

Perceptrons の由来 Perceptrons origin

□1956年 Frank Rosenblatt (フランク・ローゼンブラット) により提案されたのが、線形クラス分類を学習するための最初の反復アルゴリズムで、これがパーセプトロンである。私が最初にこのパーセプトロンに出会ったのが、今から 27 年以上も前のことで、人工知能の創始者である Marvin Minsky (マービン・ミンスキー) が書いた「Perceptrons」(MIT 出版) の Expanded Edition であった。当時は、この学習内容でも斬新なアイデアであって、少し興奮気味に読んでいたのを覚えている。でも、時が経ち、今では誰も見向きもしない理論になってしまった。小生の傍らには今でもそのボロボロになった洋書がある。

しかし、この理論が初めて学習についての定義をし、重みベクトル (Weight Vector) やバイアス (Bias) という名称を付けて、最近の SVM (Support Vector Machine) やニューラルネット、強化学習などの学習理論へと繋がってきたことは確かです。パーセプトロンの学習コンセプトは、「失敗は成功の母」である。成功は後に何も残さないが、失敗は成功への学習を残す…というのが、基本的な考え方である。では、具体的にパーセプトロンのアルゴリズムをみていきましょう。

【定義 1】 一般に入力空間を X 、出力定義域を Y とした場合、入力空間は、 $X \subseteq R^n$ になる。出力定義域は、2 値クラス分類に対して、 $Y = \{-1, 1\}$ 、 m クラス分類なら $Y = \{1, 2, \dots, m\}$ 、回帰に対して、 $Y \subseteq R$ である。トレーニング集合 (Training Set) は、トレーニング事例 (Training Examples) の集合である。なお、トレーニング事例は、トレーニングデータ (Training Data) ともいわれる。通常、入力データと出力データとが対になった、 $S = ((x_1, y_1), \dots, (x_l, y_l)) \subseteq (X, Y)^l$ で表記される。(但し、 l を事例の数とする。)

x_i を事例 (Examples) もしくはインスタンス (Instances) ともいい、 y_i を事例のラベル (Labels) とも呼ぶ。事例のラベルがすべて同じ場合、トレーニング集合 S は自明 (Trivial) であるという。なお、もし X がベクトル空間であれば、入力ベクトルは、重みベクトルとしての列ベクトルであることに注意されたい。 x_i からの行ベクトルを形成したいならば転置 x_i^T をとることができる。ベクトル空間は交換律が成り立つので…。

入力 $x_i \in X$ を列ベクトルにし、出力 $y_i \in Y$ を行ベクトルとした行列 D をつくる。

そうすると、上の $S = ((x_1, y_1), \dots, (x_l, y_l)) \subseteq (X, Y)^l$ は、対角行列になります。

すなわち、 $|D| = l$ となり標数が l 個あることを示し、この l が基底の個数になります。

そこで、固有値問題の対角行列は固有値ですから、この行列 D は、重みベクトルになります。…ということは、基底も l 個なので、固有ベクトル $l \times l$ 行列が生成されることになりま

す。

次元削減は、この l 個の固有値の大きい順に並べ、対数化をし、正規化をした上で、無視しても良い小さな基底を除き、 $k < l$ 個の次元に $k \times k$ 行列をつくる。また、視点を変えて代数的にみれば、 $y = ax + b$ の a が重みで、 b がマージンになります。

写像では、 $f: X \rightarrow Y$ になり、 f が重みになり、求める解となる。行列で書くと、 $fX = Y$ という f 行列 $l \times l$ を求めることになる。

上の「 $\dots k \times k$ 行列をつくる」というところで、同義語／類義語／関連語などの語彙の関連度を抽出したい場合にこれらは使う。しかし、選んだサンプルデータに偏りやノイズが含まれている可能性があり、精度が出ない場合がありますので、これを除去する方法として、固有ベクトル $m \times k$ と片方の固有ベクトルの転置行列 $k \times n$ の真ん中にこの $k \times k$ 行列をはさめば、得られる行列は $m \times n$ という D 行列と同じになりますが、中身が違います。何故、同じ D 行列の中身が違ってしまふかということ、最初は基底が m 個から l 個にし、大きい基底だけを採用し、 k 個にしました。この k 個に次元削減した値域集合へ写像して得られた新しい D 行列は、偏りやノイズを除去した D 行列になってます。

【主形式のパーセプトロンアルゴリズム】

線形分離可能なトレーニング集合を S 、学習率を $\eta \in R^+$ とする。

$w_0 \leftarrow 0; b_0 \leftarrow 0; k \leftarrow 0$

$R \leftarrow \max_{1 \leq i \leq l} \|x_i\|$

repeat

 for $i = 1$ to l

 if $y_i (< w_k \cdot x_i > + b_k) \leq 0$ then

$w_{k+1} \leftarrow w_k + \eta y_i x_i$

$b_{k+1} \leftarrow b_k + \eta y_i R^2$

$k \leftarrow k + 1$

 end if

 end for

until 「for ループ内で失敗をしなくなる」

return (w_k, b_k) ただし、 k は失敗の数

【定義 2】

超平面 (w, b) に関する事例 (x_k, y_k) の (関数) マージン ((Functional) Margin) を $\gamma_i = y_i (< w \cdot x_i > + b)$ という量で定義する。

ここで、マージンが $\gamma_i > 0$ であれば、 (x_i, y_i) を正しくクラス分類したことを示唆する。「トレーニング集合 S に関する超平面 (w, b) の (関数) マージン分布」は、 S の事例のマージン分布である。マージン分布の最小値を、「トレーニング集合 S に関する超平面 (w, b) の (関

数) マージン」と呼ぶ。これらの定義で、「関数マージン」を「幾何マージン」で置き換え

れば、正規化線形関数 $\left(\frac{1}{\|w\|} w, \frac{1}{\|w\|} b \right)$ に同値な量を得る。したがって、幾何マージンは、「入

力空間での決定境界から点までのユークリッド距離」を測定する。最後に「トレーニング集合 S のマージン」は、すべての超平面にわたる最大幾何マージンである。その最大を実現する超平面は、「最大マージン超平面」として知られている。そのマージンの大きさは、線形分離可能なトレーニング集合に対し正である。

【Novikoff の定理 3】

S を自明でないトレーニング集合とし、また、

$$R = \max_{1 \leq i \leq l} \|x_i\| \quad \text{とする。}$$

$1 \leq i \leq l$ に対し、 $\|w_{opt}\| = 1$ で、しかも

$$y_i (< w_{opt} \cdot x_i > + b_{opt}) \geq \gamma \quad \text{となるようなベクトル } w_{opt} \text{ が存在すると仮定する。}$$

すると、トレーニング集合 S に対し、オンラインのパーセプトロンアルゴリズムが失敗する数は、高々、

$$\left(\frac{2R}{\gamma} \right)^2 \quad \text{となる。}$$

通常、この定理はバイアスをゼロとして与えられる。その場合、失敗数の限界はこの定理より 4 倍良いものとなる。しかしながら、バイアスは、パーセプトロンのアルゴリズムで更新されるので、標準的な更新を想定した場合、失敗数は、拡張重みベクトルをもつトレーニング集合のマージンに依存する。マージンが大きければ、失敗が大きくなる…ということ

ことです。このマージンは常に γ 以下で、非常に小さくなり得る。 $b_{opt} = 0$ の場合には、 γ に等しくなる。したがって、拡張トレーニング集合によるマージンを用いた限界では、一般に 4 倍良いものとなるが、 $b_{opt} = 0$ の場合には、証明の最終行で 2 の因子を導入せず、2

倍良いに留まる。対照的に、 $R > 1$ として、 $|b_{opt}| = O(R)$ の場合には、拡張重みベクトルをもつトレーニング集合で得る限界は、この定理の限界より $O(R^2)$ 倍悪くなる。

限界の決定的な量は、『データを含む球の半径』と『分離超平面のマージン』の比の 2 乗である。これを理解するイメージは、ピタゴラスの定理と最適解は球に対して直行する…ということだけです。(数学の厳密な条件である凸球体などは省きますが…) この比は、

データの正のスケール変更に対する不変量である。スケール変更がアルゴリズムで必要とする反復数に影響しないのは明らかである。もっとも、この反復数が学習率に依存しないというのは、最初は直感的ではないかもしれない。重みとバイアスの両者をスケール変更しても、クラス分類を変化させないので、線形閾値関数の記述に自由度がある理由による。この事実を「分離可能なトレーニング集合に対する正準最大マージン超平面」を定義するために用いることができる。定義はマージンを1に固定し、重みベクトルのノルムを最小化することによる。結果のノルム値は、マージンに反比例する。

Novikoffの定理は、「マージンが正であれば、有限数の反復でアルゴリズムが収束すること」を示している。超平面が存在する場合には、同じトレーニング列 S に対し反復実行し

さえすれば、パーセプトロンのアルゴリズムは高々 $\left(\frac{2R}{\gamma}\right)^2$ 回の失敗で分離する超平面を発見し、停止する。

データが線形可能でない場合、アルゴリズムは収束しない。トレーニング列 S に対し反復実行する場合、クラス分類した点を発見する度に仮説 w_i は変更し続け、振動する。しかしながら、この状況には Novikoff の定理と同様の定理が存在する。1回の反復で発生する失敗の数に上限を定める。これはマージン分布とは異なる尺度を用いている。直感的にトレーニング標本のさらに大域的特性を説明するために、「超平面に最も近いトレーニング点からのみ得られたマージン」ではなく、「トレーニング点すべてを考慮して得られたマージン」を用いる。このマージンにより、マージンの概念を汎化する。このマージン分布の尺度は、標本の分離不可能性の測定にも用いられる。

【定義 4】

マージン $\gamma > 0$ を固定し、超平面 (w, b) とターゲットマージン γ に対する事例のマージンスラック変数 (MarginSlackVariable) を

$$\xi((x_i, y_i), (w, b), \gamma) = \xi_i = \max(0, \gamma - y_i (< w \cdot x_i > + b))$$

と定義する。非公式には、この量で「点が超平面からマージン γ をもつことに失敗する量」を測定する。もし、 $\xi_i > \gamma$ であれば、 x_i は (w, b) により、クラス分類ミスをする。

ノルム $\|\xi\|_2$ は、トレーニング集合がマージン γ をもつことに失敗する量を測定し、トレーニングデータのクラス分類ミスのすべてを考慮に入れる。

【Freund&Schapire の定理 5】

S を重複事例のない、自明でないトレーニング集合とし、 $\|x_i\| \leq R$ とする。 (w, b) を $\|w\| = 1$ を満たす任意の超平面とする。さらに $\gamma > 0$ とする。そして、

$$D = \sqrt{\sum_{i=1}^l \xi_i^2} = \sqrt{\sum_{i=1}^l \xi((x_i, y_i), (w, b), \gamma)^2} \quad \text{と定義する。}$$

すると、トレーニング集合 S に対するパーセプトロンアルゴリズムの「for ループ」の 1 回目の実行における失敗の数は、高々

$$\left(\frac{2(R+D)}{\gamma} \right)^2 \quad \text{回となる。}$$

定理を適用できるのが、「for ループ」の 1 回目の反復のみである理由を述べよう。拡張空間において、トレーニング事例 \tilde{x}_i は、重みベクトルの更新に用いられた後、重みベクトルの i 番目の拡張座標は、ゼロでない要素をもつ。したがって、これが続く反復で用いられた場合、 \tilde{x}_i の評価に影響してしまう。これらの拡張座標をも含めた形でパーセプトロンアルゴリズムの適用をすればどうかという意見があるかもしれない。しかし、 Δ の値はパラメータであるか、計算の一部として推定されるものでなければならない。

D は任意の超平面に対して定義できる。したがって、定理の失敗数の限界は、データが線形分割可能であることに依存しない。クラス分類ミスの最小数をもつような「分割不可能なデータの線形分離を発見する問題」は、NP-完全である。この問題に対しては、いくつもヒューリスティックなやり方が提案されている。例えば、ポケットアルゴリズム (PocketAlgorithm) は、反復を最も長い間生き残る w を出力する。前に述べた拡張は、分離不可能なデータのためのパーセプトロンアルゴリズムを導出するのに用いることができる。

さて、パーセプトロンのアルゴリズムの機能する仕組みには注意されたい。任意の重みベクトルを初期値として、クラス分類ミスした正のトレーニング事例を追加するか、クラス分類ミスした負のトレーニング事例を除去するかの選択を繰り返すことにより機能することとなる。こう見ることは重要である。まず、一般性を失わず、重みベクトルの初期値はゼロベクトルと仮定できる。したがって、最後の仮説をトレーニング点の線形結合

$$w = \sum_{i=1}^l \alpha_i y_i x_i \quad \text{と書くことができる。}$$

ここで、 x_i の係数の符号はクラス分類 y_i により与えられる。また、 α_i は正值で、「 x_i のクラス分類ミスが重みを更新させた回数」に比例する。失敗の少ない点は小さい α_i をもち、失敗の多い点は大きな値をもつ。この量は、「パターン x_i の埋め込み力 (EmbeddingStrength)」と呼ばれることもある。一旦、標本 S が固定されると、「ベクトル α 」を「仮説の異なる座標 (あるいは双対座標) における表現」と考えることができる。しかしながら、この拡張は、一意なものではない。なぜなら、同じ仮説 w に異なる α を対応させることができるからである。一方、直観的に α_i を「事例 x_i の内容を示唆するもの」と見ることもできる。例えば、分離不可能なデータの場合、クラス分類ミスした点の係数

は無限に大きくなる。

以上により、決定関数は双対座標を用いて、

$$\begin{aligned} h(x) &= \text{sgn}(\langle w \cdot x \rangle + b) \\ &= \text{sgn} \left(\left\langle \sum_{j=1}^l \alpha_j y_j x_j \cdot x \right\rangle + b \right) \\ &= \text{sgn} \left(\sum_{j=1}^l \alpha_j y_j \langle x_j \cdot x \rangle + b \right) \end{aligned}$$

と書き換えることができる。パーセプトロンのアルゴリズムは、完全にこの双対形式で表現することも可能である。もっとも学習率は、超平面のスケール変更をするのみで、アルゴリズムをゼロの初期ベクトルで開始することには何の影響も与えない。したがって、すでにこれを含めていないことには注意されたい。

■ 【双対形式のパーセプトロンアルゴリズム】

与えられたトレーニング集合 S に対し、

$$\alpha \leftarrow 0; b \leftarrow 0$$

$$R \leftarrow \max_{1 \leq i \leq l} \|x_i\|$$

repeat

 for $i = 1$ to l

$$\text{if } y_i \left(\sum_{j=1}^l \alpha_j y_j \langle x_j \cdot x_i \rangle + b \right) \leq 0 \text{ then}$$

$$\alpha_i \leftarrow \alpha_i + 1$$

$$b \leftarrow b + y_i R^2$$

 end if

 end for

until 「for ループ」 内で失敗をしない

return (α, b) これが関数 $h(x) = \text{sgn} \left(\sum_{j=1}^l \alpha_j y_j \langle x_j \cdot x \rangle + b \right)$ を定義する。

このパーセプトロンアルゴリズムの双対形式および決定関数の双対は、いくつもの興味のある特性をもつ。たとえば学習の困難なトレーニング点は、 α_i が大きな値をもつという事実は、 α_i の中身に従い、データを順序づけ（ランクづけ）するのに用いることができる。以上、単純なパーセプトロンアルゴリズムの解析で、実にサポートベクターマシンの理論

で用いる重要概念である「マージン」、「マージン分布」、「双対表現」を見出すことができる。

更新の数は、失敗の数と等しい。また、各更新ではベクトル α の成分の1つのみに1を加算することとなる。したがって、ベクトル α の1-ノルムは、定理3で与えられる失敗数に対する限界を満足して、

$$\|\alpha\|_1 \leq \left(\frac{2R}{\gamma}\right)^2$$

となる。したがって、「 α の1-ノルム」を「双対表現でのターゲット概念の複雑さの尺度」として見ることができる。

アルゴリズムにおいて、行列 $G = (\langle x_i \cdot x_j \rangle)_{i,j=1}^l$ というインターフェース以外では、トレーニングデータは現れない。この行列は、グラム行列 (GramMatrix) として知られる。以上である。

線形であることの簡潔さと単純さによって、判り易く、計算もし易く、処理も速いのが特徴だが、言語解析に用いるのには少し不十分である。言語解析は、表層的には線形ではなく、非線形であることは明示的であり、その上、深層的には意味概念へ写像すると連続という空間が存在する。「言語は人間どうしのコミュニケーションや脳内での推定の道具である。」…とすれば、脳内での処理は連続した空間で処理され、それを言語に変換した瞬間、穴あきの離散データになってしまう…ことに、曖昧さや誤解が存在するので、脳内での連続した推定や推量から言語という離散集合への写像の過程で、何が削減されたか…を考察するためには、ユークリッド空間ではなく、トポロジーやコホモロジー、群、行列、景そして層などが必要になってくる。PerceptronsEngine と称した深い理由が少しでも判って頂きたい一番の人たちは、これに多少でも関わったエンジニア諸君です。【第3版】