

隠れマルコフモデルデータベースの高速尤度検索

藤原 靖宏[†] 櫻井 保志[†] 山室 雅司[†]

[†] 日本電信電話株式会社 NTT サイバースペース研究所

〒 239-0847 神奈川県横須賀市光の丘 1-1

E-mail: †{fujiwara.yasuhiro,sakurai.yasushi,yamamuro.masashi}@lab.ntt.co.jp

あらまし 時系列データは金融, 環境, モバイル, マルチメディア等の様々な分野で利用される汎用的なデータモデルである. 多くのアプリケーションにおいては時系列データを HMM(Hidden Markov Model) として格納しておいて時系列信号を問い合わせとして検索されることが行われる. 本論文では複数の HMM が格納されたデータベースに時系列信号が与えられたときに, 尤度の高い HMM を高速かつ漏れなく検索する問題を対象とする. 本論文では HMM データベースの尤度検索に対して (1) HMM を縮退して近似し, 尤度の上限値により高速に解候補を絞り込む手法と (2) 検索途中の HMM の尤度計算を打ち止めて, 尤度の低い HMM を高速に枝刈りする手法を提案する. 本論文では提案手法の有効性を示すために実験を行った. その結果提案手法は naïve な手法と比較して数十倍まで高速に検索できることを確認した.

キーワード 隠れマルコフモデル, 尤度検索, 動的計画法, Viterbi アルゴリズム, 上限値

Fast Likelihood Search for Hidden Markov Model Databases

Yasuhiro FUJIWARA[†], Yasushi SAKURAI[†], and Masashi YAMAMURO[†]

[†] NTT Cyber Space Laboratories, NTT Corporation

1-1 Hikarinooka, Yokosuka-Shi, Kanagawa, 239-0847, Japan

E-mail: †{fujiwara.yasuhiro,sakurai.yasushi,yamamuro.masashi}@lab.ntt.co.jp

Abstract Time series data is a general data model utilized in several domains such as finance, environment, mobile and multimedia. Time series data are stored as HMM(Hidden Markov Model)s and HMMs are retrieved by a time signals query in many applications. In this paper, we focus on the problem to retrieve high likelihood HMMs from a database fast by given time signals with no false negatives. We propose two technics for likelihood search of HMM databases to: (1) degenerate and approximate an HMM to compute candidates by upper bound of likelihood, (2) break a processing likelihood computation to pruning out a low-likelihood HMM fast. Experiments are conducted to evaluate the usefulness of our technics. And we confirmed our technics enable up to dozens of times faster than naive method.

Key words Hidden markov model, likelihood search, dynamic programming, viterbi algorithm, upper bound

1. はじめに

時系列データは金融, 環境, モバイル, マルチメディア等の様々な分野で利用される汎用的なデータモデルである. そのためデータベースの分野では時系列データを問い合わせとして, 時系列データそのものが格納されているデータベースに対して問い合わせを行う研究が多く行われてきた. 時系列データは時間が経過するごとに発生するため, 時系列データの量は膨大になる. よって一般的に時系列データ同士の比較には多くの計算コストがかかるため, 従来より研究の主眼は大量な時系列データに関する高速検索に置かれてきた.

時系列データの検索は類似度によって行われるが, 多くの研

究において類似度としてユークリッド距離か DTW (Dynamic Time Warping) が用いられている. DTW は時系列データ間の距離を最小にするように時間軸方向に伸長させる時系列データの変換処理であり, ユークリッド距離よりノイズに強い性質がある. DTW の類似度は動的計画法により計算される.

しかし実際のアプリケーションにおいては時系列データそのものが格納されていることは少なく, HMM(Hidden Markov Model) として格納しておいて時系列信号を問い合わせとして検索されることが多く行われる. それは HMM が時系列データを確率密度関数に従う発生モデルを有する状態の遷移として表現するデータモデルであるため, 時間軸方向のみでなく時系列の成分方向のノイズにも強いからである. 多くのアプリケーションにおいて HMM が格納されている場合, 検索は問い合

わせの時系列信号が格納された HMM によって推定される尤度によって行われる。

本論文では複数の HMM が格納されたデータベースに時系列信号が与えられたときに、尤度の高い HMM を高速かつ漏れなく検索する問題を対象とする。尤度は動的計画法によって計算するが、一般的に HMM の状態数は多くまた時系列信号は長いので、尤度を計算する計算コストは高くなる。そのためナイーブに格納されたすべての HMM の尤度を計算して解く手法の計算コストは莫大になる。

本研究における HMM の高速尤度検索は以下に示すとおり様々な分野に対して応用が可能である。

- 動画コンテンツのシーンを高速に識別 [1], [2], [3], [4]
- 音楽コンテンツをフレーズを高速に切り分け [5], [6], [7]
- 様々な話者が発生した単語を高速に認識 [8], [9], [10]

本論文では HMM データベースの尤度検索に対して、(1) HMM を縮退して近似し、尤度の上限值により高速に解候補を絞り込む手法と (2) 検索途中の HMM の尤度計算を打ち止めて、尤度の低い HMM を高速に枝刈りする手法を提案する。本論文で提案する手法は以下のような特長がある。

- 高速処理：1000 個の HMM が格納されていても 1 秒以内に検索が可能
- 検索漏れなし：提案手法により検索された HMM は厳密に正確であり、提案手法を用いずに検索した集合と同じになる
- HMM の構造に制限なし：HMM には複数の構造があるが、どのような構造であっても検索が可能

提案手法では近似後の尤度を用いて検索の解候補を絞り込んでから、厳密な尤度を用いてさらに枝刈りを行う。提案手法では格納された HMM の中から必要最低限のものに対してのみ尤度を計算するので、高速処理が可能となる。

本論文では提案手法の有効性を示すために実験を行った。その結果提案手法はナイーブな手法と比較して数十倍まで高速に検索できることを確認した。

本論文の構成は以下の通りである。2. 章で関連研究をあげる。3. 章で本論文における問題設定をした後に、HMM を高速に尤度検索するための提案手法について説明する。4. で提案手法の実験結果を示し、5. 章で本研究をまとめる。

2. 関連研究

本研究は時系列信号を確率密度関数に従う発生モデルを有する状態の遷移として HMM で表現して格納しておき、尤度により高速に検索する問題を扱う。

しかし時系列データそのものを格納しておき高速に検索する研究は過去から多く行われている。Agrawal らは格納された時系列データに対してそれと長さが等しい時系列データを問い合わせる研究を行った [11]。Agrawal らの手法は時系列データを DFT (Discrete Fourier Transform) を用いて近似し、近似後の時系列データを R*-tree [12] に格納することによって高速に検索可能とするものである。Agrawal らの研究は Faloutsos らや Moon らによって格納された時系列データと長さが異なる検索へと拡張された [13], [14]。

Agrawal らの研究は類似度としてユークリッド距離を用いていたが、ノイズやアウト라이어等に強い検索を行うために、類似度として DTW を用いることが提案されている [15]。DTW の高速検索は Yi らによって初めて研究された [16]。彼らの手

Symbols	Definitions
o_i	信号列 O の時刻 i における値 ($i = 1, \dots, T$)
K	検索される HMM の個数
$A = \{a_i\}$	状態 u_i の初期状態確率 ($i = 1, \dots, N$)
$B = \{b_{ij}\}$	状態 u_i から状態 u_j への遷移確率
$C = \{c_{ij}\}$	状態 u_i における信号 v_j の信号出力確率 ($j = 1, \dots, Q$)
$P(X, O)$	時系列信号 O の HMM X に対する尤度
$\hat{P}(\hat{X}, \hat{O})$	縮退後の時系列信号 \hat{O} の HMM \hat{X} に対する尤度

表 1 主な記号とその定義

法は時系列データを FastMap [17] を用いてインデクシングし、時系列データの最大値や最小値を用いた距離関数により類似候補を絞りこむ。FastMap による絞込みは検索漏れが発生しうるため、最終的な検索結果には検索漏れがありえた。Keogh らや Zhu らによって DTW での時間軸方向の伸長での制限がある場合における検索漏れのない検索が研究された [18], [19]。彼らの手法では時系列データの時間軸方向の伸長の範囲を PAA [20] を用いて近似し、インデクスを付与する。時間軸方向の伸長での制限を用いない検索は櫻井らによって研究された [21]。彼らは時系列データをセグメントに分けたときの最大値や最小値を用いて、検索漏れがないことが保証される距離関数を新たに提案した。

また最近には特にセンサなどから取得されるノイズなどによる不確実な時系列データを対象に、時系列データがある確率密度関数に従って分布しているとして、確率による検索を行う研究も行われている。Cheng らは問い合わせを、検索結果が時系列データの集合であるかないかで、また集約演算を含むか含まないかで 4 つに分類した [22]。またそれぞれの検索に対する処理手法も提案した。さらに Cheng らは確率密度関数の範囲検索について研究を行った [23]。彼らは R-tree [24] によって検索を行うときに閾値を大きくしていく手法と、平均や標準偏差などの統計量によってデータポイントをクラスタリングする手法を用いた。Tao らは各密度関数の形に制限がない場合を対象としたときの手法として U-tree を提案した [25]。U-tree は R*-tree のように MBR (Minimum Bounding Rectangle) と検索の閾値により検索結果の候補を高速に枝刈りする手法である。

3. 提案手法

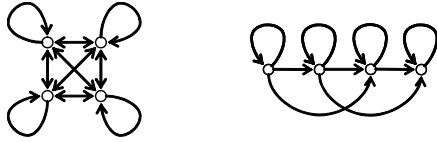
本論文では複数の HMM が格納されたデータベースに時系列データが与えられたときに、尤度の高い HMM を高速かつ漏れなく検索する問題を対象とする。実際のアプリケーションにおける問い合わせとしては k-近傍検索と範囲検索が重要であるが、本論文では特に k-近傍検索を議論の対象とする。しかし後に説明するように提案手法は範囲検索に対しても利用することができる。

3.1 前準備

HMM は時系列データを確率密度関数に従う発生モデルを有する状態の遷移として表現するデータモデルである [26], [27]。表 1 に本論文で用いる主な記号とその定義を示す。本論文では以下の問題を解く。

[問題 1] データベースに格納された HMM と、問い合わせの時系列信号 $O = (o_1, o_2, \dots, o_T)$ と、整数 K が与えられたとき、時系列信号 O に対して尤度の高い順に K 個の HMM を検索する。

HMM は以下の要素で構成される。



(a) エルゴディック HMM (b) left-right HMM

図 1 HMM の種類

初期状態確率: $A = \{a_i\}$ 時刻 1 において状態 $u_i (i = 1, \dots, N)$ に存在する確率.

状態遷移確率: $B = \{b_{ij}\}$ 時刻が 1 つ進んだときに状態 u_i から状態 u_j への遷移する確率.

信号出力確率: $C = \{c_{ij}\}$ 状態 u_i において信号 $v_j (j = 1, \dots, Q)$ を出力する確率.

HMM を説明するためにツボとボールの例を用いる. 赤や青などの色のついたボールが入っているツボが複数あることを考える. はじめにある確率によってツボを選ぶ. そしてボールをひとつ取り出し, その色を記録する. 次のツボは現在のツボの選択によって関連付けされた確率に従って選択する. そしてボールの色の記録を繰り返す. これらの操作によってボールの色の系列が得られる.

この例においてツボは状態に対応し, 色の系列は時系列信号に対応する. すなわち初期状態確率は初めに選択するツボの確率であり, 状態遷移確率は次のツボの選択確率であり, 信号出力確率はどの色のボールを選択するかの確率である.

HMM は図 1 のように状態遷移確率 B の構造によって分類することができる. 図 1 において白い円は状態を表し, 矢印は遷移を表す. すべての状態から他のすべての状態へ遷移できるタイプとして全結合 HMM とエルゴディック HMM がある. エルゴディック HMM は全結合 HMM を包含し, すべての状態が他のすべての状態へ有限であるが非周期的な回数で遷移できる特徴がある. 1 回で遷移できるエルゴディック HMM や全結合 HMM において状態遷移確率 B は例えば正の係数 b_{ij} を用いて以下ようになる.

$$B = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix} \quad (1)$$

また他のタイプとして left-right HMM がある. left-right HMM は状態を並べたときに左から右へ遷移する特徴がある. 状態遷移確率の係数は $j < i$ のとき $b_{ij} = 0$ となり, 初期状態確率の係数は $i = 1$ で $a_i = 1$ となり $i \neq 1$ で $a_i = 0$ となる. また left-right HMM においては状態間を多く遷移しないように制約が課せられてる事があり, 状態遷移確率の係数は $j > i + \Delta - 1$ のとき $b_{ij} = 0$ となる. 例えば状態遷移確率 B は $\Delta = 3$ のとき b_{ij} を用いて以下ようになる.

$$B = \begin{bmatrix} b_{11} & b_{12} & b_{13} & 0 \\ 0 & b_{22} & b_{23} & b_{24} \\ 0 & 0 & b_{33} & b_{34} \\ 0 & 0 & 0 & b_{44} \end{bmatrix} \quad (2)$$

時系列信号 O の HMM X に対する尤度 $P(X, O)$ は Viterbi アルゴリズムにより計算される. 先のツボとボールの例では尤度の計算は, ボールの系列が与えられたときにそれを出力する

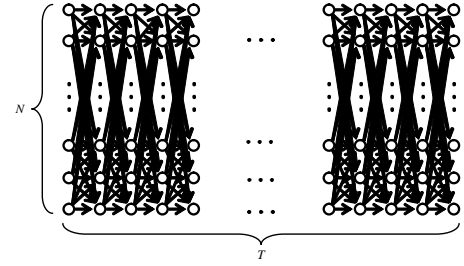


図 2 トレリス構造

ツボの遷移の最大確率を計算することに対応する. 尤度は以下のように求める.

$$P(X, O) = \max_{1 \leq i \leq N} (p_{iT}) \quad (3)$$

$$p_{ij} = \max_{1 \leq k \leq N} (p_{k(j-1)} \cdot b_{ki}) \cdot c_{ioj}, (1 \leq i \leq N; 1 \leq j \leq T)$$

ただし $j = 1$ のとき

$$\max_{1 \leq k \leq N} (p_{k(j-1)} \cdot b_{ki}) = a_i$$

Viterbi アルゴリズムでは図 2 のように状態を縦軸に, 時間を横軸に並べたときに構成される $N \cdot T$ の大きさのトレリス構造において, 各状態における確率の最大値を動的計画法によって求めている.

[例 1] HMM と時系列信号が以下のものであるとすると,

$$A = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, B = \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0.5 & 0.25 & 0.25 \\ 0 & 0 & 1 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.75 & 0.25 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 \end{bmatrix}$$

$$O = (1, 1, 2, 4)$$

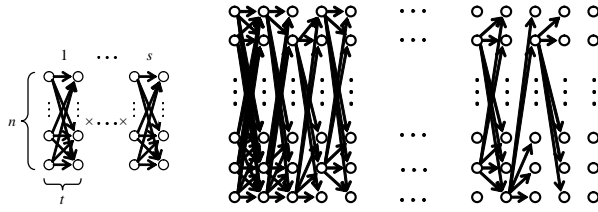
Viterbi アルゴリズムは以下のように計算される

$$\begin{aligned} p_{11} &= 1 & p_{12} &= 0.5 & p_{13} &= 0 & p_{14} &= 0 \\ p_{21} &= 0 & p_{22} &= 0.75 \cdot 0.5 & p_{23} &= (0.5)^2 \cdot 0.25 & p_{24} &= 0 \\ p_{31} &= 0 & p_{32} &= 0 & p_{33} &= 0 & p_{34} &= (0.5)^3 \cdot (0.25)^2 \end{aligned}$$

結果的に尤度は $P(X, O) = (0.5)^3 \cdot (0.25)^2$ となり, その尤度を与える状態遷移は u_1, u_1, u_2, u_3 となる.

HMM の構造に制約がない場合, すなわちエルゴディック HMM の場合の Viterbi アルゴリズムの計算量は $O(N^2T)$ となる. これはすべての状態から他のすべての状態へ遷移する確率を各時刻において計算するからである. HMM における状態の遷移はトレリス構造をバックトラックすることによって得られる. なお実際に Viterbi アルゴリズムを計算機で計算するときはアンダーフローを防ぐために各要素を対数化する. また HMM の各要素は Baum-Welch アルゴリズムによって事前に求める.

問題 1 の技術的な課題として HMM データベースを検索する計算コストが高いことが挙げられる. すなわち問題 1 をナイーブに解くには, すべての HMM の尤度を Viterbi アルゴ



(a) Trellis Degeneration (b) Quick Break

図3 提案手法

リズムにより計算しなければならない。しかし一般的に HMM における状態数 N は大きくまた時系列信号の長さ T は大きいため、莫大な計算コストが必要となってしまう。

3.2 手法概要

提案手法は以下の2つの手法によって構成される。まず2つの提案手法の概要を示し、その後にそれぞれの手法について詳しく述べる。

Trellis Degeneration 3.1で述べたとおり Viterbi アルゴリズムの計算コストはトレリス構造の大きさのために膨大になる問題がある。そのため Trellis Degeneration は図3に示すとおりトレリス構造を縮退して高速に尤度を近似計算する。Trellis Degeneration によりトレリス構造は s 個の $n \cdot t$ の大きさのトレリス構造に縮退される。縮退後のトレリス構造の尤度の計算は $O(n^2 st + nT)$ となる。

Trellis Degeneration には3つのメリットがある。まずはじめのメリットは Trellis Degeneration によって検索を行っても検索漏れは発生しないことが保証されることである。これは後に示すとおり縮退後の近似尤度はオリジナルのトレリス構造による尤度より小さくなることはないためである。そのため近似尤度を計算するのみで、検索結果の候補を少ない計算コストで得ることができる。

2番目とメリットとして Trellis Degeneration はどのようなタイプの HMM にも適用できることが挙げられる。これは Trellis Degeneration が 3.1 に示したような確率の制約を用いる手法ではないためである。

HMM はデータベースに与えられた時点で知ることができるが、時系列信号は検索が行われるまで知ることができない。そのため事前に時系列信号を縮退することは困難である。しかし提案手法はこのような状況においても高速に計算することができる。このことが3番目のメリットである。これは後に説明するように Trellis Degeneration では時系列信号を重ね合わせているが、この処理は高速に行えるからである。

Quick Break

提案手法は検索結果が正確であることを保証している。Trellis Degeneration により高速な検索が可能になるが、正確な検索結果は近似した尤度からは求めることができない。そのため提案手法では近似尤度に求めた候補に対して正確な尤度を求めている。しかし 3.1 に示したように正確に尤度を計算することには大きなコストがかかる。

Quick Break は図3のようにオリジナルのトレリス構造の尤度計算を高速に打ち切る手法である。Viterbi アルゴリズムの計算量の高さの原因のひとつにトレリス構造の状態遷移による計算の多さがある。しかしこの尤度計算の途中では検索処理としては計算する必要のない状態遷移による計算も含まれる。例えば例1において検索途中の K 番目に高い尤度が 0.25 である

ときに、時刻1における状態 u_2 と u_3 の尤度は 0.25 より小さいため、時刻2においてこれらの状態から遷移する確率は計算する必要はない。同じように時刻3において状態 u_3 からの確率は計算する必要はない。時刻3においてすべての状態の確率は 0.25 より小さいため、尤度計算を打ち切ることができる。

検索途中の K 番目に高い尤度が大きくなるほど、より効果的に計算を打ち切ることができる。ここで K 番目に高い尤度は検索処理が進むほど大きくなるので、結果的に Quick Break は HMM の数が多くても効果的に検索することができる。

また計算の途中においては尤度は単調減少するため、問い合わせの時系列信号が長くなるほど Quick Break はより効果的になる。これはユーザやアプリケーションによって問い合わせの長さが異なることを考えると、非常に有効な性質である。ただし後の実験に示すように特定のタイプの HMM に対しては必ずしも計算の打ち切りが時系列信号の後のほうで行われるわけではない。

3.3 Trellis Degeneration

Trellis Degeneration はトレリス構造を縮退して高速に尤度を計算する手法である。縮退後のトレリス構造で検索を行っても検索漏れは発生しないことが保証される。またトレリス構造の縮退は近似後の精度が高くなるように行う。

3.3.1 構造縮退

Trellis Degeneration では $N \cdot T$ のトレリス構造を s 個の大きさ $n \cdot t$ ののトレリス構造に縮退する。

N 個の状態 u は n 個の状態 \hat{u} へ統合する。同様に Q 個の出力信号 v は q 個の出力信号 \hat{v} へ統合する。 u と v はそれぞれひとつの状態と出力信号へのみに統合される。統合後の状態の確率は各要素の確率の中で最大値を用いる。すなわち統合後の初期状態確率は $\hat{a}_i = \max(a_{kl} | u_k \mapsto \hat{u}_i)$ となり、状態遷移確率は $\hat{b}_{ij} = \max(b_{kl} | u_k \mapsto \hat{u}_i, u_l \mapsto \hat{u}_j)$ となり、信号出力確率は $\hat{c}_{ij} = \max(c_{kl} | u_k \mapsto \hat{u}_i, v_l \mapsto \hat{v}_j)$ となる。ここで状態 u_k が状態 \hat{u}_i へ統合されていることを $u_k \mapsto \hat{u}_i$ と表し、出力信号 v_k が出力信号 \hat{v}_i へ統合されていることを $v_k \mapsto \hat{v}_i$ と表すこととする。

[例2] 例1の HMM において状態 u_1 と u_2 を統合した後の確率は以下ようになる。

$$A = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, B = \begin{bmatrix} 0.5 & 0.25 \\ 0 & 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0.25 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 \end{bmatrix}$$

T 個の時系列信号は等間隔に s 個に分割し、さらに t 個ごとに重ね合わせる。すなわち時系列信号は $\hat{o}_{ij} = \{o_{(i-1) \cdot \frac{T}{s} + j + (k-1) \cdot t}\}$, ($1 \leq i \leq s; 1 \leq j \leq t; 1 \leq k \leq T/st$) となる。

[例3] 例1の時系列信号において $s = 1, t = 2$ とすると、縮退後の信号は $\hat{O} = (\hat{o}_{11}, \hat{o}_{12}) = (\{1, 2\}, \{1, 4\})$ となる。

3.3.2 尤度計算

縮退後の尤度 $\hat{P}(\hat{X}, \hat{O})$ の計算には Viterbi アルゴリズムと同様に動的計画法を用いる。縮退後の HMM \hat{X} と信号列 \hat{O} が与えられとき $\hat{P}(\hat{X}, \hat{O})$ は以下のように計算する。

$$\hat{P}(\hat{X}, \hat{O}) = \prod_{k=1}^s \left\{ \max_{1 \leq i \leq n} (\hat{p}_{it}^k) \right\}^{T/st} \quad (4)$$

$$\hat{p}_{ij}^k = \max_{1 \leq l \leq n} (\hat{p}_{l(j-1)}^k \cdot \hat{b}_{li}) \cdot \max_{1 \leq l \leq T/st} (\hat{c}_{i\hat{o}_{kl}})$$

$$(1 \leq k \leq s; 1 \leq i \leq n; 1 \leq j \leq t)$$

ただし $k \neq 1$ かつ $j = 1$ のとき

$$\max_{1 \leq l \leq n} (\hat{p}_{l(j-1)}^k \cdot \hat{b}_{li}) = \max_{1 \leq l \leq n} (\hat{b}_{li})$$

また $k = 1$ かつ $j = 1$ のとき

$$\max_{1 \leq l \leq n} (\hat{p}_{l(j-1)}^k \cdot \hat{b}_{li}) = \max(\max_{1 \leq l \leq n} (\hat{b}_{li}), \hat{a}_i)$$

縮退後の尤度の計算では $n \cdot t$ の大きさのトレリス構造を計算することを s 回繰り返している．縮退後の尤度の計算量は $O(n^2 st + nT)$ となる．ここで縮退後の出力信号の個数 q は縮退後の尤度の計算量に影響しないため，出力信号を縮退しても尤度の計算量は減少しないように見える．しかし出力信号を縮退すると時系列信号の重なりが多くなり $\max_{1 \leq l \leq T/st} (\hat{c}_{i\hat{o}_{kl}})$ の計算コストが下がるため，結果的に尤度の計算量は減少する．またここで $\hat{P}(\hat{X}, \hat{O})$ は $P(X, O)$ の上限値となる．

[補助定理 1] 縮退後の HMM \hat{X} と信号列 \hat{O} が与えられとき， $\hat{P}(\hat{X}, \hat{O})$ は $P(X, O)$ より小さくなることはない．すなわち以下の関係が成立する．

$$P(X, O) \leq \hat{P}(\hat{X}, \hat{O}) \quad (5)$$

[証明 1] $u_g \mapsto \hat{u}_i$ である g に対して

$$p_{g1} = a_g \cdot c_{g\hat{o}_1} \leq \max(\max_{1 \leq l \leq n} (\hat{b}_{li}), \hat{a}_i) \cdot \max_{1 \leq l \leq T/st} (\hat{c}_{i\hat{o}_{1l}}) = \hat{p}_{i1}^1$$

となり， $2 \leq j \leq t$ のとき

$$\begin{aligned} p_{gj} &= \max_{1 \leq l \leq N} (p_{l(j-1)} \cdot b_{lg}) \cdot c_{g\hat{o}_j} \\ &\leq \max_{1 \leq l \leq n} (\hat{p}_{l(j-1)}^1 \cdot \hat{b}_{li}) \cdot \max_{1 \leq l \leq T/st} (\hat{c}_{i\hat{o}_{1l}}) = \hat{p}_{ij}^1 \end{aligned}$$

となるので $j = t$ で

$$p_{gt} \leq \hat{p}_{it}^1$$

となる．また $j = t + 1$ のとき

$$\begin{aligned} p_{gj} &\leq \max_{1 \leq l \leq n} (\hat{p}_{l(j-1)}^1) \cdot \max_{1 \leq l \leq n} (\hat{b}_{li}) \cdot \max_{1 \leq l \leq T/st} (\hat{c}_{i\hat{o}_{1l}}) \\ &\leq \max_{1 \leq l \leq n} (\hat{p}_{lt}^1) \cdot \max(\max_{1 \leq l \leq n} (\hat{b}_{li}), \hat{a}_i) \cdot \max_{1 \leq l \leq T/st} (\hat{c}_{i\hat{o}_{1l}}) \\ &= \max_{1 \leq l \leq n} (\hat{p}_{lt}^1) \cdot \hat{p}_{i1}^1 \end{aligned}$$

となるので同様に $j = T/s$ で

$$p_{g T/s} \leq \hat{p}_{it}^1 \cdot \left\{ \max_{1 \leq l \leq n} (\hat{p}_{lt}^1) \right\}^{T/st-1}$$

となる．また同様 $j = T/s + 1$ のとき

$$\begin{aligned} p_{gj} &\leq \max_{1 \leq l \leq n} (\hat{p}_{l(j-1)}^2) \cdot \max_{1 \leq l \leq n} (\hat{b}_{li}) \cdot \max_{1 \leq l \leq T/st} (\hat{c}_{i\hat{o}_{2l}}) \\ &= \left\{ \max_{1 \leq l \leq n} (\hat{p}_{lt}^1) \right\}^T / st \cdot \hat{p}_{i1}^2 \end{aligned}$$

となるので $j = T$ で

$$p_{gT} \leq \hat{p}_{it}^s \cdot \left\{ \max_{1 \leq l \leq n} (\hat{p}_{lt}^s) \right\}^{T/s-1} \cdot \prod_{k=1}^{s-1} \left\{ \max_{1 \leq l \leq n} (\hat{p}_{lt}^k) \right\}^{T/st}$$

となる．よって

$$P(X, O) = \max_{1 \leq g \leq N} (p_{gT}) \leq \prod_{k=1}^s \left\{ \max_{1 \leq l \leq n} (\hat{p}_{lt}^k) \right\}^{T/st} = \hat{P}(\hat{X}, \hat{O})$$

が成り立つ． \square

定理 1 の性質を用いて検索漏れなく HMM を検索することができる．

[補助定理 2] $\hat{P}(\hat{X}, \hat{O})$ を用いて検索することは，検索漏れが発生しないことの十分条件である．

[証明 2] k -近傍検索の K 番目の解の尤度を ϵ としたとき，検索漏れが発生しないことを保証するためには， $P(X, O) \geq \epsilon$ であるとき $\hat{P}(\hat{X}, \hat{O}) \geq \epsilon$ でなければならない．ところでここで式 (5) より

$$\epsilon \leq P(X, O) \leq \hat{P}(\hat{X}, \hat{O})$$

となるので検索漏れは発生しない． \square

3.3.3 統合処理

3.3.1 で述べたとおりトレリス構造はオリジナルの HMM の状態と出力信号を統合することにより縮退する．状態と出力信号を統合するときの組み合わせは複数通り考えられるが，統合の組み合わせによっては縮退後の尤度の近似精度は大きく変化する．検索の高速性は近似精度に依存するため，統合は縮退後の尤度の近似精度が高くなるようにしなければならない．トレリス構造の縮退においては実験より得た次の観測結果を用いる．

[観測結果 1] 統合前と統合後の確率の差が小さくなるほど近似精度が上がる．

例えば例 1 の場合において，確率が似ている状態 u_1 と u_2 (b_{11} と b_{21} はともに 0.5) を統合すると，縮退前の時系列信号に対する近似尤度は $0.5^3 \cdot 0.25^2$ となり，オリジナルの尤度と同じになる．これに対して状態 u_1 と u_3 を統合すると近似尤度は $0.5^3 \cdot 0.25$ となり，オリジナルの尤度より大きくなる．

そのため状態と出力信号の統合は，誤差を計算し統合による誤差が小さくなるようにしながら行う．状態 u_i と u_j , ($1 \leq i, j \leq k$) を統合するときの誤差 $E(u_i, u_j)$ と，出力信号 v_i と v_j , ($1 \leq i, j \leq l$) を統合するときの誤差 $E(v_i, v_j)$ を確率の差を用いて以下のように定義する．

$$\begin{aligned} E^2(u_i, u_j) &= (a_i - a_j)^2 + \sum_{\substack{g=1 \\ g \neq i, j}}^k \{(b_{ig} - b_{jg})^2 + (b_{gi} - b_{gj})^2\} \\ &\quad + \sum_{g=i, j} \{(\hat{b} - b_{gi})^2 + (\hat{b} - b_{gj})^2\} + \sum_{g=1}^l (c_{ig} - c_{jg})^2 \end{aligned}$$

ここで $\hat{b} = \max_{g=i, j} (b_{gi}, b_{gj})$ (6)

$$E^2(v_i, v_j) = \sum_{g=1}^k (c_{gi} - c_{gj})^2 \quad (7)$$

Trellis Degeneration ではまず状態を統合してから次に出力信号を統合する．状態を統合するときは，その度すべての状態の組み合わせに対して誤差を計算する．そして誤差の最も小さくなる組み合わせを求め，その組み合わせに対して統合を行う．統合は状態の数が n になるまで繰り返していく．出力信号に対しても同様に誤差を計算しながら，出力信号の種類の数 q になるまで統合していく．

3.4 Quick Break

Quick Break は検索処理途中で求めた暫定的な K 個の解候補の尤度により尤度計算を高速に打ち切る手法である。

検索処理途中で得られる暫定的な K 番目の解候補のオリジナルの尤度を ϵ_{temp} とすると、ある HMM X に対して縮退後の尤度が $\hat{P}(\hat{X}, \hat{O}) \geq \epsilon_{temp}$ であっても $P(X, O) < \epsilon_{temp}$ であれば HMM X は解候補になりえない。また Viterbi アルゴリズムは動的計画法により計算されるので、尤度計算の途中においては尤度は単調減少する。そのため Quick Break では尤度を計算するときに ϵ_{temp} より小さい尤度となる状態からの遷移による計算を打ち切る。計算を途中で打ち切っても明らかに検索漏れは発生しない。

Quick Break ではまず HMM のタイプによって遷移集合の初期設定を行う。遷移集合とは尤度を計算するときに用いる集合である。遷移集合はトレリス構造のそれぞれの時刻と状態ごとに設定され、時刻 i の状態 j において $b_{kj} > 0$ であれば u_k は状態 j の遷移集合に加えられる。尤度を計算するときは遷移集合に含まれる状態からの遷移のみを用いる。もしある状態 u_k における尤度 p_k が ϵ_{temp} より小さければ、その状態からの遷移は解候補となりえないため、すべての状態における遷移集合から状態 u_k を削除する。またすべての状態に対して遷移集合が空集合である場合は、その HMM は解候補となりえないため、尤度計算を打ち切る。

3.5 検索処理

検索処理では高速に解を求めるために近似尤度を用いて尤度の低いほとんどの HMM の枝刈りを行い、計算コストのかかるオリジナルの尤度計算は必要最低限に限定する。

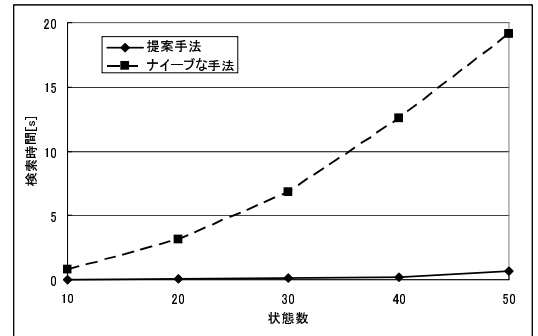
検索処理ではまずすべての HMM に対して近似尤度を計算し、近似尤度の降順によって並び替えを行う。そして K 番目までの HMM すべてに対してオリジナルの尤度を計算し、最も低い尤度を ϵ_{temp} とする。 ϵ_{temp} は検索における枝刈りに用いられるが、これは k -近傍検索の K 番目の解の尤度を ϵ としたときに、 ϵ により近い ϵ_{temp} をできるだけ少ないオリジナルの尤度計算の回数で求めるためである。近似尤度が ϵ_{temp} が大きくなる HMM は解候補となり得るため、オリジナルの尤度を計算する。もしオリジナルの尤度が ϵ_{temp} より大きい場合は、新たに ϵ_{temp} を計算しなおす。 ϵ_{temp} は検索処理の途中において更新され続けられるので、枝刈りはさらに効果的になっていく。

ここでは k -近傍検索の処理についてのみ述べたが、範囲検索も同様に処理することができる。範囲検索の場合は検索の閾値を ϵ_{temp} に置き換えて処理を行う。そのため近似尤度の並び替えをして ϵ_{temp} を求めることも、 ϵ_{temp} を更新することも行わない。

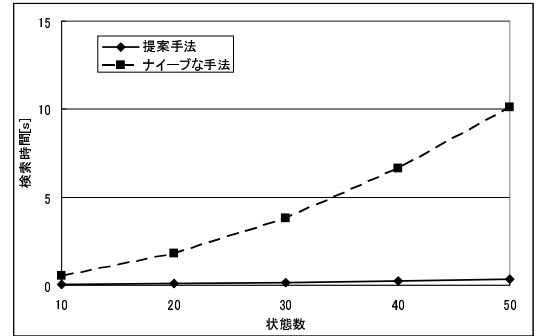
4. 評価実験

提案手法の有効性を示すために実験を行った。ナイーブに検索を行う手法と比較実験を行った。実験ではナイーブに検索を行う手法と比較を行った。また実験はエルゴディック HMM と left-right HMM に対して行った。検索処理にかかる時間において CPU の計算時間はディスクアクセスにかかる時間よりもはるかに大きいため、実験では検索における CPU の計算時間を比較した。

実験に用いた HMM はその各要素を動画コンテンツの特徴量から Baum-Welch アルゴリズムにより求めた。動画コン



(a) エルゴディック HMM



(b) left-right HMM

図 4 状態数を変化させた場合の検索時間

テンツはドラマ、風景画、CG 映画など様々なものを 24 個用意し、1000 個のシーンを切り出した。動画は 1 秒あたり 30 フレームで構成され、特徴量を各フレームごとに計算した。特徴量にはフレームの輝度ヒストグラムの特定成分を抽出したものをを用いた。

実験におけるパラメータは問い合わせの個数 $K = 3$ 、時系列信号の種類の数 $Q = 100$ とし、HMM の状態数 N と時系列信号の長さ T は実験によって変化させ、縮退後の状態数 $n = \lceil \log(N) \rceil$ 、時系列信号の長さ $s = \lceil \log(T) \rceil$ 、 $t = 2$ 、時系列信号の種類の数 $q = \lceil \log(Q) \rceil$ とした。また left-right HMM における遷移の制約は状態数 N によって変化させ $\Delta = 0.2 \cdot N$ とした。

実験は CPU が Pentium4 の 3.2GHz、メインメモリが 1GB のマシンで行った。

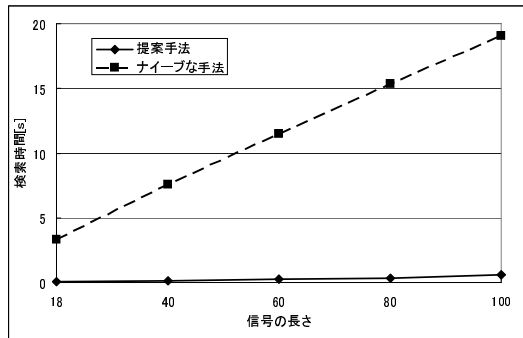
4.1 検索処理時間

検索処理時間について状態数を変化させた場合と、時系列信号の長さを変化させた場合について実験を行った。

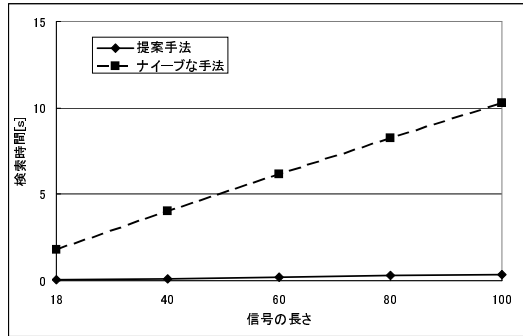
4.1.1 状態数を変化させた場合

状態数を変化させて実験を行った。なお実験において信号の長さは $T = 100$ とした。実験結果を図 4 に示す。

結果を見ると提案手法はナイーブな手法と比較してエルゴディック HMM に対して約 33~73 倍、left-right HMM に対して本実験の条件下で約 16~29 倍高速な検索が行えている。エルゴディック HMM において Viterbi アルゴリズムの計算量は $O(N^2T)$ なので、ナイーブな手法は状態数 N に対して計算量は $O(N^2)$ で増加する。しかし提案手法において計算量は Trellis Degeneration と Quick Break によって大幅に抑えられている。すなわち縮退後の尤度の計算量は $O(n^2st + nT)$ であり $n = \lceil \log(N) \rceil$ であるので、状態数に対して計算量の増加は $O((\log N)^2)$ に抑えられる。オリジナルの尤度計算を行う場合でも計算量は Quick Break によって抑えられる。また left-right



(a) エルゴディック HMM



(b) left-right HMM

図 5 時系列信号の長さを変化させた場合の検索時間

HMM においても, Trellis Degeneration と Quick Break は HMM のタイプに依存する手法ではないので, 同様に高速化が達成されている. なお後の実験に示すとおり, left-right HMM においてはエルゴディック HMM より Quick Break による高速化の影響が大きい.

4.1.2 時系列信号の長さを変化させた場合

時系列信号の長さを変化させて実験を行った. なお実験において状態数は $N = 50$ とした. 実験結果を図 5 に示す.

結果を見ると提案手法はナイーブな手法と比較してエルゴディック HMM に対して約 33~49 倍, left-right HMM に対して本実験の条件下で約 28~43 倍高速な検索が行えている. 状態数を変化させた場合と同様に, エルゴディック HMM においてナイーブな手法は信号の長さに対して計算量が $O(T)$ で増加する. しかし提案手法において縮退後の計算量の増加は $O(\log T)$ に抑えられ, オリジナルの計算量も Quick Break によって抑えられる. また提案手法は HMM のタイプに依存しないので, left-right HMM においても同様に高速化が達成されている.

4.2 Trellis Degeneration によるオリジナルの尤度計算の削減量

提案手法ではまず Trellis Degeneration で縮退した後の尤度を計算してから, 解候補となる HMM にのみに対してオリジナルの尤度を計算する. そのため縮退後の尤度で絞り込まれる解候補個数が検索の高速化に影響する. そこでこの実験では 1000 個の格納された HMM に対してオリジナルの尤度を計算したものの個数を調べた. 実験では時系列信号の長さを変化させ, 状態数は $N = 50$ とした. 実験結果を図 6 に示す.

結果を見るとエルゴディック HMM に対してはほとんどオリジナルの尤度を計算することなく検索が行えていることがわかる. これはエルゴディック HMM に対しては Trellis Degeneration による近似尤度の精度が高いからである. しかし left-right

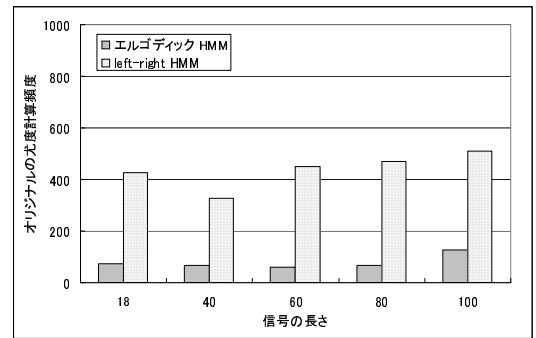
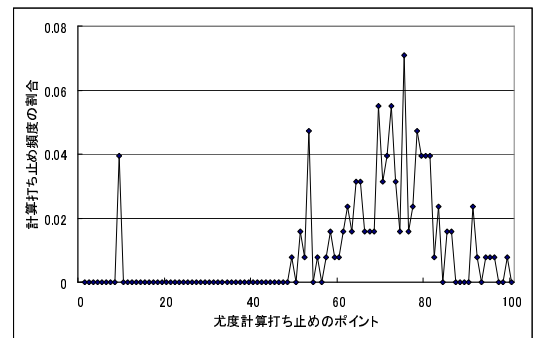
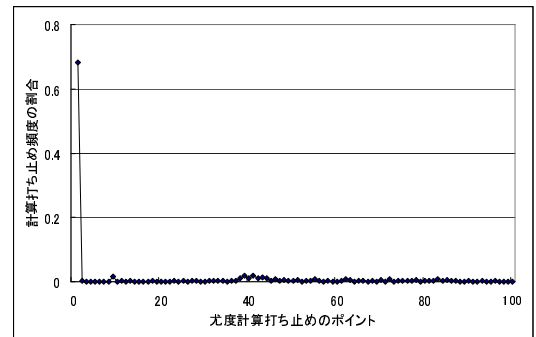


図 6 Trellis Degeneration による計算削減量



(a) エルゴディック HMM



(b) left-right HMM

図 7 Quick Break による計算削減量

HMM の結果を見るとそれほどオリジナルの尤度計算を省けていないことがわかる. これは式 (2) 示すように left-right HMM においては状態遷移確率が状態ごとに大きく異なり, 式 (6) に示すような Trellis Degeneration で状態を統合するときの誤差が大きくなるため, Trellis Degeneration による近似尤度の精度が低くなるからである. しかし図 4, 図 5 にあるとおり, 提案手法は left-right HMM に対しても高速化を達成している. これは次の実験に示すとおり left-right HMM においては Quick Break によって大幅に計算量の削減ができていたためである.

4.3 Quick Break によるオリジナルの尤度計算の削減量

提案手法では解候補となる HMM にのみに対してオリジナルの尤度を計算するが, その計算も Quick Break により打ち止めされる. オリジナルの尤度を計算するときのトレリス構造の横軸方向において, 早期のポイントで計算が打ち止められるほど高速に検索ができる. そこでこの実験では計算が打ち止められた HMM に対して, 計算が打ち切られた横軸方向のポイントの頻度の割合を調べた. なおオリジナルの尤度を計算することとなった HMM のうち, エルゴディック HMM では 127 個のなかで 122 個において, left-right HMM では 509 個のな

かで 494 個においてそれぞれ計算の打ち止めが行われた。実験では状態数は $N = 50$ ，時系列信号の長さ $T = 100$ とした。

エルゴディック HMM の結果を見ると，トレリス構造の横軸方向のポイントが増加するほど計算が打ち切られる割合が多くなるのがわかる。これは Viterbi アルゴリズムは動的計画法により計算されるので，尤度計算の途中においては尤度は単調減少するため，横軸方向のポイントが増加するほど尤度が ϵ_{temp} より小さくなる状態が増加するためである。

また left-right HMM の結果を見ると，トレリス構造の横軸方向のはじめのポイントでほとんど計算が打ち切られていることがわかる。これは left-right HMM においては初期状態確率の係数は $i = 1$ で $a_i = 1$ となり $i \neq 1$ で $a_i = 0$ となるため，トレリス構造の横軸方向のはじめのポイントで状態 u_1 の尤度が ϵ_{temp} より小さくなると，計算は打ち止めされるからである。

5. ま と め

本論文では複数の HMM が格納されたデータベースに時系列信号が与えられたときに，尤度の高い HMM を高速かつ漏れなく検索する問題について取り組んだ。本論文ではこの HMM の検索を高速，検索漏れなし，HMM の構造に制限なしで行えることを示した。

本論文では HMM を縮退して近似し，尤度の上限値により高速に候補を絞り込む手法 Trellis Degeneration と，検索途中の HMM の尤度計算を打ち止めて，尤度の低い HMM を高速に枝刈りする手法 Quick Break を提案した。

本論文では提案手法に対して動画コンテンツによるデータを用いて検証し，ナイーブな手法と比較して数十倍まで高速に検索できることを確認した。

謝 辞

動画コンテンツをご提供頂いた NTT サイバースペース研究所の阿部剛仁氏と，動画像の特徴量について議論頂いた NTT サイバースペース研究所の安藤慎吾氏には厚く感謝申し上げます。

文 献

- [1] J. Huang, Z. Liu and Y. Wang: "Joint scene classification and segmentation based on hidden markov model.", IEEE Transactions on Multimedia, **7**, 3, pp. 538–550 (2005).
- [2] P. Chang, M. Han and Y. Gong: "Extract highlights from baseball game video with hidden markov models.", ICIP (1), pp. 609–612 (2002).
- [3] S.-F. Chang, A. Divakaran and H. Sun: "Structure analysis of soccer video with hidden markov models.", ICASSP, Vol. 4, pp. 4096–4099 (2002).
- [4] M. Petkovic, W. Jonker and Z. Zivkovic: "Recognizing strokes in tennis videos using hidden markov models.", VIIP, pp. 512–516 (2001).
- [5] C. Raphael: "Automatic segmentation of acoustic musical signals using hidden markov models.", IEEE Trans. Pattern Anal. Mach. Intell., **21**, 4, pp. 360–370 (1999).
- [6] C. G. Puntonet and A. Prieto Eds.: "Independent Component Analysis and Blind Signal Separation, Fifth International Conference, ICA 2004, Granada, Spain, September 22-24, 2004, Proceedings", Vol. 3195 of Lecture Notes in Computer Science Springer (2004).
- [7] H. Takeda, N. Saito, T. Otsuki, M. Nakai, H. Shimodaira and S. Sagayama: "Hidden markov model for automatic transcription of midi signals.", IEEE Workshop on Multimedia Signal Processing, pp. 428–431 (2002).
- [8] F. Jelinek: "A fast sequential decoding algorithm using a stack.", IBM J. Res. Develop., **13**, pp. 675–685 (1969).
- [9] L. R. Bahl and F. Jelinek: "Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition.", IEEE Trans. Informat. Theory, **IT-21**, pp. 404–411 (1975).
- [10] L. R. Bahl, F. Jelinek and R. L. Mercer: "A maximum likelihood approach to continuous speech recognition.", IEEE TPAMI, **PAMI-5**, pp. 179–190 (1983).
- [11] R. Agrawal, C. Faloutsos and A. N. Swami: "Efficient similarity search in sequence databases.", FODO, pp. 69–84 (1993).
- [12] N. Beckmann, H.-P. Kriegel, R. Schneider and B. Seeger: "The r*-tree: An efficient and robust access method for points and rectangles.", SIGMOD Conference, pp. 322–331 (1990).
- [13] C. Faloutsos, M. Ranganathan and Y. Manolopoulos: "Fast subsequence matching in time-series databases.", SIGMOD Conference, pp. 419–429 (1994).
- [14] Y.-S. Moon, K.-Y. Whang and W.-S. Han: "General match: a subsequence matching method in time-series databases based on generalized windows.", SIGMOD Conference, pp. 382–393 (2002).
- [15] D. J. Berndt and J. Clifford: "Using dynamic time warping to find patterns in time series.", KDD Workshop, pp. 359–370 (1994).
- [16] B.-K. Yi, H. V. Jagadish and C. Faloutsos: "Efficient retrieval of similar time sequences under time warping.", ICDE, pp. 201–208 (1998).
- [17] C. Faloutsos and K.-I. Lin: "Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets.", SIGMOD Conference, pp. 163–174 (1995).
- [18] E. J. Keogh: "Exact indexing of dynamic time warping.", VLDB, pp. 406–417 (2002).
- [19] Y. Zhu and D. Shasha: "Warping indexes with envelope transforms for query by humming.", SIGMOD Conference, pp. 181–192 (2003).
- [20] E. J. Keogh, K. Chakrabarti, M. J. Pazzani and S. Mehrotra: "Dimensionality reduction for fast similarity search in large time series databases.", Knowl. Inf. Syst., **3**, 3, pp. 263–286 (2001).
- [21] Y. Sakurai, M. Yoshikawa and C. Faloutsos: "Ftw: Fast similarity search under the time warping distance.", PODS, pp. 326–337 (2005).
- [22] R. Cheng, D. V. Kalashnikov and S. Prabhakar: "Evaluating probabilistic queries over imprecise data.", SIGMOD Conference, pp. 551–562 (2003).
- [23] R. Cheng, Y. Xia, S. Prabhakar, R. Shah and J. S. Vitter: "Efficient indexing methods for probabilistic threshold queries over uncertain data.", VLDB, pp. 876–887 (2004).
- [24] A. Guttman: "R-trees: A dynamic index structure for spatial searching.", SIGMOD Conference, pp. 47–57 (1984).
- [25] Y. Tao, R. Cheng, X. Xiao, W. K. Ngai, B. Kao and S. Prabhakar: "Indexing multi-dimensional uncertain data with arbitrary probability density functions.", VLDB, pp. 922–933 (2005).
- [26] L. E. Baum and T. Petrie: "Statistical inference for probabilistic functions of finite markov chains.", Ann. Math. Stat., **37**, pp. 1554–1563 (1966).
- [27] L. R. Rabiner and B. H. Juang: "An introduction to hidden markov models.", IEEE ASSP Magazine, **3**, pp. 4–16 (1986).