正則化

- 線形計画識別器

  正則化項を l_1 ノルム $||\boldsymbol{w}||_1 = \sum_{i=1}^{d'} |w_i|$ にする

  $$\min_{\boldsymbol{w},b} \frac{1}{n} \sum_{i=1}^{n} c(f(\boldsymbol{x}_i), y_i) + \lambda ||\boldsymbol{w}||_1$$

  スパースな解 (0に近い w_i を 0に押し切る)

- 様々なヴァリエーション
  - 第1項: 損失関数
  - 第2項: 正則化項

逐次最小最適化アルゴリズム (SMO algorithm) その1

$$\boxed{\max W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \quad \text{s.t.} \ 0 \le \alpha_i \le C, \ \sum_{i=1}^{n} \alpha_i y_i = 0}$$

1. 変化させるパラメータを2つに限定

   一般性を失わずに α_1, α_2 と仮定

   変更前 $(\alpha_1^{\text{old}}, \alpha_2^{\text{old}}) \Rightarrow$ 変更後 $(\alpha_1^{\text{new}}, \alpha_2^{\text{new}})$

   $$\alpha_1^{\text{new}} y_1 + \alpha_2^{\text{new}} y_2 = \text{定数} = \alpha_1^{\text{old}} y_1 + \alpha_2^{\text{old}} y_2 \quad \left(\because \sum_{i=1}^{n} \alpha_i y_i = 0 \right)$$

2. α_2^{new} を変数とみなす

   $$\alpha_1^{\text{new}} = \alpha_1^{\text{old}} + (\alpha_2^{\text{old}} - \alpha_2^{\text{new}}) y_1 y_2$$

逐次最小最適化アルゴリズム (SMO algorithm) その2

$$\boxed{\max W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \quad \text{s.t.} \ 0 \le \alpha_i \le C, \ \sum_{i=1}^{n} \alpha_i y_i = 0}$$

- ボックス制約 $0 \le \alpha_i \le C$ から

  $$U \le \alpha_2^{\text{new}} \le V$$

  1. $y_1 \ne y_2$ の場合

     $$\begin{aligned} U &= \max(0, \alpha_2^{\text{old}} - \alpha_1^{\text{old}}) \\ V &= \min(C, C - \alpha_1^{\text{old}} + \alpha_2^{\text{old}}) \end{aligned}$$

  2. $y_1 = y_2$ の場合

     $$\begin{aligned} U &= \max(0, \alpha_1^{\text{old}} + \alpha_2^{\text{old}} - C) \\ V &= \min(C, \alpha_1^{\text{old}} + \alpha_2^{\text{old}}) \end{aligned}$$

逐次最小最適化アルゴリズム (SMO algorithm) その3

$$\boxed{\max W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \quad \text{s.t.} \ 0 \le \alpha_i \le C, \ \sum_{i=1}^{n} \alpha_i y_i = 0}$$

- 簡略化のための表記法

  1. (ある途中段階での) 識別関数の出力とクラスの差

     $$E_i = f(\boldsymbol{x}_i) - y_i = \left(\sum_{j=1}^{n} \alpha_j y_j K(\boldsymbol{x}_j, \boldsymbol{x}_i) + b \right) - y_i \quad i = 1, 2$$

  2. 目的関数の2次導関数

     $$\begin{aligned} \kappa &= K(\boldsymbol{x}_1, \boldsymbol{x}_1) + K(\boldsymbol{x}_2, \boldsymbol{x}_2) - 2K(\boldsymbol{x}_1, \boldsymbol{x}_2) \\ &= ||\boldsymbol{\phi}(\boldsymbol{x}_a) - \boldsymbol{\phi}(\boldsymbol{x}_b)||^2 \end{aligned}$$

逐次最小最適化アルゴリズム (SMO algorithm) その4

$$\boxed{\max W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \quad \text{s.t.} \ 0 \le \alpha_i \le C, \ \sum_{i=1}^{n} \alpha_i y_i = 0}$$

- 変化するパラメータが2の場合の解

  1. $\frac{\partial W}{\partial \alpha_2} = 0$ から

     $$\alpha_2^{\text{tmp}} = \alpha_2^{\text{old}} + \frac{y_2(E_1 - E_2)}{\kappa} \qquad \leftarrow \text{要確認}$$

  2. $U \le \alpha_2^{\text{new}} \le V$ なので

     $$\alpha_2^{\text{new}} = \begin{cases} V & \text{if} \quad V < \alpha_2^{\text{tmp}} \\ \alpha_2^{\text{tmp}} & \text{if} \quad U \le \alpha_2^{\text{tmp}} \le V \\ U & \text{if} \quad \alpha_2^{\text{tmp}} < U \end{cases}$$

  3. 最後に $\alpha_1^{\text{new}} = \alpha_1^{\text{old}} + (\alpha_2^{\text{old}} - \alpha_2^{\text{new}}) y_1 y_2$

逐次最小最適化アルゴリズム (SMO algorithm) その5

$$\boxed{\max W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \quad \text{s.t.} \ 0 \le \alpha_i \le C, \ \sum_{i=1}^{n} \alpha_i y_i = 0}$$

- 変化させるパラメータ α_1,α_2 (に対応する点 $\boldsymbol{x}_i$,$\boldsymbol{x}_j$ (の添字)) の選択

  1. KKT(Karush-Kuhn-Tucker) 条件を満たさない点を探す

     見つかればそれを α_1 とする

  2. $|E_1 - E_2|$ が最大化となる $\boldsymbol{x}_2$ を選択

     (a) 符合チェック

     (b) 全探索

逐次最小最適化アルゴリズム (SMO algorithm) その6

$$\max W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \quad \text{s.t. } 0 \le \alpha_i \le C, \ \sum_{i=1}^{n} \alpha_i y_i = 0$$

1. 変化させるパラメータ α_1, α_2 の選択
2. α_1, α_2 の更新
3. 停止条件のチェック

31

最大エントロピー法 (Maximum Entropy Method)

- いびつな3面体

  各面の出る確率を $\boldsymbol{p} = (p_1,\ p_2,\ p_3)$ とする

  条件:

  1. $p_1 + p_2 + p_3 = 1$
  2. $p_3 = 2p_1$

- 各面の出る確率をいくらとするのが妥当か?

32

最大エントロピー法 (ME)

- 最大エントロピー原理

  エントロピーが最大となるモデルを選択

- いびつな3面体の例では
  - 目的関数

    $$\max_{\boldsymbol{p}} H \quad ここで\ H = \sum_{i=1}^{3} -p_i \log p_i$$

  - 制約条件
    1. $p_1 + p_2 + p_3 = 1$
    2. $p_3 = 2p_1$

33

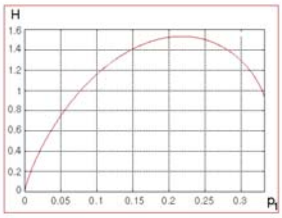
最大エントロピー法 (MaxEnt)

- 制約条件から

  $$p_3 = 2p_1,\ p_2 = 1 - 3p_1$$

- 目的関数

  $$\frac{dH}{dp_1} = \sum_{i=1}^{3} \frac{\partial H}{\partial p_i}\frac{dp_i}{dp_1} = -\sum_{i=1}^{3} (\log p_i + 1)\frac{dp_i}{dp_1}$$



- グラフから (数値計算)

  $$\hat{\boldsymbol{p}} = \mathop{\mathbf{argmax}}_{\boldsymbol{p}} H = (0.218,\ 0.346,\ 0.436)$$
  $$\hat{H} = \max_{\boldsymbol{p}} H = 1.5310$$

34

素性関数を用いた制約表現 (その1)

- 学習データ

  $$\{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$$

- 素性関数

  $$\phi_i : (x, y) \mapsto \{0, 1\}, \quad i \in \{1, 2, \cdots, n\}$$

- 経験的確率分布

  $$\tilde{P}(x, y) = \frac{C(x, y)}{N}$$

35

素性関数を用いた制約表現 (その2)

- 経験的確率分布による素性 f_i の期待値

  $$E_{\tilde{P}}[\phi_i] = \sum_{x,y} \tilde{P}(x, y)\phi_i(x, y)$$

- (推定すべき) モデル $P(x, y)$ による素性 ϕ_i の期待値

  $$E_P[\phi_i] = \sum_{x,y} P(x, y)\phi_i(x, y)$$

- 両者は一致しているべき

  $$\sum_{x,y} P(x, y)\phi_i(x, y) = \sum_{x,y} \tilde{P}(x, y)\phi_i(x, y)$$

36

最大エントロピーモデル

- 目的関数 (最大化)

$$H(P) = -\sum_{x,y} P(x,y) \log P(x,y)$$

- 制約

$$\sum_{x,y} P(x,y)\phi_i(x,y) = \sum_{x,y} \tilde{P}(x,y)\phi_i(x,y) \quad \forall i \in \{1,2,\cdots,n\}$$

$$P(x,y) \geq 0 \qquad \forall(x,y)$$

$$\sum_{x,y} P(x,y) = 1$$

制約を満たしつつ、$P(x,y)$ をできるだけ散らす

37

ラグランジュの未定乗数法

- ラグランジュ乗数 w_i (≥ 0) を導入

$$\begin{aligned} &L(P, \boldsymbol{w}) \\ &= H(P) + \sum_{i=1}^{n} w_i (E_P[\phi_i] - E_{\tilde{P}}[\phi_i]) \\ &= -\sum_{(x,y)} P(x,y) \log P(x,y) + \sum_{i=1}^{n} w_i \sum_{x,y} \{P(x,y) - \tilde{P}(x,y)\} \phi_i(x,y) \end{aligned}$$

- 偏微分が 0 (必要条件)

$$\frac{\partial L}{\partial P(x,y)} = -\log P(x,y) - 1 + \sum_{i=1}^{n} w_i \phi_i(x,y) = 0$$

$$\Leftrightarrow P(x,y) = C \exp\left(\sum_{i=1}^{n} w_i \phi_i(x,y)\right)$$ 十分性は省略

- 定数 C は残りの制約から決定

38

パラメトリック形式

- 以下のパラメトリック形式で表せる (ラグランジュ未定乗数法)

$$P_{\boldsymbol{w}}(x,y) = \frac{1}{Z_{\boldsymbol{w}}} \exp(\langle \boldsymbol{w} \cdot \boldsymbol{\phi}(x,y) \rangle)$$

$$Z_{\boldsymbol{w}} = \sum_{x,y} \exp(\langle \boldsymbol{w} \cdot \boldsymbol{\phi}(x,y) \rangle)$$

$\boldsymbol{w}$ は各素性の重み、$\boldsymbol{\phi}(x,y)$ は素性ベクトル

$$\langle \boldsymbol{w} \cdot \boldsymbol{\phi}(x,y) \rangle = \sum_{i=1}^{n} w_i \phi_i(x,y)$$

- 解探索アルゴリズム

GIS(Generalized Iterative Scaling),
BFGS(Broyden-Fletcher-Goldfarb-Shanno)

39

条件付き最大エントロピーモデル

- 識別のためには $P(y|x)$ が必要

$P(x,y)$ を $\tilde{P}(x)P(y|x)$ で置き換える

- 目的関数

$$H(P) = -\sum_{x,y} \tilde{P}(x) P(y|x) \log P(y|x)$$

- 制約

$$\underline{\sum_{x} \tilde{P}(x)} \sum_{y} P(y|x)\phi_i(x,y) = \sum_{x,y} \tilde{P}(x,y)\phi_i(x,y) \quad \forall i \in \{1,2,\cdots,n\}$$

↑訓練データに出現する x のみが対象

40

パラメトリック形式の解

- x に依存する解

$$P_{\boldsymbol{w}}(y|x) = \frac{1}{Z_{\boldsymbol{w}}(x)} \exp(\langle \boldsymbol{w} \cdot \boldsymbol{\phi}(x,y) \rangle)$$

$$Z_{\boldsymbol{w}}(x) = \sum_{y} \exp(\langle \boldsymbol{w} \cdot \boldsymbol{\phi}(x,y) \rangle)$$

$\boldsymbol{w}$ は各素性の重み、$\boldsymbol{\phi}(x,y)$ は素性ベクトル

$$\langle \boldsymbol{w} \cdot \boldsymbol{\phi}(x,y) \rangle = \sum_{i=1}^{n} w_i \phi_i(x,y)$$

41

パラメータ推定

- 確率分布

$$P(y|x; \boldsymbol{w}) = \frac{1}{Z_w(x)} \exp(\langle \boldsymbol{w} \cdot \boldsymbol{\phi}(x,y) \rangle)$$

- 以下を最小化

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\mathbf{argmin}} \sum_{i=1}^{n} -\log P(y_i|x_i; \boldsymbol{w})$$

※ i に対する合計なので x の頻度が考慮される

42

識別器としての ME

- 入力 x に対する確率最大のラベル y の推定
  - 確率分布

$$P(y|x;\boldsymbol{w}) = \frac{1}{Z_w(x)} \exp(\langle \boldsymbol{w} \cdot \boldsymbol{\phi}(x,y) \rangle)$$

  - 識別 (※ y は多値)

$$\begin{aligned}\hat{y} &= \mathop{\mathbf{argmax}}_y P(y|x;\boldsymbol{w}) \\ &= \mathop{\mathbf{argmax}}_y \log P(y|x;\boldsymbol{w}) \\ &= \mathop{\mathbf{argmax}}_y \log \frac{1}{Z_w(x)} \exp(\langle \boldsymbol{w} \cdot \boldsymbol{\phi}(x,y) \rangle) \\ &= \mathop{\mathbf{argmax}}_y \langle \boldsymbol{w} \cdot \boldsymbol{\phi}(x,y) \rangle - \log Z_w(x)\end{aligned}$$

ME の正則化

- 正則化項を加えて過学習を防ぐ
- l_2 ノルム

$$\min_{\boldsymbol{w}} \sum_{i=1}^{n} -\log P(y_i|x_i;\boldsymbol{w}) + \frac{C}{2}||\boldsymbol{w}||_2$$

事前分布にガウス分布 $\exp(-w_i^2/2\sigma^2)$ を仮定

- l_1 ノルム

$$\min_{\boldsymbol{w}} \sum_{i=1}^{n} -\log P(y_i|x_i;\boldsymbol{w}) + C||\boldsymbol{w}||_1$$

事前分布にラプラス分布 $\exp(-\sigma|w_i|)$ を仮定

損失関数の比較 (SVM, ME, etc.)

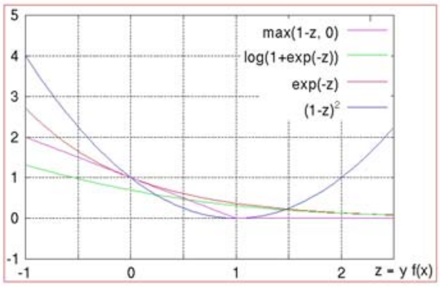
- 損失関数+正則化項

$$\min_{\boldsymbol{w}} \frac{1}{n} \sum_{i=1}^{n} \mathbf{Loss}(f(\boldsymbol{x}_i), y_i) + CR(\boldsymbol{w}) \qquad \text{※ } f(\boldsymbol{x}) \text{ は } \boldsymbol{w} \text{ を含む}$$

1. SVM
   $\mathbf{HingeLoss}(f(\boldsymbol{x}), y) = \mathbf{max}(1 - yf(\boldsymbol{x}), 0)$
2. ME $(y \in \{-1, 1\})$
   $\mathbf{LogLoss}(f(\boldsymbol{x}), y) = \log\left(1 + e^{-yf(\boldsymbol{x})}\right)$
3. AdaBoost $(y \in \{-1, 1\})$
   $\mathbf{ExpLoss}(f(\boldsymbol{x}), y) = e^{-yf(\boldsymbol{x})}$
4. 2 乗誤差 $(y \in \{-1, 1\})$
   $\mathbf{SquareLoss}(f(\boldsymbol{x}), y) = (1 - yf(\boldsymbol{x}))^2 \qquad \because y^2 = 1$

損失関数のグラフ

SVM: $\mathbf{max}(1 - yf(\boldsymbol{x}), 0)$ **AdaBoost:** $e^{-yf(\boldsymbol{x})}$
ME: $\log(1 + e^{-yf(\boldsymbol{x})})$ **MeanSquare:** $(1 - yf(\boldsymbol{x}))^2$



課題

- 河原先生分と綴じて提出
- 7 月 25 日〆切
- 手書き

1. アンケート (河原先生分も含む)
   (a) 興味を持ったトピック
   (b) もっと詳しく堀下げて欲しいトピック
   (c) 新たに扱って欲しいトピック