機械学習　レポート

19M30258 田中　紘

実装に関しては、github のリポジトリ においてあります。

1. Problem 1

1.1. 1.1

batch steepest gradient method をにおける重み w の更新式は，学習率 μ を用いて以下のように定義される．

$$w^{(t+1)} = w^{(t)} - \mu \frac{\partial J(w^{(t)})}{\partial w}$$

以下の終了条件を満たすまで、更新を続ける。

$$|J(w^t) - J(w^{(t+1)})| < \epsilon$$

1.2. 1.2

Newton method における重みw の更新式は，ヘッセ行列 $\nabla J(w)$ を用いて以下のように定義される．

$$w^{(t+1)} = w^{(t)} - \nabla J(w^{(t)})^{-1} \cdot J(w^{(t)})$$

以下の終了条件を満たすまで、更新を続ける。

$$|J(w^t) - J(w^{(t+1)})| < \epsilon$$

1.3. 1.3

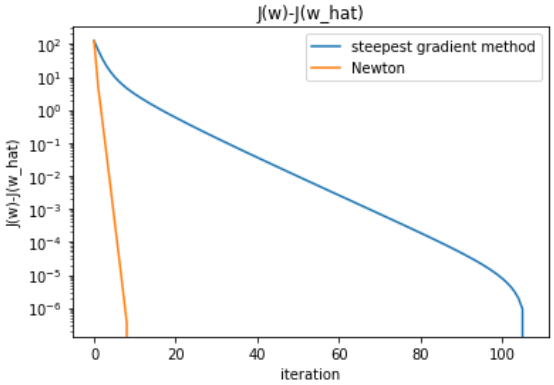
それぞれの手法について，訓練終了時の重みを$\hat{w}$とし、各反復ごとの$J(w^{(t)})$と、$J(w^{(t)}) - J(\hat{w})$の値の変化をプロットする．
横軸は反復回数，縦軸は$J(w^{(t)}) - J(\hat{w})$の値を対数スケールで表す．

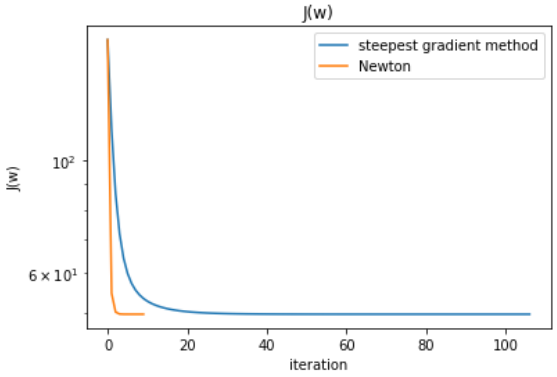
各種、設定について述べる。 デートセットに関して、Toy Datasets の Dataset IV を用いた。

両手法とも,重みwの初期値として$w = (1,1,1,1,1)$,正規化項に出現する定数について$\lambda = 1$, 終了判定に用いる定数について$\epsilon = 1e - 6$ とした。

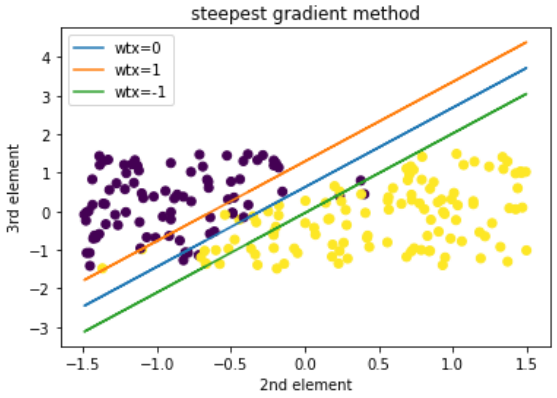
batch steepest gradient method の学習率は 0.01 と設定した。

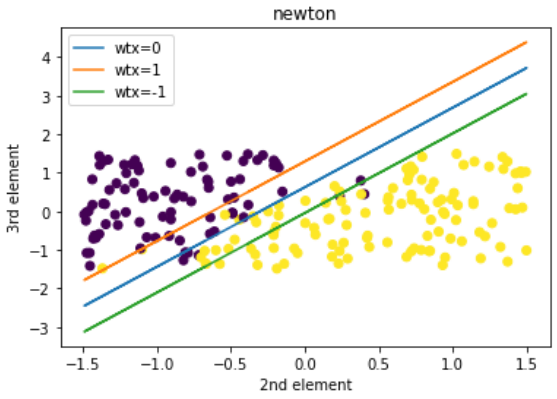
結果は以下のようになった。batch steepest gradient method が 110 回ほどの反復で終了条件を満たしたのに対し、Newton method では 5 回以内の反復でほぼ値が収束し,10 回程度の反復で終了条件を満たしており、収束の速さがわかる.また、最終的にほぼ同じ損失に収束したことがわかる.





また、以下のように、データセットとそれぞれの回帰直線を第 2 成分と第 3 成分を用いて可視化した。





両手法が重みについても、ほぼ同じ値に収束し、分類ができていることがわかる。

1.4. 1.4

batch steepest gradient method の実装、評価のみ行った。 表記の簡潔化のため、正解ラベルyから one-hot 化した行列Yを考える。

$$(Y)_{i,j} = [[y_i == j]]$$

多クラスロジスティック回帰についての損失関数に関して、以下のようにかける。

$$J(W) = -\sum_i \log(softmax(X_i W)_{y_i}) + \lambda \parallel W \parallel_2^2$$

$$\frac{\partial J(W)}{\partial W} = X^T(Y - softmax(XW)) + 2\lambda W$$

batch steepest gradient method をにおける重み w の更新式は，学習率 μ を用いて以下のように定義される．

$$W^{(t+1)} = W^{(t)} - \mu\frac{\partial J(W^{(t)})}{\partial W}$$

$$= W^{(t)} - \mu(X^T(Y - softmax(XW)) + 2\lambda W)$$

以下の終了条件を満たすまで、更新を続ける。

$$ |J(W^t) - J(W^{(t+1)})| < \epsilon $$

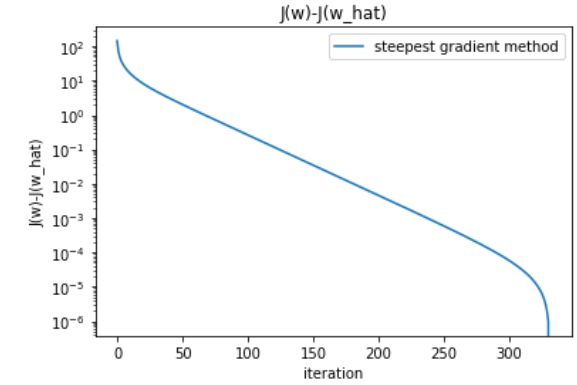
batch steepest gradient method について，1.3 と同様、訓練終了時の重みを$\hat{w}$とし、各反復ごとの$J(w^{(t)})$と、$J(w^{(t)}) - J(\hat{w})$の値の変化をプロットする．
横軸は反復回数，縦軸は$J(w^{(t)}) - J(\hat{w})$の値を対数スケールで表す．

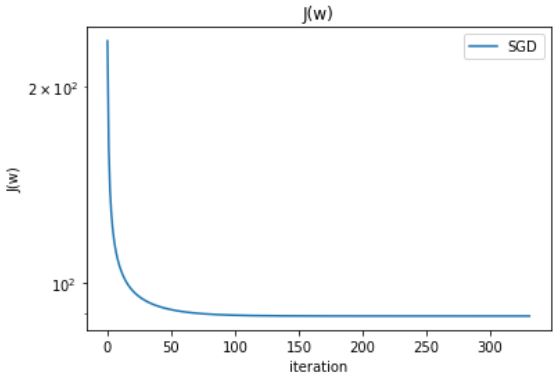
各種、設定について述べる。 データセットに関して、Toy Datasets の Dataset V を用いた。

両手法とも,重みwの初期値として$w = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$,正規化項に出現する定数について$\Lambda = 1$ とした。

学習率は 0.01 と設定した。

結果は以下のようになった。 2 クラスの時と比べて収束までの反復回数が増えていることがわかる。





2. Problem 3

2.1. 3.1

X, Yを次のよう定義する。

$$X = (\mathbf{x_1}, \mathbf{x_2}, \ldots \mathbf{x_n}).T$$

$$(Y)_{i,j} = \begin{cases} 0 & (i \neq j) \\ y_i & (i = j) \end{cases}$$

主問題は以下のようになる。

$$\begin{aligned} \underset{\mathbf{w},\boldsymbol{\xi}}{\text{minimize}} \quad & \lambda\mathbf{w}^T\mathbf{w} + \mathbf{1}\boldsymbol{\xi} \\ \text{subject to} \quad & \boldsymbol{\xi} \geq \mathbf{0} \\ & \boldsymbol{\xi} \geq \mathbf{1} - (\mathbf{w}^T X^T Y)^T \end{aligned}$$

ラグランジェ関数:Lは以下のようになる。

$$L(\mathbf{w}, \boldsymbol{\xi}, \boldsymbol{\mu}, \boldsymbol{\nu})$$

$$= \lambda\mathbf{w}^T\mathbf{w} + \mathbf{1}^T\boldsymbol{\xi} - \boldsymbol{\mu}^T\boldsymbol{\xi} - \boldsymbol{\nu}^T(\boldsymbol{\xi} - (\mathbf{1} - (\mathbf{w}^T X^T Y)^T))$$

ラグランジェ双対関数:$\tilde{L}$は以下のようになる。

$$\begin{aligned} \tilde{L}(\boldsymbol{\mu}, \boldsymbol{\nu}) &= \inf_{\mathbf{w},\boldsymbol{\xi} \in D} L(\mathbf{w}, \boldsymbol{\xi}, \boldsymbol{\mu}, \boldsymbol{\nu}) \\ &= \inf_{\mathbf{w},\boldsymbol{\xi} \in D} \lambda(\mathbf{w} - \frac{1}{2\lambda}X^T Y\boldsymbol{\nu})^T(\mathbf{w} - \frac{1}{2\lambda}X^T Y\boldsymbol{\nu}) + (\mathbf{1} - \boldsymbol{\mu} - \boldsymbol{\nu})^T\boldsymbol{\xi} \\ &\quad - \frac{1}{4\lambda}(X^T Y\boldsymbol{\nu})^T(X^T Y\boldsymbol{\nu}) + \mathbf{1}^T\boldsymbol{\nu} \\ &= \begin{cases} -\frac{1}{4\lambda}(X^T Y\boldsymbol{\nu})^T(X^T Y\boldsymbol{\nu}) + \mathbf{1}^T\boldsymbol{\nu} & ((\mathbf{1} - \boldsymbol{\mu} - \boldsymbol{\nu}) > \mathbf{0}) \\ -\infty & (else) \end{cases} \end{aligned}$$

双対問題は$\boldsymbol{\nu} \geq 0, \boldsymbol{\mu} \geq 0$条件下での$\tilde{L}$の最大化問題である。そのため、$((\mathbf{1} - \boldsymbol{\mu} - \boldsymbol{\nu}) > \mathbf{0})$を満たす必要がある。また、$\boldsymbol{\mu}$については無視できる。 そのため、$\mathbf{K} = (X^T Y)^T(X^T Y), \boldsymbol{\alpha} = \boldsymbol{\nu}$とすると、双対問題は以下のようにかける。

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{maximize}} \quad & -\frac{1}{4\lambda}\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha} + \boldsymbol{\alpha}^T\mathbf{1}^T \\ \text{subject to} \quad & \mathbf{1} \geq \boldsymbol{\alpha} \geq \mathbf{0} \end{aligned}$$

2.2. 3.2

$$\frac{\partial L}{\partial w} = 2\lambda \mathbf{w} - X^T Y \boldsymbol{\nu}$$

KKT 条件から、$\frac{\partial L}{\partial w} = 0$より、

$$\mathbf{w} = \frac{X^T Y \boldsymbol{\nu}}{2\lambda}$$

双対問題の解を$\boldsymbol{\alpha}$としたので、

$$\widehat{\mathbf{w}} = \frac{X^T Y \boldsymbol{\alpha}}{2\lambda} = \frac{1}{2\lambda}\sum_i \alpha_i y_i \mathbf{x}_i$$

となる。

2.3. 3.3

$\boldsymbol{\alpha}$の更新式に間違いがあると感じた。

$$\boldsymbol{\alpha}^{(t)} = P_{[0,1]^n}(\boldsymbol{\alpha}^{(t-1)} - \eta_t(\frac{1}{2\lambda}\mathbf{K}\boldsymbol{\alpha} - \mathbf{1}))$$

と表記されているが、正しくは

$$\boldsymbol{\alpha}^{(t)} = P_{[0,1]^n}(\boldsymbol{\alpha}^{(t-1)} - \eta_t(\frac{1}{2\lambda}\mathbf{K}\boldsymbol{\alpha}^{(t-1)} - \mathbf{1}))$$

であると思う。

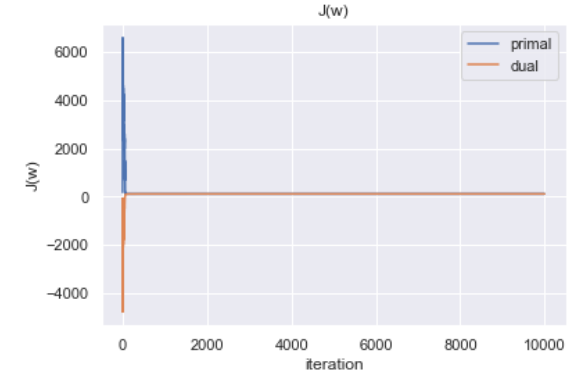
上の式による$\boldsymbol{\alpha}$の更新を、双対問題の目的関数を$D(\boldsymbol{\alpha})$とし、以下の終了条件を満たすまで繰り返す。

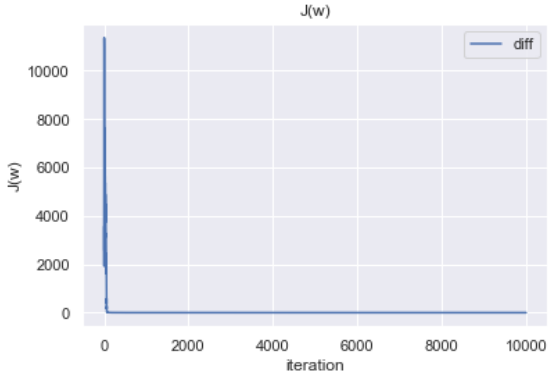
$$|D(\alpha^t) - D(\alpha^{(t-1)})| < \epsilon$$

$\boldsymbol{\alpha}$の更新のたび、3.2 で求めた式で、$\mathbf{w}$の更新を行う。

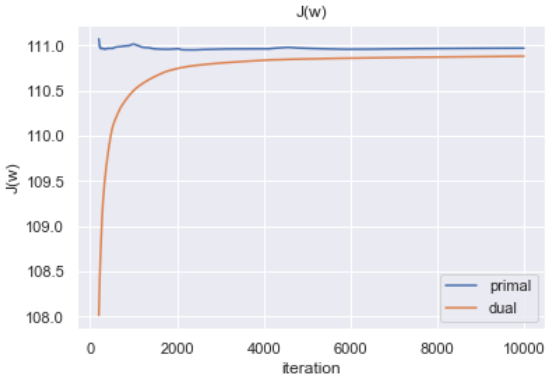
更新のたび、主問題の目的関数、双対問題の目的関数を求め、それぞれの値と差分をプロットする。

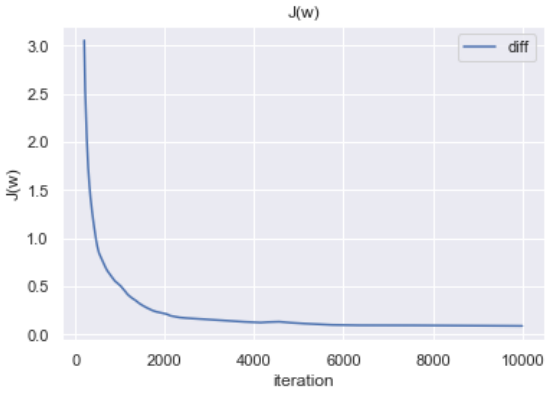
データセットに関して、Toy Datasets の Dataset II を用いた。 各種変数に関して、重みw, αの初期値として$\mathbf{w} = \mathbf{1}, \boldsymbol{\alpha} = \mathbf{1}$,正規化項に出現する定数について$\lambda = 1$, 終了判定に用いる定数について$\epsilon = 1e-6$ とした。





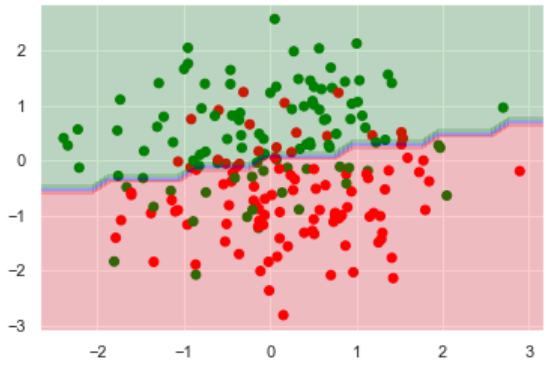
初期の値の変化が大きすぎるので、反復 200 回以降のプロットをした。





主問題の目的関数、双対問題の目的関数の差が十分小さくなり、最適化がなされた。

dataset II の分類の様子は以下のようになった。



3. Problem 5

3.1. 5.1

X, Yを次のよう定義する。

$$X = (\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}).T$$

$$(Y)_{i,j} = \begin{cases} 0 & (i \neq j) \\ y_i & (i = j) \end{cases}$$

X から$\tilde{d}$個のピボット$(X_{z_1}, X_{z_1}, \ldots X_{z_{\tilde{d}}})$をランダムサンプリングにより取得し、ガウシアンカーネルϕ用いて、$\tilde{X}$を次のように定義する。

$$\tilde{X} = (\phi(\mathbf{x_1}), \phi(\mathbf{x_2}), \ldots, \phi(\mathbf{x_n}))^T = (\widetilde{\mathbf{x_1}}, \widetilde{\mathbf{x_2}}, \ldots, \widetilde{\mathbf{x_n}})^T$$

損失関数は$\sum_{i=0}^{n} max(0,1-y_i \mathbf{w}\tilde{\mathbf{x}}_\mathbf{i})$, 正則化項は$\lambda \parallel \mathbf{w} \parallel_2^2$とする。

主問題は以下のようになる。

$$\underset{\mathbf{w},\boldsymbol{\xi}}{\text{minimize}} \quad \lambda \mathbf{w}^T \mathbf{w} + \mathbf{1}\boldsymbol{\xi}$$

$$\text{subject to} \quad \boldsymbol{\xi} \geq \mathbf{0}$$

$$\boldsymbol{\xi} \geq \mathbf{1} - (\mathbf{w}^T \tilde{X}^T Y)^T$$

ちょうど Problem 3 において、Xを$\tilde{X}$に置き換えた形になっているので、同様にして最適化できる。

3.2. 5.2

Problem 3 と同様に実装した。

デートセットに関して、Toy Datasets の Dataset I を用いた。 各種変数に関して、重みw, αの初期値として$\mathbf{w} = \mathbf{1}, \boldsymbol{\alpha} = \mathbf{1}$,正規化項に出現する定数について$\lambda = 1$, 終了判定に用いる定数について$\epsilon = 1e-6$ とした。

ガウシアンカーネルに与えるパラメータα, ピボット数$\tilde{d}$,データセットの要素数nを次の範囲で全通り変更しながら、分類の様子、最適化の様子を観察した。

$$\alpha \in [0.1,0.2,0.4,0.8,1]$$

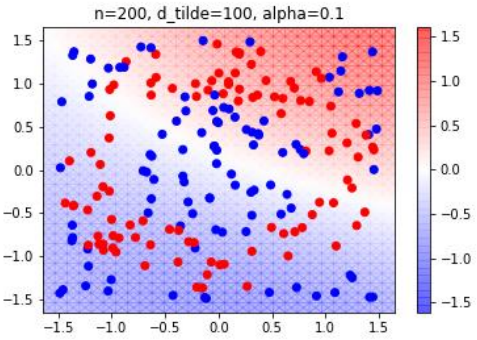
$$\tilde{d} \in [2,10,100,200,400]$$

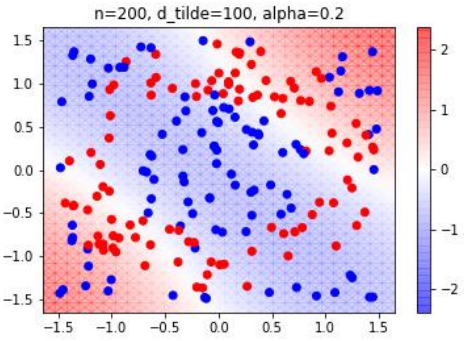
$$n \in [100,200,400,800]$$

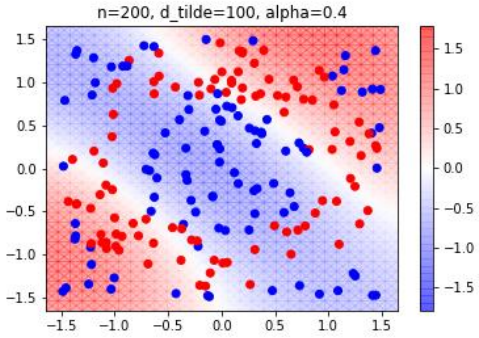
以下、それぞれのパラメータについて考察する。

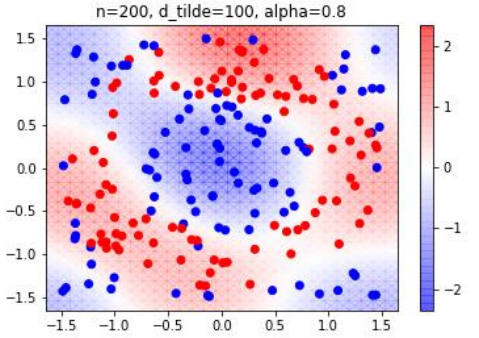
3.2.1. αに関して

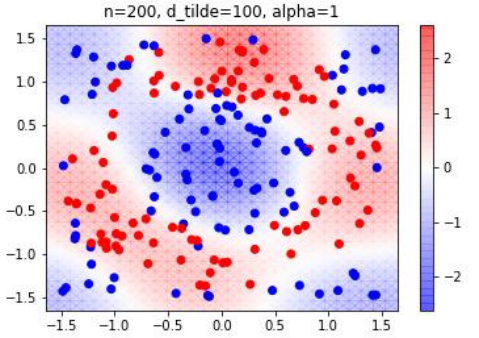
αが分類結果に与える影響について、以下は$\tilde{d} = 100, n = 200$で固定し、αを変更させていった時の分類の様子である。



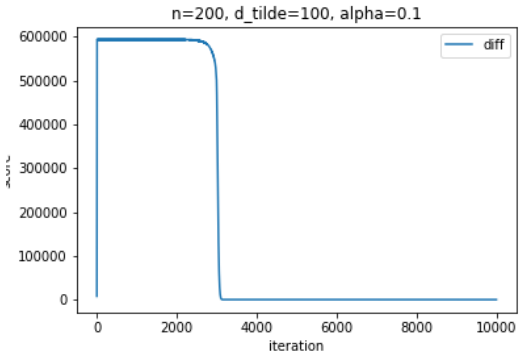


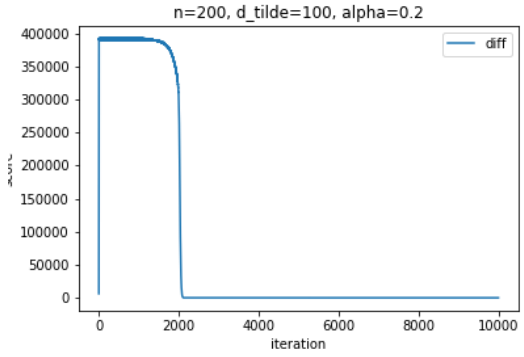


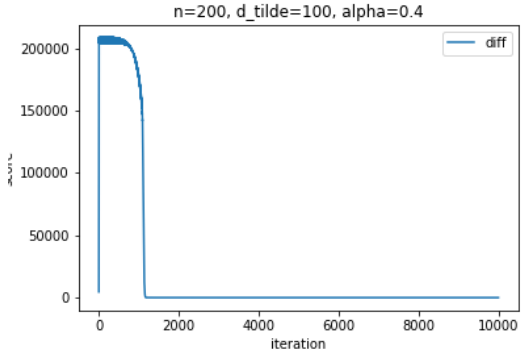


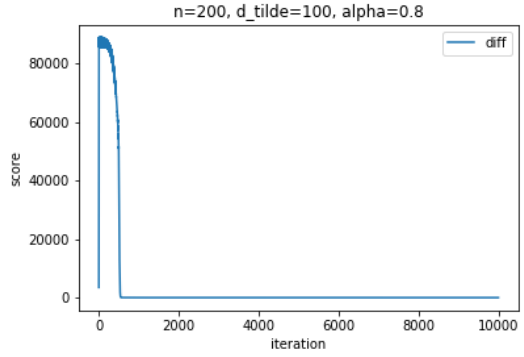


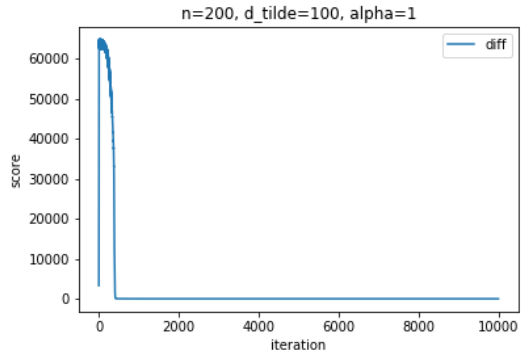
αが大きくなるにつれ分類の結果が向上し,0.8 と 1 ではほとんど同じ結果が得られていることが見て取れる。 また、最適化の様子を主問題の目的関数と双対問題の目的関数の差分についてプロットしたものを見る。









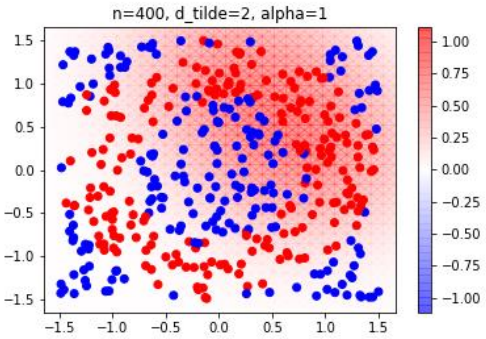


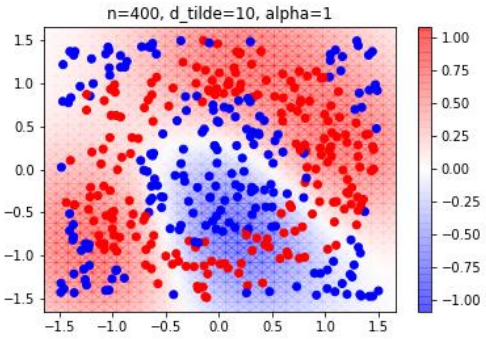
αが大きくなるにつれ収束が早くなっていることが見て取れる。

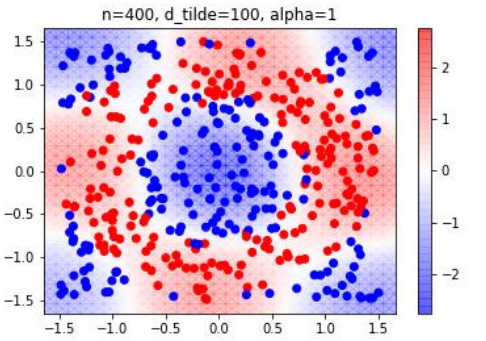
他のパラメータを動かした時の様子について、$\tilde{d}, n$を変更した場合もおおよそ同じ傾向が観れた。

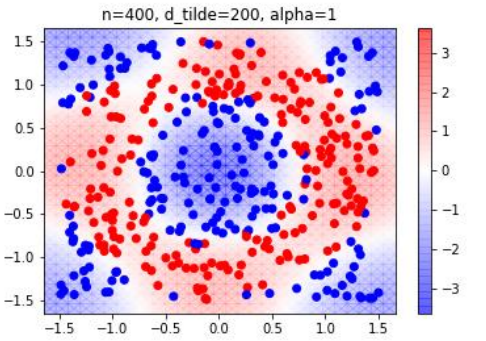
3.2.2. ピボット数$\tilde{d}$に関して

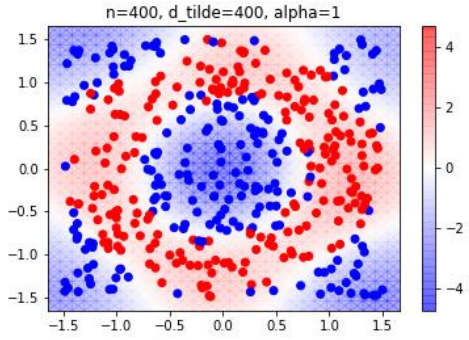
$\tilde{d}$が分類結果に与える影響について、以下は$\alpha = 1, n = 400$で固定し、$\tilde{d}$を変更させていった時の分類の様子である。





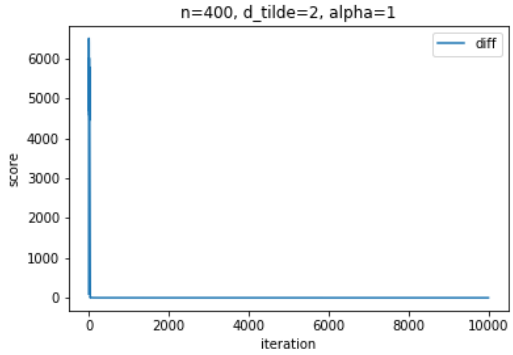


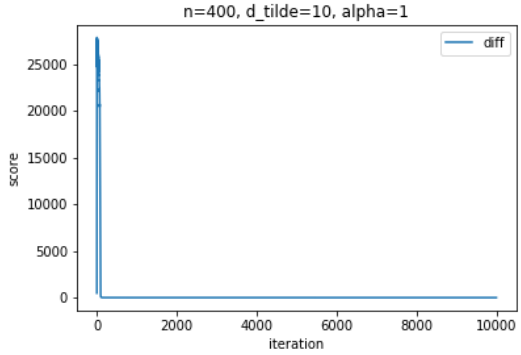


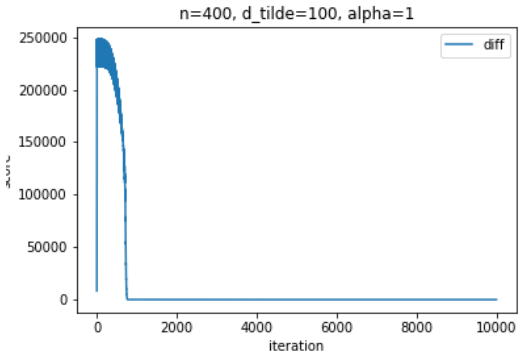


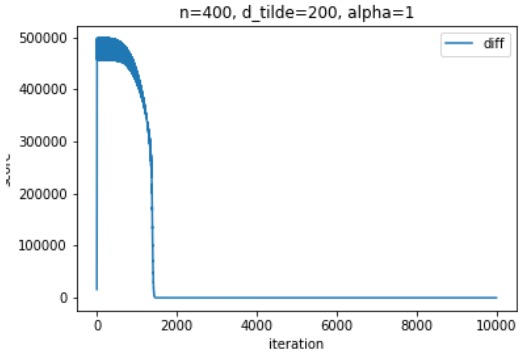
$\tilde{d}$が大きくなるにつれ分類の結果が向上していることが見て取れる。

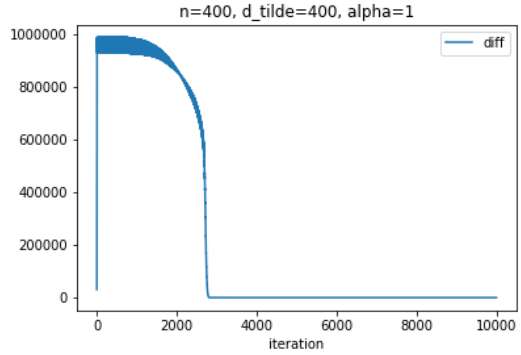
また、最適化の様子を主問題の目的関数と双対問題の目的関数の差分についてプロットしたものを見る。







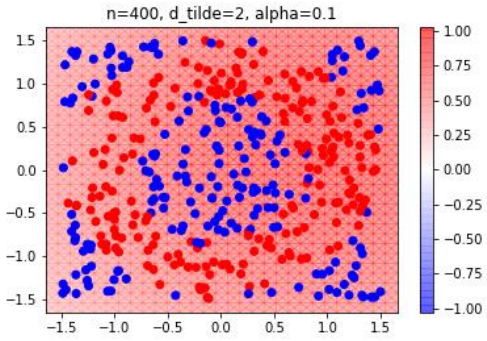


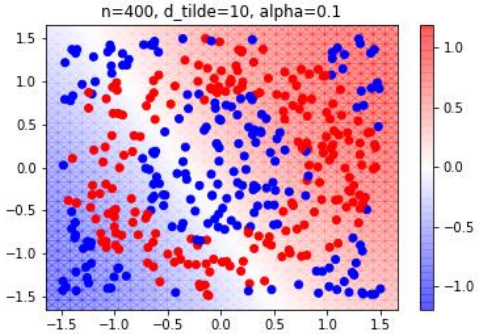


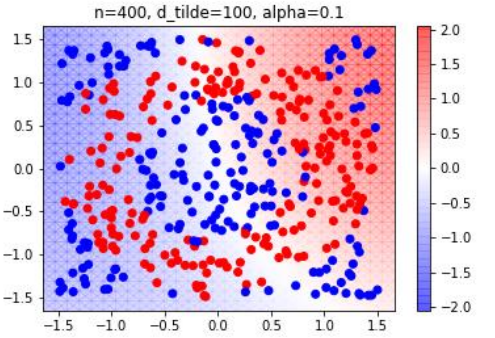
特徴量の次元が増えるため、$\tilde{d}$が大きくなるにつれ収束が遅くなっていることが見て取れる。

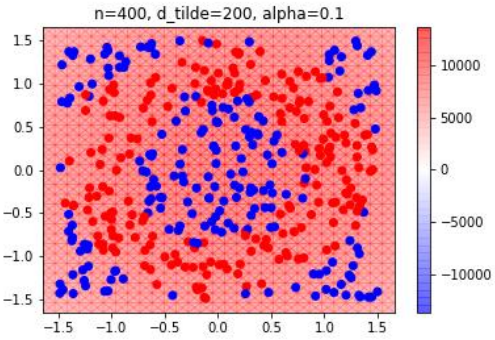
他のパラメータを動かした時の様子について、$n > \tilde{d}$となる場合を除いて、nを変更した場合もおおよそ同じ傾向が観れた。

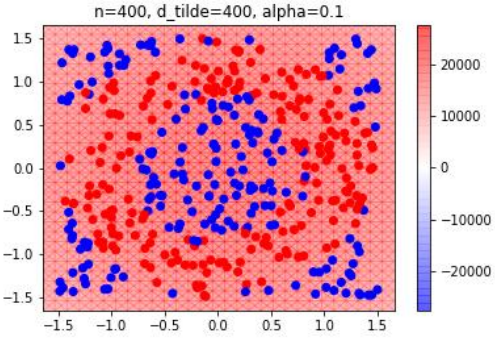
αを動かした時の様子に関して、以下は、$\alpha = 0.1, n = 400$で固定し、$\tilde{d}$を変更させていった時の分類の様子である。







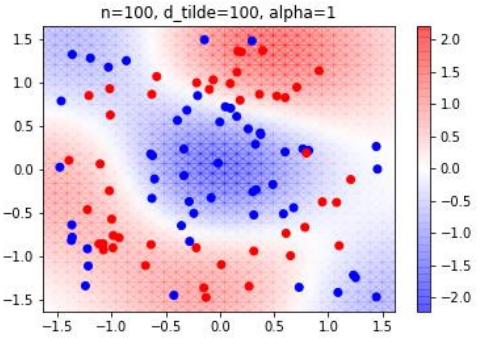


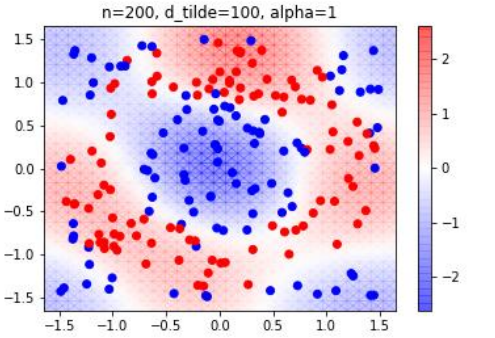


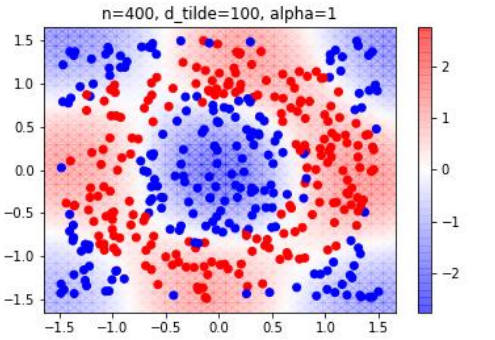
このように、αが小さすぎる場合、$\tilde{d}$を大きくしても学習がうまくいかない様子が観れた。

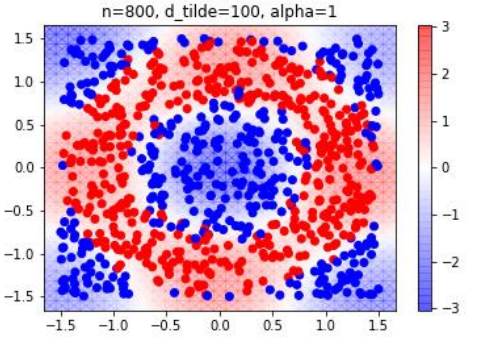
3.2.3. 要素数nに関して

nが分類結果に与える影響について、以下は$\alpha = 1, \tilde{d} = 200$で固定し、$\tilde{d}$を変更させていった時の分類の様子である。



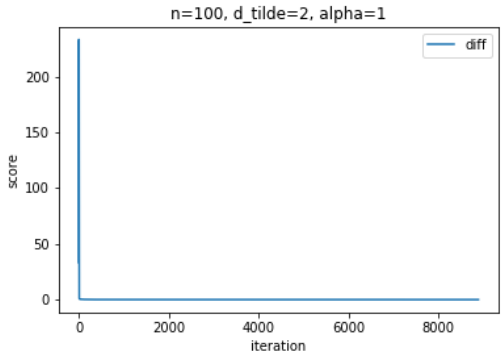


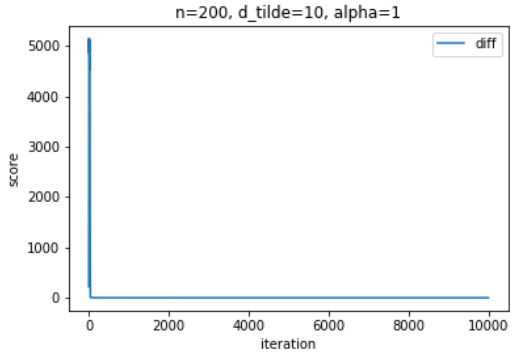


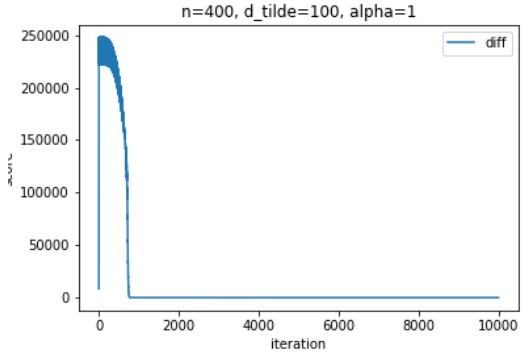


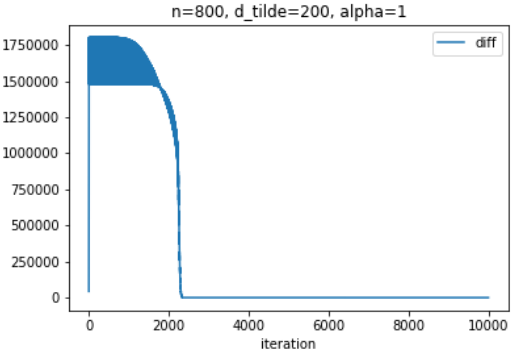
nが大きくなるにつれ元のドーナツ型のような分布を学習している様子が見て取れる。

また、最適化の様子を主問題の目的関数と双対問題の目的関数の差分についてプロットしたものを見る。









nが大きくなるにつれ収束が遅くなっていることが見て取れる。

十分大きな$\alpha, \tilde{d}$については、おおよそ同じ傾向が観れた。