Machine Learning Midterm Assignment

Keisuke Okumura, 18M31013

2018 年 7 月 16 日

本レポートで用いたソースコードは全て https://github.com/Kei18/H-ML-report.git に掲載してある.

Problem 1

ロジスティック回帰では次の問題を解くことになる.

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}}\, J(\boldsymbol{w})$$

$$J(\boldsymbol{w}) := \sum_{i=1}^{n} (\ln\left(1 + \exp\left(-y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i\right)\right)) + \lambda \boldsymbol{w}^{\mathrm{T}} \boldsymbol{w}$$

最急降下法 (steepest gradient method) では, 以下の更新式に従って学習を行う. α は定数とする.

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \alpha \left.\frac{\partial J}{\partial \boldsymbol{w}}\right|_{\boldsymbol{w}=\boldsymbol{w}^{(t)}}$$

$$\begin{aligned}\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} &= \frac{1}{n} \sum_{i=1}^{n} \frac{\exp(-y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i)(-y_i \boldsymbol{x}_i)}{1 + \exp(-y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i)} + 2\lambda \boldsymbol{w} \\ &= \frac{1}{n} \sum_{i=1}^{n} \frac{-y_i \boldsymbol{x}_i}{1 + \exp(y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i)} + 2\lambda \boldsymbol{w}\end{aligned}$$

また, ニュートン法 (newton method) では以下の更新式に従って学習を行う. α は定数とする.

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} + \alpha \boldsymbol{d}^{(t)}$$

$$\nabla^2 J(\boldsymbol{w}^{(t)}) \boldsymbol{d}^{(t)} = -\nabla J(\boldsymbol{w}^{(t)}) \Longleftrightarrow \boldsymbol{d}^{(t)} = -(\nabla^2 J(\boldsymbol{w}^{(t)}))^{-1} \nabla J(\boldsymbol{w}^{(t)})$$

$\nabla^2 J(\boldsymbol{w})$ は次のように計算される.

$$\nabla^2 J(\boldsymbol{w}) = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{\exp(-y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i)}{(1 + \exp(-y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i))^2} \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}} \right) + 2\lambda \boldsymbol{I}$$

100 個のデータからなるデータセットを Toy Dataset II を参考に生成し, 最急降下法, 及びニュートン法を実行した結果を図. 1 に示す. また, その時の学習の進行の様子を図. 2 に示す. パラメータ更新の繰り返し数は 50 回, $\alpha = 0.1$, $\lambda = 0.1$ として学習を行った.

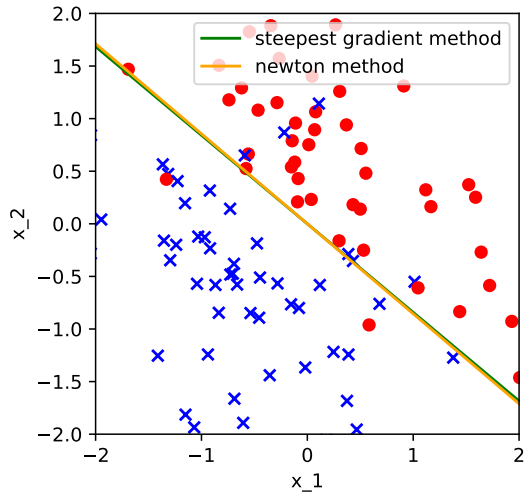


図 1: ロジスティック回帰の学習結果. $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} = 0$ を描画.

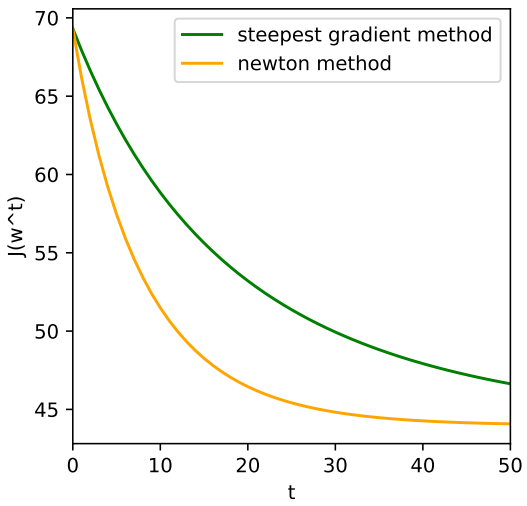


図 2: 損失関数の値の変化.

Problem 2

本問題では, *lasso* を用いて次の問題を考える.

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\mathrm{argmin}}((\boldsymbol{w}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{w}-\boldsymbol{\mu}) + \lambda||\boldsymbol{w}||_1)$$

解を得るために, Proximal gradient method (PG) を用いる. $\psi = (\boldsymbol{w}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{w}-\boldsymbol{\mu})$ とする. 更新式は次のようになる.

$$w_i^{(t+1)} = prox_{\eta_t\lambda||\cdot||_1}\left(\boldsymbol{w}^{(t)} - \eta_t\nabla\psi(\boldsymbol{w}^{(t)})\right)$$

ここで, 学習率 $\eta_t = \gamma^{-1}$ とする. γ は ψ のリプシッツ定数とし, $2\boldsymbol{A}$ の固有値の最大値をとることができる. 上式は L1 ノルムの近接写像であるから, Soft Threshold 関数を用いて, 次のように変換される.

$$w_i^{(t+1)} = ST_{\frac{\lambda}{\gamma}}\left(w_i^{(t)} - \left\{\frac{1}{\gamma}\nabla\psi(\boldsymbol{w}^{(t)})\right\}_i\right), \quad ST_q(\mu) = \begin{cases}\mu - q & (if\ \mu > q)\\ 0 & (if\ |\mu| \le q)\\ \mu + q & (if\ \mu < -q)\end{cases}$$

ψ を二次形式とみなし $\boldsymbol{A}$ を対称行列とすると, $\nabla\psi$ は次のようになる.

$$\nabla\psi = (\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}})(\boldsymbol{w}-\boldsymbol{\mu}) = 2\boldsymbol{A}(\boldsymbol{w}-\boldsymbol{\mu})$$

以上より, 更新式は次式で表される.

$$w_i^{(t+1)} = ST_{\frac{\lambda}{\gamma}}\left(w_i^{(t)} - \left\{\frac{1}{\gamma}2\boldsymbol{A}(\boldsymbol{w}^{(t)}-\boldsymbol{\mu})\right\}_i\right)$$

また, Accelerated proximal gradient (APG) を用いた場合, 更新式は次のように変化する.

$$\boldsymbol{w}^{(t+1)} = prox_{\eta_t\lambda||\cdot||_1}\left(\boldsymbol{v}^{(t)} - \eta_t\nabla\psi(\boldsymbol{v}^{(t)})\right)$$

$$\boldsymbol{v}^{(t)} = \boldsymbol{w}^{(t)} + q(t)(\boldsymbol{w}^{(t)} - \boldsymbol{w}^{(t-1)})$$

$q(t)$ はスカラー関数である．今回は次式を用いる．

$$q(t) = \frac{t-1}{t+2}$$

以上を踏まえて, 以下の条件で学習を行う．

$$\boldsymbol{A} = \begin{pmatrix} 3 & 0.5 \\ 0.5 & 1 \end{pmatrix}, \boldsymbol{\mu} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

繰り返し数を 50 回, λ を 2, 4, 6 と変えた時の学習の様子を図. 3 に示す．さらに, Python の数理最適化ライブラリ CVXOPT を使用して最適解 $\hat{\boldsymbol{w}}$ を導出し, $||\boldsymbol{w}^{(t)} - \hat{\boldsymbol{w}}||_1$ を計算して結果を図. 4 に示す．

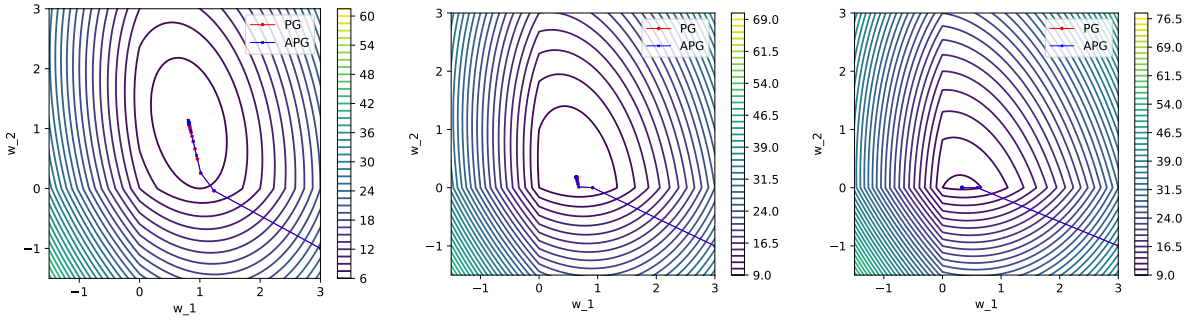


図 3: *lasso* を用いて学習を行った様子．左から $\lambda = 2, 4, 6$ となっている．

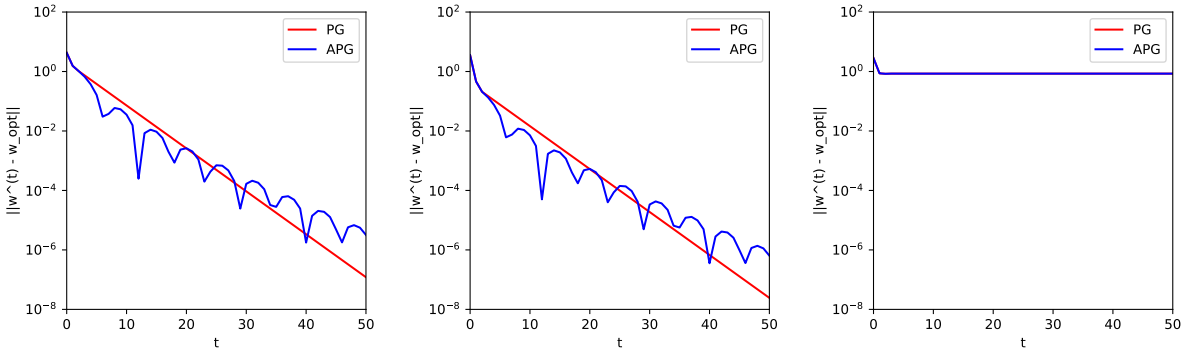


図 4: *lasso* を用いて学習した時の $||\boldsymbol{w}^{(t)} - \hat{\boldsymbol{w}}||_1$ の値の変化．左から $\lambda = 2, 4, 6$ となっている．$\lambda = 6$ の時, PG と APG の線はほとんどかぶっている．

AdaGrad を使用した場合, *lasso* の更新式は次のようになる．

$$\boldsymbol{w}^{(t)} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \left((\boldsymbol{w} - \boldsymbol{w}^{(t-1)})^{\mathrm{T}} \boldsymbol{g}_t + \lambda ||\boldsymbol{w}||_1 + \frac{1}{2\eta_0} ||\boldsymbol{w} - \boldsymbol{w}^{(t-1)}||^2_{\boldsymbol{H}_t} \right)$$

ただし, $\boldsymbol{g}_t = \nabla \psi(\boldsymbol{w}^{(t-1)})$ である．$\boldsymbol{H}_t$ は次のように定義される．

$$\boldsymbol{H}_t = \boldsymbol{G}_t^{1/2} + \epsilon \boldsymbol{I}, \quad \{\boldsymbol{G_t}\}_{i,j} = \delta_{ij} \sum_{\tau=1}^{t} \{\boldsymbol{g}_\gamma\}_i^2$$

更新式は Soft Threshold 関数を用いると, 次のように表される.

$$\boldsymbol{w}_i^{(t)} = ST_{\eta_0 \lambda \{\boldsymbol{H}_t^{-1}\}_{ii}} \left(\boldsymbol{w}_i^{(t-1)} - \eta_0 \{\boldsymbol{H}_t^{-1} \boldsymbol{g}_t\}_i \right)$$

PG, APG, AdaGrad の比較を行う. 以下の条件で学習を行う.

$$\boldsymbol{A} = \begin{pmatrix} 250 & 15 \\ 15 & 4 \end{pmatrix}, \boldsymbol{\mu} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \lambda = 0.89$$

AdaGrad のパラメータとして, $\eta_0 = 500/\gamma$, $\epsilon = 0.02$ を用いた.

以上を踏まえて学習を行う. 繰り返し数を 500 回とした時の学習の様子と, CVXOPT により導出された最適解 $\hat{\boldsymbol{w}}$ を用いて $||\boldsymbol{w}^{(t)} - \hat{\boldsymbol{w}}||_1$ を計算し, その結果を図. 5 に示す.

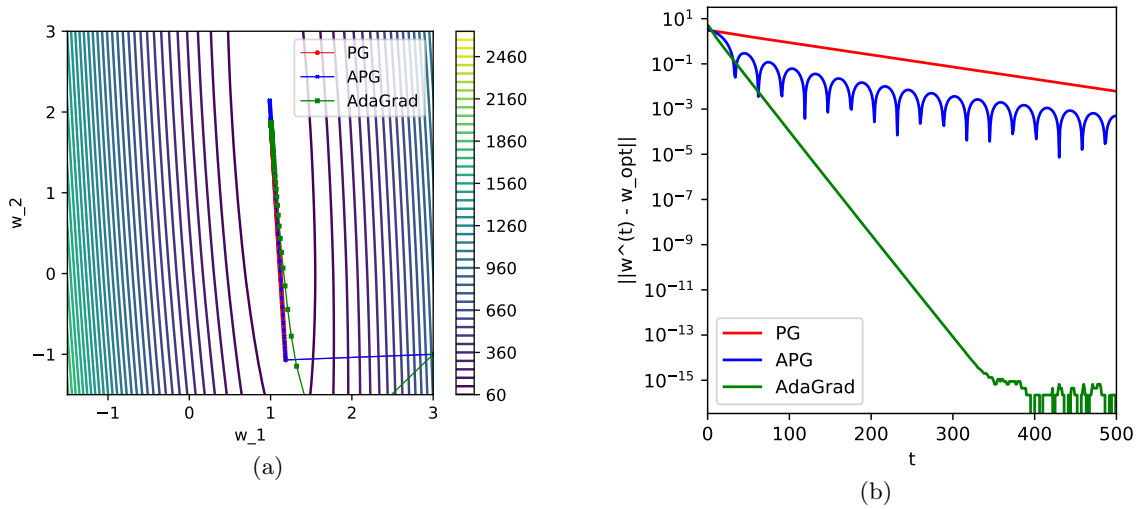


図 5: **a)** *lasso* を用いて学習を行った様子. **b)** $||\boldsymbol{w}^{(t)} - \hat{\boldsymbol{w}}||_1$ の値の変化.

Problem 3

support vector machine では次の問題を考える. $\lambda > 0$ とする.

$$\hat{\boldsymbol{w}} = \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \left(\sum_{i=1}^{n} max(0, 1 - y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i) + \lambda ||\boldsymbol{w}||_2^2 \right) \tag{1}$$

$\xi_i \geq max(0, 1 - y_i \boldsymbol{w}\mathrm{T}\boldsymbol{x}_i) \geq 0$ となる $\boldsymbol{\xi} = (\xi_1, ..., \xi_n)^{\mathrm{T}}$ を用いて, 上記の問題を以下の最適化問題に置き換える. $C = 1/2\lambda$ とする.

$$\begin{aligned} minimize_{\boldsymbol{w}, \boldsymbol{\xi}} \quad & \frac{1}{2} \boldsymbol{w}^{\mathrm{T}} \boldsymbol{w} + C \boldsymbol{1}^{\mathrm{T}} \boldsymbol{\xi} \\ subject\ to \quad & \xi_i \geq 1 - y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i \\ & \xi_i \geq 0 \end{aligned}$$

Lagrangian を求めると以下のようになる.

$$L(\boldsymbol{w}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2} \boldsymbol{w}^{\mathrm{T}} \boldsymbol{w} + C \boldsymbol{1}^{\mathrm{T}} \boldsymbol{\xi} + \sum_i \alpha_i (1 - y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i - \xi_i) + \sum_i \beta_i (-\xi_i)$$

KKT 条件を求める.

$$\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{w}} = 0 \iff \hat{\boldsymbol{w}} &= \sum_i \alpha_i y_i \boldsymbol{x}_i \\
\frac{\partial L}{\partial \boldsymbol{\xi}} = 0 \iff \hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{\beta}} &= C\mathbf{1} \\
\hat{\alpha}_i &\geq 0 \\
\hat{\beta}_i &\geq 0 \\
\hat{\alpha}_i(1 - y_i \hat{\boldsymbol{w}}^{\mathrm{T}} \boldsymbol{x}_i - \hat{\xi}_i) &= 0 \\
\hat{\beta}_i(-\hat{\xi}_i) &= 0
\end{aligned}$$

次に, Lagrange の双対関数を求める. $\hat{\boldsymbol{w}} = \sum_i \alpha_i y_i \boldsymbol{x}_i$, $\hat{\boldsymbol{\beta}} = C\mathbf{1} - \hat{\boldsymbol{\alpha}}$ を利用する.

$$\begin{aligned}
\tilde{L}(\boldsymbol{\alpha}, \boldsymbol{\beta}) &= \inf_{\boldsymbol{w}, \boldsymbol{\xi} \in D} L(\boldsymbol{w}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \\
&= \frac{1}{2} \left(\sum_i \alpha_i y_i \boldsymbol{x}_i \right)^{\mathrm{T}} \left(\sum_i \alpha_i y_i \boldsymbol{x}_i \right) + C\mathbf{1}^{\mathrm{T}} \boldsymbol{\xi} \\
&\quad + \sum_i \alpha_i \left(1 - y_i \left(\sum_j \alpha_j y_j \boldsymbol{x}_j \right)^{\mathrm{T}} \boldsymbol{x}_i - \xi_i \right) \\
&\quad + \sum_i (C - \alpha_i)(-\xi_i) \\
&= \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j \\
&= -\frac{1}{2} \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{K} \boldsymbol{\alpha} + \boldsymbol{\alpha}^{\mathrm{T}} \mathbf{1}
\end{aligned}$$

したがって双対問題は次のように表される.

$$\begin{aligned}
maximize_{\boldsymbol{\alpha}} \quad & -\frac{1}{2} \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{K} \boldsymbol{\alpha} + \boldsymbol{\alpha}^{\mathrm{T}} \mathbf{1} \\
subject\ to \quad & \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1} = \frac{1}{2\lambda} \mathbf{1}
\end{aligned}$$

ただし, $\{\boldsymbol{K}\}_{i,j} = y_i y_j \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j$ とする. ここで $\boldsymbol{\alpha} \to \boldsymbol{\alpha}/2\lambda$ とすると, 上記の双対問題を以下のように変形することができ, これが解くべき問題である.

$$\begin{aligned}
maximize_{\boldsymbol{\alpha}} \quad & -\frac{1}{4\lambda} \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{K} \boldsymbol{\alpha} + \boldsymbol{\alpha}^{\mathrm{T}} \mathbf{1} && (2) \\
subject\ to \quad & \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{1} && (3)
\end{aligned}$$

また, 導出の過程で出てきた $\hat{\boldsymbol{w}}$ にも $\boldsymbol{\alpha} \to \boldsymbol{\alpha}/2\lambda$ を適用すると, 最適解は次のように書き下すことができる.

$$\hat{\boldsymbol{w}} = \frac{1}{2\lambda} \sum_i \alpha_i y_i \boldsymbol{x}_i$$

続いて, projected gradient method を用いて (2) 式の解を求める. $\psi(\boldsymbol{\alpha}) = \frac{1}{4\lambda} \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{K} \boldsymbol{\alpha} - \boldsymbol{\alpha}^{\mathrm{T}} \mathbf{1}$ とすると, ラグランジュの双対問題は $\psi(\boldsymbol{\alpha})$ の最小化問題とみなすことができる.

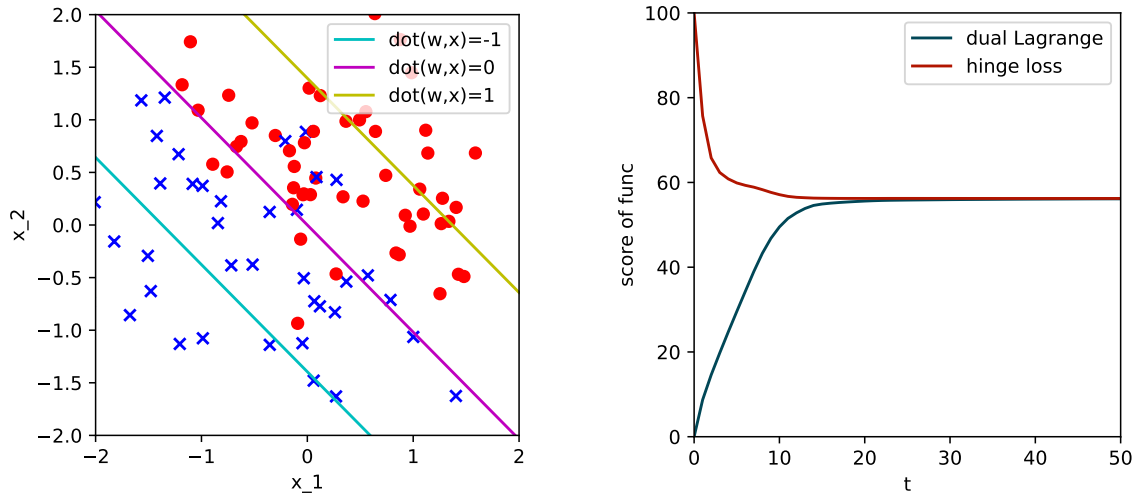
projected gradient の更新式は以下の通り.

$$\begin{aligned}\boldsymbol{\alpha}^{(t)} &= P_{[0,1]^n}(\boldsymbol{\alpha}^{(t-1)} - \eta_t \nabla\psi(\boldsymbol{\alpha}^{(t-1)})) \\ &= P_{[0,1]^n}\left(\boldsymbol{\alpha}^{(t-1)} - \eta_t\left(\frac{1}{2\lambda}\boldsymbol{K}\boldsymbol{\alpha} - \mathbf{1}\right)\right)\end{aligned}$$

$P_{[0,1]^n}$ は $[0,1]$ への射影演算子であり, 具体的には次のような写像である.

$$\{P_{[0,1]^n}(\tilde{\boldsymbol{\alpha}})\}_i = \begin{cases} 0 & (if\ \tilde{\alpha}_i < 0) \\ \tilde{\alpha}_i & (if\ 0 \leq \tilde{\alpha}_i \leq 1) \\ 1 & (if\ 1 < \tilde{\alpha}_i) \end{cases}$$

以上を踏まえて学習を行う. 繰り返し数を 50 回, $\lambda = 10$, $\eta_t = 0.1$ とした時の学習の様子を図 6 に示す.



(a) (b)

図 6: **a)** support vector machine を用いて学習を行った様子. $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} = -1, 0, 1$ を描画. **b)** ヒンジ損失関数と正則化項の和 - (1) 式と, ラグランジュの双対関数 - (2) 式の値の変化の様子. duality gap が 0 に収束していく様子が見られる.

Problem 4

ヒンジ損失関数と L1 正則化項を用いて, binary classification の問題を考える. 具体的には次の問題を考える.

$$\hat{\boldsymbol{w}} = \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \left(\sum_{i=1}^{n} max(0, 1 - y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i) + \lambda ||\boldsymbol{w}||_1 \right) \tag{4}$$

$0 \leq \max(0, 1 - y_i \boldsymbol{w}\mathrm{T}\boldsymbol{x}_i) \leq \xi_i$ となる $\boldsymbol{\xi} = (\xi_1, ..., \xi_n)^{\mathrm{T}}$ と, $e_i \geq |w_i| \geq 0$ となる $\boldsymbol{e} = (e_1, ..., e_n)^{\mathrm{T}}$ を用いて, 上記の問題を以下の最適化問題に置き換える.

$$
\begin{aligned}
minimize_{\boldsymbol{w},\boldsymbol{x}} \quad & \mathbf{1}^{\mathrm{T}}\boldsymbol{\xi} + \lambda \mathbf{1}^{\mathrm{T}}\boldsymbol{e} \\
subject\ to \quad & \xi_i \geq 1 - y_i \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i \\
& \xi_i \geq 0 \\
& e_i \geq -w_i \\
& e_i \geq w_i \\
& e_i \geq 0
\end{aligned}
$$

ここで, $\boldsymbol{z}$, $\boldsymbol{c}$, $\boldsymbol{A}$, $\boldsymbol{b}$ を以下のように定義する.

$$
\boldsymbol{z} = \begin{pmatrix} \boldsymbol{\xi} \\ \boldsymbol{e} \\ \boldsymbol{w} \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} \mathbf{1}^n \\ \lambda \mathbf{1}^d \\ \mathbf{0}^d \end{pmatrix}
$$

$$
\boldsymbol{A} = \begin{pmatrix}
-1 & & 0 & 0 & \cdots & 0 & -y_1 x_{11} & \cdots & -y_1 x_{1d} \\
& \ddots & & \vdots & & \vdots & \vdots & & \vdots \\
0 & & -1 & 0 & \cdots & 0 & -y_n x_{n1} & \cdots & -y_n x_{nd} \\
-1 & & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
& \ddots & & \vdots & & \vdots & \vdots & & \vdots \\
0 & & -1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & \cdots & 0 & -1 & & 0 & -1 & & 0 \\
\vdots & & \vdots & & \ddots & & & \ddots & \\
0 & \cdots & 0 & 0 & & -1 & 0 & & -1 \\
0 & \cdots & 0 & -1 & & 0 & 1 & & 0 \\
\vdots & & \vdots & & \ddots & & & \ddots & \\
0 & \cdots & 0 & 0 & & -1 & 0 & & 1 \\
0 & \cdots & 0 & -1 & & 0 & 0 & \cdots & 0 \\
\vdots & & \vdots & & \ddots & & \vdots & & \vdots \\
0 & \cdots & 0 & 0 & & -1 & 0 & \cdots & 0
\end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix}
-1 \\ \vdots \\ -1 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0
\end{pmatrix}
$$

上記を用いて, 最小化問題は次のように書き下すことができる.

$$
\begin{aligned}
minimize_{\boldsymbol{z}} \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{z} \\
subject\ to \quad & \boldsymbol{A}\boldsymbol{z} \leq \boldsymbol{b}
\end{aligned}
$$

続いて, proximal gradient method を用いた学習では, $\psi = \sum_{i=1}^{n} max(0, 1 - y_i \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)$ とおくと, 更新式は次のようになる.

$$
\boldsymbol{w}^{(t+1)} = prox_{\eta_t \lambda ||\cdot||_1} \left(\boldsymbol{w}^{(t)} - \eta_t \nabla \psi(\boldsymbol{w}^{(t)}) \right)
$$

ところで ψ は微分可能ではない. そこで劣微分を用いる. $l_i = max(0, 1 - y_i \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)$ とおくと, $\boldsymbol{g}_i \in \partial l_i(\boldsymbol{w})$ は次のように定義される.

$$
\boldsymbol{g}_i = \begin{cases}
0 & (if\ y_i \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i > 1) \\
-\theta y_i \boldsymbol{x}_i & (if\ y_i \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i = 1),\ \ (0 \leq \theta \leq 1) \\
-y_i \boldsymbol{x}_i & (if\ y_i \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i < 1)
\end{cases}
$$

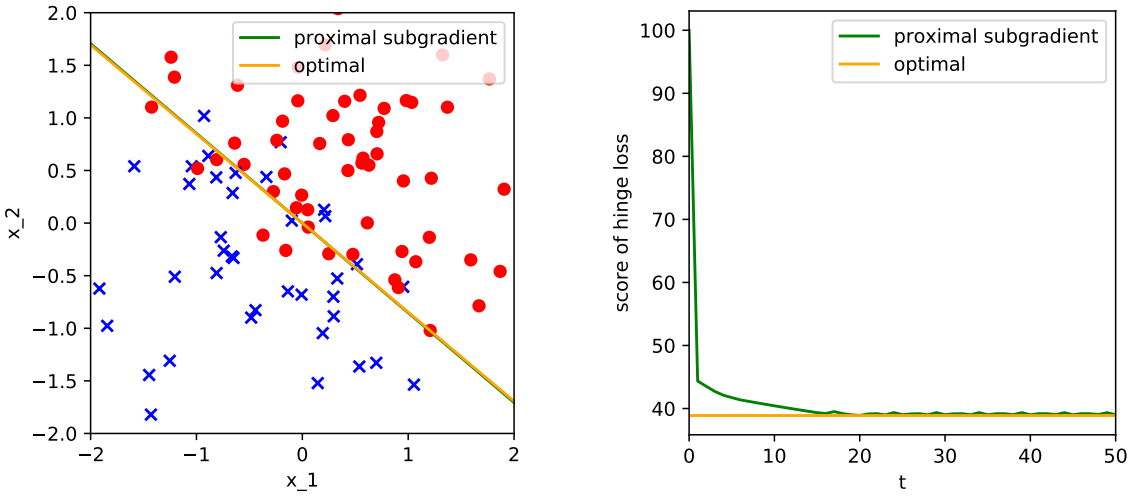
劣微分を用いると, 更新式は次のようになる.

$$w_i^{(t+1)} = prox_{\eta_t \lambda ||\cdot||_1} \left(\boldsymbol{w}^{(t)} - \eta_t \sum_{i=0}^{n} \boldsymbol{g}_i \right)$$

上式は L1 ノルムの近接写像であるから, Soft Threshold 関数を用いて, 次のように変換される.

$$w_i^{(t+1)} = ST_{\eta_t \lambda} \left(w_i^{(t)} - \left\{ \eta_t \sum_{j=0}^{n} \boldsymbol{g}_j \right\}_i \right)$$

100 個のデータからなるデータセットを Toy Dataset II を参考に生成し, proximal subgradient method を用いて学習した結果およびヒンジ損失関数と正則化項の和の値の変化を図. 7 に示す. 同時に CVXOPT を用いて算出した最適解も示す. パラメータ更新の繰り返し数は 50 回, $\eta_t = 0.05$, $\lambda = 1$, $\theta = 0.5$ として学習を行った.



(a) (b)

図 7: **a)** proximal subgradient method を用いて学習を行った様子. $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} = 0$ を描画. **b)** ヒンジ損失関数と正則化項の和の値の変化.

授業に対するコメント

- 講義スライドの PDF で文字検索 (Preview.app, OSX) ができない. スライド枚数が多く, どこに何があるか検索したいことが多々あるので対応してほしい. (e.g. AdaGrad, lasso)
- 深層学習の部分 (岡崎先生のパート) を機械学習の講義から独立させて, 集中講義等でも良いので深層学習だけを扱った講義を作成してもいいのではないか. 下坂先生のパートは非常にボリュームがあり, 1Q 分の授業内容に十分足り得ると思うし, 率直にもっと講義を聞きたかった. また, 講義をもう少しじっくり進めてほしいこと, 理論だけでなく実装も同時に行えるような進め方をしてほしいこと等の印象もある.