実データで学ぶ人工知能講座

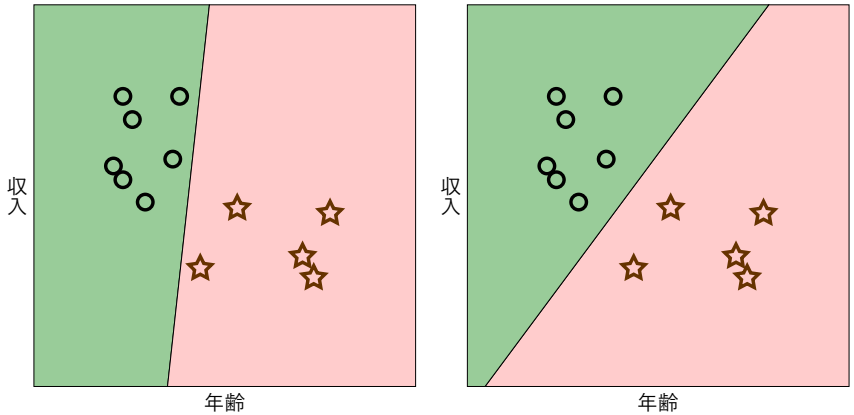
ノイズとフィードバックと汎化能力

マシュー　ホーランド
Matthew J. Holland
matthew-h@ar.sanken.osaka-u.ac.jp

大阪大学 産業科学研究所 助教

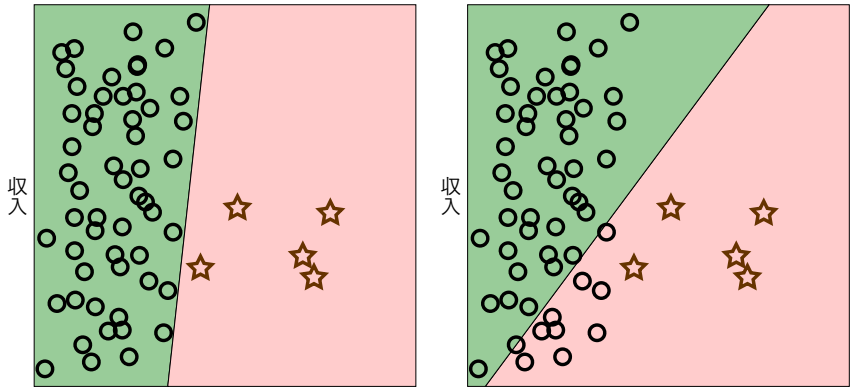
学習機の出来栄えを考える

2 つの候補, いずれも全問正解している. これ以上の優劣をどう評価する？



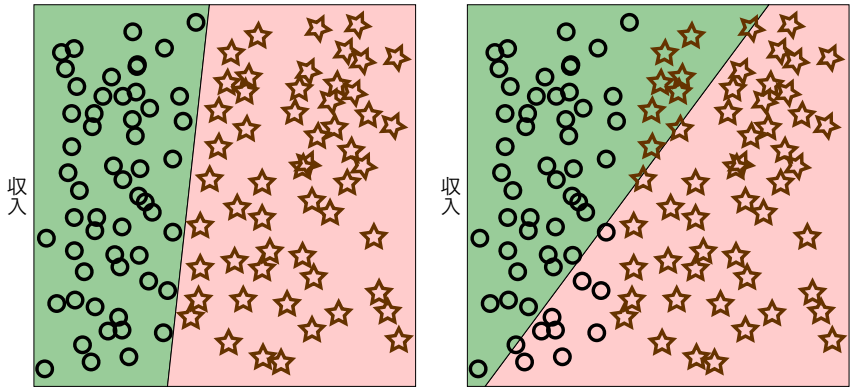
学習機の出来栄えを考える

これから入ってくるデータが重要である.



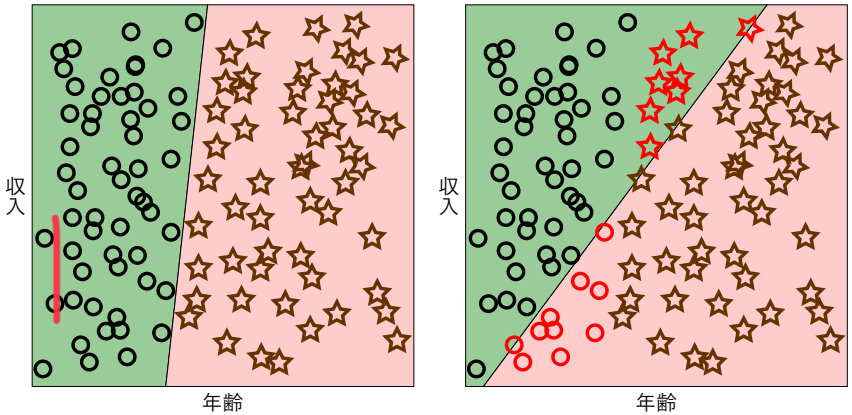
学習機の出来栄えを考える

これから入ってくるデータが重要である.



学習機の出来栄えを考える

これから入ってくるデータが重要である.



より現実的なモデルへ

A 氏 (箕面在住)

年齢	39 歳
性別	男
年収	650 万
持ち家	なし
借金元本金額	200 万
現職勤続年数	2 年
債務不履行歴	なし
今回の不履行	

B 氏 (豊中在住)

年齢	39 歳
性別	男
年収	650 万
持ち家	なし
借金元本金額	200 万
現職勤続年数	2 年
債務不履行歴	なし
今回の不履行	

C 氏 (吹田在住)

年齢	39 歳
性別	男
年収	650 万
持ち家	なし
借金元本金額	200 万
現職勤続年数	2 年
債務不履行歴	なし
今回の不履行	

より現実的なモデルへ

A 氏 (箕面在住)

年齢	39 歳
性別	男
年収	650 万
持ち家	なし
借金元本金額	200 万
現職勤続年数	2 年
債務不履行歴	なし
今回の不履行	なし

B 氏 (豊中在住)

年齢	39 歳
性別	男
年収	650 万
持ち家	なし
借金元本金額	200 万
現職勤続年数	2 年
債務不履行歴	なし
今回の不履行	なし

C 氏 (吹田在住)

年齢	39 歳
性別	男
年収	650 万
持ち家	なし
借金元本金額	200 万
現職勤続年数	2 年
債務不履行歴	なし
今回の不履行	あり

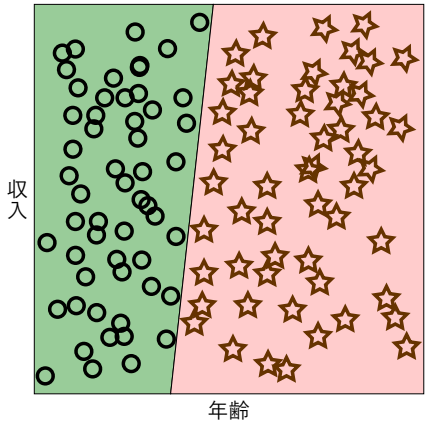
より現実的なモデルへ [1]

パーセプトロンの仮定には無理がある

- 最大の問題は「不確実性がない」と暗黙のうちに仮定していること.
- 境界線を一ミリでも超えたら 100%デフォルトなどは考えられない.

つまり, 下記のような「魔法の予測方法」が存在するという仮定になってしまう.

$$y = \mathrm{sign}\left(\sum_{j=1}^{d} w_j^* x_j - w_0^*\right)$$



[1] 線形分離可能性を前提にすると, 前スライドの状況はあり得ない.

より現実的なモデルへ

ノイズ (=不確実性) を許容する

- 従来の仮定では, 魔法のような関数が許されてしまう.

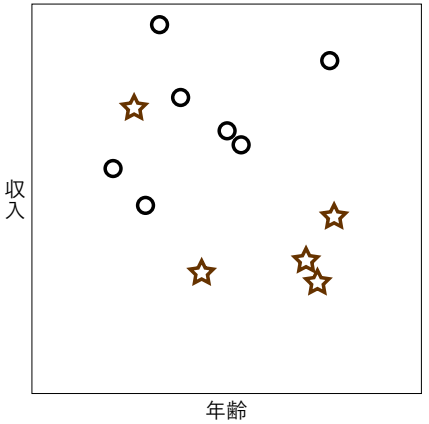
$$y = f^*(x), \qquad x \in \mathbb{R}^d.$$

- その代わりに, 複雑かつ不確実な関係を表せる確率モデルを導入する.
- 入力分布：$\mathbf{P}\{X = x\}$
- 出力の条件つき分布：$\mathbf{P}\{Y = y \,|\, X = x\}$

確率を導入すると, 下記のようなモデルが可能になる.

$$y = f^*(x) + \epsilon$$
$$y = f^*(x)(1 + \epsilon)$$
$$y = f^*(x + \epsilon)$$
$$y = f^*_\epsilon(x)$$

より現実的なモデルへ



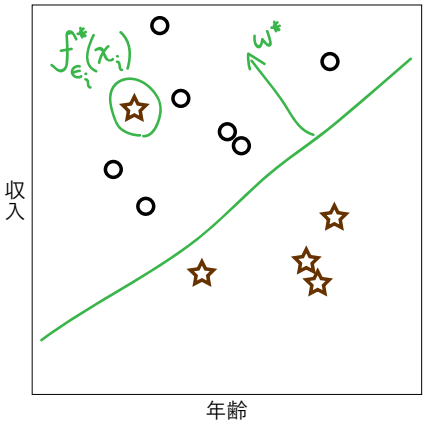
左図は決定論的な線形モデルでは説明できない.

線形性の仮定を諦めるか, ノイズを認めるか. 後者だと, たとえば

$$y = \mathrm{sign}\left(\sum_{j=1}^{d} w_j^* x_j + \epsilon - w_0^*\right)$$

というようなモデルならば, 線形性を残したまま, ノイズ ϵ の如何によって, 左図のデータが十分ありうる.

より現実的なモデルへ



左図は決定論的な線形モデルでは説明できない.

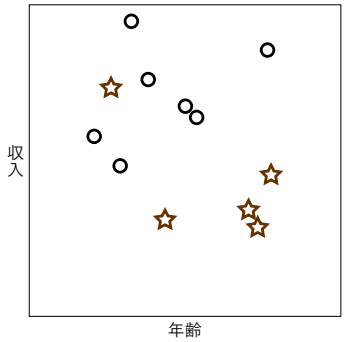
線形性の仮定を諦めるか, ノイズを認めるか. 後者だと, たとえば

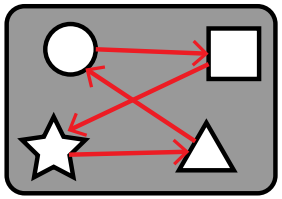
$$y = \mathrm{sign}\left(\sum_{j=1}^{d} w_j^* x_j + \epsilon - w_0^*\right)$$

というようなモデルならば, 線形性を残したまま, ノイズ ϵ の如何によって, 左図のデータが十分ありうる.

より現実的なモデルを見据えたアルゴリズムづくり

PLA ではデータを平面で分けられないと終了しない. そこで簡単な改善策を考える.[2]

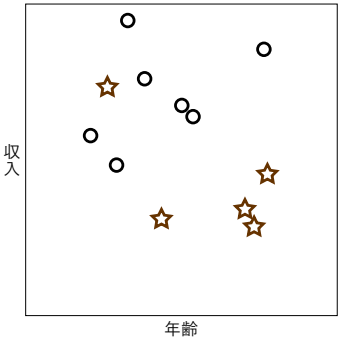




[2] このアイディア (Pocket algorithm と呼ぶ) は至って単純で, 古くから重宝されている (Gallant, 1990).

何を可視化してる?

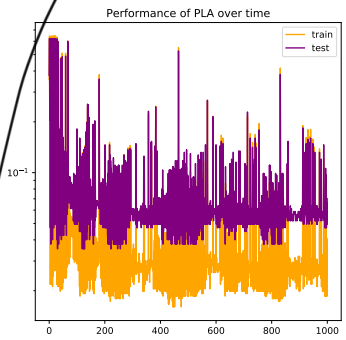
より現実的なモデルを見据えたアルゴリズムづくり



パーセプトロン学習則 **(PLA)**

1: initialize $t \leftarrow 0$, $w_{(0)}$ randomly.
2: initialize $\mathcal{I} \leftarrow \{1, \ldots, n\}$.
3: **while** $|\mathcal{I}| > 0$ **do**
4: $\quad \mathcal{I} \leftarrow \{i : h_{(t)}(x_i) \neq y_i\}$
5: $\quad$ randomly select $k \in \mathcal{I}$.
6: $\quad w_{(t+1)} \leftarrow w_{(t)} + y_k x_k$
7: $\quad t \leftarrow t + 1$
8: **return** $w_{(t)}$

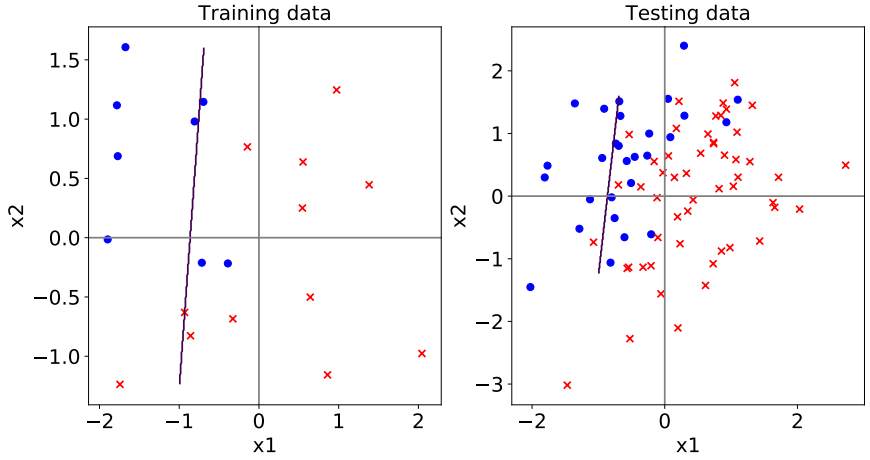
より現実的なモデルを見据えたアルゴリズムづくり



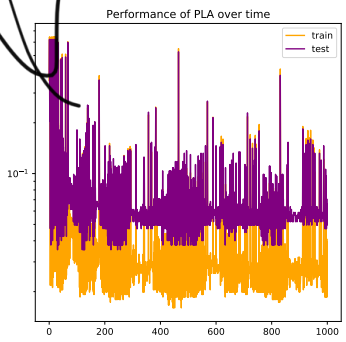
パーセプトロン学習則 **(PLA)**

1: initialize $t \leftarrow 0$, $w_{(0)}$ randomly.
2: initialize $\mathcal{I} \leftarrow \{1, \ldots, n\}$.
3: **while** $|\mathcal{I}| > 0$ **do**
4: $\quad \mathcal{I} \leftarrow \{i : h_{(t)}(x_i) \neq y_i\}$
5: $\quad$ randomly select $k \in \mathcal{I}$.
6: $\quad w_{(t+1)} \leftarrow w_{(t)} + y_k x_k$
7: $\quad t \leftarrow t + 1$
8: **return** $w_{(t)}$

より現実的なモデルを見据えたアルゴリズムづくり



より現実的なモデルを見据えたアルゴリズムづくり



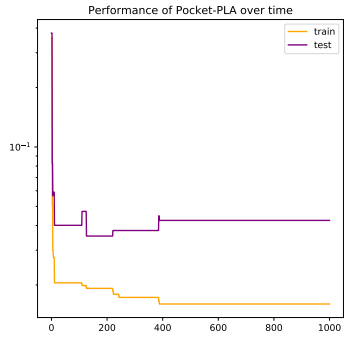
パーセプトロン学習則 **(PLA)**

1: initialize $t \leftarrow 0$, $w_{(0)}$ randomly.
2: initialize $\mathcal{I} \leftarrow \{1, \ldots, n\}$.
3: **while** $|\mathcal{I}| > 0$ **do**
4: $\quad \mathcal{I} \leftarrow \{i : h_{(t)}(x_i) \neq y_i\}$
5: $\quad$ randomly select $k \in \mathcal{I}$.
6: $\quad w_{(t+1)} \leftarrow w_{(t)} + y_k x_k$
7: $\quad t \leftarrow t + 1$
8: **return** $w_{(t)}$

Pocket Algorithm の説明

分類の誤りを最小にする

より現実的なモデルを見据えたアルゴリズムづくり



PLA ＋ポケット則

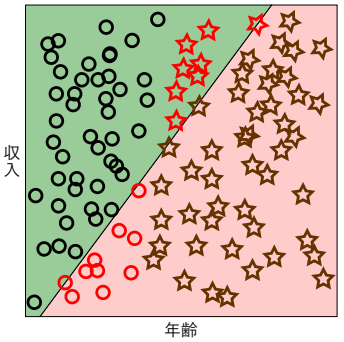
1: initialize $w_{\text{best}} = w_{(0)}$, $r_{\text{best}} = n$.
2: initialize $\mathcal{I} \leftarrow \{1, \ldots, n\}$.
3: **for** $t = 0, 1, 2 \ldots$ **do**
4: $\quad \mathcal{I} \leftarrow \{i : h_{(t)}(x_i) \neq y_i\}$
5: $\quad$ **if** $|\mathcal{I}| < r_{\text{best}}$ **then**
6: $\quad\quad r_{\text{best}} \leftarrow |\mathcal{I}|$
7: $\quad\quad w_{\text{best}} = w_{(t)}$
8: $\quad$ randomly select $k \in \mathcal{I}_{(t)}$
9: $\quad w_{(t+1)} \leftarrow w_{(t)} + y_k x_k$
10: **return** w_{best}

何をもって評価すべきか

予測能力を重視
$\widehat{h}(X) \approx f^*(X)$ の関数近似ではなく $\widehat{h}(X) \approx Y$ を追求.

- 魔法の関数からの距離ではなく, ノイズも加味した予測性能.
- 確率変数 X と Y はこれから入ってくるデータ.
- 評価時には, データの分布を考慮しないと効率が悪い.

データ分布と汎化能力
これから入ってくるデータでの性能を考えると, たとえば予測誤差の期待値を指標とするなど.

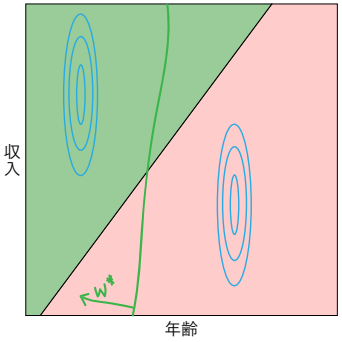


何をもって評価すべきか

予測能力を重視
$\widehat{h}(X) \approx f^*(X)$ の関数近似ではなく $\widehat{h}(X) \approx Y$ を追求.

- 魔法の関数からの距離ではなく, ノイズも加味した予測性能.
- 確率変数 X と Y はこれから入ってくるデータ.
- 評価時には, データの分布を考慮しないと効率が悪い.

データ分布と汎化能力
これから入ってくるデータでの性能を考えると, たとえば予測誤差の期待値を指標とするなど.

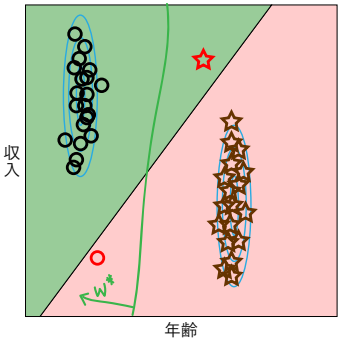


何をもって評価すべきか

予測能力を重視
$\widehat{h}(X) \approx f^*(X)$ の関数近似ではなく $\widehat{h}(X) \approx Y$ を追求.

- 魔法の関数からの距離ではなく, ノイズも加味した予測性能.
- 確率変数 X と Y はこれから入ってくるデータ.
- 評価時には, データの分布を考慮しないと効率が悪い.

データ分布と汎化能力
これから入ってくるデータでの性能を考えると, たとえば予測誤差の期待値を指標とするなど.



何をもって評価すべきか：一つの具体例

指紋画像 x に基づく判断：

- $h(x) = +1$ なら「本人」.
- $h(x) = -1$ なら「不審者」.

	$y = +1$	$y = -1$
$h(x) = +1$	正解	偽陽
$h(x) = -1$	偽陰	正解

何をもって評価すべきか：一つの具体例

指紋画像 x に基づく判断：

- $h(x) = +1$ なら「本人」.
- $h(x) = -1$ なら「不審者」.

	$y = +1$	$y = -1$
$h(x) = +1$	正解	偽陽
$h(x) = -1$	偽陰	正解

	$y = +1$	$y = -1$
$h(x) = +1$	0	5
$h(x) = -1$	10	0

Table: 女子大生のスマホの場合

何をもって評価すべきか：一つの具体例

指紋画像 x に基づく判断：

- $h(x) = +1$ なら「本人」.
- $h(x) = -1$ なら「不審者」.

	$y = +1$	$y = -1$
$h(x) = +1$	正解	偽陽
$h(x) = -1$	偽陰	正解

	$y = +1$	$y = -1$
$h(x) = +1$	0	5
$h(x) = -1$	10	0

Table: 女子大生のスマホの場合

	$y = +1$	$y = -1$
$h(x) = +1$	0	1000
$h(x) = -1$	5	0

Table: 銀行の金庫室の場合

何をもって評価すべきか

学習機の「出来栄え」を数値化すべく, 設計者が損失関数を使う.[3]

損失関数の例

- 二乗誤差：$L(h; x, y) = (h(x) - y)^2$
- 識別誤差：$L(h; x, y) = I\{h(x) \neq y\}$
- 過誤依存の誤差：

$$L(h; x, y) = \begin{cases} a_+, & \text{if FP} \\ a_-, & \text{if FN} \\ 0, & \text{else.} \end{cases}$$

学習時と最終評価時とで, 異なる損失関数を使うことも多々ある.

[3] ロス関数, コスト関数とも呼ばれる.

何をもって評価すべきか

損失値の分布を見る

データ $Z_1, \ldots, Z_n$ に基づいて, いわば「損失値のデータセット」を作る.

$$\{L(h; Z_1), \ldots, L(h; Z_n)\}, \qquad h \in \mathcal{H}.$$

任意の候補 h に対して, 新たなデータセットが得られるイメージ.

これは当然, 損失値の確率分布からの標本 (サンプル) と捉えられる.

代表値を考える

「最良の分布にしなさい」というプログラムはなかなか書けない...

候補 h の損失値を算出 $\to$ 代表値を決める $\to$ 代表値を小さくすべく h を更新

最適化が楽になる上に, 統計的な知見も働く. 一石二鳥の戦略.

何をもって評価すべきか

汎化能力を推し量る

典型的な代表値は算術平均である.

$$\widehat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} L(h; Z_i) \approx R(h)$$

この推定量が近似する $R(h)$ とは：

$$R(h) \coloneqq \mathbf{E}_{\mathsf{P}}\, L(h; Z) = \int L(h; z)\, \mathsf{P}(\mathrm{d}z).$$

前者は観測できるが, 後者は近似しかない. これが真の汎化能力の指標である.

まとめ

機械学習は統計的推論と最適化の共同作業である.

現実的なモデルから有効なアルゴリズム設計

- 甘い仮定に基づくアルゴリズムの性能は不安定になりやすい.
- データを生成する「自然界の過程」は基本的には不確実である.
- 単純な確率モデルでも, 表現できるデータの幅が大きく拡がる.

フィードバック設計は多面的

- 叩き出す数値と, 本当の目的の達成度合いの乖離は要注意.
- 統計の側面：損失値が多数あるなか, 適切な代表値を選ぶことが多い.
- 最適化の側面：代表値を効率よく, 安定的に小さくすることが重要.

※関心のある方は付属の演習課題にも取り組んでみてください.

参考文献

Gallant, S. I. (1990). Perceptron-based learning algorithms. *IEEE Transactions on Neural Networks*, 1(2):179–191.