

マルチテンプレート辞書のLVQ学習窓の検討\*

2L-6

新村貴彦† 出本 浩† 坂野 鋭†  
株式会社NTTデータ‡

1. はじめに

Kohonenによって提案された学習ベクトル量子化(LVQ: Learning Vector Quantization)は強力なパターン認識辞書設計アルゴリズムとして知られ<sup>[1]</sup>, 様々な改良が行われてきた<sup>[2]</sup>. 特に学習パターン選択の問題は早期から指摘され, Kohonen自身LVQ2.1の中で識別境界付近を重点的に学習するためにアポロニウスの円を用いた学習窓を提案し, 実験的に有効性を示している.

しかし, アポロニウスの円による学習窓(以下アポロニウス円窓)は, 特徴空間で広い範囲に分布する学習パターンを選択するため学習が進むと窓内の正解パターンが増える傾向があり, 誤り訂正の効果が低下するという問題がある. そこで, 我々は誤りパターンの割合を増やすことを考え, アポロニウス円窓に誤読パターンを重視する新たな窓関数を付加する手法を提案する.

以下, 提案手法を説明し, 文字認識実験を通して有効性を示す. さらに, 学習窓が選択したデータの統計的な性質を解析することで提案手法が有効である理由を考察する.

2. LVQと学習窓

LVQには種々の改良があるが, ここでは実験で用いるLVQ3について説明する. 基本的にLVQ3はユークリッド距離を用いた最近接近傍規則を誤り訂正で学習する方法である.

LVQでは最初に1カテゴリあたり1つもしくはそれ以上のテンプレートをクラスタリング等の手法で用意する. 学習過程では入力された学習パターンに対して最も近い2つのテンプレートを選び, 正解テンプレートについては学習パターンに近づけるように, 誤りテンプレートは学習パターンから遠ざけるように更新する. すなわち,

$$\vec{\mu}_{ct+1} = \vec{\mu}_{ct} + \alpha (\vec{x}_c - \vec{\mu}_{ct}), \quad (1)$$

$$\vec{\mu}_{et+1} = \vec{\mu}_{et} - \alpha (\vec{x}_c - \vec{\mu}_{et}) \quad (2)$$

である. 但しここで

$\vec{\mu}_c, \vec{\mu}_e$                     正解, 誤判別テンプレート  
 $\alpha, \epsilon$                         係数  
 $\vec{x}_c$                             学習パターン

である. LVQ3にはテンプレート位置の発散を押さえる工夫が追加されている. 学習パターンに最も近いテンプレートが2つとも誤りならば正解カテゴリと誤りカテゴリから学習パターンに近いテンプレートをそれぞれ選ぶ.

※ A proposition of the LVQ window with a multitemplate dictionary

† Takahiko SHIMMURA, Hiroshi IZUMOTO, Hitoshi SAKANO    ‡ NTT DATA CORPORATION

E-mail: shinmura@rd.nttdata.co.jp

また2つが正解ならば2つとも学習パターンに近づけるように更新する. つまり

$$\vec{\mu}_{ct+1} = \vec{\mu}_{ct} + \alpha \epsilon (\vec{x}_c - \vec{\mu}_{ct}) \quad (3)$$

である.

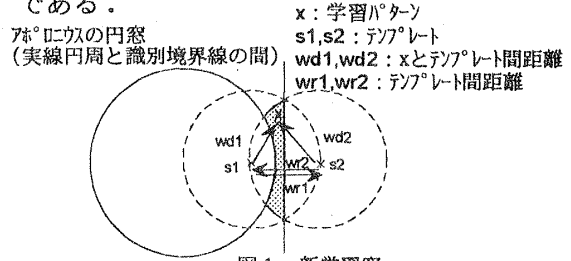


図1 新学習窓

アポロニウスの円窓は識別境界付近の学習パターンを選び出すために用いられる. その範囲は円周の外とテンプレート間の2等分線で挟まれた範囲である. つまり

$$\frac{wd1}{wd2} \geq r \quad (4)$$

$$wd1 \leq wd2 \quad (5)$$

で定義される. rは窓サイズと呼ばれる. 学習の範囲を決定するパラメータである. しかし, 学習が進み窓内の誤りパターンが少なくなると誤り訂正のためのパターンが少なくなり学習が進まなくなる. 提案する窓は, 誤りパターンの集中する範囲に設定する. 具体的には, 図1に示すようにテンプレート間の距離を半径とする円を各々のテンプレートを中心に描き, この範囲とアポロニウス円窓の重なる範囲を学習窓とする. すなわち

$$\frac{wd1}{wd2} \geq r \quad (6)$$

$$wd1 \leq wd2 \quad (7)$$

$$(wd1 \leq wr1) \wedge (wd2 \leq wr2) \quad (8)$$

である.

この学習窓はアポロニウス円窓に比べ誤判別されるデータの占有率が高いと予測されるため, 誤り訂正が効率的に進むと考えられる.

以後, アポロニウスの円窓による方法をLVQ3, 新学習窓によるそれをLVQ3.2と呼ぶことにする.

3. 手書き文字認識の実験

提案手法の有効性を確かめるために手書き片仮名認識実験を行った. 実験条件を表1に示す. なお予備実験よりアポロニウス円窓のサイズと更新係数の値を最適値に定めた.

初期テンプレートはカテゴリごとのサンプルの平均値を用い, 出本の方法によりマルチテンプレートを作成した<sup>[5]</sup>. また, 距離計算では重み付距離を使用した. 表2に認識率を示す. この結果からLVQ3.2の方がよいことが分かる.

4. 考察

4.1 窓内学習データ数の変化

学習パターン数と認識率の関係を知るために窓中の学習データについて正しく認識されたもの(正)、誤って認識されたもの(誤)の数の変化を調べた(図2)。LVQ3では両者の合計数が多く、正解パターンがそのほとんどである。誤りパターン数が圧倒的に少なく、学習回数を重ねても誤りパターン数も合計パターン数も減らない。一方、LVQ3.2ではパターンの合計数が少なく、誤りパターン数の占有率が比較的高い。また、学習回数を重ねると誤りパターン数が減るにつれ合計数も減る。これは誤りパターンによる識別境界の学習が進んでいるためと考えられる。

4.2 窓内学習カテゴリの変化

窓の違いがテンプレートの学習にどのような違いを与えたかを調べた。窓内のデータの識別結果を正解、誤りに分けカテゴリ別にまとめ、個数の多い上位を表3に示した。LVQ3.2では誤りカテゴリが正解カテゴリの類似カテゴリになっており、両カテゴリの識別困難な領域を学習していると考えられる。しかし、LVQ3必ずしも学習対象に類似カテゴリが選択されているわけではなく学習が効率的に行われていないことが分かる。実際に”ボボ”のテンプレートを特徴空間の2軸で観察するとLVQ3では更新が滞っているが、LVQ3.2ではそれが活発であった。

以上から、誤りパターンの占有率を高くしていくことで、LVQ3.2では誤り訂正学習が効率的に進んでいると考えられる。この原因を図3のように考えた。同じカテゴリの学習データでも識別結果が誤りだと黒丸で書いてある。学習前にa)であって学習が進むと誤りデータが減る、b)。次にLVQ3では正解データ数と誤りデータ数に極端な差ができる、2つのテンプレートの更新に引っ張られて、誤りデータの学習、つまり識別境界の移動が適切に行われず、c)のようになる。逆にLVQ3.2では誤りデータ数の占有率が高いため、正解データに引き止められることがなく、誤りデータの学習が進みやすいと考えられる。

5. まとめ

新しい学習窓を使ったアルゴリズムLVQ3.2を提案し、その効果を文字認識実験により確認した。また、選択された学習パターンの振る舞いを調べ、提案手法が有効である理由を考察した。

謝辞

日頃有益な議論をして頂いている当社同僚諸氏に感謝します。

参考文献

[1] Teuvo Kohonen, "IMPROVED VERSIONS OF LEARNING VECTOR QUANTIZATION," Proc. of IJCNN Vol.1, 1990, 1-545  
 [2] コホネン, "自己組織化マップ," シェアリング・フェアテック東京株式会社, 1996年6月15日  
 [3] 佐藤, "一般学習ベクトル量子化の提案," 信学技法, NC95-60, 1995-10.  
 [4] 坂野, 他, "遺伝的アルゴリズムによる文字識別系の解析," 信学会論文誌, D-II Vol. J80-D-II No.7 pp.1687-1694  
 [5] 出本, 他, "LVQを用いた手書き漢字認識辞書作成方法の検討," 信学技法, PRU91-50, JUL 1991

	学習試料	評価試料
字種	カタカナ74カテゴリ	同じ
パターン数	1カテゴリ-5100	1カテゴリ-300~600
次元数	44 (ePDC)	同じ
作成方法	正準判別で次元圧縮	同じ

	初期辞書	LVQ3	LVQ3.2
平均	95.04	96.27	97.03

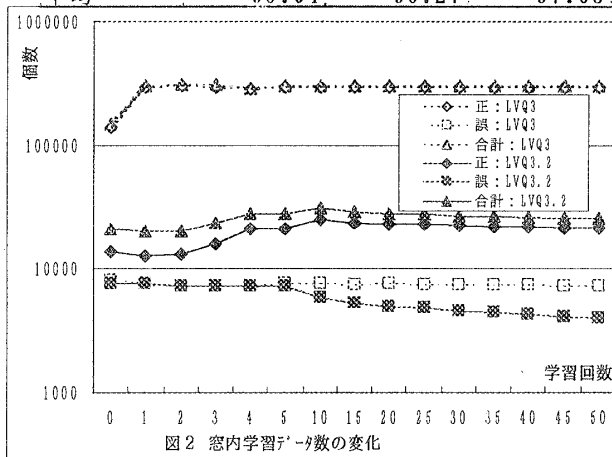


図2 窓内学習データ数の変化

		5	10	20	30	40	50
LVQ3	正	ヤ	ヤ	ヤ	ヤ	ヤ	ス
	誤	カ	カ	カ	カ	カ	ボ
LVQ3.2	正	フ	フ	フ	フ	フ	フ
	誤	ボ	ボ	ボ	ボ	ボ	ボ

s0: 正解テンプレート  
 s1: 対抗テンプレート  
 ○, ●: カテゴリ-s0データ  
 △, ▲: カテゴリ-s1データ

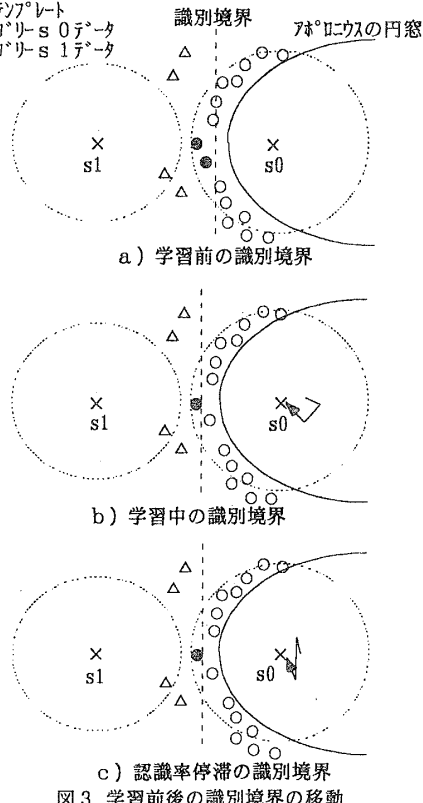


図3 学習前後の識別境界の移動