最適化と学習アルゴリズム

勾配降下法
直線探索
劣勾配降下法
ニュートン法
確率的勾配降下法

学習における計算効率の問題

- 既に説明した回帰、識別（分類）の解法で得た閉じた式は逆行列計算が必要。例えば、線形回帰の場合の正規方程式の解である重みベクトルは $\mathbf{w}=(X^TX)^{-1}X^Ty$
- 学習データが高次元になると(X^TX)の次元も大きくなり、逆行列の計算はコストが高い（次元数の３乗）
- 要は損失(=Loss)を最小化したい
  - 正規方程式の場合は2乗誤差だった。
- 逆行列計算を避けて最適化する方法が実用上重要。

記法：損失(Loss)

入力データ：$\mathbf{x}_i$, 重みベクトル：$\mathbf{w}$はD次元ベクトルとする

$\mathbf{x}_i$に対する分類の正解：$y_i (= -1 \text{ or } 1)$ ただしデータはm個あるとする

$$L(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^{m} loss(\mathbf{w} \text{ on}(\mathbf{x}_i, y_i))$$

例：

2乗損失の場合　$L(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^{m}\left\| \langle \mathbf{w}, \mathbf{x}_i \rangle - y_i \right\|_2^2 = \frac{1}{m}\sum_{i=1}^{m}\left(\langle \mathbf{w}, \mathbf{x}_i \rangle - y_i \right)^2$

ヒンジ損失の場合　$L(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^{m}\left[1 - y_i \langle \mathbf{w}, \mathbf{x}_i \rangle\right]_+$

ここで　$[z]_+ = \max(0, z)$

記法：
勾配(Gradient)とヘッセ行列（Hessian)

$\mathbf{w}$はD次元ベクトルとする

pノルム：$\|\mathbf{w}\|_p = \left(w_1{}^p + \cdots w_D{}^p\right)^{1/p}$ $\quad \|\mathbf{w}\|_p^p = \left(w_1{}^p + \cdots w_D{}^p\right)$

$\mathbf{w} = \mathbf{w}^{(k)}$における$L$の勾配

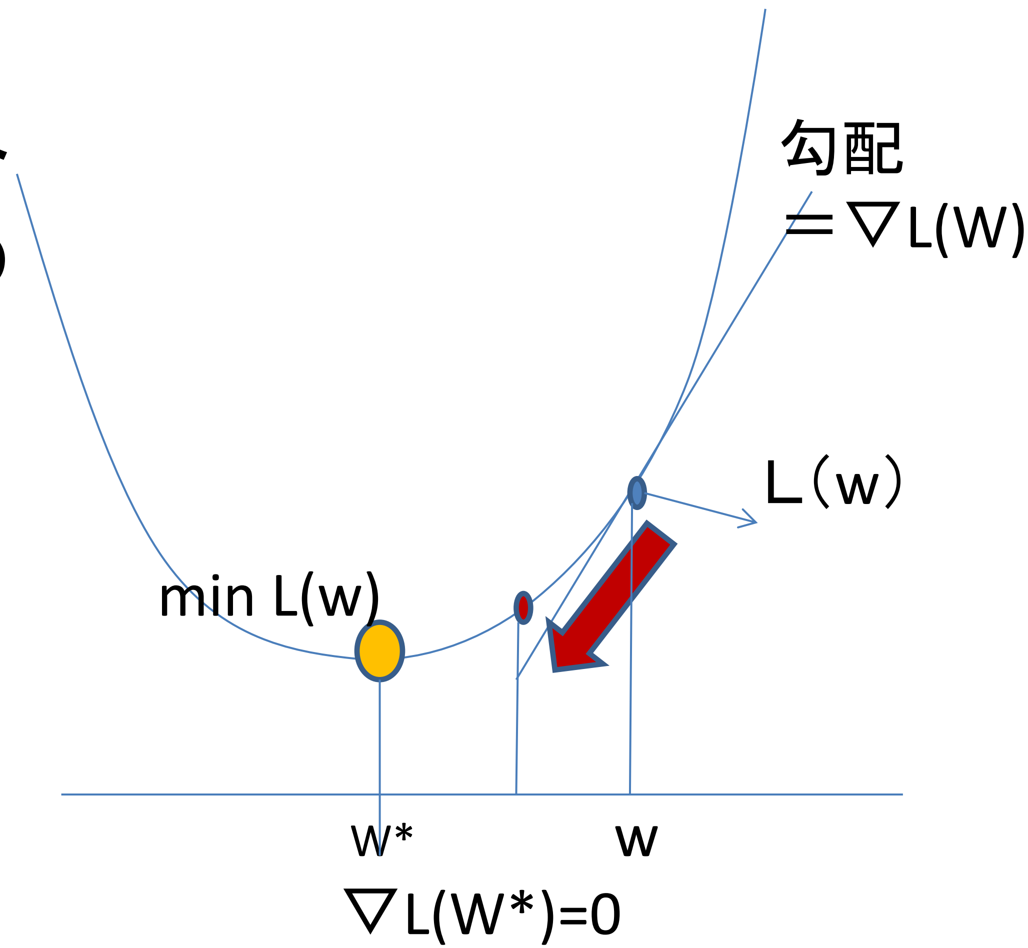
$$\nabla L\left(\mathbf{w}^{(k)}\right) = \left(\left.\frac{\partial L(\mathbf{w})}{\partial w_1}\right|_{w_1 = w_1{}^{(k)}}, \ldots, \left.\frac{\partial L(\mathbf{w})}{\partial w_n}\right|_{w_D = w_D{}^{(k)}} \right)^{\mathrm{T}}$$

$\mathbf{w} = \mathbf{w}^{(k)}$における$L$のヘッセ行列

$$\mathrm{H}L\left(\mathbf{w}^{(k)}\right) = \begin{pmatrix} \left.\dfrac{\partial^2 L(\mathbf{w})}{\partial w_1 \partial w_1}\right|_{w_1 = w_1{}^{(k)}} & \cdots & \left.\dfrac{\partial^2 L(\mathbf{w})}{\partial w_1 \partial w_D}\right|_{w_1 = w_1{}^{(k)}, w_D = w_D{}^{(k)}} \\ \vdots & \ddots & \vdots \\ \left.\dfrac{\partial^2 L(\mathbf{w})}{\partial w_D \partial w_1}\right|_{w_D = w_D{}^{(k)}, w_1 = w_1{}^{(k)}} & \cdots & \left.\dfrac{\partial^2 L(\mathbf{w})}{\partial w_D \partial w_D}\right|_{w_D = w_D{}^{(k)}} \end{pmatrix}$$

勾配降下法

- 右図で分るように L（w）が凸の場合は、勾配∇L(w)の逆方向に進むと W*=argmin L(W) に向かって進む。
  - 図は1次元だが実際は多次元なので、勾配の方向が大切なことに注意



定式化

- 勾配降下法(Gradient Descent)の定式化する

$\mathbf{w}$：重みベクトル　　$L(\mathbf{w})$: 重みベクトル$\mathbf{w}$のときの損失

$\mathbf{w}^{(k)}$: k回目の繰り返し結果の更新された重みベクトル

最適な重みベクトル : $\mathbf{w}^* = \arg\min_{\mathbf{w}} L(\mathbf{w})$

を求めるには勾配の逆方向に進むので下の式となる。

重みベクトルの更新式：$\boxed{\mathbf{w}^{(k+1)} = \arg\min_{\mathbf{v}} L\left(\mathbf{v} - \left(\mathbf{w}^{(k)} - \alpha^{(k)} \nabla L\left(\mathbf{w}^{(k)}\right)\right)\right)}$

$L(*) = \|*\|_2^2$: ２乗損失だと$\mathbf{w}^{(k+1)} = \arg\min_{\mathbf{v}} \left\|\mathbf{v} - \left(\mathbf{w}^{(k)} - \alpha^{(k)} \nabla L\left(\mathbf{w}^{(k)}\right)\right)\right\|_2^2$

損失を微分して０と置けば閉じた解が求まり

$\boxed{\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \alpha^{(k)} \nabla L\left(\mathbf{w}^{(k)}\right)}$　となる

2乗損失でないとこう簡単な式ではない

最急降下法アルゴリズム

- $L(\mathbf{w})$を最小化するような$\mathbf{w}$を求めるために、$\mathbf{w}$の現在の値$\mathbf{w}^{(k)}$から勾配$\nabla L(\mathbf{w})$の逆方向に進むアルゴリズム

$$\text{初期値} = \mathbf{w}^{(0)}; k = 0; \quad \mathbf{w}^* = \arg\min L(\mathbf{w});$$

*step*1: $\mathbf{w}^{(k)}$が$\mathbf{w}^*$に十分近いなら終了

（具体的には$\mathbf{w}^{(k)}$が$\mathbf{w}^{(k-1)}$に十分近くて収束した）

*step*2: $\nabla L(\mathbf{w}^{(k)})$を計算

*step*3: $\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \alpha^{(k)} \nabla L(\mathbf{w}^{(k)})$ 　　$\alpha^{(k)}$は進む幅(step size)

*step*4: $k = k + 1$ 　として*step*1へ

$$L(\mathbf{w}) = \left\| y_n - \langle \mathbf{w}, \phi(\mathbf{x}_n) \rangle \right\|_2^2$$ 　つまり2乗損失なら

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \alpha^{(k)} \left(y_n - \langle \mathbf{w}^{(k)}, \phi(\mathbf{x}_n) \rangle \right) \phi(\mathbf{x}_n)$$

最急降下法の収束の評価

- 前提条件
  - 凸　&　$\nabla^2 L(\mathbf{w})$ positive
  - Lipschitz条件： $\left|L\left(\mathbf{w}^{(k)}\right)-L\left(\mathbf{w}^{(k+1)}\right)\right| \le G\left\|\mathbf{w}^{(k)}-\mathbf{w}^{(k+1)}\right\|_2$

$$\Rightarrow \left\|\nabla L\left(\mathbf{w}^{(k)}\right)\right\|_2 < G$$

- テイラー展開により近似

$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \alpha^{(k)} \nabla L\left(\mathbf{w}^{(k)}\right)$とし、$L\left(\mathbf{w}^{(k+1)}\right)$を2次のテイラー近似すると

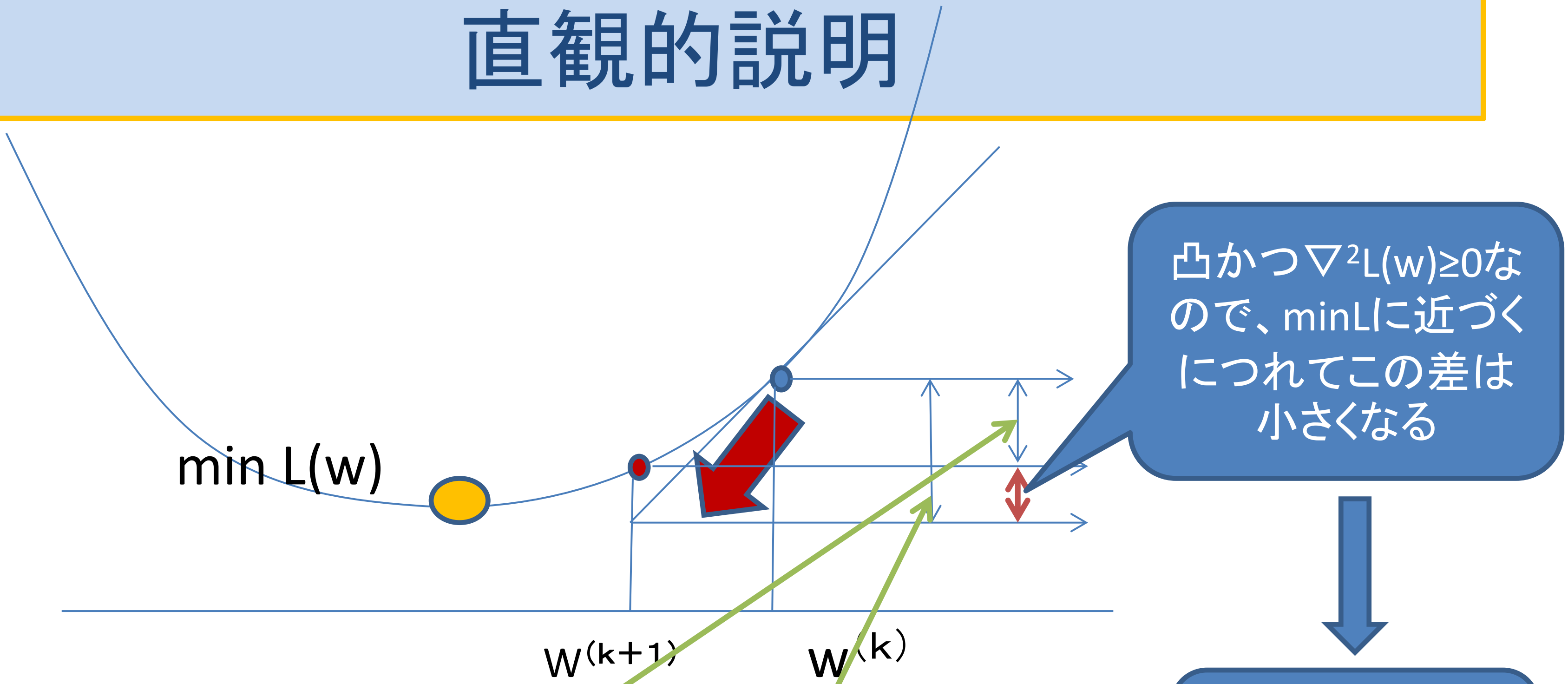
$$L\left(\mathbf{w}^{(k+1)}\right)-L\left(\mathbf{w}^{(k)}\right) \approx -\alpha^{(k)}\left\|\left(\nabla L\left(\mathbf{w}^{(k)}\right)\right)\right\|_2^2 + \frac{\left(\alpha^{(k)}\right)^2}{2}\left\|\left(\nabla L\left(\mathbf{w}^{(k)}\right)\right)\right\|_2^2 \left\|\nabla^2 L\left(\mathbf{w}^{(k)}\right)\right\|_2$$

$\left\|\nabla^2 L\left(\mathbf{w}^{(k)}\right)\right\|_2 \ge 0$により

$$L\left(\mathbf{w}^{(k+1)}\right)-L\left(\mathbf{w}^{(k)}\right) \ge -\alpha^{(k)}\left\|\left(\nabla L\left(\mathbf{w}^{(k)}\right)\right)\right\|_2^2$$

$$\alpha^{(k)}\left\|\left(\nabla L\left(\mathbf{w}^{(k)}\right)\right)\right\|_2^2 \ge L\left(\mathbf{w}^{(k)}\right)-L\left(\mathbf{w}^{(k+1)}\right) \quad (1)$$

直観的説明



$$\left|L\left(\mathbf{w}^{(k)}\right)-L\left(\mathbf{w}^{(k+1)}\right)\right| \le G\left\|\mathbf{w}^{(k)}-\mathbf{w}^{(k+1)}\right\|_2$$

$Lipshitz$条件より以下のようなGがある

$$\Rightarrow \boxed{\left|L\left(\mathbf{w}^{(k)}\right)-L\left(\mathbf{w}^{(k+1)}\right)\right|} \le \boxed{\left\|\nabla L\left(\mathbf{w}^{(k)}\right)\right\|_2\left\|\mathbf{w}^{(k)}-\mathbf{w}^{(k+1)}\right\|_2} \le G\left\|\mathbf{w}^{(k)}-\mathbf{w}^{(k+1)}\right\|_2$$

$$\Rightarrow \left\|\nabla L\left(\mathbf{w}^{(k)}\right)\right\|_2 < G$$

最急降下法の収束の評価:つづき

$$\alpha^{(k)}\left\|\left(\nabla L\left(\mathbf{w}^{(k)}\right)\right)\right\|_2^2 \geq L\left(\mathbf{w}^{(k)}\right)-L\left(\mathbf{w}^{(k+1)}\right) \quad (1)$$

$\mathbf{w}^{(k+1)}$が$\mathbf{w}^{*}$に十分近い　($\mathbf{w}^{(k+1)} \approx \mathbf{w}^{*}$)ないし等しいとすると

$$\alpha^{(k)}\left\|\left(\nabla L\left(\mathbf{w}^{(k)}\right)\right)\right\|_2^2 \geq L\left(\mathbf{w}^{(k)}\right)-L\left(\mathbf{w}^{*}\right)$$

ここで, $B \leq \left\|\mathbf{w}^{*}\right\|_2$ というBを選び、$\alpha^{(k)}=\dfrac{B}{G\sqrt{k}}$とおく

Lipschitz条件　$\left\|\nabla L\left(\mathbf{w}^{(k)}\right)\right\|_2 < G$　より

$$\Rightarrow L\left(\mathbf{w}^{(k)}\right)-L\left(\mathbf{w}^{*}\right) \leq \frac{B}{G\sqrt{k}}\cdot G^2=\frac{BG}{\sqrt{k}}$$

誤差をε以下にするのはオーダとして：$O(\varepsilon)=O\left(\dfrac{BG}{\sqrt{k}}\right)$

$\Rightarrow$ $\boxed{O\left(\dfrac{B^2G^2}{\varepsilon^2}\right)\text{回の更新で誤差}\varepsilon\text{以下に収束}}$

捕捉

$\alpha^{(k)}$をやたらと大きくすると無意味！

$$\left|L\left(\mathbf{w}^{(k)}\right)-L\left(\mathbf{w}^{*}\right)\right|\le G\left\|\mathbf{w}^{(k)}-\mathbf{w}^{*}\right\|_2$$

$B\le\left\|\mathbf{w}^{*}\right\|_2 \quad \Rightarrow \quad B\le\left\|\mathbf{w}^{(k)}-\mathbf{w}^{*}\right\|_2$くらいにはなるだろう

$$\Rightarrow\left|L\left(\mathbf{w}^{(k)}\right)-L\left(\mathbf{w}^{*}\right)\right|\approx BG\le G\left\|\mathbf{w}^{(k)}-\mathbf{w}^{*}\right\|_2$$

このあたりを想定してというBを選び、$\alpha^{(k)}=\dfrac{B}{G\sqrt{k}}$とおく

という雰囲気

step size αを決める:直線探索

- $\alpha^{(k)}$が小さすぎると収束が遅々として進まず、
  大きすぎると最悪の場合最適値を飛び越す
- 最適な$\alpha^{(k)}$を求める問題:直線探索

$$\min_{\alpha^{(k)}} L\left(\mathbf{w}^{(k)} + \alpha^{(k)}\mathbf{d}\right) \quad \mathbf{d}\text{は降下方向(つまり}-\nabla L\left(\mathbf{w}^{(k)}\right)\text{方向)の単位ベクトル}$$

$$\text{subject to } \alpha^{(k)} > 0$$

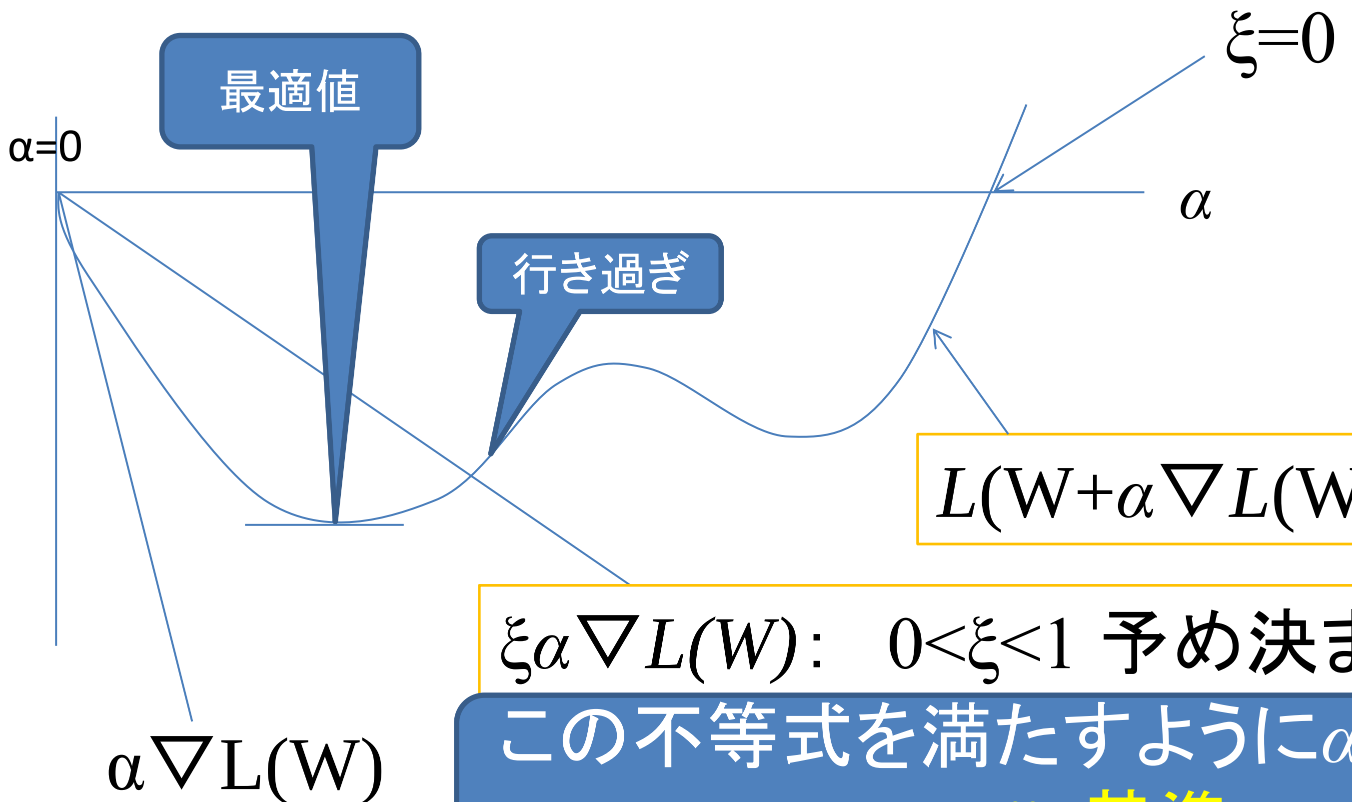
- 厳密に解くのは困難

直線探索の代案

- kの減少関数にする→ i.e. $\alpha^{(k)} \propto 1/\sqrt{k}$
- しかし、あまりにもヒューリスティック
- もう少し客観的な基準として

- Armijo基準とWolfe基準がある

Armijo基準とWolfe基準



$L(\mathrm{W}+\alpha\nabla L(\mathrm{W}))\text{-}L(\mathrm{W})$

$\xi\alpha\nabla L(W)$:　$0<\xi<1$ 予め決まった値

この不等式を満たすようにαを選ぶ：
Armijo基準

$$L(\mathrm{W}+\alpha\mathbf{d})\text{-}L(\mathrm{W}) \le \xi\alpha\nabla L(\mathrm{W})^{\mathrm{T}}\mathbf{d}$$

Wolfe基準：
α が小さくなりすぎないため。α =0だと明らかに成立しない(!∇$L(\mathrm{W})^{\mathrm{T}}\mathbf{d}<0$)

$$\mu\nabla L(\mathbf{w})^{\mathrm{T}}\mathbf{d} \le \nabla L(\mathbf{w}+\alpha\mathbf{d})^{\mathrm{T}}\mathbf{d}$$

$$\xi<\mu<1$$

直線探索の代案:2分割法

- k回目の繰り返しにおいて以下のアルゴリズムで$\alpha^{(k)}$を減少させる

Step1　t=1,　$\alpha_t^{(k)}=\alpha^{(k)}$

Step2　if 停止基準(Armijo基準等）が満たされる

then return　$\alpha_t^{(k)}$

else $\alpha_t^{(k)}=\alpha^{(k)}/2$

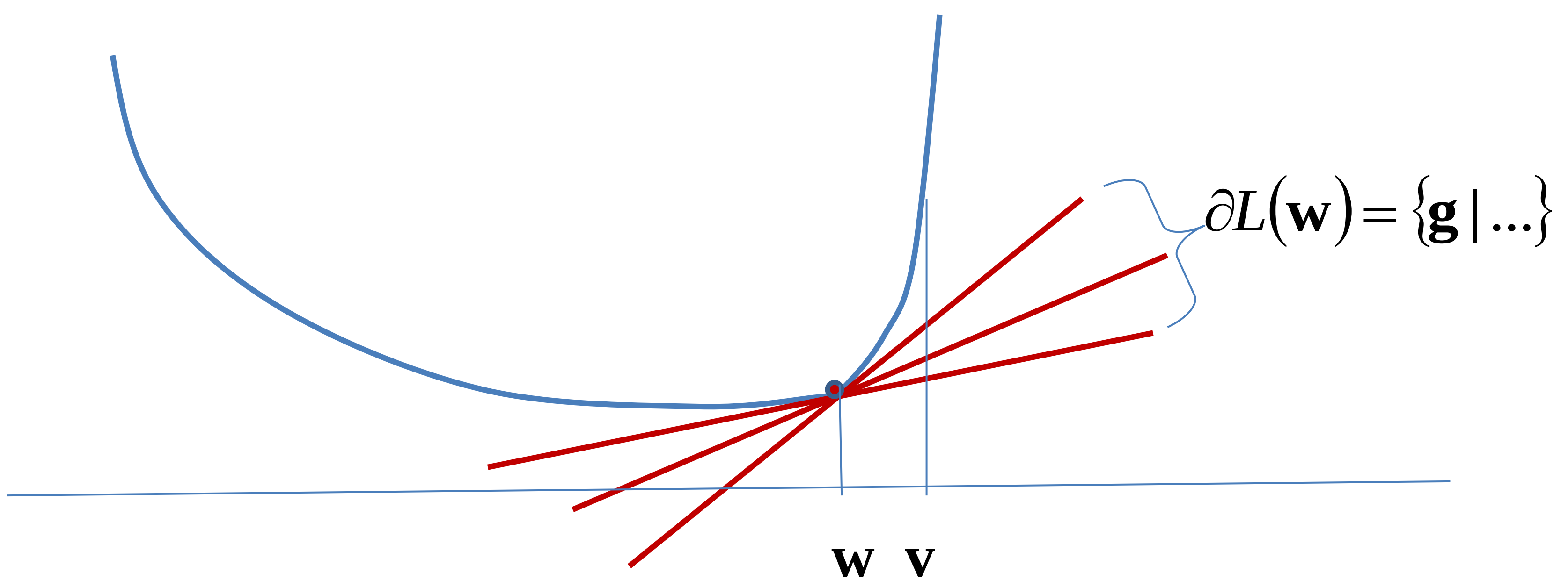
Step3　k← k+1

Step4　Step 2へ

微分可能でない場合：劣勾配(sub-gradient)の利用

凸関数 : convext : L の subgradient

$$\partial L(\mathbf{w}) = \left\{ \mathbf{g} \mid \forall \mathbf{v} \geq 0 \quad L(\mathbf{v}) \geq L(\mathbf{w}) + (\mathbf{v} - \mathbf{w})^T \mathbf{g} \right\}$$



例

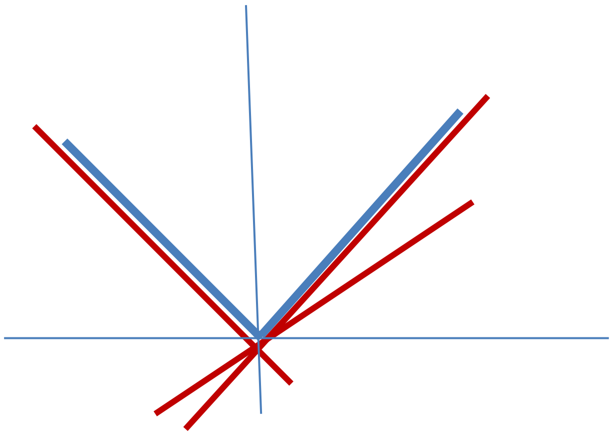
$L1 \quad loss$

$$L(w) = \|w\|_1 = |w|$$

$$\nabla L(w) = -1 \qquad w < 0$$

$$\nabla L(w) = [-1,1]$$

$$\nabla L(w) = 1 \quad w > 0$$



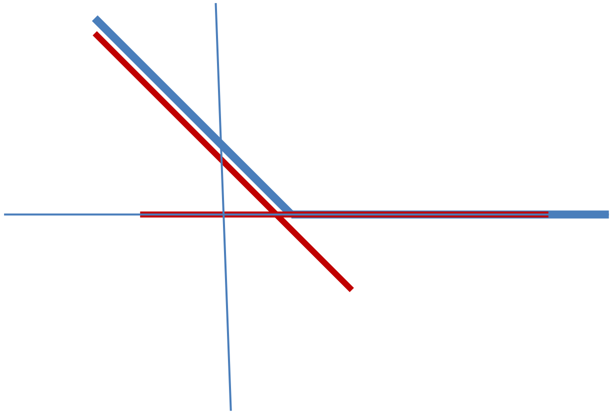
$hinge \quad loss$

$$L(w) = [1 - w]_+$$

$$\nabla L(w) = -1 \qquad w < 1$$

$$\nabla L(w) = [-1,0]$$

$$\nabla L(w) = 0 \quad w > 1$$



劣勾配降下法
Sub-Gradient Descent

初期値 $= \mathbf{w}^{(0)}; k = 0;$ $\quad \mathbf{w}^* = \arg\min L(\mathbf{w});$

$step1:$ $\quad \mathbf{w}^{(k)}$が$\mathbf{w}^*$に十分近いなら終了

（具体的には$\mathbf{w}^{(k)}$が$\mathbf{w}^{(k-1)}$に十分近くて収束した）

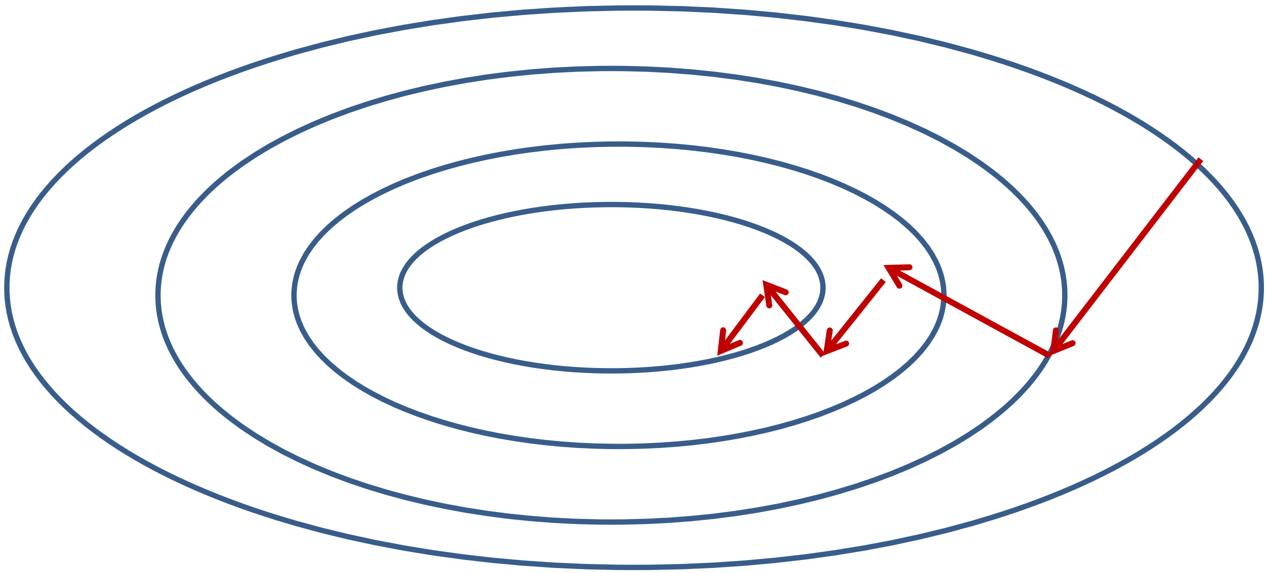
$\text{step 2 sub-gradient}: \partial L(\mathbf{w})$

$$= \{\mathbf{g} \mid \forall \mathbf{v} \geq 0 \quad L(\mathbf{v}) \geq L(\mathbf{w}) + (\mathbf{v} - \mathbf{w}) \cdot \mathbf{g}\}$$ を計算

$$step3: \quad \mathbf{w}^{(k+1)} = \underset{\mathbf{v},\ \mathbf{g} \in \partial L(\mathbf{w})}{\arg\min}\ L\left(\mathbf{v} - \left(\mathbf{w}^{(k)} - \alpha^{(k)} \mathbf{g}\right)\right)$$

$step4:$ $\quad k = k + 1$ $\quad$ として$step1$へ

最急降下法がうまくいかない場合



- ジグザグに動きながら最適解に近づくので収束する速度が遅い（効率が良くない）

- 勾配以外の情報も使ってみる→ニュートン法

ニュートン法

- 最小化したいのは損失 $L(\mathbf{w})$
  - ただし、直接には最適化できない場合を考えている
- 今$\mathbf{w}$の現在値$\mathbf{w}^{(k)}$が与えられていて、$\boldsymbol{d}$を加えることによってより最適値に近い$\mathbf{w}^{(k+1)}$を得たい。
- ここで$\mathbf{w}^{(k)}$が定数で$\boldsymbol{d}$を変数と思って
- $L(\mathbf{w}^{(k+1)}) = L(\mathbf{w}^{(k)} + \boldsymbol{d})$

の左辺を直接最適化する代わりに右辺を最小にする $\boldsymbol{d}$ を求める。

- この結果を$\boldsymbol{d}^{(k)}$とし、よりよいLを$L(\mathbf{w}^{(k)} + \boldsymbol{d}^{(k)})$として、繰り返すことによって真の最適値に近づこうとする
- ここで$L(\mathbf{w}^{(k)} + \boldsymbol{d})$を２次のテーラー近似する場合がニュートン法

➤2次のテーラー展開

$$L\left(\mathbf{w}^{(k)}+\mathbf{d}\right) \approx L\left(\mathbf{w}^{(k)}\right)+\nabla L\left(\mathbf{w}^{(k)}\right)^{\mathrm{T}}\mathbf{d}+\frac{1}{2}\mathbf{d}^{\mathrm{T}}\mathrm{H}L\left(\mathbf{w}^{(k)}\right)\mathbf{d}$$

➤右辺を$\mathbf{d}$の関数と見て最小化しニュートン方向$\mathbf{d}^{(t)}$を求める

$$\mathbf{d}^{(k)} = \underset{\mathbf{d}}{\arg\min}\ \text{右辺} = -\mathrm{H}L\left(\mathbf{w}^{(k)}\right)^{-1}\nabla L\left(\mathbf{w}^{(k)}\right)$$

ニュートン法のアルゴリズム

$step0:$ $k=0;$ 初期値 $=\mathbf{w}^{(0)}; \mathbf{w}^{*}=\arg\min L(\mathbf{w});$

$step1:$ $\mathbf{w}^{(k)}$が$\mathbf{w}^{*}$に十分近いなら終了

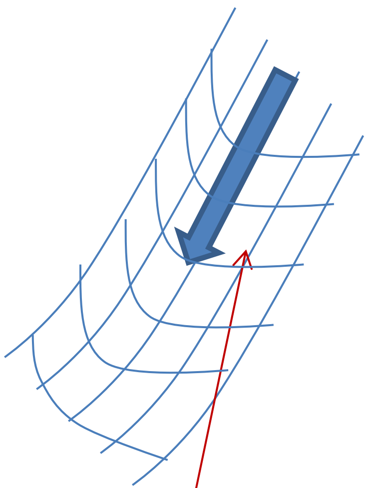
$step2:$ ニュートン方向 $\mathbf{d}^{(k)}=-\mathrm{H}L\left(\mathbf{w}^{(k)}\right)^{-1}\nabla L\left(\mathbf{w}^{(k)}\right)$を計算

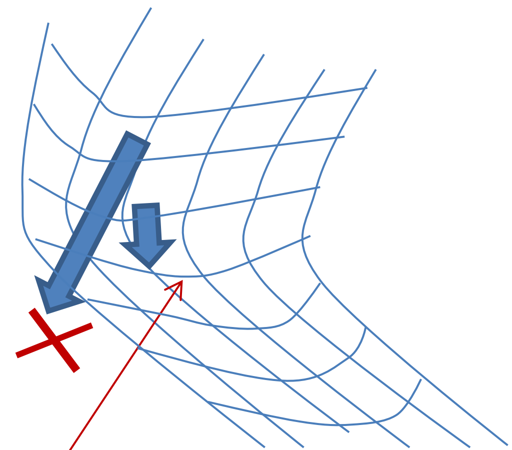
$step3:$ $\mathbf{w}^{(k+1)}=\mathbf{w}^{(k)}-\alpha^{(k)}\mathrm{H}L\left(\mathbf{w}^{(k)}\right)^{-1}\nabla L\left(\mathbf{w}^{(k)}\right)$ $\alpha^{(k)}$はstep size

$step4:$ $k=k+1$ として$step1$へ

$\alpha^{(k)}$は直線探索で決めてもよい

ニュートン法の直感的な解釈





この辺りは▽の変化すなわちヘッセ行列 $\mathrm{H}L(\mathrm{W})$が小さいので大きく進む方が効率がよい
→ $\mathrm{H}L(\mathrm{W})^{-1}$: 大

この辺りは▽の変化すなわちヘッセ行列 $\mathrm{H}L(\mathrm{W})$が大きいので大きく進むと不味い方向に行ってしまう。
→ $\mathrm{H}L(\mathrm{W})^{-1}$: 小

ニュートン法の問題点

- ニュートン法は勾配降下法より多くの場合で性能がよいが、ヘッセ行列の逆行列を繰り返しの度に計算しなければならない
- N×Nの行列の逆行列の計算には$O(N^3)$のコストがかかる。Nが大きくなると実用的でなくなる
- ヘッセ行列の逆行列を近似する行列を行列のかけ算だけによる更新式を使って$O(N^2)$で計算する方法もある。
  - BFGS公式のヘッセ逆行列版(cf. 共立出版「最適化法」p.122)

確率的勾配降下法
Stochastic Gradient Descent(SGD)

- ここまで述べてきた方法は全データを一挙に学習に用いるバッチ処理
- 1データないし少数のデータ毎に重みベクトルw学習する方法：オンライン学習に近い
  - 最終結果としては、1ないし少数データ毎の学習で得られたwの平均
  - メモリに全データを展開しなくてよい可能性もある→省メモリ→しかし、全データを何回も繰り返し使うなら省メモリにならない-
  - →1つのデータは1回しか見ない→ストリームマイニング的

- 2種類のSGD
  - Sample Approximation(SA):1データ毎の学習

SGD(SA) と　バッチ学習

$\mathbf{w}_0$

$$\mathbf{w}_1 = \mathbf{w}_0 - \alpha^{(0)} \nabla loss(\mathbf{w}_0 \text{ on } (\mathbf{x}_1, y_1))$$

$$\mathbf{w}_2 = \mathbf{w}_1 - \alpha^{(1)} \nabla loss(\mathbf{w}_1 \text{ on } (\mathbf{x}_2, y_2))$$

$$\mathbf{w}_3 = \mathbf{w}_2 - \alpha^{(2)} \nabla loss(\mathbf{w}_2 \text{ on } (\mathbf{x}_3, y_3))$$

$\vdots$

$$\mathbf{w}_i = \mathbf{w}_{i-1} - \alpha^{(i-1)} \nabla loss(\mathbf{w}_{i-1} \text{ on } (\mathbf{x}_i, y_i))$$

$\vdots$

$$\mathbf{w}_m = \mathbf{w}_{m-1} - \alpha^{(m-1)} \nabla loss(\mathbf{w}_{m-1} \text{ on } (\mathbf{x}_m, y_m))$$

$\Downarrow$

$$\boxed{\mathbf{w} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{w}_i}$$

$\mathbf{x}_1, y_1$
$\mathbf{x}_2, y_2$
$\mathbf{x}_3, y_3$
:
$\mathbf{x}_\mathrm{i}, y_\mathrm{i}$
:
$\mathbf{x}_\mathrm{m}, y_\mathrm{m}$

このデータ生成が確率的

$\mathbf{w}^{(k)}$

$\nabla loss(\mathbf{w}^{(k)} \text{ on } (\mathbf{x}_1, y_1))$

$\nabla loss(\mathbf{w}^{(k)} \text{ on } (\mathbf{x}_2, y_2))$

$\nabla loss(\mathbf{w}^{(k)} \text{ on } (\mathbf{x}_3, y_3))$

$\vdots$

$\nabla loss(\mathbf{w}^{(k)} \text{ on } (\mathbf{x}_i, y_i))$

$\vdots$

$\nabla loss(\mathbf{w}^{(k)} \text{ on } (\mathbf{x}_m, y_m))$

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \alpha^{(k)}\left(\frac{1}{m}\sum_{i=1}^{m} \nabla loss(\mathbf{w}^{(k)} \text{ on } (\mathbf{x}_i, y_i))\right)$$

さらにkを$+1$して上の処理を繰り返す

Sample Average Approximation(SAA)
SA inside SAA

- 全データからランダムにm個のデータをサンプルしてm個のデータからなるデータ集合を作る。
- このようにして異なるデータ集合をk個作る。
- データ集合(i)にSGD(SA)を適用して学習し、重みベクトルw(i)を計算
- 最終結果 $$\mathbf{w} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{w}(i)$$

例：SGDによるサポートベクターマシン

$$\min_{\|\mathbf{w}\|_2<B} \frac{1}{m}\sum_{i=1}^{m}\left[1-y_i\left(\mathbf{w}^T\mathbf{x}_i\right)\right]_+ = \min_{\mathbf{w}}\left(\frac{1}{m}\sum_{i=1}^{m}\left[1-y_i\left(\mathbf{w}^T\mathbf{x}_i\right)\right]_+ + \frac{\lambda}{2}\|\mathbf{w}\|_2^2\right)$$

に注意するとSVDは以下のようになる

初期値：$\mathbf{w}^{(0)}$, 全データ数 : n

以下をk回繰り返す

$\{i$を$[1..n]$からランダムに生成

$\text{if } y_i\left(\mathbf{w}^T\mathbf{x}_i\right)<1 \text{ then } \mathbf{w} \leftarrow \mathbf{w} + \alpha y_i \mathbf{x}_i$

$\text{if } \|\mathbf{w}\|_2 \geq B \text{ then } \mathbf{w} \leftarrow B\mathbf{w}/\|\mathbf{w}\|_2$

$\mathbf{w}_{sum} = \mathbf{w}_{sum} + \mathbf{w}\}$

結果：$\mathbf{w} = \mathbf{w}_{sum}/k$