SGD，FOBOS，CDの比較実験

東京大学　飯田紘士，松島慎，中川裕志

概要

- 文書の分類問題
  - 文書とそのカテゴリが与えられた教師付きデータから学習を行う
  - カテゴリが未知の文書に対して、カテゴリを予測する予測器を作ることが目的

- 今回行ったこと
  - 予測器と実際のデータの差で定義される目的関数を最小化することで学習を行う。最適化の手法としてSGD，FOBOS，CDを用いた結果をそれぞれ比較した。（これらの手法は既存手法）

問題設定

- 問題設定

$$F(w) = \sum_{i=1}^{m} l_i(w) + \lambda \|w\|_1$$

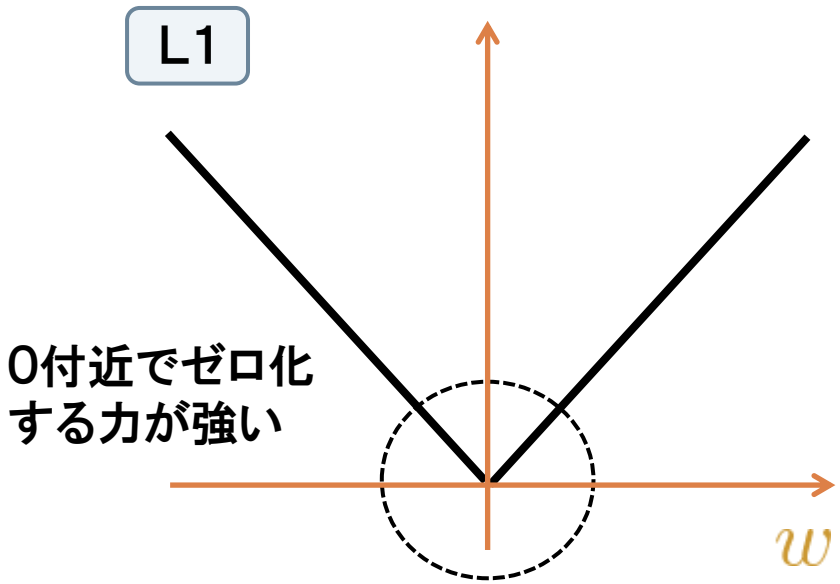
訓練データへの適合

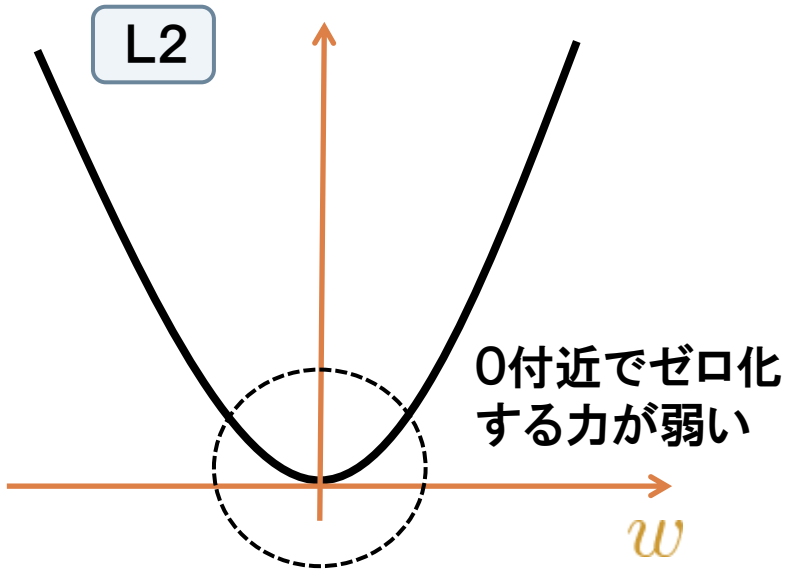
汎化能力
スパースな解をえる

- 損失項　＋　正則化項の最適化
- 正則化項として、できるだけスパースな解が得られる L1　正則化項を用いた

正則化項について

- 正則化項は w 大きさを抑える効果がある
- L1正則化　$\|w\|_1 = |w_1| + |w_2| + \ldots$
- L2正則化　$\|w\|_2 = (w_1^2 + w_2^2 + \ldots)^{\frac{1}{2}}$





SGD，FOBOS，CDの比較実験

- 目的関数の最適化問題を解くための手法

SGD (Stochastic Gradient Descent), FOBOS (Forward-Backward Splitting) ： オンライン学習

CD (Coordinate Descent) ： バッチ学習

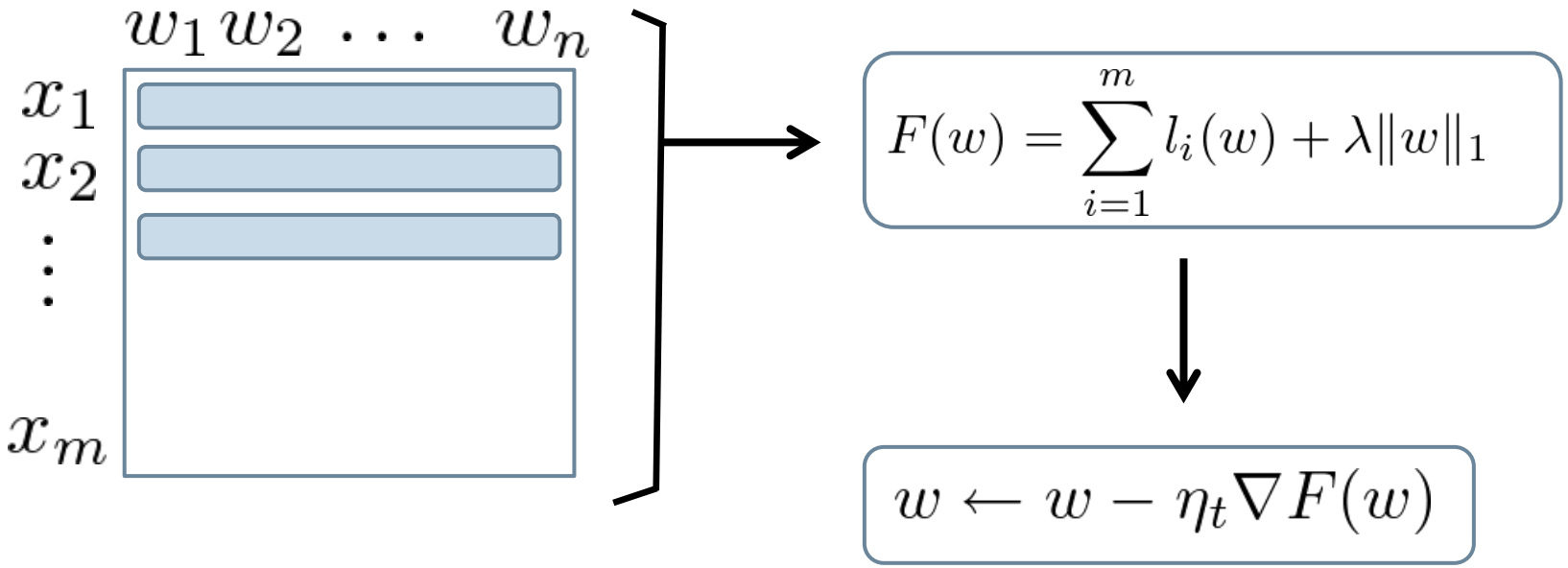
⟶ オンライン学習として用いられるSGD，FOBOSと呼ばれる手法と、バッチ学習として用いられるCDによる最適化をそれぞれ行い、CDが優れていることを確認した

発表の内容

- それぞれの手法の説明
  - SGD
  - FOBOS
  - CD
- 実験結果

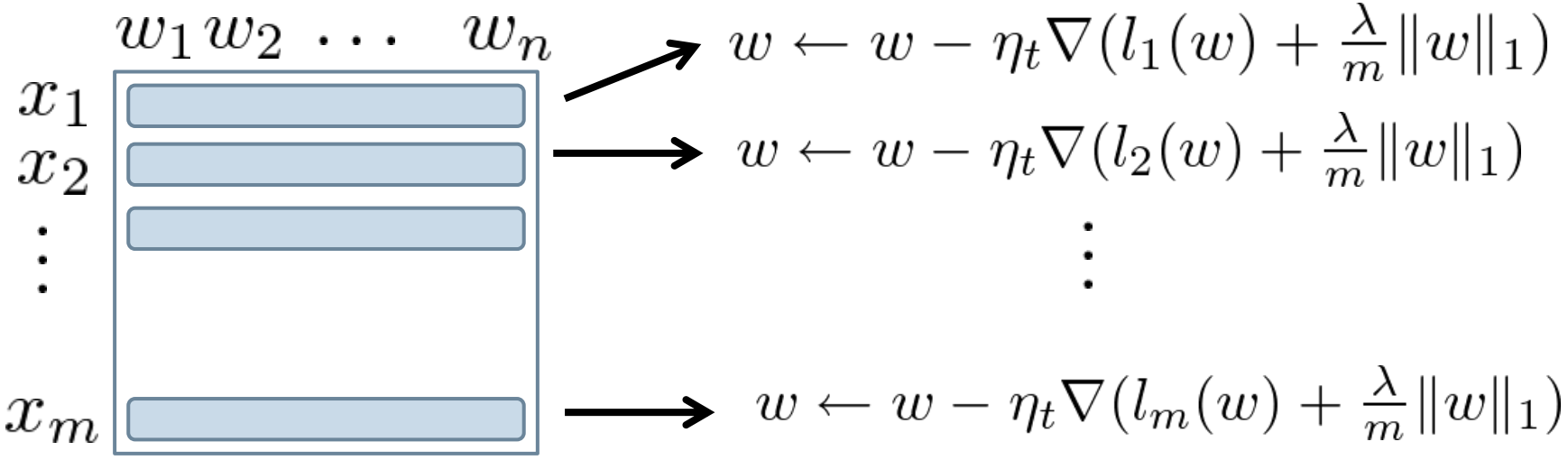
GD（Gradient Descent）

- 全データを見た後、w を一回更新する



SGD（Stochastic GD）

- 一つのデータみるたびに w を一回更新する

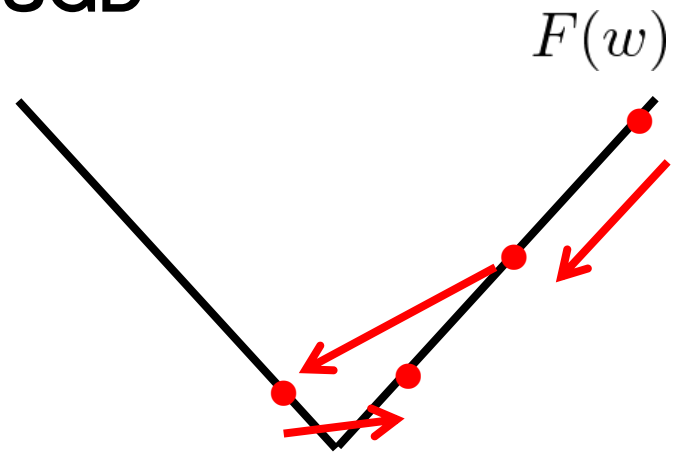


- 本当は $\sum l_i(w) + \lambda\|w\|_1$ を最適化したい。
- 各ステップでは間違った更新を行う可能性がある。

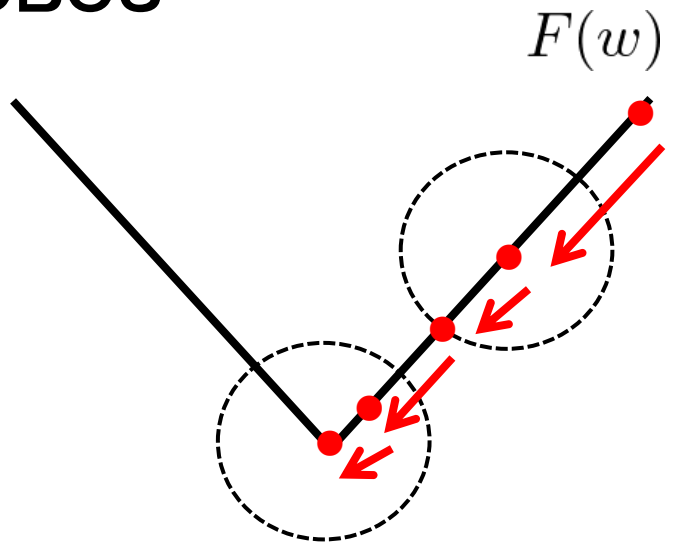
FOBOS（SGDの改良）

- SGDでは（特に劣勾配の時）ぴったり最適解にたどりつくのに時間がかかる。

SGD



FOBOS



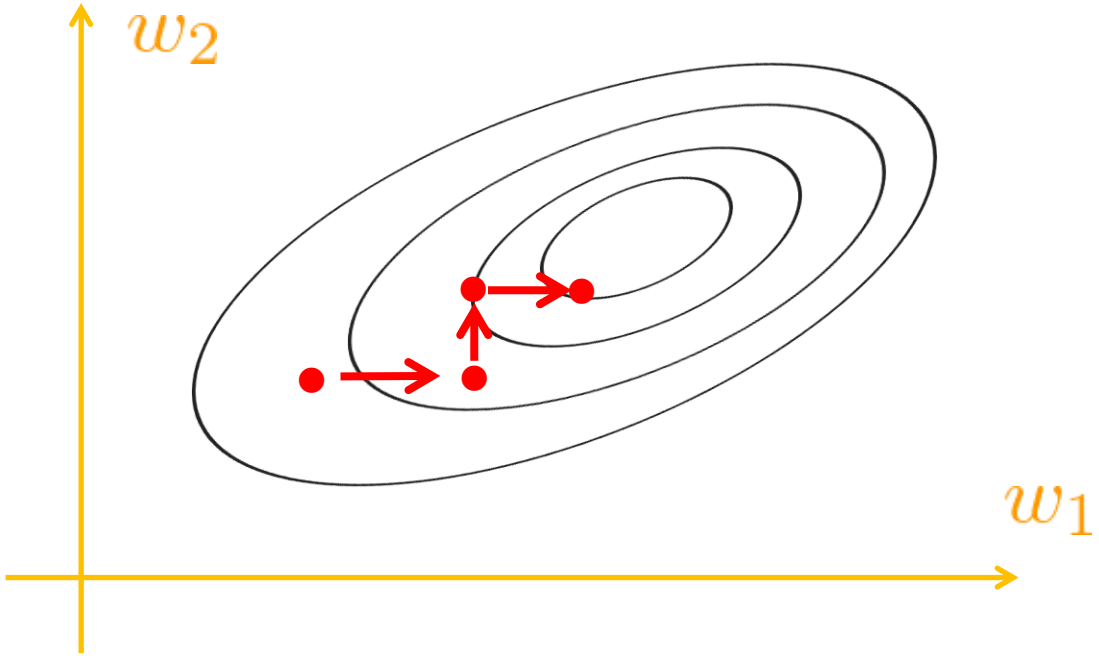
FOBOS

$$w_{t+1/2} = w_t - \eta_t \nabla l_t(w_t)$$
$$w_{t+1} = \underset{w}{\operatorname{argmin}} \frac{1}{2}\|w - w_{t+1/2}\|_2 + \eta_{t+1/2}\|w\|_1$$

- 一般的に COMID と呼ばれる手法の特殊な形
- 2つ目の式は閉じた形で解が求められるから、学習は容易

CD

- 現在の w から第 j 成分だけ変化させて最適化
- GDなどでは、一回の更新で w の全成分を更新していたが、CDでは一成分ごと



CD

- Newton法を普通に行うと

$$\min_{d\in\mathbb{R}^n} f(w+d) \longrightarrow \min_{d\in\mathbb{R}^n} f(w) + d^T\nabla f(w) + \frac{1}{2}d^T H(w)d$$

$$\longrightarrow \boxed{d = -H^{-1}(w)\nabla f(w)}$$

逆行列の計算

- 一つの成分毎に行うと

$$\min_{d\in\mathbb{R}} f(w+de_j) \longrightarrow \min_{d\in\mathbb{R}} f(w) + (de_j)^T\nabla f(w) + \frac{1}{2}(de_j)^T H(w)(de_j)$$

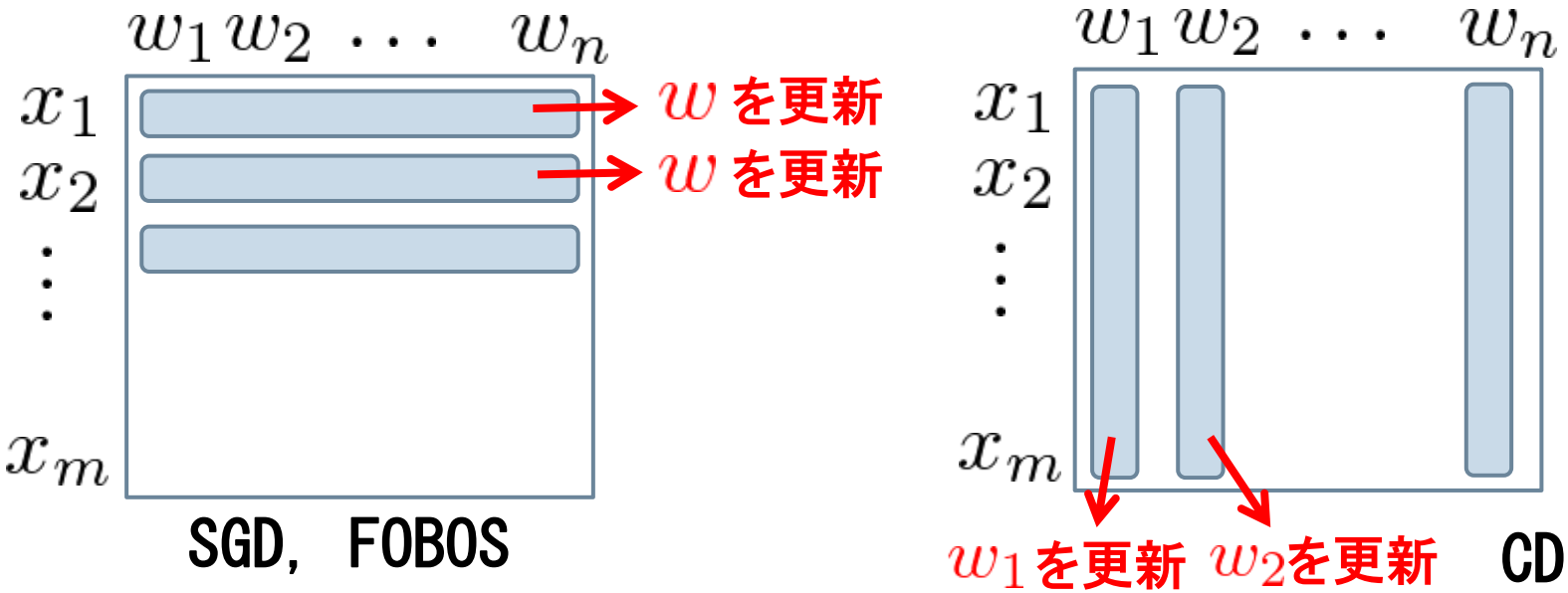
$$\longrightarrow \nabla_j f(w) + (H(w))_{jj}d = 0$$

$$\longrightarrow \boxed{d = -\frac{\nabla_j f(w)}{\nabla_{jj} f(w)}}$$

実数

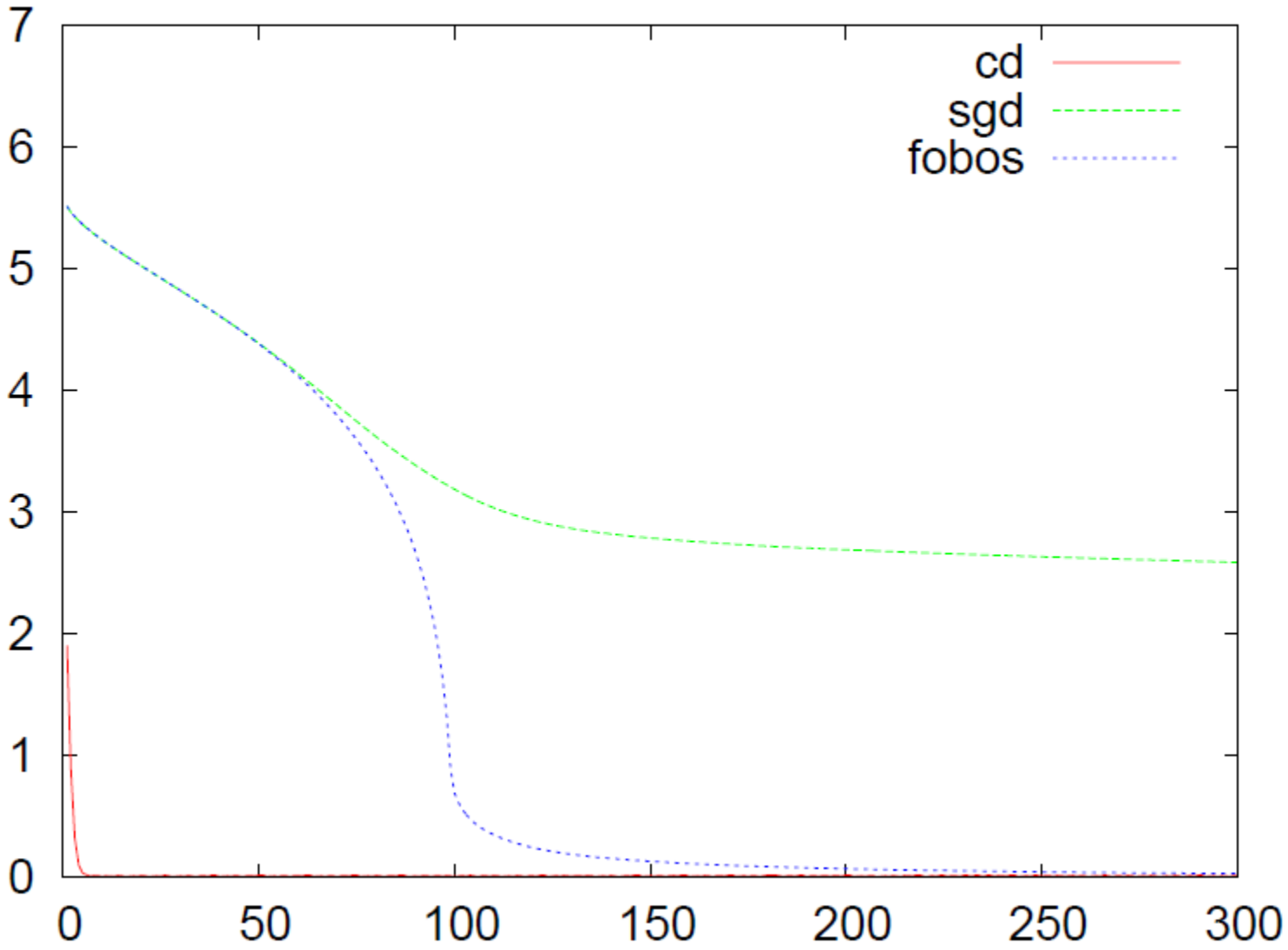
実験の設定

- 全ての訓練データをちょうど一度ずつみた時を1周として、T 周した時点での w を比較する
- SGD, FOBOSは一周でm回重みを更新するが、CDでは一周して初めて全ての成分が更新される



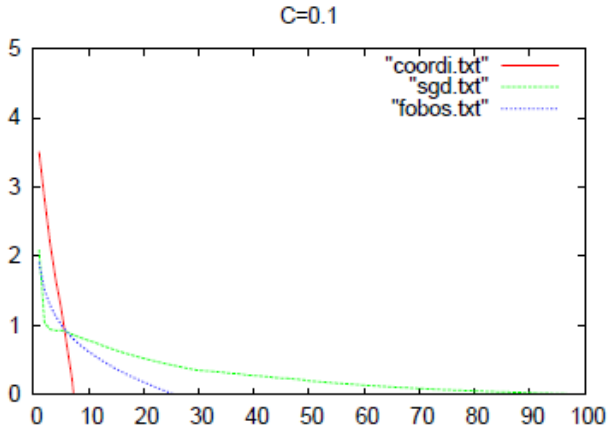
実験の設定

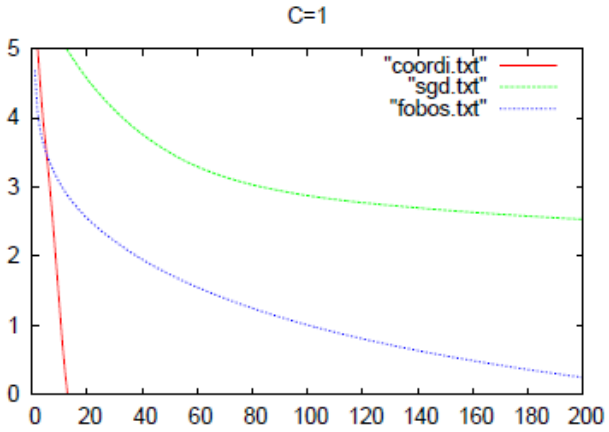
- LIBSVMデータセットの news20 を使用
- 一つの文は30～60word程度で、合計で1577個の文書がある。次元（単語数）は25510で、データは疎行列として与えられている
- 損失関数としてヒンジ二乗損失関数、正則化項としてL1 正則化を採用
  - $l_i(w)=\max(0,1-w^{\mathrm{T}}y_i x_i)^2$
- 横軸をデータセットを何周学習したか（T）を表し、縦軸は最適な w^* と学習途中の w_Tを用いて目的関数の差を計算した値
  - X-axis := T
  - Y-axis := $\log(F(w_T)-F(w^*))$

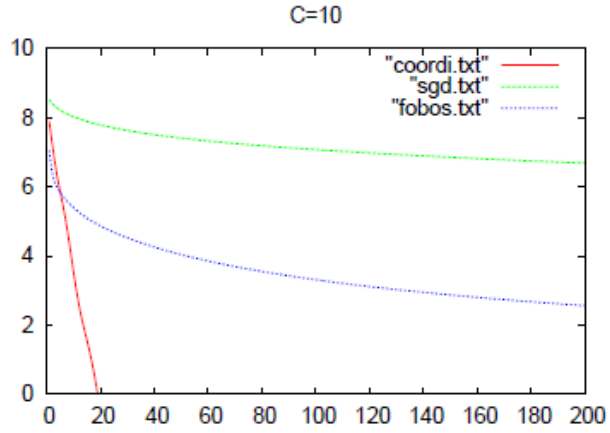

cd
sgd
fobos
0
1
2
3
4
5
6
7
0
50
100
150
200
250
300

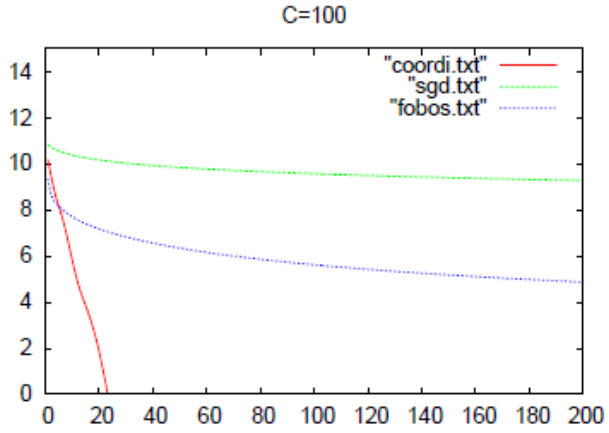
実験の補足

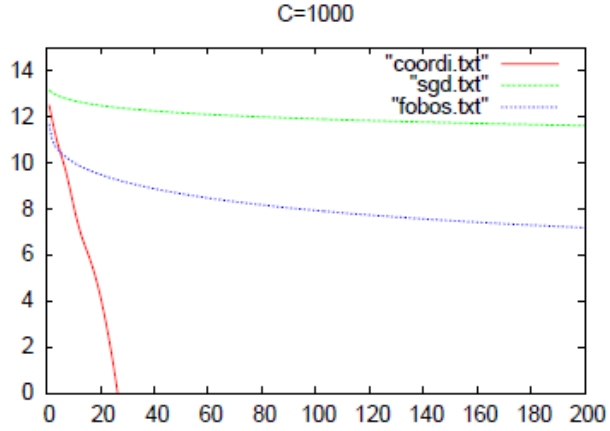
- 異なるデータセット(heart_scale)に対し実験を行った
- 目的関数の中で、損失関数と正則化項の大きさの比を制御するパラメーター、C を変えて実験した結果をのせる
  - $C = 1/\lambda m$
- 扱った問題は、損失関数がヒンジ二乗損失関数で、正則化項が L1 正則化項であたえられた最適化問題(先ほどと同様)
- 文書分類問題ではありません(心臓疾患の予測)

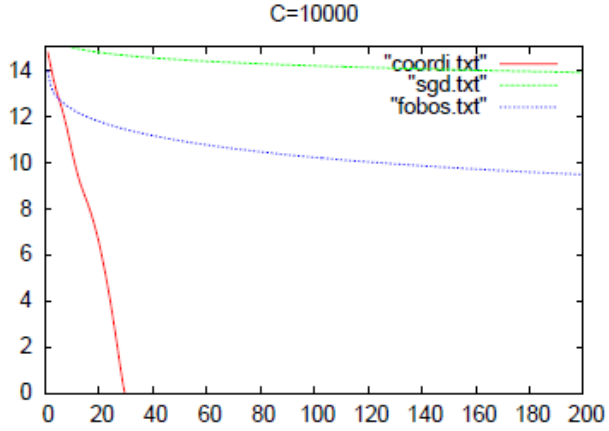

C=0.1
"coordi.txt"
"sgd.txt"
"fobos.txt"
0 1 2 3 4 5
0 10 20 30 40 50 60 70 80 90 100


C=1
"coordi.txt"
"sgd.txt"
"fobos.txt"
0 1 2 3 4 5
0 20 40 60 80 100 120 140 160 180 200


C=10
"coordi.txt"
"sgd.txt"
"fobos.txt"
0 2 4 6 8 10
0 20 40 60 80 100 120 140 160 180 200


C=100
"coordi.txt"
"sgd.txt"
"fobos.txt"
0 2 4 6 8 10 12 14
0 20 40 60 80 100 120 140 160 180 200


C=1000
"coordi.txt"
"sgd.txt"
"fobos.txt"
0 2 4 6 8 10 12 14
0 20 40 60 80 100 120 140 160 180 200


C=10000
"coordi.txt"
"sgd.txt"
"fobos.txt"
0 2 4 6 8 10 12 14
0 20 40 60 80 100 120 140 160 180 200