}$ は n 個の成分を持つベクトルである。スカラーの基底関数を $\phi_j(\boldsymbol{x}),\ j = 1, \ldots, k$ とすれば、計画行列 X の要素は $(X)_{ij} = \phi_j\left(\boldsymbol{x}^{(i)}\right)$ である。$\boldsymbol{\phi}(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}), \ldots, \phi_k(\boldsymbol{x})]^\mathsf{T},\ \boldsymbol{\theta} = [\theta_1, \ldots, \theta_k]^\mathsf{T}$ として、訓練データによる y の推定量を

$$\hat{y} = \sum_{j=1}^{k} \theta_j \phi_j(\boldsymbol{x}) = \boldsymbol{\theta}^\mathsf{T} \boldsymbol{\phi}(\boldsymbol{x})$$

と展開できるとする。さらに各訓練データ $\boldsymbol{x}^{(i)}$ に対してこの推定値を計算したものを $\hat{y}^{(i)} = \boldsymbol{\theta}^\mathsf{T} \boldsymbol{\phi}\left(\boldsymbol{x}^{(i)}\right)$ とする。それを並べたベクトルを $\hat{\boldsymbol{y}} = \left[\hat{y}^{(1)}, \ldots, \hat{y}^{(m)}\right]^\mathsf{T}$ と表記すれば、$\hat{\boldsymbol{y}} = X\boldsymbol{\theta}$ と簡潔に書ける。

y の訓練データと推定量の平均二乗誤差（mean squared error）はパラメータ $\boldsymbol{\theta}$ の関数であり、これを $\mathrm{MSE}(\boldsymbol{\theta})$ と記す。

$$\mathrm{MSE}(\boldsymbol{\theta}) = \frac{1}{m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)}\right)^2 = \frac{1}{m} \|\hat{\boldsymbol{y}} - \boldsymbol{y}\|_2^2 = \frac{1}{m} \|X\boldsymbol{\theta} - \boldsymbol{y}\|_2^2$$

1 バッチ勾配降下法

関数 $J(\boldsymbol{\theta})$ はパラメータのベクトル $\boldsymbol{\theta}$ によって特徴付けられている。この関数をバッチ勾配降下法（batch gradient descent, BGD）を用いて最小化することがここでの目的である。

まず初期値 $\boldsymbol{\theta}_0$ をランダムに選び、その点での勾配 $\boldsymbol{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_0)$ を計算する。そして

$$J(\boldsymbol{\theta}_1) \leftarrow J(\boldsymbol{\theta}_0) - \boldsymbol{\eta}^\mathsf{T} \boldsymbol{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_0)$$

のように関数 $J(\boldsymbol{\theta})$ を更新する。さらにその点 $\boldsymbol{\theta}_1$ を初期値として同じ操作を反復すれば、最小点 $\underset{\boldsymbol{\theta}}{\operatorname{argmin}}\ J(\boldsymbol{\theta})$ に収束する。係数のベクトル $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_k]^\mathsf{T}$ の成分あるいはベクトル自体のことを、ステップ幅あるいは機械学習の文脈では学習率（learning rate）とよばれる。

数値計算上では、関数の微分はその定義の近似を用いればよい。

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \approx \frac{f(x + \Delta x) - f(x)}{\Delta x}.$$

学習率の値は大きすぎず小さすぎないのがよい。もし学習率が大きすぎると、ステップごとに最小点を通り過ぎてしまい、収束が遅くなるか収束できなくなるからである。反対に小さすぎると 1 ステップが小さいため収束が遅くなる。

関数 $J(\boldsymbol{\theta})$ が m 個の訓練データから得られる平均二乗誤差 $\mathrm{MSE}(\boldsymbol{\theta})$ であることはよくある。その場合、勾配降下法はコストが高い方法である。訓練データの数を m，パラメータの数を k とすれば、平均二乗誤差は m 項からなる。パラメータのそれぞれの成分についての微分を計算するので、1 ステップあたり $m \times k$ 回の計算が必要である。ステップは 1000 回行うことが普通であるので、例えば $m = 10000,\ k = 10$ であるとすれば全体で 10^8 回の計算が必要になってしまい、データ数が大きいと遅いアルゴリズムであることがわかる。

2　確率的勾配降下法

バッチ勾配降下法ではコスト関数は各データに関する項の和の形

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{m} J_i(\boldsymbol{\theta})$$

をしていた。このニュアンスを強調して上のアルゴリズムをバッチ勾配降下法（batch gradient descent）とよぶ。それに対して訓練データからランダムに 1 つ選んで、そのコスト関数のみについての勾配を計算して更新してゆく手法を確率的勾配降下法（stochastic gradient descent）という。訓練データはステップごとに選び直す。

確率的な性質を持つため、最小点に向かってゆく仮定は複雑になる。平均的には最小値に向かって緩やかに小さくなっていくのだが、最小値周りに達すると勾配は様々な方向を向いているので、1 箇所に落ち着くことがない。そのため確率的勾配降下法による最終的なパラメータは十分よいものだが最適ではない。

しかしコスト関数が複数の極小値を持つような複雑な関数の場合、確率的な性質のおかげで、関数の谷となっている部分から抜け出し、真の最小値にたどり着く可能性は大きくなる。

まとめると確率的勾配降下法は、その無作為性によって局所的な最小値から逃れる可能性はあがるが、最小値に落ち着かないというジレンマをもつ。その問題を解決するために、勾配の大きさに応じて学習率を変化させる方法がある。勾配が大きければ大きく前進し、勾配がゼロに近くなったら慎重に前進するのだ。金属が結晶化する過程がヒントになっていて、このアルゴリズムを焼きなまし法（simulated annealing）とよぶ。

3　ミニバッチ勾配降下法

ミニバッチ勾配降下法（minibatch gradient descent）はバッチ勾配降下法と確率的勾配降下法の中間的なアルゴリズムである。ステップごとに訓練データから複数個のデータをランダムに抜き出して、それらのコスト関数の勾配から最小値を探し出す。ランダムに選ばれた訓練データの集合をミニバッチとよぶ。N ステップ目の、m' 個の訓練データからなるミニバッチを $\mathbb{B}_N$ と表せば、N ステップ目のコスト関数は

$$J_N(\boldsymbol{\theta}) = \sum_{\boldsymbol{x}^{(i)} \in \mathbb{B}_N} J_i(\boldsymbol{\theta}) = \frac{1}{m'} \sum_{\boldsymbol{x}^{(i)} \in \mathbb{B}_N} \left(\hat{y}^{(i)} - y^{(i)} \right)^2$$

とかける。