構造化ストリングデータにおける知識獲得および生成

碓井大祐[†,☆] 荒木宏行[†] 阿江忠[†]

構造を持つストリングからの知識獲得と獲得された知識を用いたストリング生成の一手法を提案する．要素はベクトルから成り，その要素に付与するカテゴリをサンプルにより学習させる．このような要素の列がストリングを形成しており，そのストリング自体もサンプルにより学習させ，いくつかのストリングセット（各セットをケースと呼ぶ）の構造を表す特徴的なシーケンスを獲得する．ストリングの学習はデータマイニング的手法を用いるが，ストリングにおけるシーケンスを考慮するため従来のものとは異なったものである．このような構造化されたデータの学習結果に遺伝的アルゴリズムを適用し，新たなデータの生成を行う．

Knowledge Acquisition and Generation on Structured String Data

DAISUKE USUI,[†,☆] HIROYUKI ARAKI[†] and TADASHI AE[†]

We propose a method of knowledge acquisition from structured string data and a string generation using acquired knowledge. In this method, a component is represented by a vector. and a category including several samples is learned. A string data consists of components, and learning of string data is also made on samples. The acquired sequence represents a structure of string sets. A data-mining-like method is used for learning, but it is different from the original one, because we need to consider a string (sequence) consisting of components. We apply a genetic algorithm (GA) for generation of new data using results obtained from such structured data.

1. はじめに

本稿では構造化ストリングデータにおける知識獲得とその知識を用いたデータ生成について述べる．

従来の生成に関する研究として，Sheldon Kleinらによる自動作文の研究[1]，CT（Computed Tomography）に代表される画像生成に関する研究[2]，高橋らによる分子設計支援システム“TUTORS”[3]など，様々なものがあるが，これらはすべて，生成のために explicit な知識表現を必要としており，その知識の範囲内でのみ目標とする生成が行われる．そのため，与えられる知識は問題となる領域をすべて表現するために大量の記述が必要となる．また，知識が与えられた範囲外での生成は行えない．また，模擬育種法（simurated breeding）による作曲支援システム[4]もあり，これはデータ列を GA（Genetic Algorithm）[5),6)]的手法により進化させることにより作曲を行うというものである．しかし，データ列の進化の際，逐次ユーザによりデータの選択をせねばならず，ユーザに対する負担が大きくなるという問題がある．これらのことを考慮したうえで，本研究では生成の前にサンプルからの学習を行うことにした．多戦略学習[7]の観点からみると，提案方法は要素レベルをニューラルネットを用いてクラスタリング学習をし，ストリングレベルでカテゴリに対しエントロピーと閾値を基準とした学習を行うことで，少数のサンプル例から学習可能な帰納学習である．さらに，2レベルの構造により以下のように効率良くサンプル例に明示されていない表現空間を生成できると思われ，構成的帰納学習[8]としても有効であると思われる．まず，同一カテゴリのすべての要素を用いてストリングレベルの学習をする必要性はないので，要素レベルで新しい要素のものが生成でき，さらに，獲得されたシーケンス自体も GA を用いて変化するので，コストという条件を満たす範囲で新しいシーケンスも生成できると思われる．また，文献 4) で問題となったデータ選択も，獲得された知識を用いることにより自動的に行うことができる．以下に本手法の概要を述べる．

知識発見における問題として，膨大な探索空間によ

† 広島大学工学部
Faculty of Engineering, Hiroshima University
☆ 現在，日本航空電子工業
Presently with Japan Aviation Electronics Industry, Limited

る計算効率の低下があげられる[9)~11)]．その解決策の1つとして属性指向のdomain knowledgeを用いる方法がある[12),13)]．これは属性ごとにその属性値の階層概念を構築しておくことで，計算過程において各属性値を抽象化し，その探索空間を減少させるというものである．しかし，そのようなdomain knowledgeは専門家の知識として与えられており，新たな属性に対する情報を与えることが困難な場合が多い．そこで本研究ではデータ探索の前にdomain knowledgeの代わりとなる属性に関する分類ルールを学習させることにした．その上でデータ間のインターセクションをとることにより獲得される知識は，エキスパートシステムなどにおけるような明確なものでなく，抽象化されたimplicitなものであり，その知識自体の空間量もかなり小規模なものとなる．そして獲得された知識をデータ生成のための探索空間と見なし，その空間内でGAを用いることにより新たなデータの生成を試みる．データ探索により得られた空間を内包的空間とすると，そこからデータを生成することはその空間の性質を含みつつ，さらに拡張された外延的空間を探索していくことになるので，データベースには示されなかったデータが生成される可能性がある．

本研究で対象とするデータは構造を持つストリングデータであり，それは要素レベルとストリングレベルの2段階の構造を持ったものである．このような構造化データを用いることで自然で容易な学習，生成を目指している．また，そのようなものの応用例としてツアープランニングを考える．

2. 構造化ストリングデータ

構造化ストリングデータは要素レベルとストリングレベルの2レベルの構造から成る．ストリングは要素の並びにより構成されており，学習用のサンプルセットはいくつかのケースごとに与えられ，その各ケースの構造を得ることが本研究における学習の目的である．この際，ストリングを構成する各要素はベクトル値を持っている．そのベクトル値に従って要素はその上位概念であるカテゴリに分類される．その分類規則に従って，ストリング中の要素をカテゴリに置き換えることにより，学習の対象となるデータ空間は飛躍的に減少され，ストリングレベルでの学習が簡単化できる．構造化ストリングデータは以下のように記述される．

要素集合を $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_i,\cdots,\boldsymbol{x}_I\}$ とすると，各要素は $\boldsymbol{x}_i=(a_{i1},a_{i2},\cdots,a_{iv})$ というベクトル値を持ち，それぞれの要素はいくつかのカテゴリ $\boldsymbol{Y}_j$ $(j=1,\cdots,J)$ に分類される．すなわち，$\boldsymbol{X}=\bigcup_j \boldsymbol{Y}_j$（ただし，$\bigcap_j \boldsymbol{Y}_j=\phi$）である．また，ストリング集合を $\boldsymbol{S}=\{s_1,\cdots,s_l,\cdots,s_L\}$ とすると，各ストリングは $s_l=\boldsymbol{x}'_{l1}\boldsymbol{x}'_{l2}\cdots\boldsymbol{x}'_{lp}\cdots\boldsymbol{x}'_{lP}$ と記述される☆．ただし，$\boldsymbol{x}'$ は $\boldsymbol{X}$ の中の任意の要素を表す．ストリング集合はいくつかの部分集合 $\boldsymbol{S}_k$ $(k=1,\cdots,K)$ に分類され，$\boldsymbol{S}=\bigcup_k \boldsymbol{S}_k$（ただし，$\bigcap_k \boldsymbol{S}_k=\phi$）のように与えられる．このストリング集合がサンプルセットであり，その部分集合をケースと呼ぶ．学習の際，これらのストリング中の要素はカテゴリに置き換えられ，連続した同一カテゴリの数を乗数 m で表すことにより，ストリングは次のような形になる．

$$s_l=\boldsymbol{Y}'^{m_{l1}}_{l1}\boldsymbol{Y}'^{m_{l2}}_{l2}\cdots\boldsymbol{Y}'^{m_{lq}}_{lq}\cdots\boldsymbol{Y}'^{m_{lQ}}_{lQ} \qquad (1)$$

ここで，$\boldsymbol{Y}'$ は $\boldsymbol{Y}$ の中の任意のカテゴリを意味する．

このような構造化ストリングの応用例としてツアープランニングを考える．ツアープランニングはユーザの希望するツアーを生成するシステムであり，要素を都市，ストリングをツアーと見なすことで構造化ストリングデータが適用できる．

3. 知識の獲得

ここでは，いくつかのケースごとに与えられた構造化ストリングデータのサンプルセットから，各ケースを記述するシーケンスの知識テーブルを獲得する．

3.1 要素レベルの学習

要素はベクトルを持つので距離空間に配置できる．そして，各要素はカテゴリの代表である参照ベクトルにより適切なカテゴリに分類される．その参照ベクトルはLVQ（Learning Vector Quantization）により学習される．

LVQはベクトル量子化法を発展させたものであり（ベクトル量子化法はパターン空間を有限個の参照ベクトルで記述するものである（**図1**）[14)]），参照ベクトルをニューラルネットワークにおけるニューロン間のシナプス結合に対応させ学習することにより適切な参照ベクトルを求める方法である．

LVQの学習では，各カテゴリの参照ベクトルに学習用のカテゴリが付加された要素のトレーニングセットを与えることにより，各カテゴリの参照ベクトルの状態が学習される．その結果，各要素はベクトル間の距離が一番近い参照ベクトルを持つカテゴリに分類される．

☆ $\forall s_l\in\boldsymbol{S}\rightarrow s_l\in\boldsymbol{X}^*$
ただし，$\boldsymbol{X}^*$ は $\boldsymbol{X}$ の要素から構成されるストリングの集合である．

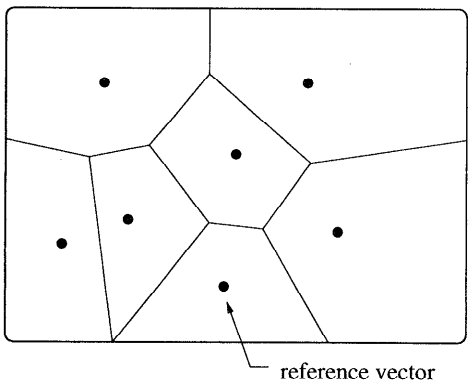


図1 ベクトル量子化
Fig. 1 Vector quantization.

3.2 ストリングレベルの学習

ストリングレベルでの学習では，各ケースごとにストリング間のインターセクションをとることにより特徴的なシーケンスを抽出する．通常，ストリング間のインターセクションはその組合せをすべて見ていかねばならないため多大なコストを必要とする．それに対する近似的手法[15]も提案されているが，ここではデータマイニング[16]的手法を用いる．通常，このような手法は膨大な計算コストを必要とするが，我々の手法では要素レベルの学習により学習対象となるデータ空間を飛躍的に削減できるため，1点固定でのインターセクションが可能である．具体的には以下に示すように行う．

(1) 各ストリングの要素を先に学習したカテゴリに置き換え，式(1)の形にする．

(2) 各ケースセットにおいて，すべてのストリングに共通なカテゴリ $\boldsymbol{Y}_k^C$ を見つける．通常，1つのケースに対し共通カテゴリは複数見られるのでそれぞれを $\boldsymbol{Y}_k^{C'}$ とおき，それぞれの共通カテゴリの乗数 $m^{c'}$ によるエントロピー（式(2)）をとる．その結果，エントロピーが最大となるものをそのケースの代表カテゴリ $\boldsymbol{Y}_k^R$ とし，それぞれのストリングにおける代表カテゴリの位置を q^R とする．

$$E(\boldsymbol{Y}_k^{C'}) = -\sum_l \frac{m_{klq}^{c'}}{N_{kq}} log \frac{m_{klq}^{c'}}{N_{kq}} \qquad (2)$$

ただし，$N_{kq} = \sum_l^L m_{klq}^{c'}$ である．

(3) 各ケースセットの中で代表カテゴリの乗数 m_{klq}^R の等しいストリングどうしでインターセクションをとる．通常のインターセクションは $A^3B^2C^3 \cap A^2B^4 = A^2B^2$ のように完全な一致のみ残すことを意味する．しかし，ここでは構造化ストリングデータを用いているので，ある程度のあいまいさを許すことにより以下のような方法でシーケンスを獲得できる．

- 0～1.0の値を持ったconfidence値を与える．これにより残るべきシーケンスが変化し獲得される知識の曖昧さが決まる．
- 各ケースごとに，代表カテゴリの乗数 m_{klq}^R の等しいストリング $s_{kl'}$ を代表カテゴリの位置 q^R でそろえる（l' は共通の乗数を持つストリングを指す）．代表カテゴリを中心にして順番に同列中に存在する各カテゴリの数 $\sum m'_{kl'(q^R+i')}$（ここで，m' は共通のカテゴリを表し，$i' = (\ldots,-1,0,1,\ldots)$ である．）を同列中のカテゴリの合計数 $\sum m_{kl'(q^R+i')}$ で割り，最大となる値がconfidence値以上となるカテゴリを残す．これによりシーケンスを表現するカテゴリの並び *cline* が決まる．
- 残ったカテゴリの乗数の合計 $\sum m'_{kl'(q^R+i')}$ をそのインターセクションにおけるストリング数 $\sum_{l'} s_{kl'}$ で割ったものをそのカテゴリの乗数とする．

(4) 最後にインターセクションによってできた新たなストリングセットの各ストリング $s_{kl'}$ に対し，

$$rate\,(k,l',q^R+i') = \frac{\sum m'_{kl'q^R+i'}}{\sum_{l'} s_{kl'} \times m_{kl'qR}} \qquad (3)$$

というカテゴリの乗数に対する比率計算を行い，各ケースの代表カテゴリとその各シーケンスの *cline* とその比率 *rate* がケースごとの知識テーブルに収められる．

3.3 Example1

簡単な例を考える．要素の集合として以下のものを扱う．

$$\boldsymbol{X} = \{\, a\,,\ b\,,\ c\,,\ d\,,\ e\,,\ f\,\}$$

これらがLVQ学習の結果を用いて以下のカテゴリ $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ に分類されたとする．

$$\boldsymbol{A}=\{\, a\,,\ b\,\} \quad \boldsymbol{B}=\{\, c\,,\ d\,\} \quad \boldsymbol{C}=\{\, e\,,\ f\,\}$$

これらの要素によるストリングセットをケースごとに与え，それぞれのケースがどのような構造を持つか学習していく．

case1として以下のサンプルセットが与えられた場合の知識獲得を3.2節に示された流れに沿って行う．

$\boldsymbol{S}_1$={

$s_1 = d\,a\,a\,b\,e\,f\,,\quad s_2 = b\,a\,b\,f\,f\,e\,,$

$s_3 = a\,b\,e\,e\,e\,,\quad s_4 = c\,b\,e\,f\,f\,,$

$s_5 = a\,c\,d\,e\,f\,,\quad s_6 = b\,f\,f$

}

表1 case1の知識テーブル
Table 1 Knowledge table of case1.

sequence No.	cline	rate
1	AC	0.67 : 1
2	BAC	0.11 : 0.67 : 1

(1) これらのストリングは，各要素を一致するカテゴリに置き換えることにより式(1)の形をとり次のようになる．

S_1={
$s_1 = B^1\ A^3\ C^2$, $s_2 = A^3\ C^3$
$s_3 = A^2\ C^3$, $s_4 = B^1\ A^1\ C^3$
$s_5 = A^1\ B^2\ C^2$, $s_6 = A^1\ C^2$
}

(2) 次に式(2)により各カテゴリのエントロピーをとる．各ストリングに共通なカテゴリはAとCなので，

$E(A) = 1.673$
$E(C) = 1.771$

よりCがcase1の代表カテゴリになる．

(3) ここでconfidence=0.6とする．まず，Cの乗数が2であるs_1, s_5, s_6を見る．共通カテゴリCの列から順に同列内でのカテゴリの割合を見る．最初の列は0.67でAが残り，次はA，Bとも0.5なので削除される．そしてAの乗数は(3+1)/3=1.33となる．同様に，Cの乗数が3のストリングについても計算する．

$s_1 = B^1\ A^3\ C^2$
$s_5 = A^1\ B^2\ C^2$
$s_6 = A^1\ C^2$
$s'_1 = A^{1.33}\ C^2$

(4) 最後に式(3)を用いて各ストリングにおけるカテゴリの乗数の比率を計算し，**表1**のような知識テーブルを得る．

これらの作業を，与えられたすべてのケースセットに対し行う．

4. 獲得知識の評価方法

学習の後，獲得された知識がユーザの意図をどれだけ反映しているか検証する必要がある．そのために，各ケースごとにサンプルセットとは異なるデータセット（テストデータ）を用意し，獲得知識によりそれらがどれだけ正確に分類されるかを見る．

4.1 分類方法

1つのストリングに対し獲得された知識テーブルを用いてすべてのケースとの距離をとり，その距離が最小となるケースにそのストリングは分類される．テストデータ中のあるストリングs'に対する分類は以下のように行う．

(1) s'を知識テーブルの形に変換する．まず，s'の要素を式(1)のようにカテゴリに置き換える．この時点でカテゴリの乗数を取り除いたものがs'の$cline$となる．次に代表カテゴリに当たるs'のカテゴリの乗数ですべての乗数を割り，これらをs'のカテゴリ比$rate(s', q)$とする（これは対象となるケースごとに異なる）．

(2) ケースkの各シーケンスl_sに対し，

(a) 代表カテゴリを中心に$cline$を比較し一致しなかったカテゴリの数をカウントし，これを$cline$の距離$d^1_{kl_s}$とする．

(b) 一致したカテゴリの比について

$$d^2_{kl_s} = \sum_j |rate(k, l_s, q) - rate(s', q)|$$

を計算する．

(c) シーケンスとの距離を

$$d_{kl_s} = d^1_{kl_s} + \alpha\ d^2_{kl_s} \qquad (4)$$

とし，最小となる値d_{kl_s}をケースkとの距離D_kとする．

(3) D_kを最小とするケースkにs'は分類される．

これをすべてのテストデータに対して行い，正確に分類されたデータの割合により知識は評価される．

4.2 Example2

次のようなストリングs'が与えられたとき，3.3節で得られたcase1の2番目のシーケンス（表1）に対する距離をとってみる．

$s' = a\ c\ b\ a\ e\ f\ e\ e$

(1) $s' = A^1\ B^1\ A^2\ C^4$
case1の代表カテゴリはCなので$cline$は$ABAC$で，$rate$は0.25 : 0.25 : 0.5 : 1.0である．

(2) Cの部分をそろえて距離をとる．$ABAC$とBACより

(a) d^1_{12}=1

(b) d^2_{12}=|0.25 − 0.11| + |0.50 − 0.67| = 0.31

(c) よって，$\alpha = 0.1$とすると$d_{12} = 1.031$となる．同様にして$d_{11} = 2.017$となるので，$D_1 = 1.031$である．

このαの値は実験では0.05〜0.1のとき最も良い結果が得られた．

5. GAによるストリングの生成

GAは生物の遺伝子における進化に着想を得て考えられたアルゴリズムであり，用意されたデータ列の集

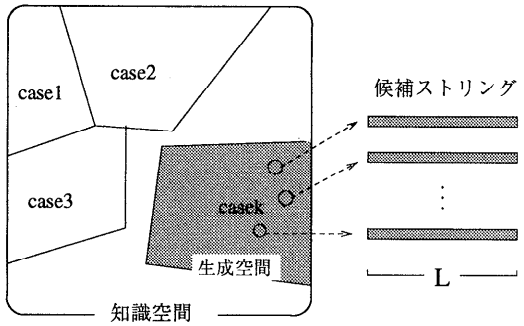


図2 問題1の生成イメージ
Fig. 2 Image of generation by problem 1.

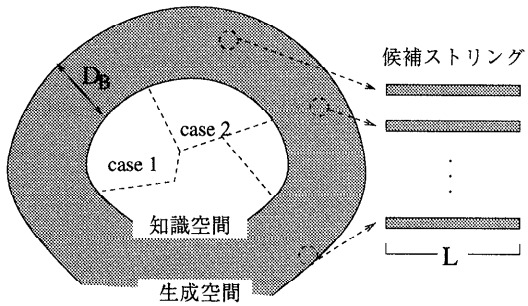


図3 問題2の生成イメージ
Fig. 3 Image of generation by problem 2.

合に対し淘汰，交叉，突然変異の遺伝的操作を何世代にもわたり繰り返すことにより希望する解を得ようというものである[5]．本研究ではこの手法を獲得した知識に用いることでユーザの要求を満たすストリングの生成を行う．

5.1 生成問題

2つの生成問題を考える．

問題1 ある1つのケース k に属す長さ L_s のストリングをそのコストが最適値 C になるよう生成せよ．

問題2 長さが L_s でコストが最適値 C となる新たなケースに属するストリングを生成せよ．

それぞれの問題による生成のイメージ図を**図2**，**図3**に示す．これらはサンプルから学習された空間を内包的空間とすると，前者を内包的探索，後者を外延的探索を意味するものである．

ここで，C には有効誤差 δ を持たせるものとする．また，コストの設定は用いるストリングの性質により異なるが，ここでは単純にすべての要素間に距離を持たせる．つまり，要素数 × 要素数の距離マトリクスを与え，そのストリング s_l 内での要素間の距離の合計値を $cost(s_l)$ とする．これによりこれらの問題におけるストリングの適応度を

$$|C - cost(s_{max})| - |C - cost(s_l)| \quad (5)$$

と定義する．ここで s_{max} とは，各世代のプール内において，$|C - cost(s_l)|$ を最大にするストリングである．この適応度は再生手続きの際，プール内から親を選択するときに必要となる．

5.2 生成手法

GAを用いる前に，いくつかのパラメータを設定する．プールサイズ N_p，最大世代数 G_{max}，要求するストリングの候補数 S_d，交叉の起こる確率 R_c，突然変異の起こる確率 R_m，突然変異の局所率 R_l とする．本稿では $R_c = 1.0$，$R_m = 0.3$ とする．この設定は通常のGAよりもかなり大きい値となっているが，ここでのGAの適用目的は新たなストリングの生成であり，同一ストリングの生成が意味をなさないためである．また，突然変異の局所率とはある要素がその要素の属するカテゴリの要素に変異する確率で，問題1では $R_l = 0.8$，問題2では $R_l = 0.1$ となっている．これはある要素がその要素の属するカテゴリ以外の要素に変異することは学習された知識構造を変化させることを意味するためである．

5.2.1 問題1

問題1に対する手順は以下のようになる．

(1) 第1世代プールの生成：

ケース k の知識テーブル内のシーケンスからストリングを生成する．各シーケンス中のカテゴリからそのカテゴリに属する要素をランダムに N_r 個ずつ生成することにより長さ L_s のストリングができる．ここで $N_r \equiv N_{k,l,q^R+i'} = L_s \times rate(k,l,q^R+i')/\sum_i rate(k,l,q^R+i')$ を4捨5入した整数である．ケース k のシーケンスの数を N_{ks} とするとそれぞれのシーケンスから N_p/N_{ks} 個（少数点以下切捨て）ずつストリングを生成する．これらの集合を第1世代のプールとする．

(2) 再生：

この段階でストリングの適応度を計算し，交叉する親を選択する．ストリングの選択にはルーレット式親選択法を[6]，交叉には1点交叉を用いている．交叉のあとに確率 R_m で突然変異を起こし，前世代のプール中のストリングから新たなストリングを生成する．生成されたストリングとプール内のストリングを合わせて次世代への候補集団とする．

(3) 淘汰：淘汰は2段階で行われる．

(a) 4章の分類規則に従い各ストリングがどのケースに属するかを見る．ここで，ケース k に属さないストリングは削除される．

(b) 残ったすべてのストリングの適応度を計算し，

大きい値を持つものから順に次世代のプールに入れる．ただし，同じストリングは排除する．

(4) 終了：次のどちらかの条件を満たせば終了となる．

(a) プール内で $|C - cost(s_l)| < \delta$ となるストリングの数が S_d 個以上となったとき，適応度の大きい順に S_d 個のストリングをユーザに示し，ユーザがそれらを認めた場合．

(b) 世代数が G_{max} となった場合．

そうでなければ，(2)，(3)を繰り返す．

5.2.2 問 題 2

基本的な手順は問題1と同じだが，その中の(1)と(3)(a)は異なり，以下のようになる．

(1) 第1世代プールの生成：

すべてのケースの知識テーブルのシーケンスからストリングを生成する．その方法は問題1と同様だが，すべてのシーケンスから $N_p/\sum_k N_{ks}$ 個ずつストリングを生成するものとする．

(3) 淘汰：

(a) ここでは，生成されるストリングが学習された知識空間から離れすぎないよう，限界距離 D_B を与える．これは，離れすぎるとストリングの構造がランダムになり意味がなくなるのでそれを防ぐためである．そのうえで，それぞれのストリングはすべてのケースとの距離をとられ，最小となる距離 $D_k \leq D_B$ となるストリングのみ残す．

6. ツアープランニング

以上で述べた手法をツアープランニングに適用する．ツアーの学習，生成に用いた都市はそれぞれ **表2** のような属性値を持っており，92都市用意した．これらの内22都市をサンプルとしてLVQで学習させ，分類した結果が **表3** である．表3中での各カテゴリの意味はそれぞれ，***A***：大都市，***B***：首都，***C***：地方の中心都市，***D***：商業都市，***E***：観光地，***F***：工業都市，となっている．ここで問題になるのが距離マトリクスで，92×92のマトリクスは膨大な数になる．そこで，都市を地図上で12のブロックに区分し（**図4**），そのブロック間の移動時間（半日単位）を **図5** のようなグラフにより表現した後マトリクスにし，これによりコストを求めることにした．また，ツアー構造の学習は4つのケースを用いて行った．サンプル数は各ケースとも10個で，それぞれのケースの意図は，case1：ビジネスツアー（工業都市や商業都市を廻る），case2：ショッピングツアー（商業都市や大都市を廻る），case3：観光ツアー（観光都市や歴史的都市を廻る），case4：レジャーツアー（山，海，娯楽施設などを廻る），というものである．その学習結果を用いた実際の生成例を **表4** に示す．ここで，*の付いたツアーはユーザが修正すれば使えると判断したものであり，各ツアーの右端の( )内の数値はそのツアーが生成された世代数である．

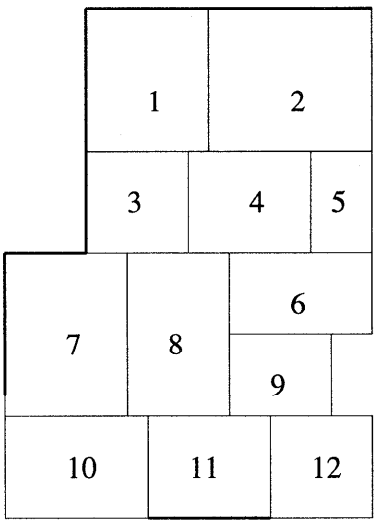


図4 12ブロックに区分した中央ヨーロッパ地図
Fig. 4 A Central European Map partitioned into 12 blocks.

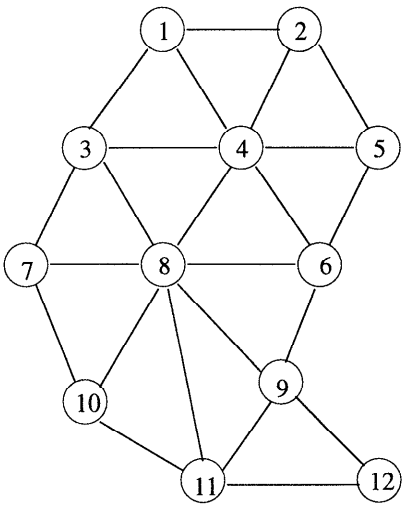


図5 ブロック間の移動グラフ
Fig. 5 A graph of blocks.

7. おわりに

本稿では構造化ストリングデータにおける知識獲得と，それを基にしたデータ生成の一手法を提案した．

構造化ストリングデータを用いることで学習におけ

表2 ツアープランニングに用いた都市データ
Table 2 City data for tour-planning.

都市名	治安	経済	工業	農業	文化&学問	水産業	景観	スポーツ&リゾート
Amsterdam	0.7	1.0	0.1	0.2	0.8	0.5	1.0	0.5
Groningen	0.4	0.4	0.4	0.8	0.6	0.2	0.3	0.7
Den-Haag	1.0	0.6	0.2	0.5	0.9	0.8	0.9	0.9
Frankfurt	0.4	1.0	0.2	0.2	0.5	0.0	0.4	0.2
⋮					⋮			

表3 ツアープランニングに用いた92都市（学習後）
Table 3 92 cities for tour-planning (after learning).

カテゴリ	都市
A	Amsterdam Munchen Zurich Geneve Bruxelles Paris Marseille
B	Den-Haag Bonn Berlin Bern Luxembourg
C	Groningen Eindhoven Koln Stuttgart Nurnberg Grenoble Tours Dresden Leiptig Hamburg Luzern Fribourg Bordeaux Erlangen Antwerpen Liege Strasbourg Dijon Rouen Renne Lausanne
D	Frankfurt Maastricht Dortmund Essen Dusseldorf Lyon Kaiserslautern Saarbrucken
E	Delft Bremen Koblenz Wiesbaden Mainz Darmstadt Deauville Orleans Manheim Heideberg Karlsruhe Freiburg Wurzburg Monte-Carlo Nice Rothenburg Augsburg Regensburg St. Moritz Interlaken Cannes Angers Lugano Brugge Namur Waterloo Metz Nancy Brest Cherbourg Sanremo Colmar Mulhouse Besancon Amiens Le-Haver Le-Mans Caen Neuchatel
F	Rotteldam Aachen Basel Lille Tourlouse Torino

表4 ツアープランニングの生成例
Table 4 Examples of generated tours on tour-planning.

case1	Aachen Basel Rotteldam Dusseldorf Dusseldorf Maastricht (2) *Aachen Kaiserslautern Aachen Dusseldorf Lyon Lyon (3) Groningen Lille Essen Dortmund Frankfurt Saarbrucken (6)
case2	Frankfurt Dortmund Munchen Lugano Geneve Geneve (3) Saarbrucken Dortmund Munchen Paris Paris Geneve (3) Saarbrucken Essen Bruxelles Bruxelles Amsterdam Munchen (4)
case3	Toulon Koblenz Nancy Den-Haag Dortmund Bonn (3) Le-Haver St. Moritz Luxembourg Luxembourg Den-Haag Den-Haag (3) Augsburg St. Moritz Luxembourg Amsterdam Essen Essen (4)
case4	*Nurnberg Manheim St. Moritz Paris Munchen Paris (3) Koln Besancon Brest Namur Paris Paris (3) Renne Le-Mans Deauville Munchen Bruxelles Paris (5)
new case	Koln Rothenburg Bonn Amsterdam Saarbrucken Lyon (3) Saarbrucken Saarbrucken Munchen Paris Bern Berlin (4) Kaiserslautern Den-Haag Munchen Bruxelles Bruxelles Rotteldam (5)

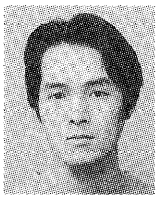
る対象データ空間を飛躍的に減少させることができた．I 個の要素が J 個のカテゴリに分類され，サンプルセットの平均ストリング長を L_m とすると対象データ空間は通常では I^{L_m} であるのに対し，構造化ストリングデータを用いると J^{L_m} となる．ツアープランニングでは $I = 92$，$J = 6$ であり，$L_m = 10$ とすると，4.3×10^{19} のデータ空間が 6.0×10^7 にまで減少している．これにより，1点固定でのインターセクションという形でストリングからの特徴抽出を簡単化することができた．また，その学習結果を用いることで容易に新たなデータを生成できた．応用システムとしてのツアープランニングではブロック化した距離マトリクスを用いたため制約が緩くなったが，数世代に数個の割合で複数のケースの特徴を持つユーザの好みのツアーが生成されることが確認できた．

参考文献

1) Rich, E.: *Artificial Intelligence*, McGraw-Hill

(1983). 廣田 薫, 宮村 鮫 (訳): 人工知能 II, マグロウヒルブック (1984).
2) 塚本克治 (編・著): AI—情報処理から知能処理へ, アスキー出版局 (1988).
3) 第 2 回「大学と科学」公開シンポジウム組織委員会 (編): 知識情報の世界を拓く, 朝日出版社 (1988).
4) 藤原 寛, 阿江 忠: 構造をもつデータ列の進化—作曲を例として, 人工知能学会並列人工知能研究会資料, SIG-PPAI-9402-4 (1994).
5) 人工生命研究会 (編): 人工生命, 共立出版 (1994).
6) Davis, L.: *Handbook of Genetic Algorithms*, A Division of Wadsworth (1990). 嘉数侑昇, 三上貞芳, 皆川雅章, 川上 敬, 高取則彦, 鈴木恵二 (訳): 遺伝アルゴリズムハンドブック, 森北出版 (1994).
7) 沼尾正行: 複数の情報媒体を用いた学習—多戦略学習とその情報源による分析, 人工知能学会誌, Vol.9, No.6, pp.837–842 (1994).
8) 滝 寛和: 構成的帰納学習とバイアス, 人工知能学会誌, Vol.9, No.6, pp.818–822 (1994).
9) Kramer, S.: CN2-MCI: A Two-Step Method for Constructive Induction, *Proc. ML-COLT'94* (1994).
10) Yasdi, R.: Learning Classification Rules from Database in the Context of Knowledge Acquisition and Representation, *IEEE Trans. Knowledge and Data Engineering*, Vol.3, No.3, pp.293–306 (1991).
11) Stanfill, C. and Waltz, D.: Toward Memory-Based Reasoning, *Comm. ACM*, Vol.29, No.12, pp.1213–1228 (1986).
12) Han, J., Cai, Y. and Cercone, N.: Data-Driven Discovery of Quantitative Rules in Relational Databases, *IEEE Trans. Knowledge and Data Engineering*, Vol.5, No.1, pp.29–40 (1993).
13) Chu, W.W. and Chen, Q.: A Structured Approach for Cooperative Query Answering, *IEEE Trans. Knowledge and Data Engineering*, Vol.6, No.5, pp.738–749 (1994).
14) Kawada, M., Wu, X. and Ae, T.: A Construction of Neural-Net Based AI Systems, *Proc. 1st IEEE International Conference on Engineering of Complex Computer Systems*, pp.424–427 (1995).
15) 中川裕之, 伊藤 実, 橋本昭洋: 遺伝的アルゴリズムによるマルチプルストリングアライメント, 情報処理学会論文誌, Vol.37, No.8, pp.1543–1552 (1996).
16) Agrawal, R. and Sami, A.: Database Mining:A Performance Perspective, *IEEE Trans. Knowledge and Data Engineering*, Vol.5, No.6, pp.914–925 (1993).

(平成 9 年 3 月 19 日受付)
(平成 11 年 1 月 8 日採録)

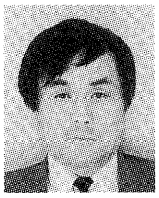


碓井 大祐

1972 年生. 1995 年広島大学工学部第 II 類 (電気系) 卒業. 1997 年同大学院情報工学科博士課程前期修了. 同年日本航空電子工業入社.

荒木 宏行

1969 年生. 1992 年熊本大学理学部物理学科卒業. 1995 年広島大学大学院理学研究科博士課程前期修了. 1998 年同大学院情報工学科博士課程後期中退. 同年同大学工学部助手となり現在に至る.



阿江 忠 (正会員)

1941 年生. 1964 年東北大学工学部通信工学科卒業. 1969 年同大学院博士課程修了. 同大学助手, 広島大学助教授を経て, 1982 年より広島大学工学部第 II 類 (電気系) 教授.