

拡張SODPマッチング法について

6N-9

花田 英輔 松永 純一 河口 英二
(九州大学大学院総合理工学研究科)

1. はじめに

我々は先にDPマッチング法の計算量を削減する目的で準最適DPマッチング法(SODPアルゴリズム)を提案した^{(1),(2)}。これは、最適パスを決定するために局所的な最適なパスをつなぎ合わせて全体のパスとしようとするものであり、通常のDPマッチング法⁽³⁾に比べ、計算量が1/4程度で済むことを報告した^{(1),(2)}。今回はそれを更に手直したマッチング法(拡張SODPマッチング法と呼ぶ。)を提案する。

2. SODPアルゴリズム

2つのパターン間の時間的歪みを正規化するためにDPマッチング法は有効であるが、計算量は膨大である。SODPアルゴリズムは、図1に示す様な前方の限られた領域内において最適パス(局所的最適パス)を見だし、そのパス上で起点を前方へ1段進めた上で次の領域について最適パス探索を行うという方法である。今、2つの音声パターンの特徴ベクトルの並びを、

$$S = S_1 S_2 \cdots S_{l-1} S_l S_{l+1} \cdots S_L$$

$$W = W_1 W_2 \cdots W_{k-1} W_k W_{k+1} \cdots W_K$$

とする。その類似度を $\omega(k, l)$ であらわし、パスを決定する評価関数 R^* を

$$R^* = \max(R(k, l), R(k+1, l), R(k, l+1))$$

とする。ここで、

$$R(k, l) = \omega(k, l) + R^*(k+1, l+1)$$

$$R(k+1, l) = \alpha + \omega(k+1, l) + R^*(k+2, l+1)$$

$$R(k, l+1) = \beta + \omega(k, l+1) + R^*(k+1, l+2)$$

である。 α および β は、パターンの時間的歪み補正に対するコストを表すパラメータで、定数として扱う。この漸化式の第 n 回目に次の式を代入したものを第 n 次近似とする。(この場合近似度 n という。)

$$R^*(k, l) = \omega(k, l)$$

$$R^*(k+1, l) = \alpha + \omega(k+1, l)$$

$$R^*(k, l+1) = \beta + \omega(k, l+1)$$

3. 拡張SODPアルゴリズム

以上のようにSODPアルゴリズムはある点を起点として n 段からなる局所的最適パスを決定した後、次の局所的最適パスを求めるための起点をそのパス上で1段だけ進めるものであった。これに対し拡張SODPアルゴリズムでは、得られた局所的最適パスの類似度(実際にはベクトル間距離を用いたのでパス距離と呼ぶ)の評価値に応じて次の計算の起点を直前の最適パス上で1段から n 段までの範囲で進める(この段数を前進ステップと呼ぶ。)ことにする。従って、この方式は適応型と呼ぶことができる。

一般的に、直前の最適パスのパス距離が非常に小さい場合、その付近では引続きパス距離が小さいことが期待できる。従って、このときの次の局所的最適パス決定のための前進ステップは多少大きく取っても構わないと思われる。このようにできれば、全体として明らかに計算量を減らすことが可能である。

逆に、直前の最適パスのパス距離が大きな場合は、なるべく小さな前進ステップを用いたほうが無難である。

実験では、あらかじめパス距離と前進ステップの関係を各種定めておき、それらと比較した。

4. SODPアルゴリズムとの比較実験

SODP方式、拡張SODP方式について離散単語認識の比較実験を行った。

語彙は日本の都市名100個とし、ベクトル間距離にはチェビシェフ距離を用いた。また、近似度は3, 4の2つについて行った。比較はアルゴリズムの処理時間に関するものであり、SODP、拡張SODPのいずれについてもマッチングのみに要した時間を計測した。

その結果、SODP方式に比べ $n=3$ の場合はおおむね30%、 $n=4$ の場合はパス距離とステップの関係式により2

8%~47%という短縮率となった。なお、いずれの関係式の場合においても $n = 4$ の場合の短縮率の方が $n = 3$ のそれよりも大きい。

認識率に与える影響としては、 $n = 3$ の場合若干の認識率低下がみられたが、 $n = 4$ の場合はほとんど影響が無かった。今後は、バス距離とステップの関係式についてより深く研究して行く必要があると思われる。

5. 座標変換

SODPや拡張SODPアルゴリズムをインプリメントする際、局所的最適パスを決定するためには図1に示すような一定範囲の格子点に関するベクトル間距離を求めなければならぬ。このとき、求めるべき格子点を算出するために要する計算時間は無視できないものである。そこで格子点算出の計算量を軽減するために図2のようにベクトル座標系を左に45度回転させることとした。これにより格子点の周期性がより明らかになり、格子点算出アルゴリズムを簡単にできる。このような座標系の回転による計算量軽減の効果を調べるために、SODPを用いた格子点算出時間の比較を行った。この実験の語彙は日本の地名50語を用い、SODPの近似度は3および、4とした。

この結果、座標系回転前に比べ近似度3の場合で約30%、近似度4の場合約45%の時間短縮がみられた。格子点算出の総回数は語彙数の2乗に比例することになるので語彙が大きくなるほど差は大きくなることが予測される。

参考文献

- (1)安木他：「SODPアルゴリズムを用いた連続音声認識システム」，昭和58年度電子通信学会情報・システム部門全国大会
- (2)松永他：「準最適DPマッチング法(SODPアルゴリズム)による単語音声認識について」，情報処理学会第28回(昭和59年前期)全国大会
- (3)H.Sakoe and S.Chiba, "Dynamic programming algorithm optimization for spoken word recognition", IEEE Trans. ASSP-26, pp.43-49, FEB.1978.

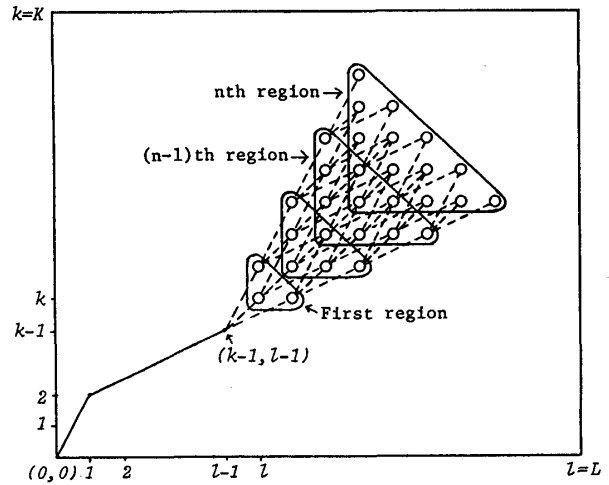


図1. SODPアルゴリズム

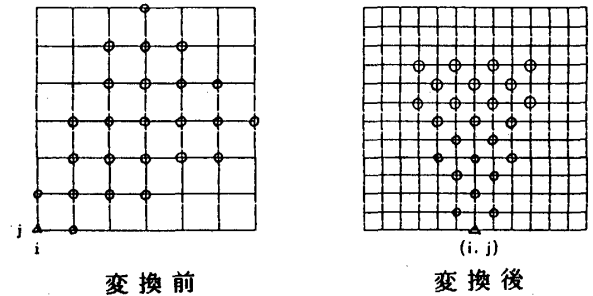


図2. 座標変換(○印は計算すべき格子点, △印は計算の基準点)