第 3 章
モデリングの基礎

3.1 行動データマイニング

(e) 行動データマイニングの概念

図 3.1 に示すように，行動調査の調査法は従来の紙ベースから GPS ベースの時間的・空間的に細かな情報を用いた調査に変わりつつある．それに伴い，計画の単位もマクロからメゾ，ミクロとより細かな単位での評価が可能となっている．ただし，GPS 機能を用いた行動調査の場合，多量の位置データは取得できるものの被験者に付加的に入力を行ってもらわない限り移動手段やトリップ目的などの情報を得ることはできない．またデータの粒度が細かくなるに従い，観測誤差や欠損といった現象をモデル中で扱うことの重要性も高まっている．

このような課題に対し，データの特徴量（位置，時刻，速度，加速度）を用いて移動手段を自動的に判別する問題や，誤差や欠損を補正しながらデータを扱うための手法として，データマイニングや機械学習の手法が有用となる．データマイニングは多量のデータから隠れた法則や有用なパターンを見出したり行動データを正規化して従前のモデルに持ち込むための前処理を行ったりする際に用いられる．本章では，代表的な行動データ分析の対象とその手法について概説する．

(f) 代表的な分析手法

クラスタリング

クラスタリングは，統計・パターン認識・データベース・データマイニングなどの分野で広く用いられている古典的な手法である．非常に多くのデー

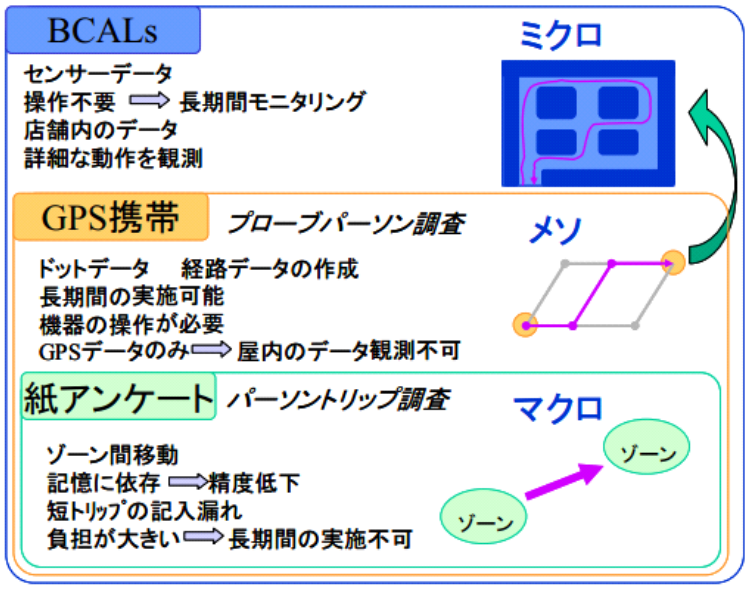


図 3.1　行動データ種別 (羽藤ら (2005))

タが集まったとき，これを自動的にいくつかの似ているもの同士グループにわけることができれば，データの概要を効率的に概観しセグメンテーションに基づく細かな情報提供や施策を行える可能性がある．クラスタリング手法は大きく，最短距離法などの階層的手法と k-means などの分割最適化手法に分けられる．ここでは，階層的手法の代表例である凝集型クラスタリングと k-means 法，さらに確率的なクラスタリングについて述べる．

- 凝集型クラスタリング　今，対象とするデータが N 個あるとする．凝集型クラスタリングでは，1 個の対象だけを含む N 個のクラスタがある初期状態から，クラスタ間の距離（非類似度）関数に基づき最も距離の近い二つのクラスタを逐次的に統合していく手法である．この統合をすべての対象が一つのクラスタに含まれるまで繰り返すことで階層構造を獲得する．下（バラバラの状態）から上（統合された状態）に順に統合していくのでボトムアップクラスタリングとも呼ばれる．クラスタリングの結果はデンドログラムと呼ばれる樹形図によって表示されることが一般的である．

  凝集型クラスタリングをアルゴリズムの形で表すと次のようになる．ただし，$sim(C_1, C_2)$ は 2 つのクラスタ間の距離を表す．

  ステップ 0

  入力：事例集合 $D = \{x^{(1)}, x^{(2)}, \cdots, x^{(N)}\}$

  クラスタ：$C = \{c_1, c_2, \cdots, c_N\}$

  ステップ 1

  1 つのクラスタに 1 つの事例を割り当てる．

  $c_1 = \{x^{(1)}\}, c_1 = \{x^{(2)}\}, \cdots, c_N = \{x^{(N)}\}$

  ステップ 2

  最も似ているクラスタ対を見つける．

  $$(c_m, c_n) = \arg \max_{c_i, c_j \in C} sim(c_i, c_j)$$

  ステップ 3

  見つかったクラスタ対を融合する．

$merge(c_m, c_n)$

ステップ 4

$|C| = 1$ になるまでステップ 2，3 を繰り返す．

ここで，クラスタ C_1 とクラスタ C_2 の距離関数 $sim(C_1, C_2)$ の違いにより以下のような手法がある．

- 最短距離法または単連結法
  この手法は，2 つのクラスタが与えられたときその中で最も近い事例対の類似度をその 2 つのクラスタの類似度とする方法である．

$$sim(C_1, C_2) = \min_{x_1 \in C_1, x_2 \in C_2} sim(x_1, x_2) \tag{3.1}$$

- 最長距離法または完全連結法
  この手法は，2 つのクラスタが与えられたときその中で最も遠い事例対の類似度をその 2 つのクラスタの類似度とする方法である．

$$sim(C_1, C_2) = \max_{x_1 \in C_1, x_2 \in C_2} sim(x_1, x_2) \tag{3.2}$$

- 群平均法
  群平均法では，与えられた 2 つのクラスタの重心間の類似度を 2 つのクラスタの類似度とする方法である．

$$sim(C_1, C_2) = \frac{1}{n_1 n_2} \sum_{x_1 \in C_1} \sum_{x_2 \in C_2} sim(x_1, x_2) \tag{3.3}$$

- ウォード法
  ウォード法は各対象からその対象を含むクラスタのセントロイドまでの距離の二乗の総和を最小化する手法である．

$$sim(C_1, C_2) = E(C_1 \cup C_2) - E(C_1) - E(C_2) \tag{3.4}$$

ただし，

$$E(C_i) = \sum_{x \in C_i} (sim(x, c_i))^2 \tag{3.5}$$

最短距離法，最長距離法，および群平均法は任意の対象完の距離 $sim(C_1, C_2)$ が与えられている場合に適用できる．対象が属性ベクトルで記述されている場合には属性ベクトル完のユークリッド距離などを求めて適用する．ウォード法は対象が属性ベクトルで与えられている場合にのみ適用できる．

- k-means 法 k-means 法は，分割の良さに関する評価関数を設定しその評価関数を最適にする分割を探索する手法である．クラスタ数 k は事前に分析者が決定する．各クラスタは平均ベクトルなどの代表ベクトルで表現される．

  k-means 法をアルゴリズムの形で表すと次のようになる．

  ステップ 0

  入力：事例集合 $D = \{x^{(1)}, x^{(2)}, \cdots, x^{(N)}\}$

  クラスタ数：k

  無作為に $m_1, m_2, \cdots, m_k$ を決定.

  ステップ 1

  すべての $x^{(i)} \in D$ に対して以下の操作を行う．

  $\forall c, c_max = \arg\max_c sim(x^{(i)}, m_c)$

  $x^{(i)}$ を c_{max} に加える．

  ステップ 2

  各クラスタの代表ベクトルを再計算する．

  $\forall c, m_c = \frac{1}{|c|} \sum_{x^{(i)} \in c} x^{(i)}$

  ステップ 3

  収束するまでステップ 1,2 を繰り返す.

  k-means 法は初期値によって結果が変化するため，適切な初期値を設定することが計算時間短縮，よい解を求めることの双方に重要となる．

例えば，凝集型クラスタリングの結果を初期値として利用する方法などが使われることもある．

- 確率的クラスタリング確率モデルに基づいたクラスタリング手法は，データマイニング，統計，パターン認識の分野で多く研究されている．これは，i 番目のクラスタについて，パラメータ θ_i をもつ対象の確率（密度）分布 $f_i(x|\theta_i)$ と，クラスタの混合比 $\forall\alpha_i > 0, \sum_i^k \alpha_i = 1$ であらわされる次の混合分布を用いる手法である．

$$Pr[x|\theta_1, \ldots, \theta_k] = \sum_i^k \alpha_i f_i(x|\theta_i) \tag{3.6}$$

$f_i(x|\theta_i)$ には，対象の属性が実数地の場合は正規分布，カテゴリ値の場合は多項分布が用いられることが多い．与えられた対象集合 X が，この混合分布に基づき発生する場合の尤度を最大にする最尤推定によりパラメータを求めて混合比と確率分布の積が最大になるクラスタへ各対象を分類する．最尤推定には一般に各クラスタへの確率的割り当てと各クラスタの分布のパラメータ推定を交互に行う EM アルゴリズムが用いられる．

この手法ですべてのクラスタと属性で標準偏差が等しく共分散が単位行列である正規分布を用いた場合と k-means 法では類似した結果が得られ，混合分布を用いた手法は k-means の制約を緩めた手法であると考えることができる．

以下では，$f_i(x|\theta_i)$ に混合正規分布を用いた例について具体的に記述する．

【混合正規分布を用いたクラスタリング】

いま，$f_i(x|\theta_i)$ は正規分布であると仮定する．ただし，分散 σ^2 は既知であり，かつクラスタによって変化しないものとする．よって，簡単のためクラスタ c として $f_i(x|\theta_i) = f_i(x|c)$ と書くことにする．$\boldsymbol{m_c}$ をクラスタ c の平均ベクトルとすると，

$$f_i(x^{(i)}|c) = \frac{1}{\sqrt{(2\pi\sigma^2)^d}} exp(-\frac{|\boldsymbol{x^{(i)}} - \boldsymbol{m_c}|^2}{s\sigma^2}) \tag{3.7}$$

ここで,

$$\begin{aligned} f_i(c|x^{(i)}) &= \frac{P(\boldsymbol{x}, c)}{P(\boldsymbol{x})} = \frac{P(c, \boldsymbol{x^{(i)}})}{\sum_c P(c, \boldsymbol{x^{(i)}})} = \frac{P(c)P(\boldsymbol{x^{(i)}}|c)}{\sum_c P(c)P(\boldsymbol{x^{(i)}}|c)} \\ &= \frac{P(c)exp(-\frac{|\boldsymbol{x^{(i)}} - \boldsymbol{m_c}|}{2\sigma^2})}{\sum_c P(c)exp(-\frac{|\boldsymbol{x^{(i)}} - \boldsymbol{m_c}|}{2\sigma^2})} \end{aligned} \tag{3.8}$$

クラスタ c の代表ベクトル $\boldsymbol{m_c}$ は確率の重みづけで計算することになるから，以下のようになる.

$$\boldsymbol{m_c} = \frac{\sum_{\boldsymbol{x^{(i)}} \in D} f_i(c|\boldsymbol{x^{(i)}})\boldsymbol{x^{(i)}}}{\sum_{\boldsymbol{x^{(i)}} \in D} f_i(c|\boldsymbol{x^{(i)}})} \tag{3.9}$$

以上の計算をアルゴリズムにまとめると以下のようになる.

ステップ 0

入力：ベクトル集合 $D = \{x^{(1)}, x^{(2)}, \cdots, x^{(N)}\}$

クラスタ数：k

無作為に $m_1, m_2, \cdots, m_k$ を決定.

ステップ 1

すべての $x^{(i)} \in D$，c に対して以下の操作を行う.

$$f_i(c|x^{(i)}; \boldsymbol{m'}) = \frac{P(c)exp(-\frac{|\boldsymbol{x^{(i)}} - \boldsymbol{m'_c}|}{2\sigma^2})}{\sum_c P(c)exp(-\frac{|\boldsymbol{x^{(i)}} - \boldsymbol{m'_c}|}{2\sigma^2})} \tag{3.10}$$

ステップ 2

各クラスタの代表ベクトルを再計算する.

$$\boldsymbol{m_c} = \frac{\sum_{\boldsymbol{x^{(i)}} \in D} f_i(c|\boldsymbol{x^{(i)}}; \boldsymbol{m'})\boldsymbol{x^{(i)}}}{\sum_{\boldsymbol{x^{(i)}} \in D} f_i(c|\boldsymbol{x^{(i)}}; \boldsymbol{m'})} \tag{3.11}$$

ステップ 3

収束するまでステップ 1,2 を繰り返す.

今回は簡単のため σ^2 を既知としたが，σ^2 も未知の場合を扱うためには，代表ベクトル$\boldsymbol{m}'$ の再計算に加え，σ^2 も再計算を行う.

相関ルール

相関ルールとは

相関ルールは 1993 年に Agrawal らが提起しデータマイニングの発展の一端を担った．元来はスーパーマーケットでの買い物の傾向を分析して商品の配置計画や販売促進に役立てようとするバスケット分析を指向して提起された手法論であるが，その後一般的なデータ解析にも適用可能な柔軟な手法として広く用いられている．相関ルールは次のような指標を用いて評価される．アイテム集合 $X \subseteq L$ とする．トランザクションとは，(購買データであれば) 一人の顧客が購買したアイテムのリスト D を表す．count(X) は，トランザクション $\boldsymbol{D}$ の中で X を含むものの個数を表す.

- 支持度 (support)
  全トランザクション数に対する，X と Y とを共に含むトランザクション数の比．$Pr[X, Y]$ に相当する.

$$support(X) = \frac{count(X)}{|D|} \tag{3.12}$$

$$support(X \Rightarrow Y) = \frac{count(X \bigcup Y)}{|D|} \tag{3.13}$$

- 確信度 (confidence)
  X を満たすトランザクション数に対する，X と Y を共に含むトランザクション数の比．$Pr[Y|X]$ に相当する.

$$confidence(X \Rightarrow Y) = \frac{count(X \bigcup Y)}{count(X)} = \frac{support(X \bigcup Y)}{support(X)} \tag{3.14}$$

【例】「A を買った人の 80%が B, C も買っておりその 3 種類の商品すべてを買った人は全顧客の 6%である」というとき，各指標は以下のようになる.

$[A] \Rightarrow [B, C]$:$sup = 6\%, conf = 80\%$

最低支持度（minsup）と最低確信度（minsconf）を指定して，データベースからすべての相関ルールを抽出するといった方法が用いられることが多い．

相関ルールの問題点

相関ルールの問題点として以下のような点が挙げられている．

- 相関ルールには統計的な相関がないため抽出したルールは無意味である
- アイテムが密になると，アイテムの組み合わせ数が膨大となりルールの抽出の計算が困難である．また，大量のルールが抽出された場合，それらの視察が実質的に困難になる．

統計的な相関がないという問題に関しては，条件部を空とした場合との確信度の比であるリフト値という新たな指標が考案された．

$$list(X \Rightarrow Y) = \frac{confidence(X \bigcup Y)}{support(Y)} \tag{3.15}$$

また，膨大なルールの計算方法についても効率的な計算方法の工夫が提案されている．

(g)　系列パターンの類似性分析

パタン認識，音声認識，画像認識，ゲノム解析など膨大なデータを扱う多くの問題においてデータ文字列をデータベース中の文字列にマッチングさせる問題は非常に重要な問題である．2 つのパターンの類似度は編集距離や相同性スコアと呼ばれる．

行動データにおいても，活動目的や活動パタンを離散的に表す際にこのような概念を用いて類似度を計算することが試みられている．（活動データの分析で具体的な例を示す.）この方法は，系列データの類似度を計算できるという点で行動データの生成過程のモデリングやシミュレーションにおいても適用可能性は高いと考えられる．

類似度計算におけるマッチングの概念

ここでの類似度の概念は，ある離散的なパターン列 $\boldsymbol{s}$ と $\boldsymbol{t}$ が存在するとき，それらのパターン列類似度が以下の 3 つの操作によって $\boldsymbol{s}$ を $\boldsymbol{t}$ に変換するのに必要な最小コストとして定義される．

- パターン列中の記号 s_i が別の文字 t_j に置換される．
- パターン列中の記号 s_i が脱落する．
- パターン列中の記号 t_j が挿入される．

ここで, 上記の置換, 削除, 挿入の各操作のコストをそれぞれ $c_s(i,j), c_d(i), c_i(j)$ と置くと，位置 (i,j) における類似度 $d(i,j)$ は以下のような再帰式で記述することができる．

$$d(i,j) = \min\{d(i-1,j-1) + c_s(i,j), d(i,j-1) + c_i(j) + d(i-1,j) + c_d(i)\} \quad (3.16)$$

このような再帰式を動的計画法を用いて解くことで，最適な操作パターンと類似度を求めることができる．

(h)　因果構造，関係性のモデル (確率的グラフィカルモデル)

確率的グラフィカルモデルとは

本項では，確率分布の図式的な表現である確率的グラフィカルモデルを扱う．確率的グラフィカルモデルでは，確率変数間の関係性をモデル化するため，画像のノイズ除去や修復問題など部分的な観測から全体を復元する問題や，全体を精度よく表現するための効率的な観測などに関して画像処理や行動データの因果推論など広く研究が進められている．われわれが分析対象とする行動データに関しても，確率変数同士関係を持つ場合の推論問題は多く存在し，これらの手法の適用性は高いと考えられる．
確率的グラフィカルモデルとは以下のような特徴を持っている．

- 確率モデルの構造を視覚化する簡単な方法を提供し，新しいモデルの設計方針を決めるのに役立つ．
- グラフ構造を調べることで，条件付き独立性などモデルの性質に関する知見が得られる．
- 推論や学習を行う際，数学的な表現を暗に伴うグラフ上の操作として表現することができる．

グラフはリンクによって接続されたノードの集まりからなる．確率的グラフィカルモデルでは，各ノードが確率変数，リンクがこれらの変数間の確率

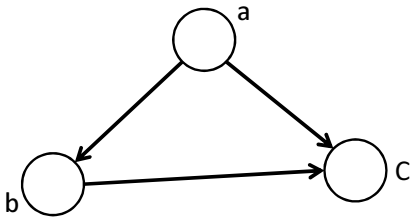


図 3.2 グラフィカルモデルの例

的関係を表現し，グラフは「全確率変数上の同時分布が一部の変数のみに依存する因子の積としてどのように分解可能か」という情報を表現する．

代表的なグラフィカルモデルには，グラフのリンクが特定の方向性をもつベイジアンネットワーク（有向グラフィカルモデル）とリンクが方向性を持たないマルコフ確率場（無向グラフィカルモデル）が存在する．有向グラフは確率変数間の因果関係を表現でき，無向グラフは確率変数間の緩い束縛関係を表現することができる．また，推論問題を解く際には有効グラフや無向グラフを因子グラフと呼ばれる別の表現に変換することが多い．次項からは，先に述べた代表的な 2 つのモデルと推論問題を解く手法について述べる．

ベイジアンネットワーク

まず，簡単な a, b, c 上の任意の同時分布 $p(a, b, c)$ を考えよう．確率の乗法定理を用いることにより，同時分布は

$$\begin{aligned} p(a, b, c) &= p(c|a, b)p(a, b) \\ &= p(c|a, b)p(a|b)p(a) \end{aligned} \tag{3.17}$$

となる．このような分解は任意の同時分布に対して可能である．

このような分解の例を図 3.2 に示した．ノード a からノード b へのリンクが存在するとき，a は b の親ノード，b は a の子ノードと呼ぶ．この例で興味深いのは，左辺は 3 変数 a, b, c に関して対象であるのに右辺は対象でないことである．

次に，K 変数の同時分布 $p(x_1, \ldots, x_K)$ の場合を考えよう．確率の乗法定理を繰り返し適用することにより，同時分布は各変数上の条件付き確率分布

の積として書ける．

$$p(x_1,\ldots,x_K) = p(x_K|x_1,\ldots,x_K)\cdots p(x_2|x_1)p(x_1) \qquad (3.18)$$

K の値を決めれば，この同時分布は K 個のノードをもつ有効グラフとして表現される．有向グラフと対応する（変数の）分布の間の一般的な関係について以下に述べる．グラフによって定義される同時分布はグラフ上で親に対応する変数によって条件づけられた，各ノード変数上の条件付き分布の（全ノード分の）積によって与えられる．よって，K 個のノードを持つグラフに対応する同時分布は以下の式で与えられる．

$$p(\boldsymbol{x}) = \prod_{k=1}^{K} p(x_k|pa_k) \qquad (3.19)$$

ただし，pa_k は x_k の親ノード集合を表し，$\boldsymbol{x}$=$x_1,\ldots,x_K$ を表す．この等式は，与えられた有効グラフィカルモデルに対応する同時分布の分解特性を表現している．

多項式曲線フィッティングの例

ここでは，ベイズ多項式回帰モデルについて考える．このモデルの確率変数は，多項式係数ベクトル $\boldsymbol{w}$ および観測データ $\boldsymbol{t}$=$(t_1,\ldots,t_N)^T$ である．このとき，同時分布は事前分布 p($\boldsymbol{w}$) と $\boldsymbol{N}$ 個の条件付き分布 $\boldsymbol{p(t_n|\ w)}$ $\boldsymbol{(n = 1,\ldots,N)}$ の積によって与えられる．

$$\boldsymbol{p(t,w) = p(w)\prod_{n=1}^{N} p(t_n|w)} \qquad (3.20)$$

モデルパラメータを陽に書く場合は次のようになる．

$$\boldsymbol{p(t,w|x,\alpha,\sigma^2) = p(w|\alpha)\prod_{n=1}^{N} p(t_n|w,x_n,\sigma^2)} \qquad (3.21)$$

ただし，$\boldsymbol{\alpha}$ はガウス事前分布の精度を表すハイパーパラメータ，$\boldsymbol{\sigma^2}$ はノイズの分散である．

マルコフ確率場

有向グラフィカルモデルは，全変数集合上の同時分布を局所的な条件付き

分布の積に分解することが可能なモデルであった．無向グラフであるマルコフ確率場においても，因数分解および条件付き独立性が規定される．

ここで，1 つのリンクによって直接接続されない 2 つのノード $\boldsymbol{x_i}$ および $\boldsymbol{x_j}$ について考える．これらの変数は，グラフ上の他のすべてのノードが与えられたもとで，条件付き独立でなければならない．この条件付き独立性は以下のように表現される．

$$\boldsymbol{p(x_i, x_j | x_{\setminus\{i,j\}}) = p(x_i | x_{\setminus\{i,j\}}) p(x_j | x_{\setminus\{i,j\}})} \tag{3.22}$$

ここで，$\boldsymbol{x_{\setminus\{i,j\}}}$ はすべての変数集合 $\boldsymbol{x}$ から $\boldsymbol{x_i}$ および $\boldsymbol{x_j}$ を取り除いたものである．したがって，この条件付き独立性がグラフに属するすべての可能な分布について成り立つためには，$\boldsymbol{x_i}$ と $\boldsymbol{x_j}$ とが同じ因子に含まれないように同時分布が因数分解される必要がある．

このようなことから，クリークと呼ばれるグラフ上の概念を導入すると便利である．クリークの定義は，すべてのノードの組にリンクが存在するようなグラフの部分集合である．さらに，極大クリークとは，もう 1 つノードを加えるとクリークでなくなってしまうようなクリークとして定義される．

クリークを $\boldsymbol{C}$ と書き，クリーク内の変数集合を$\boldsymbol{x_C}$と書く．すると同時分布は，グラフの極大クリーク上のポテンシャル関数の積の形で書ける．

$$\boldsymbol{p(x) = \frac{1}{Z} \prod_C \psi_c(x_C)} \tag{3.23}$$

ここで，$\boldsymbol{Z}$ は規格化定数（分配関数）であり，以下で与えられる．

$$\boldsymbol{Z = \sum_x \prod_C \psi_c(x_C)} \tag{3.24}$$

ポテンシャル関数は，周辺分布や条件付き確率分布のように確率的解釈が可能なものに限定されない．したがって，一般にそれらの積は正しく規格化されない．よって規格化定数が陽に組み込まれているのである．

有効グラフの場合は，因子分解の各因子である条件付き確率分布が規格化されていたため，同時分布は自動的に規格化されていた．無向グラフでは，$\boldsymbol{M}$ 個の $\boldsymbol{K}$ 状態離散変数ノードをもつモデルの場合，規格化因子を計算するためには $\boldsymbol{K_M}$ 個の状態に対する足し算が必要であり，計算量は最悪の場合モ

デルのサイズに対して指数的に増大する．ただし，分配関数はポテンシャル関数 $\boldsymbol{\psi_c(x_C)}$ を支配する関数でありパラメータを学習する際には計算が必要となるが，局所的な条件付き分布の計算には不要である．なぜなら，条件付き分布は 2 つの周辺確率の比であり比を計算する際には分配関数はキャンセルされるからである．

ここで，無向グラフの条件付き独立性と因数分解との関係について考えよう．これを定式化するためには，ポテンシャル関数 $\boldsymbol{\psi_c(x_C)}$ が狭義に正（任意の $\boldsymbol{x_C}$ に対して 0 でも負でもない）場合に制限する必要がある．この制限のもとでは，因数分解と条件付き独立性との間の精密な関係を定義することができる．

ポテンシャル関数は協議に正に限ったので，指数関数を用いて以下のように表す．

$$\boldsymbol{\psi_c(x_C) = exp\{-E(x_C)\}} \tag{3.25}$$

ここで，$\boldsymbol{E(x_C)}$ はエネルギー関数と呼ばれこの指数関数表現はボルツマン分布と呼ばれる．同時分布はポテンシャル関数の積で与えられるので全エネルギーは極大クリークごとのエネルギーの和として与えられる．先にも述べたように，有向グラフの同時分布は確率分布の積に分解されたのに対し無向グラフの因子であるポテンシャル関数は確率的解釈を持たない．規格化制約がないのでポテンシャル関数を自由に選ぶことができる．

有向グラフモデルと無向グラフモデルの関係

まず，有効フラフで記述されるモデルを無向グラフに変換する問題を考えよう．単純なグラフの場合であればこの変換は容易に実行できる．図　の例を考えてみる．有向グラフの同時分布は，条件付き分布の積として以下の形に分解できる．

$$\mathrm{p}(\boldsymbol{x}) = \boldsymbol{p(x_1)p(x_2|x_1)p(x_3|x_2)\cdots p(x_N|x_{N-1})} \tag{3.26}$$

これを無向グラフによる表現に変換してみよう．この無向グラフにおける極大クリークは単なる隣接ノード対であるので同時分布は以下のように書ける．

$$\mathrm{p}(\boldsymbol{x}) = \frac{\boldsymbol{1}}{\boldsymbol{Z}}\boldsymbol{\psi_{1,2}(x_1,x_2)\psi_{2,3}(x_2,x_3)\cdots\psi_{N-1,N}(x_{N-1},x_N)} \tag{3.27}$$

したがって，容易に以下のような対応付けができる．

$$\begin{aligned}\boldsymbol{\psi_{1,2}(x_1, x_2) = p(x_1)p(x_2|x_1)}\\ \boldsymbol{\psi_{2,3}(x_2, x_3) = p(x_3|x_2)}\\ \vdots\\ \boldsymbol{\psi_{N-1,N}(x_{N-1}, x_N) = p(x_N|x_{N-1})}\end{aligned} \tag{3.28}$$

(3.29)

一般に，有効グラフを無向グラフに変換するには以下のようにすればよい．

1. グラフの各ノードに対してそのすべての親同士の対に無向リンクを付加する．さらにもともとのリンクから矢印の方向性を取り除いてモラルグラフを作る．
2. モラルグラフのすべてのクリークポテンシャル関数を 1 に初期化する．
3. もともとの有向グラフの条件付き分布因子を 1 つ取ってきて，対応するクリークポテンシャルの 1 つにかける．モラル化を行ったため，各条件付き分布因子の変数をすべて含む極大クリークが少なくとも一つ必ず存在する．

グラフィカルモデルの枠組みは，有効リンクと無向リンクを両方持つグラフにも矛盾なく拡張でき，そのようなグラフを連鎖グラフと呼ぶ．

推論問題

ここでは，グラフィカルモデルにおける推論の問題を考える．これは，グラフのいくつかのノードが観測値に固定されたとき，残ったノード（のうちいくつか）に関する事後分布を計算したい，という問題である．

効率の良い推論アルゴリズムを見つけたりそれらのアルゴリズムの構造の見通しを良くする際にグラフの「構造」を利用することができる．具体的には，多くのアルゴリズムが局所的なメッセージのグラフ全体にわたる伝播として表現できる．本節では，木構造グラフにおける厳密推論のためのアルゴリズムについて述べる．

積和アルゴリズム

積和アルゴリズムとは，ノードあるいはノード部分集合上の局所的な周辺分布を計算する手法である．次節で述べる max-sum アルゴリズムはこの手法の修正したものであり，最も確からしい状態を見つけるのに用いられる．

以下の議論では，モデルの持つすべての変数は離散的であるとする．よって周辺化は和演算に対応するが，ここで得られる枠組みは周辺化が積分演算によって得られる線形ガウスモデルにも同様に適用可能である．ループのない有向グラフにおける厳密推論のためのアルゴリズムとして確率伝播法が存在するが，これは積和アルゴリズムの特別な場合と考えてよい．ここでの目的は以下の 2 つである．

1. 周辺分布を求めるための効率よい推論アルゴリズムを得る．
2. 複数の周辺分布を計算したい場合に，計算の重複をなくして効率化する．

初めに，ある特定の変数ノード $\boldsymbol{x}$ 上の周辺分布 $\boldsymbol{p(x)}$ を求める問題から考える．今，すべての変数が隠れている（観測されていない）と仮定する．現実には，一部の変数が観測されたもとでそれらの観測値に条件づけられた事後分布を計算した場合が多いが，拡張は容易である．

ステップ 1

変数ノード $\boldsymbol{x}$ を因子グラフの根ノードとみなし，以下の式を用いてグラフの葉ノード $\boldsymbol{f}$ におけるメッセージを初期化する．

$$\boldsymbol{\mu_{x\longrightarrow f}(x) = 1} \tag{3.30}$$

$$\boldsymbol{\mu_{f\longrightarrow x}(x) = f(x)} \tag{3.31}$$

ステップ 2

メッセージパッシングが以下の式で再帰的に適用される．

$$\begin{aligned}
&\mu_{f_s \longrightarrow x}(x) \\
&= \sum_{x_1} \cdots \sum_{x_M} f_s(x, x_1, \ldots, x_M) \prod_{m \in ne(f_s) \backslash x} [\sum_{X_s m} G_m(x_m, X_s m)] \\
&= \sum_{x_1} \cdots \sum_{x_M} f_s(x, x_1, \ldots, x_M) \prod_{m \in ne(f_s) \backslash x} \mu_{x_m \longrightarrow f_s}(x_m) \quad (3.32)
\end{aligned}$$

$$\begin{aligned}
\mu_{x_m \longrightarrow f_s}(x_m) &= \prod_{l \in ne(f_s) \backslash x} [\sum_{X_m l} F_l(x_m, X_m l)] \\
&= \prod_{l \in ne(f_s) \backslash x} \mu_{f_l \longrightarrow x_m}(x_m) \qquad (3.33) \\
& \qquad (3.34)
\end{aligned}$$

ステップ 3

根ノードがすべての隣接ノードからのメッセージを受け取ったら，求める周辺分布は以下の式を用いて計算される．

$$p(x) = \sum_{s \in ne(x)} F_s(x, X_s) \qquad (3.35)$$

max-sum **アルゴリズム**

積和アルゴリズムを用いれば, 因子グラフによって表現される同時分布 $p(x)$ に関して，その成分変数上の周辺分布を効率よくとくことができる．これ以外に 1) 確率が最大となる変数の組を見つける，2) その時の確率値を求める，という 2 つのタスクが必要になる場合がある．このタスクを実現するための手法が max-sum アルゴリズムであり，積和アルゴリズムと深く関係するアルゴリズムによって実現される．なお，max-sum アルゴリズムは動的計画法のグラフィカルモデルへの応用と考えることもできる．

ここで考えたいタスクは，同時分布を最大にするベクトル x^{max} を求める問題として以下のように定義できる．

$$x^{max} = \arg\max_{x} p(x) \qquad (3.36)$$

このときの同時確率値は，次の式で与えられる．

$$\boldsymbol{p(x^{max})} = \max_{\boldsymbol{x}} \boldsymbol{p(x)} \tag{3.37}$$

まずは，単純に成分に関する最大値演算を書き出してみると次のようになる．

$$\max_{\boldsymbol{x}} \boldsymbol{p(x)} = \max_{\boldsymbol{x_1}} \cdots \max_{\boldsymbol{x_M}} \boldsymbol{p(X)} \tag{3.38}$$

ただし，$\boldsymbol{M}$ は変数の総数である．次に，同時分布を因子の積で表現して $\boldsymbol{p(x)}$ に代入する．ここで，最大値演算に関する以下の法則を用いると，積演算と和演算が交換可能になる．

$$\boldsymbol{\max(ab, ac) = a \max(b, c)} \tag{3.39}$$

対数をとって，以下のように書ける．

$$\boldsymbol{\ln(\max_{x} p(x)) = \max_{x} \ln p(x)} \tag{3.40}$$

3.1.1　行動データの正規化

人の行動を分析する際トリップに区分してから考えることが一般的である.PP 調査でもパーソントリップ調査と比較するためにトリップ単位で集計することが多い. 移動の状態を分析する上で基礎となる情報としては、1) 移動・滞在状態を分けること、2) 移動時にどの交通機関を利用しているか、3) 移動時どの経路を移動しているかなどが大きな要素となる. 例えば図 3.3 の時間と位置の生データが得られているとする. 図 3.4 は図 3.3 を平面上にプロットしたものである．この状態ではラベルづけされていないため、移動・滞在、交通機関がわからない. また取得データには誤差が含まれているため、正確にどの経路を移動したのかはわからない. 従って図 3.5 のようにまず移動、滞在識別を行うことで、トリップを作成する. 次に図 3.6 のように移動中の交通機関を識別する. またトリップ全体でどの経路を利用したかを図 3.7 のように特定する.

(a)　移動滞在識別

GPS データを移動滞在識別に用いる場合、そのままでは移動中の点 (移動点) なのか、滞在中の点 (滞在点) なのか不明である. そこで 1) 前の時刻までの測位店重心からの距離を計算して一定距離以上離れたら移動と判定する方

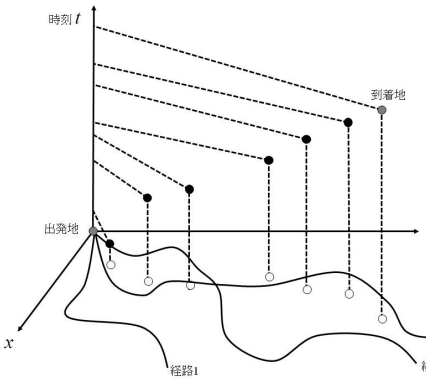


図 3.3 タイムスペース図

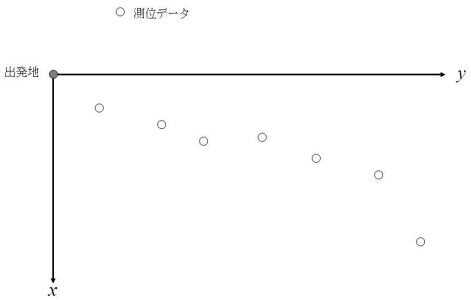


図 3.4 平面上にプロットした図

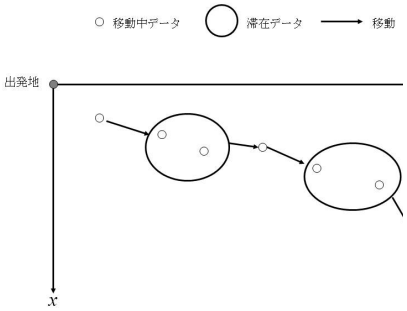


図 3.5 移動滞在識別

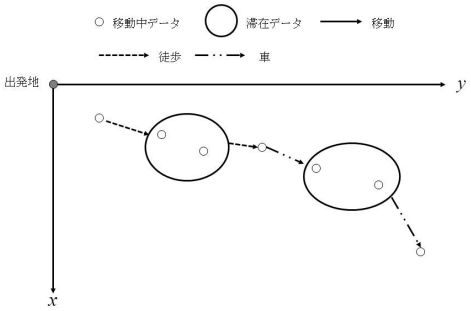


図 3.6 交通機関識別

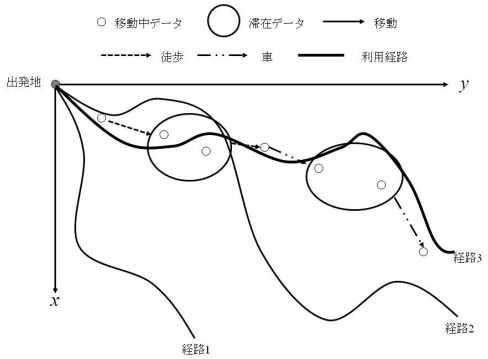


図 3.7　交通機関識別

法や、2) 測位点間の速度を計算して一定速度以上であれば移動と判定する手法によって移動滞在の識別を行うことができる. ここでは距離を用いた移動滞在識別手法について簡単に説明する [1].

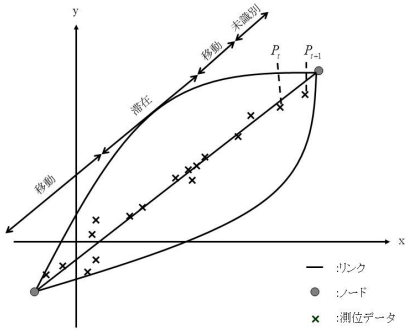


図 3.8　移動滞在識別

取得された GPS データが 2 次元座標上にあり、図 3.8 のように時間的に連続する 2 つの位置データ $\boldsymbol{P_t}$、$\boldsymbol{P_{t+1}}$ があるとする. このときデータ間の距離は式 (3.41) のようになる.

$$d = \sqrt{(x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2} \tag{3.41}$$

この距離があらかじめ与えた閾値 D 内にあるとき、時刻 t $t+1$ の間に移動はないものとみなす. 時刻 t までの識別が終わっているとして、時刻 $t+1$ の識別を考える. 時刻 $t+1$ の点識別は、時刻 t の点が滞在点であるか、移動点であるかによって異なる. また時刻 t が滞在点であることは時刻 $t-1$ と時刻 t の条件によって決められるが、移動点であるかどうかは時刻 t と $t+1$ との関係を計算しない限り決まらない. 従って、移動点には暫定的な移動点と、確定した移動点の 2 種類があることになる. 以下にそれぞれの場合のアルゴリズムを示す.

時刻 t の点が滞在の場合

ステップ 1

時刻 t 以前の時間的に連続する N 個の点 (時刻 ($i = t-N+1, .., .t$)) は滞在点であるとする. それらの点の重心位置 $(\bar{x}, \bar{y})$ と識別対象である時刻 $t+1$ の点 (x_{t+1}, y_{t+1}) との距離を式 (3.42) により求める.

$$\begin{aligned} \bar{x} &= \frac{1}{N} \sum_{i=t-N+1}^{t} x_i \\ \bar{y} &= \frac{1}{N} \sum_{i=t-N+1}^{t} y_i \end{aligned} \tag{3.42}$$

ステップ 2-1

距離が閾値 D 以内なら時刻 $t+1$ の点は時刻 t までの点と同じ位置での滞在点とする. 平均値、及び、連続する点の数を更新し $N = N+1$ とする.

ステップ 2-2

距離が閾地を越えているなら、時刻 $t+1$ の点を暫定的な移動点とする.

時刻 t の点が暫定的移動点の場合

ステップ 1

時刻 t の点と時刻 $t+1$ の距離を式 (3.41) により求める.

ステップ 2-1

距離が閾値 $\boldsymbol{D}$ 以内なら時刻 $\boldsymbol{t}$ の点と時刻 $\boldsymbol{t+1}$ の点はいずれも同一地点での滞在点である. このとき 2 点の平均位置座標を求めておく.

ステップ 2-2

距離が閾値 $\boldsymbol{D}$ 以上なら時刻 $\boldsymbol{t}$ の点と時刻 $\boldsymbol{t+1}$ の点は異なる位置の点であり、時刻 $\boldsymbol{t}$ の点は真の移動点とし、時刻 $\boldsymbol{t+1}$ の点は暫定的な移動点とする.

補足制約

上記のルール以外にも時間によって移動滞在を分ける以下の制約条件を設ける.

1: 滞在時間制約

短時間の滞在を除外する. 同一地点に 5 分以上滞在しない限り、滞在とはみなさない.

2: 移動時間制約

短時間トリップを除外する. 移動時間が 5 分以上でなければトリップとみなさない.

以上のような流れで測位データを下に移動滞在判別をすることでトリップ候補を自動的に作成することができる.

(b)　交通機関識別

移動滞在を識別し、トリップ候補の作成が完了したのち交通機関の自動識別を行う. 交通機関自動識別の手法するために扱うデータとしては 1)GPS データによる識別 2) 加速度データによる識別などがある.1) の GPS による交通機関識別では移動滞在識別のときと同様に閾値を各交通機関ごとに閾値を設けることで識別を行う.2) の、加速度データによる識別では交通機関ごとに現れる特徴を利用して識別を行う. それぞれの識別手法についてについて詳しく説明していく. またここでは”滞在”、”徒歩”、”自転車”、”車”の 4 交通機関識別について扱うこととする.

GPS データによる識別

GPS データによる識別では GPS データから各点の間の移動速度に交通機関

に閾値を設けて交通機関識別を行う. 各交通機関ごとに速度の発生確率に特徴があるのでそれをもとに識別を行う. 以下に具体的なアルゴリズムを示す.

ステップ 1

最も識別しやすい徒歩トリップから識別を開始し、ステップ 1 へ戻ってきたら、自転車→車の順でトリップを抽出する.

ステップ 2

移動滞在識別により作成された各トリップ候補について各測位データ間の移動速度が一定値以下になるまでトリップを再分割.

ステップ 3

ステップ 2 終了時にトリップ開始時刻、終了時刻、交通機関を確定する. 残りのデータでステップ 1 へ戻り、車まで終われば終了.

加速度データによる識別

加速度による識別も各交通機関ごとの加速度の特徴をもとに識別を行う。図 3.16、3.17、3.18 は交通機関ごとの 3 軸方向 (進行方向、左右方向、鉛直方向) 加速度の推移を表わしたものである. 徒歩、自転車、車の順に 3 軸加速度の分散が大きいのがわかる. このように交通機関ごとに異なった加速度の特徴をもつので、これを利用して識別を行う. ここでは SVM、HMM を用いた交通機関識別の流れについて説明する.

1: SVM(サポートベクターマシン)

SVM は 2 クラス識別問題を扱うための手法であり、文字認識や音声認識などに幅広く用いられている.SVM の特徴としては主に以下の 3 点が挙げられる.

1. マージン最大化を基準とすることで、高い判別性能を実現.
2. カーネル関数を用いた柔軟なモデリング.
3. パラメータ推定が凸 2 次最適化問題として定式化されるため、高速な最適化アルゴリズムが利用可能.

式 (3.43) で表わされる線形モデルの 2 値分類問題を解くことでその特徴について説明を行う.

$$\boldsymbol{y}\left(\mathbf{x}\right) = \mathbf{w}^{\mathbf{T}}\boldsymbol{\phi}\left(\mathbf{x}\right) + \mathbf{b} \tag{3.43}$$

ここで $\boldsymbol{\phi}(\mathbf{x})$ はある固定された特徴空間変換関数である。訓練データは $\mathbf{x_1}, ..., \mathbf{x_N}$ の $\boldsymbol{N}$ 個の入力ベクトルとそれに対応した出力値 $\boldsymbol{t_1}, ..., \boldsymbol{t_N} \in \{-1, 1\}$ からなり、未知データ $\mathbf{x}$ は $\boldsymbol{y}(\mathbf{x})$ が正か負のどちらかによって分類が行われる. つまり式 (3.44) が成立する.

$$\begin{aligned} \boldsymbol{t_n} = +1 \rightarrow \boldsymbol{y}\left(\mathbf{x_n}\right) > 0 \\ \boldsymbol{t_n} = -1 \rightarrow \boldsymbol{y}\left(\mathbf{x_n}\right) < 0 \end{aligned} \tag{3.44}$$

したがって必ず式 (3.45) が成立する.

$$\boldsymbol{t_n y}\left(\boldsymbol{x_n}\right) > 0 \tag{3.45}$$

今、図 (3.9) のように訓練データを正確に分類する解が $\boldsymbol{y_1}$、$\boldsymbol{y_2}$ のように複数存在する場合、汎化誤差が最も小さくなるような解を求めることが最も望ましい.SVM では、マージンという概念を用いてそのような解を求める. ここでマージンとは境界面と訓練データの間の最短距離を表わす. このマージンを最大化する分類境界を選択する. 分類境界面 $\boldsymbol{y}(\mathbf{x}) = 0$ から点 $\mathbf{x}$ までの距離は $|\boldsymbol{y}\left(\mathbf{x}\right)|/\|\mathbf{w}\|$ で表わされる. また今式 (3.45) がすべての $\boldsymbol{n}$ について成り立つので、分類境界から点 $\boldsymbol{x_n}$ までの距離は式 (3.46) のように表わされる.

$$\frac{\boldsymbol{t_n y}\left(\boldsymbol{x_n}\right)}{\|\mathbf{w}\|} = \frac{\boldsymbol{t_n}\left(\mathbf{w^T}\boldsymbol{\phi}\left(\mathbf{x_n}\right) + \mathbf{b}\right)}{\|\mathbf{w}\|} \tag{3.46}$$

今求めたいのはマージンを最大化するパラメータ $\mathbf{w}$ と $\boldsymbol{b}$ である. 従って、解は式 (3.47) を解くことで得られる.

$$\underset{\mathbf{w,b}}{\mathbf{arg}}\ \mathbf{max}\left\{\frac{\mathbf{1}}{\|\mathbf{w}\|}\ \mathbf{min}\left[\boldsymbol{t_n}\left(\mathbf{w^T}\boldsymbol{\phi}\left(\mathbf{x_n}\right) + \mathbf{b}\right)\right]\right\} \tag{3.47}$$

次に直接この問題を解くのは複雑なので、より簡単な形へ変形することを考える. まずパラメータを $\mathbf{w} \rightarrow \boldsymbol{\kappa}\mathbf{w}$、$\boldsymbol{b} \rightarrow \boldsymbol{\kappa b}$ のように定数倍しても、点 $\boldsymbol{x_n}$ から分類境界までの距離は変化しない. 従って定数倍することで、境界に最も近い点について式 (3.48) が成立する.

$$\boldsymbol{t_n}\left(\mathbf{w^T}\boldsymbol{\phi}\left(\mathbf{x_n}\right) + \mathbf{b}\right) = \mathbf{1} \tag{3.48}$$

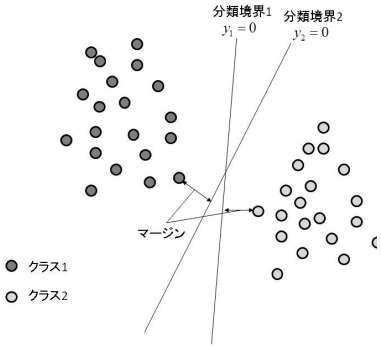


図 3.9 マージンの概念

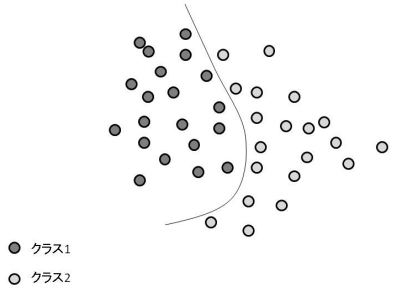


図 3.10 訓練データが線形分離不可能な場合

このスケーリングの下ではすべてのデータについて制約式 (3.49) が成り立つ.

$$\boldsymbol{t_n}\left(\mathbf{w^T}\boldsymbol{\phi}\left(\mathbf{x_n}\right)+\mathbf{b}\right)\geq\mathbf{1} \tag{3.49}$$

このとき式 (3.47) は式 (3.50) の最適化問題に帰着する.

$$\underset{\mathbf{w,b}}{\mathbf{arg\ min}}\ \frac{\mathbf{1}}{\mathbf{2}}\left\|\mathbf{w}\right\|^{\mathbf{2}} \tag{3.50}$$

ここまでの議論では、訓練データが特徴空間 $\boldsymbol{\phi}\left(\boldsymbol{x}\right)$ において線形分離可能であり、そこから得られる SVM はもともとの入力空間 $\mathbf{x}$ において訓練データを完全に分離すると仮定してきた. しかし、図 3.10 のように実際にはそのような仮定が満たされないことも多い. このような状況に対応するためには、一部の訓練データの誤分類を許すように SVM を修正する必要がある. 誤って分類したデータについては無限大のペナルティを与え、正しく分類したデータにはペナルティを与えない誤算関数を用いる. この誤差関数を、データ点がマージン内に侵入した場合にはマージン境界からの距離に応じたペナルティを与えることで、マージン境界の間違った側に存在することを許すように定式化を修正する. 具体的にはまずスラック変数 $\boldsymbol{\xi_n}\geq\mathbf{0}\ \ \left(\boldsymbol{n}=\mathbf{1},...,\boldsymbol{N}\right)$ を導入する. スラック変数は各訓練データごとに定義される変数で、データが正しく分類されかつマージン境界の上または外側に存在する場合は $\boldsymbol{\xi}=\mathbf{0}$、それ以外の場合には $\boldsymbol{\xi_n}=\left|\boldsymbol{t_n}-\boldsymbol{y}\left(\boldsymbol{x_n}\right)\right|$ で与えられる. 従って、ちょうど分

類境界 $\boldsymbol{y = (x)}$ 上にあるデータについては $\boldsymbol{\xi_n = 1}$、誤分類されたデータについては $\boldsymbol{\xi > 1}$ が成り立つ. このスラック変数を用いて識別関数式 (3.49) を式 (3.51) のように修正する.

$$\boldsymbol{t_n y\,(x_n) \geq 1 - \xi_n} \tag{3.51}$$

ただし、$\boldsymbol{\xi \geq 0}$ である. 図 3.11 のように $\boldsymbol{\xi = 0}$ が成り立つデータは正しく分類されており、かつマージン境界上、あるいは正しい側に存在する. また $\boldsymbol{0 < \xi_n \leq 1}$ となるデータ点はマージン内部にあるが正しく分類されている. そして $\boldsymbol{\xi > 1}$ となるデータ点は分類境界に対して誤った側にあり、従って誤分類されている. これはハードマージンの制約のソフトマージンへの緩和といわれることがある. このとき式 (3.50) に対する最小化すべき最適化問題は式 (3.52) のようになる.

$$\boldsymbol{C \sum_{n=1}^{N} \xi_n + \frac{1}{2} \|\mathbf{w}\|^2} \tag{3.52}$$

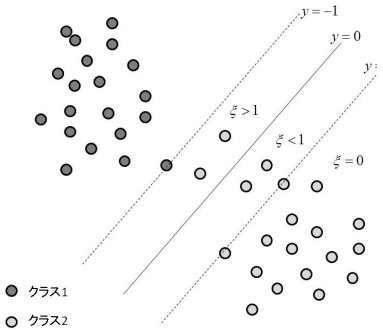


図 3.11　ソフトマージン

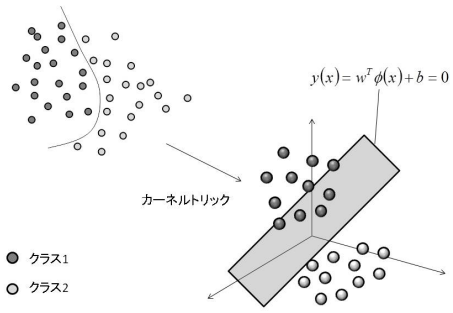


図 3.12　カーネルトリック

ここで $\boldsymbol{C > 0}$ はスラック変数を用いて表わされるペナルティとマージンの大きさ間のトレードオフを制御するパラメータである.$\boldsymbol{C \to \infty}$ の極限においては 分離可能なデータに対するハードマージン問題と同じになる. 以上のようなソフトマージンを用いることで、線形分離不可能な訓練データに対

しての識別も可能になる. しかし、本質的に非線形で複雑な場合、このソフトマージンを用いてもうまく訓練データを識別することができない. そこでカーネルトリックという手法を用いることでこの問題を解決する. 非線形な写像 $\boldsymbol{\Phi}$ を用いて入力空間をより高次の空間に写像し、写像先の空間で線形分離を行うことで分離を容易にする. 写像 $\boldsymbol{\Phi}$ を用いた計算は本来であれば元の次元より高次元でのベクトル計算を行う必要があるが、カーネルトリックを用いることで高次元でのベクトル計算を避けることが可能となる. カーネルトリックとは写像 $\boldsymbol{\Phi}$ に関する計算 $\boldsymbol{\Phi}(\boldsymbol{x}) \cdot \boldsymbol{\Phi}(\boldsymbol{x'})$ をカーネル関数 $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x'})$ を用いて置き換えることで写像 $\boldsymbol{\Phi}$ の球解を避け、カーネル関数のみの計算に変換することである. カーネル関数は複数ある. ここでは代表的なカーネルの一つである $\boldsymbol{RBF(RadialBasisFunction)}$ カーネルを式 (3.53) に示す.

$$\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x'}) = \exp\left(-\gamma \times |\boldsymbol{x} - \boldsymbol{x'}|^2\right) \tag{3.53}$$

ここで、$\boldsymbol{x}$、$\boldsymbol{x'}$ はベクトル、γ はパラメータを表わし、$\boldsymbol{RBF}$ カーネルを用いる場合はパラメータ γ に適切な値を与える必要がある.

SVM を用いた交通機関識別

SVM は訓練あり学習を用いる識別手法の一つである. すなわち訓練データから識別クラスごとの特徴量を学習することで、未知データの識別行う. 具体的には交通機関ごとの 3 軸方向加速度データ (それぞれの方向について最大、最小、合成、平均加速度データが取得されている) から交通機関ごとの特徴を学習することで識別を行う. 表 3.2 に、入出力データを、図 3.13 に交通機関識別のイメージを示す. **2: HMM(隠れマルコフモデル)**

表 3.1 交通機関識別入出力データ

入力データ	3 軸方向加速度 (最大、最小、合成、平均) 運動強度 歩数
出力データ	交通機関 (滞在、徒歩、自転車、車など)

HMM(隠れマルコフモデル) は不確定な時系列データをモデル化するための有効な統計的手法である. ここでは HMM の概要とモデルの推定、最適状態系列の推定の説明を行う [2]、[3]。HMM は出力シンボルによって一意に状態

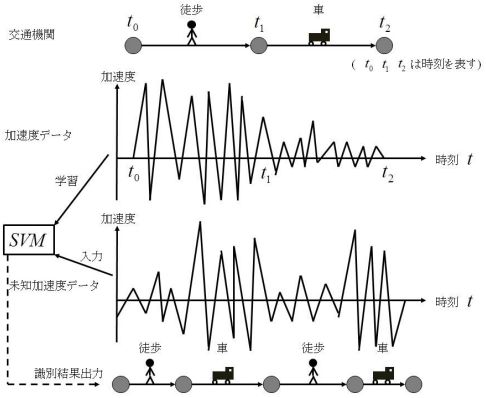


図 3.13　サポートベクターマシン交通機関識別イメージ

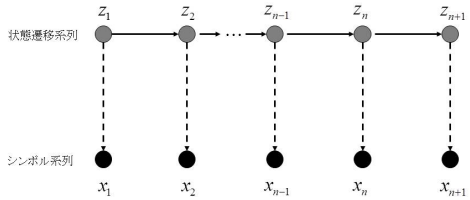


図 3.14　HMM イメージ図

遷移先が決まらないという意味での非決定性確率有限オートマトンとして定義される. 出力シンボル系列が与えられても状態遷移系列は唯一には決まらない. 観測できるのはシンボル系列だけであることから,”隠れ”マルコフモデルと呼ばれる. 図 3.14 に HMM のイメージを示す.

HMM はパラメータとして状態遷移確率、シンボル出力確率、初期状態確率を持つ. シンボル出力確率の計算方法によって離散型 HMM と連続分布型 HMM に分かれる. またシンボル出力確率が状態で出力される Moore マシン

と状態遷移で出力される Mealy マシンに分類できる. 以下では Mealy タイプの離散型 HMM について述べる. ただし、Moore タイプと Mealy タイプは相互に変換可能である. 説明で用いる変数を表 3.2 に定義する. ここで HMM の

表 3.2 変数の定義

$\boldsymbol{T}$	観測系列の長さ
$\boldsymbol{o_1, o_2, ..., o_T}$	観測系列
$\boldsymbol{N}$	状態数
$\boldsymbol{L}$	観測シンボル数
$\boldsymbol{S = \{s\}}$	状態集合
$\boldsymbol{s_t}$	時刻 $\boldsymbol{t}$ における状態 (番号)
$\boldsymbol{i, j}$	状態番号
$\boldsymbol{v = \{v_1, v_2, ..., v_L\}}$	出力可能なシンボル集合

要素を正確に定式化し、観測系列がどのように生成されるか説明を行う.HMM は以下のように特徴づけられる.

1. モデル (実際の状態) の数 $\boldsymbol{N}$. それぞれの隠れた実際の状態をそれぞれ $\boldsymbol{S = \{S_1, S_2, \cdots, S_N\}}$ と表わし、時刻 $\boldsymbol{t}$ における状態を $\boldsymbol{q_t}$ と表わす.
2. 観測シンボルの数 $\boldsymbol{M}$. これはモデル化されたシステムが出力する状態を表わす. それぞれの観測シンボルを $\boldsymbol{V = \{v_1, v_2, \cdots, v_N\}}$ と表わす.
3. 状態遷移確率分布 $\boldsymbol{A = \{a_{ij}\}}$. ここで $\boldsymbol{a_{ij}}$ は式 (3.54) のようになる.

$$\boldsymbol{a_{ij} = P\left[q_{t+1} = S_j | q_t = S_i\right] \quad 1 \leq i,\ j \leq N} \tag{3.54}$$

ある状態からすべての状態へ遷移できるモデルを想定する場合すべての $\boldsymbol{i}$、$\boldsymbol{j}$ について $\boldsymbol{a_{ij}}$ である. ある状態から特定の状態にしか遷移できないモデルであれば、$\boldsymbol{a_{ij} = 0}$ となる場合がある.

4. 状態 $\boldsymbol{j}$ のときの観測シンボル確率分布 $\boldsymbol{B = \{b_j\,(k)\}}$ ただし式 (3.55) が成り立つ.

$$\begin{aligned} \boldsymbol{b_j\,(k) = P\left[v_k\ at\ t | q_t = S_j\right]} \quad & \boldsymbol{1 \leq j \leq N} \\ & \boldsymbol{1 \leq k \leq M} \end{aligned} \tag{3.55}$$

5. 初期状態確率 $\boldsymbol{\pi = \{\pi_i\}}$ ただし、式 (3.56) が成り立つ.

$$\boldsymbol{\pi_i = P\left[q_1 = S_i\right], \quad 1 \leq i \leq N} \tag{3.56}$$

$\boldsymbol{N}$、$\boldsymbol{M}$、$\boldsymbol{A}$、$\boldsymbol{B}$ と $\boldsymbol{\pi}$ が与えられたとき、HMM が式 (3.57) のような観測系列を生成したとする.

$$\boldsymbol{O = O_1 O_2 \cdots O_T} \tag{3.57}$$

それぞれの観測シンボル $\boldsymbol{O_t}$ は $\boldsymbol{V}$ の一つであり、$\boldsymbol{T}$ は観測系列の数を表わす. このとき式 (3.57) で表わされる観測系列が生成される過程は以下のようになる.

過程 1

初期状態確率 $\boldsymbol{\pi}$ に従って初期状態 $\boldsymbol{q_1 = S_i}$ を選択する

過程 2

$\boldsymbol{t = 1}$ とする

過程 3

状態 $\boldsymbol{S_i}$ のときの観測シンボル確率分布 $\boldsymbol{b_i(k)}$ に従って $\boldsymbol{O_t = v_k}$ とする.

過程 4

状態 $\boldsymbol{S_i}$ のときの状態遷移確率分布 $\boldsymbol{a_{ij}}$ に従って新しい状態 $\boldsymbol{q_{t+1} = S_j}$ に遷移する.

過程 5

もし、$\boldsymbol{t < T}$ ならば $\boldsymbol{t = t + 1}$ とし、過程 3 へ、そうでなければ過程を終了する

以上の過程は観測系列の生成や観測系列が HMM によって以下に生成されたかに利用される.HMM を規定するためには実際の状態と観測状態の数 $\boldsymbol{N}$、$\boldsymbol{M}$、観測系列、確率値 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{\pi}$ が必要であることがわかる. そこで簡単ために HMM を式 (3.58) のように略記する.

$$\boldsymbol{\lambda = (A, B, \pi)} \tag{3.58}$$

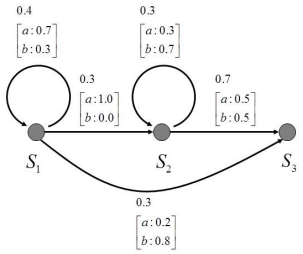


図 3.15 3 状態 left to right HMM

ここで HMM の例として状態遷移が一定方向に進む left to right モデルの例を図 3.15 に示す. 図 3.15 において HMM は 3 つの状態から構成され、2 種類のラベル $\boldsymbol{a}$、$\boldsymbol{b}$ のみからなるラベル系列を出力する. 初期状態確率は $\boldsymbol{\pi_1 = 1.0}$、$\boldsymbol{\pi_2 = 0}$、$\boldsymbol{\pi_3 = 0}$、最終状態を $\boldsymbol{S_3}$ とし、図 3.15 のような遷移のみを行うとする. 図 3.15 において **0.4** など遷移アークに添えられている数字は状態遷移確率を意味し、[] 内の数字の上段はラベル $\boldsymbol{a}$ の出力確率、下段はラベル $\boldsymbol{b}$ の出力確率を意味する. 例えば状態 $\boldsymbol{S_1}$ を例にとると、$\boldsymbol{S_1}$ から $\boldsymbol{S_1}$ 自身に **0.4** の確率で遷移し、遷移の際に確率 0.7 でラベル $\boldsymbol{a}$ を出力し、0.3 の確率で $\boldsymbol{b}$ を出力する. 他の状態、遷移についてもどうようである. ここでラベル系列が **"$\boldsymbol{aab}$"** を出力する確率を考える. この HMM で許される状態系列において **"$\boldsymbol{aab}$"** を出力する可能性のあるのものは、$\boldsymbol{S_1 \to S_1 \to S_2 \to S_3}$、$\boldsymbol{S_1 \to S_2 \to S_2 \to S_3}$、$\boldsymbol{S_1 \to S_1 \to S_1 \to S_3}$ である. それぞれについて **"$\boldsymbol{aab}$"** を出力する確率は以下のようになる.

$$\boldsymbol{S_1 \to S_1 \to S_2 \to S_3} \qquad \boldsymbol{0.4 \times 0.7 \times 0.3 \times 1.0 \times 0.7 \times 0.5 = 0.0294}$$
$$\boldsymbol{S_1 \to S_2 \to S_2 \to S_3} \qquad \boldsymbol{0.3 \times 1.0 \times 0.3 \times 0.3 \times 0.7 \times 0.5 = 0.00945}$$
$$\boldsymbol{S_1 \to S_1 \to S_1 \to S_3} \qquad \boldsymbol{0.4 \times 0.7 \times 0.4 \times 0.7 \times 0.3 \times 0.2 = 0.004704}$$

従ってこの HMM が **"$\boldsymbol{aab}$"** を出力する確率は 3 つの合計で以下のようになる.

$$\mathbf{0.0294 + 0.00945 + 0.004704 = 0.038154}$$

HMM では状態系列に意味を持たないが、最尤の経路を推定することはできる. この例では **"aab"** を出力する可能性が最も高い状態系列は、先ほどの計算より、$\boldsymbol{S_1 \rightarrow Sd_1 \rightarrow S_2 \rightarrow S_3}$ とわかる.

HMM の基本問題

HMM を現実の問題に適用することを想定したとき重要な基本問題として以下の 3 つが挙げられる.

問題 1　観測系列 $\boldsymbol{O = O_1O_2 \cdots O_T}$ とモデル $\boldsymbol{\lambda = (A, B, \pi)}$ が与えられたとき、観測系列が与えられた HMM モデルでどれくらいの確率で観測されるか

問題 2　観測系列 $\boldsymbol{O = O_1O_2 \cdots O_T}$ とモデル $\boldsymbol{\lambda = (A, B, \pi)}$ が与えられたとき、どのようううに最も可能性の高い状態遷移系列を推定するか.

問題 3　モデルパラメータ $\boldsymbol{\lambda(\pi, A, B)}$ をどのように調整するのか.

問題 1 は評価の問題で、つまりモデルと観測系列が与えられたときモデルによって観測系列が生成された確率をどのように計算するかという問題である. あるいは、与えられたモデルが観測された系列にどれくらいマッチしているかという問題ととらえることができる. 問題 2 はモデルの隠れた部分を推定する問題で、つまり正しい系列を探索する問題である. 問題 3 はモデルが最もマッチするよなパラメータを求める問題である. モデルパラメータを修正するための系列を訓練系列と呼ぶこととする. 訓練問題は観測された訓練データから最も適したパラメータを採用する問題なので、HMM を適用する上で重要になってくる.

モデル評価問題

与えられたモデル $\boldsymbol{\lambda}$ において観測系列 $\boldsymbol{O = O_1O_2 \cdots O_T}$、の観測確率、$\boldsymbol{P(O|\lambda)}$ を計算する. 最も単純な方法は、図 3.15 の例で示したように長さ $\boldsymbol{T}$ の系列の状態を数え挙げる方法がある. 今、ある一つの状態系列が式 (3.59) のように表わされるとする.

$$\boldsymbol{Q = q_1q_2 \cdots q_T} \tag{3.59}$$

ここで $\boldsymbol{q_1}$ は初期状態である. 式 (3.59) で表わされる状態系列のとき観測系列 $\boldsymbol{O}$ が観測される確率は式 (**??**) のようになる.

$$\boldsymbol{P}\left(\boldsymbol{O}|\boldsymbol{Q},\boldsymbol{\lambda}\right)=\prod_{\boldsymbol{t=1}}^{\boldsymbol{T}}\boldsymbol{P}\left(\boldsymbol{O_t}|\boldsymbol{q_t},\boldsymbol{\lambda}\right) \tag{3.60}$$

ただし、各観測同士は互いに独立だと仮定する. 式 (**??**) を書き下すと式 (3.61) が成り立つ.

$$\boldsymbol{P}\left(\boldsymbol{O}|\boldsymbol{Q},\boldsymbol{\lambda}\right)=\boldsymbol{b}_{\boldsymbol{q}_1}\left(\boldsymbol{O_1}\right)\cdot\boldsymbol{b}_{\boldsymbol{q2}}\left(\boldsymbol{O_2}\right)\cdots\boldsymbol{b}_{\boldsymbol{qT}}\left(\boldsymbol{O_T}\right) \tag{3.61}$$

状態系列 $\boldsymbol{Q}$ の確率は式 (3.62) のようになる.

$$\boldsymbol{P}\left(\boldsymbol{Q}|\boldsymbol{\lambda}\right)=\boldsymbol{\pi}_{\boldsymbol{q}_1}\boldsymbol{a}_{\boldsymbol{q}_1\boldsymbol{q}_2}\boldsymbol{a}_{\boldsymbol{q}_2\boldsymbol{q}_3}\cdots\boldsymbol{a}_{\boldsymbol{q}_{T-1}\boldsymbol{q}_T} \tag{3.62}$$

$\boldsymbol{O}$ と $\boldsymbol{Q}$ が同時に起こる確率は式 (3.61)、(3.62) を使って式 (3.63) のように表わされる.

$$\boldsymbol{P}\left(\boldsymbol{O},\boldsymbol{Q}|\boldsymbol{\lambda}\right)=\boldsymbol{P}\left(\boldsymbol{O}|\boldsymbol{Q},\boldsymbol{\lambda}\right)\boldsymbol{P}\left(\boldsymbol{Q}|\boldsymbol{\lambda}\right) \tag{3.63}$$

$\boldsymbol{O}$ が得られる確率は式 (3.63) をすべてのあり得る状態の系列について足し合わせたものなので式 (3.64) のようになる.

$$\begin{aligned}\boldsymbol{P}\left(\boldsymbol{O}|\boldsymbol{\lambda}\right)&=\sum_{\boldsymbol{all\,Q}}\boldsymbol{P}\left(\boldsymbol{O}|\boldsymbol{Q},\boldsymbol{\lambda}\right)\boldsymbol{P}\left(\boldsymbol{Q}|\boldsymbol{\lambda}\right)\\&=\sum_{\boldsymbol{q_1},\boldsymbol{q_2},\cdots,\boldsymbol{q_T}}\boldsymbol{\pi}_{\boldsymbol{q}_1}\boldsymbol{b}_{\boldsymbol{q}_1}\left(\boldsymbol{O_1}\right)\boldsymbol{a}_{\boldsymbol{q_1q_2}}\boldsymbol{b}_{\boldsymbol{q_2}}\left(\boldsymbol{O_2}\right)\\&\qquad\cdots\cdot\boldsymbol{a}_{\boldsymbol{q_{T-1}q_T}}\boldsymbol{b}_{\boldsymbol{q_T}}\left(\boldsymbol{Q_T}\right)\end{aligned} \tag{3.64}$$

以上の式展開の解釈は次のようになる. まず時刻 $\boldsymbol{t=1}$ では確率 $\boldsymbol{\pi}_{\boldsymbol{q}_1}$ で状態 $\boldsymbol{q_1}$ になり、確率 $\boldsymbol{b}_{\boldsymbol{q}_1}\left(\boldsymbol{O_1}\right)$ でシンボル $\boldsymbol{O_1}$ を生成する.$\boldsymbol{t=t+1}$ にし ($\boldsymbol{t=2}$)、確率 $\boldsymbol{a}_{\boldsymbol{q_1q_2}}$ で状態 $\boldsymbol{q_1}$ から状態 $\boldsymbol{q_2}$ へ推移する. また確率 $\boldsymbol{b}_{\boldsymbol{q_2}}\left(\boldsymbol{O_2}\right)$ でシンボル $\boldsymbol{O_2}$ を生成する. このような過程を確率 $\boldsymbol{a}_{\boldsymbol{q_{T-1}q_T}}$ で状態 $\boldsymbol{q_{T-1}}$ から状態 $\boldsymbol{q_T}$ に遷移し、シンボル $\boldsymbol{O_T}$ が確率 $\boldsymbol{b}_{\boldsymbol{q_T}}\left(\boldsymbol{O_T}\right)$ で生成されるまで繰り返す.

隠れ最適状態推定問題

この問題は観測系列 $\boldsymbol{O}=\left\{\boldsymbol{O_1}\,\boldsymbol{O_2}\,\cdots\boldsymbol{O_T}\right\}$ が与えられたときに最適状態

$\boldsymbol{Q = \{q_1\, q_2\, \cdots q_T\}}$ を探索する問題である. この隠れ最適状態を推定するために利用されうる手法の一つに Viterbi アルゴリズムがある. ここで Viterbi アルゴリズムに必要な量を式 (3.65) により定義する.

$$\boldsymbol{\delta_t\,(i) = \max_{q_1,q_2,\cdots,q_{t-1}} P\left[q_1 q_2 \cdots q_t = i, O_1 O_2 \cdots O_t | \lambda\right]} \tag{3.65}$$

$\boldsymbol{\delta_t\,(i)}$ は状態 $\boldsymbol{S_i}$ で終わる時刻 $\boldsymbol{t}$ までの観測を説明する最もその確率が大きいパス (遷移系列) である. このとき式 (3.66) が成り立つ.

$$\boldsymbol{\delta_{t+1}\,(i) = \left[\max_i\ \delta_t\,(i)\, a_{ij}\right] \cdot b_j\,(O_{t+1})} \tag{3.66}$$

実際に状態系列を得るためには式 (3.66) を最大化する状態を各 $\boldsymbol{i}$、$\boldsymbol{j}$ について把握していなければいけない. これを配列 $\boldsymbol{\psi_t\,(j)}$ を経由して行う. 具体的なアルゴリズムを以下に示す.

ステップ 1: 初期化

$$\boldsymbol{\delta_1\,(i) = \pi_i b_i\,(O_1)\,, \quad 1 \le i \le N} \tag{3.67}$$

$$\boldsymbol{\psi_1\,(i) = 0} \tag{3.68}$$

ステップ 2: 反復

$$\boldsymbol{\delta_t\,(j) = \max_{1\le i\le N}\left[\delta_{t-1}\,(i)\, a_{ij}\right] b_j\,(O_t)\,, \quad \begin{array}{l} 2 \le t \le T \\ 1 \le j \le N \end{array}} \tag{3.69}$$

$$\boldsymbol{\psi_t\,(j) = \arg\max_{1\le i\le N}\left[\delta_{t-1}\,(i)\, a_{ij}\right], \quad \begin{array}{l} 2 \le t \le T \\ 1 \le j \le N \end{array}} \tag{3.70}$$

ステップ 3: 終了

$$P^* = \max_{1 \le i \le N} \left[\delta_T(i)\right] \tag{3.71}$$

$$q_T^* = \arg\max_{1 \le i \le N} \left[\delta_T(i)\right] \tag{3.72}$$

ステップ 4: 状態系列のバックトラッキング

$$q_t^* = \psi_{t+1}\left(q_{t+1}^*\right), \quad t = T-1, T-2, \cdots, 1 \tag{3.73}$$

ステップ 4 で求めたものが最適な経路 (状態系列) となる.

パラメータ調整問題

パラメータ調整問題は HMM の中でも最も難しい問題である. パラメータを調整することで、観測系列が観測される確率が最も大きくなるようにする.Baum-Welch アルゴリズムを用い訓練データから局所的に観測系列の観測確率を最大にするパラメータを選択することができる.Baum-Welch アルゴリズムを説明するに当たりまず変数の定義を行っておく.$\xi_t(i,j)$ を観測系列において、与えられたモデルで時刻 t のときに状態 S_i でかつ時刻 $t+1$ のときに状態 S_j である確率であるとする. つまり式 refdmeq51) のように表わされるとする.

$$\xi_t(i,j) = P\left(q_t = S_i, q_{t+1} = S_j | O, \lambda\right) \tag{3.74}$$

この $\xi_t(i,j)$ は式 (3.75)、(3.76) であるとき式 (3.77) のように表わされる.

$$\alpha_t(i) = P\left(O_1 O_2 \cdots O_t, q_t = S_i | \lambda\right) \tag{3.75}$$

$$\beta_t(i) = P\left(O_{t+1} O_{t+2} \cdots O_T | q_t = S_i, \lambda\right) \tag{3.76}$$

$$\begin{aligned} \xi_t(i,j) &= \frac{\alpha_t(i) a_{ij} b_j\left(O_{t+1}\right) \beta_{t+1}(j)}{P(O|\lambda)} \\ &= \frac{\alpha_t(i) a_{ij} b_j\left(O_{t+1}\right) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j\left(O_{t+1}\right) \beta_{t+1}(j)} \end{aligned} \tag{3.77}$$

ここで時刻 t で状態 S_i である確率 $\gamma_t(i)$ を式 (3.78) で定義する.

$$\gamma_t(i) = P\left(q_t = S_i | O, \lambda\right) \tag{3.78}$$

このとき $\xi_t(i,j)$ と $\gamma_t(i)$ は式 (3.79) のような関係がある.

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{3.79}$$

時刻 t に対して $\gamma_t(i)$ を足し合わせた $\sum_{t=1}^{T-1} \gamma_t(i)$ は状態 S_i から遷移すると期待される数になり、また $\xi_t(i,j)$ を同様に時刻について足し合わせた $\sum_{t=1}^{T-1} \xi_t(i,j)$ 状態 S_i から状態 S_j に遷移すると期待される数になる. 以上のような関係式を用い、パラメータ π、A、B の再推定を行う. 再推定されたパラメータを $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ とすると式 (3.80)〜(3.82) のようになる.

$$\bar{\pi}_i = \gamma_1(i) \tag{3.80}$$

$$\frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{3.81}$$

$$\frac{\sum_{\substack{t=1 \\ s.t\ O_t = v_k}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \tag{3.82}$$

再推定された $\bar{\lambda}$ の評価は以下のように行う.

1. $\bar{\lambda} = \lambda \rightarrow$ (局所的な収束状態)
2. $P(O|\bar{\lambda}) > P(O|\lambda) \rightarrow$シンボル系列 O を出力するより最適なモデル λ を推定

Baum-Welch アルゴリズムは、学習データの尤度を最大にするようにパラメータを学習する. ただし、基本的には gradient 学習によるパラメータ収束の学習方法であるためｎ、local minimum の方向にしか学習が進まない. そのため初期値が重要になる.

HMM を用いた交通機関識別

HMM ではシステムがパラメータ未知のマルコフ過程 (未来の挙動が、過去の挙動とは無関係に現在の値だけで決定される性質) であると仮定そ観測可能な

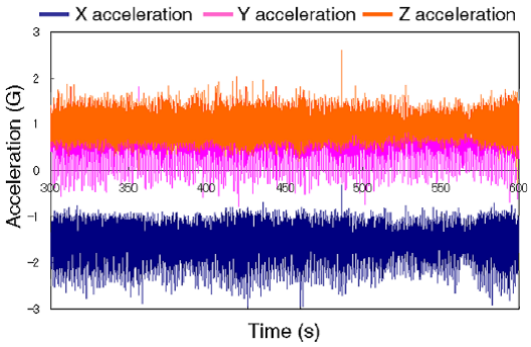


図 3.16 3 軸加速度推移: 徒歩 (Hato[4] から引用)

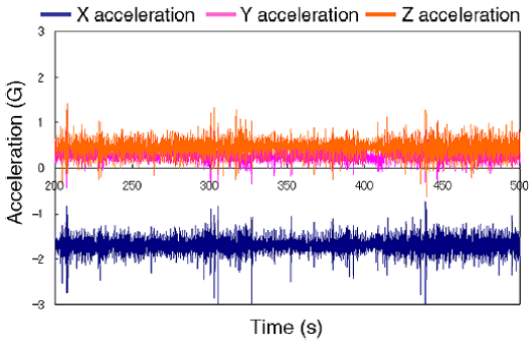


図 3.17 3 軸加速度推移: 自転車 (Hato[4] から引用)

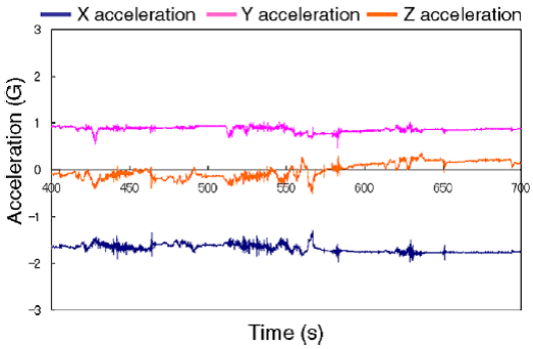


図 3.18　3 軸加速度推移: 車 (Hato[4] から引用)

情報から未知パラメータを推定する. モデル中の状態数は”滞在”、”徒歩”、”自転車”、”車”の 4 状態であり、各状態において出力シンボルの数は”移動速度”、”単位時間あたりの歩数”、”鉛直方向加速度振幅”の 3 つ (各時点における状態数を表わす) から $\mathbf{3 \times 4 \times 3 = 48}$ シンボルとした. 出力値の閾値の設定は、サンプルデータから各交通機関における各盲目の確率密度関数を重ね合わせ、境界となる値に設定する. 図 3.19 は交通機関別速度の発生確率であり、車の閾値に 3.4m/s、自転車と徒歩の閾値に 1.6m/s、徒歩と滞在の閾値に 0.6m/s を設定し 4 クラスにした. 図 3.20 は交通機関別単位時間あたり歩数の発生確率であり、徒歩と自転車の閾値に 1.7step/sec、自転車と徒歩及び滞在の閾値に 0.6step/sec を設定し 3 クラスにした. 図 3.21 は交通機関別鉛直方向加速度振幅の発生確率であり、徒歩と自転車の閾値に 0.7G, 自転車と自動車の閾値に 0.43G、自動車と滞在の閾値に 0.10G を設定し 4 クラスにした.

次に Baum-Welch アルゴリズムの適用について説明する. このアルゴリズムの目的は状態遷移確率 $\boldsymbol{A}$ を求めることである. 推定の際に必要となる初期状態確率 $\boldsymbol{\pi}$ をランダムに与え、初期状態遷移確率 $\boldsymbol{A_0}$ を等状態遷移の場合は 0.99999、他状態遷移の場合は 0.00001 とした. これは交通機関が同じという状態が連続して起こりやすいからである. 初期シンボル出力確率 $\boldsymbol{B_0}$ は真値データとの照合に得られる出力確率を与えた. この Baum-Welch アルゴリズムは gradient 学習によるパラメータ収束を用いているため、局所最大にしか

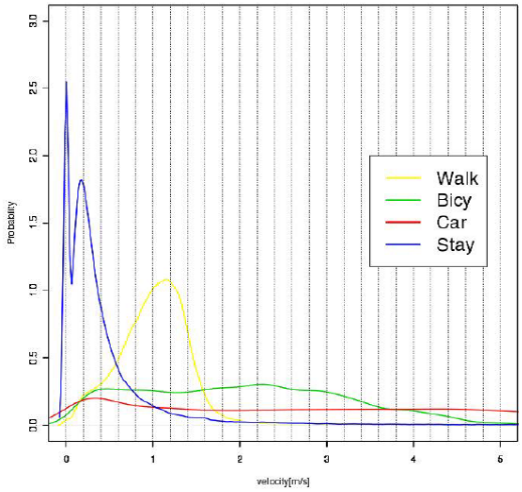


図 3.19 交通機関別、速度の発生確率 (大村 [2] から引用)

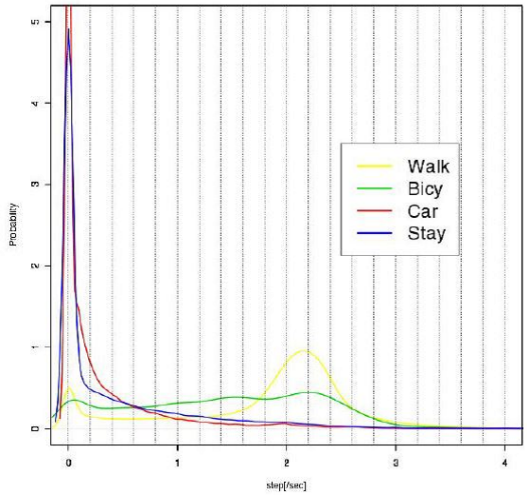


図 3.20 交通機関別、単位時間あたり歩数の発生確率 (大村 [2] から引用)

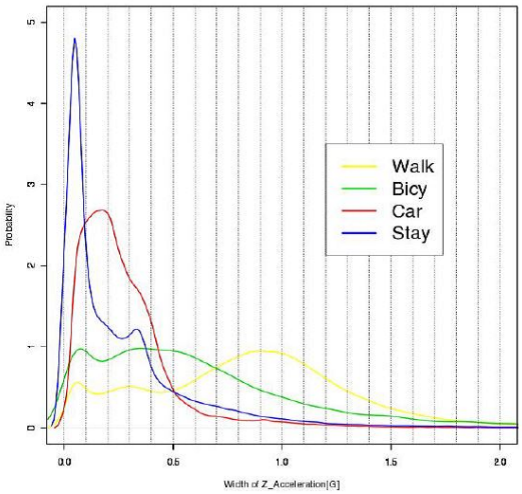


図 3.21　交通機関別、鉛直方向加速度振幅の発生確率 (大村 [2] から引用)

学習は進まない. ここで扱っている HMM はすべての状態がすべての状態に推移する可能性のあるモデルであるので推定が困難である. 従って以下で説明する Viterbi アルゴリズムでは状態遷移確率 $\boldsymbol{A}$ を等状態遷移の場合は 0.99999、他状態遷移の場合は 0.00001、シンボル出力確率 $\boldsymbol{B}$ を真値データとの照合により得られる確率とした. また Viterbi アルゴリズムの性質上確率 0 は $\mathbf{10^{-10}}$ とした.Viterbi アルゴリズムではモデル $\boldsymbol{\lambda}$ がシンボル系列 $\boldsymbol{O}$ を出力する最も可能性の高い状態遷移系列を推定し、その系列に対する尤度を求めるアルゴリズムである. モデル $\boldsymbol{\lambda}$ を決定するパラメータ群は Baum-Whelch アルゴリズムぬよるパラメータ推定が行えなかったので、初期状態確率 $\boldsymbol{\pi}$ は等確率で $\boldsymbol{\pi = (0.25, 0.25, 0.25, 0.25)}$ とする. このとき入力する系列データは一人の一日分のデータであるが、識別精度を考慮し、ある個人の調査期間の全データとする.

(c)　マップマッチング

マップマッチング基礎概念 高度道路交通システム (ITS) の分野において、経路ガイダンス、道路利用者への課金、交通流制御などを行うために、あるいは利用者に事故やバス到着時刻情報を知らせるためには自動車の位置情

報が必要となる. 近年位置を正確に取得するための技術として GPS(Global Positioning System) が目覚ましい技術発展を遂げてきた. また、ストリートキャニオンなどの遮蔽物の多いところや地下トンネルのような GPS だけでは十分に位置を特定できない場所でも、方位情報や加速度情報を用いたデッドレコニング (自律航法) と呼ばれる技術によって正確な位置情報を取得できるようになってきた. このような技術を用いて取得された位置データを用い、ある時点で移動者がどの経路を利用しているのか、またその経路上のどこにいるのかを特定する技術がマップマッチングである. ここではマップマッチング手法のレビューを行う.

マップマッチング基本ステップ

まずマップマッチングの基本的なステップについて説明を行う.

ステップ 1

道路ネットワークデータを準備する (図 3.22)

ステップ 2

道路ネットワーク上に測位位置データをプロットする (図 3.23)

ステップ 3

道路ネットワーク上のリンクやノードと測位位置データとの関係 (距離、角度) を定量化し通過リンクを特定する (図 3.24)

ステップ 4

リンク単位の所要時間、走行速度などパフォーマンス指標を各リンクについて算出 (図 3.25)

マップマッチングの基本的なステップは以上の通りである. リンクを特定する際の定量化の方法や、測位方法の補正などによって様々なマップマッチング手法がある.

マップマッチング手法

この節では様々なマップマッチングを行う際の手法を紹介する. マップマッチングの精度向上のために様々なアプローチが考案されてきた. アプローチの仕方を分類すると 1) 幾何解析マップマッチング手法、2) 位相幾何解析マッ

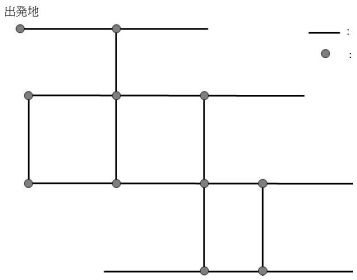


図 3.22　道路ネットワークの準備

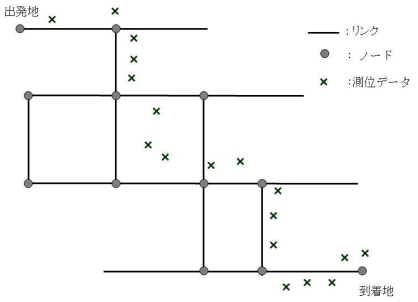


図 3.23　測位データプロット

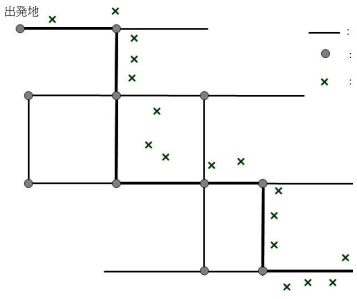


図 3.24　通過リンク特定

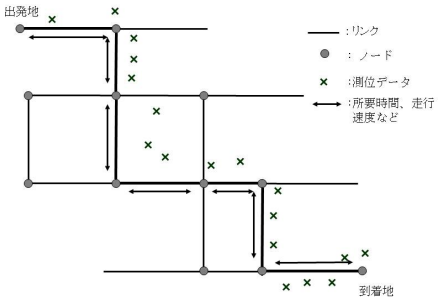


図 3.25　リンクごとの所要時間、走行速度算出

プマッチング手法、3) 確率的マップマッチング手法、4) 高度な技術を利用したマップマッチング手法の 4 つに分類される. この節では各手法ごとに簡単な紹介を行う.

1): 幾何解析マップマッチング手法

幾何解析マップマッチング手法はリンクの道路ネットワークデータの形状のみを考慮した手法である. 幾何解析手法で最も利用される手法は **point to point** といわれるマッチング手法である [6]. この手法は測位点をネットワーク上の最も近いノードにマッチングするという簡単な手法であり、実行も容易で計算もとても速い. 以下にステップを示す.

ステップ 1

測位点と道路ネットワーク上のすべてのノード (交差点など) の距離を計算する

ステップ 2

測位点を最も距離が小さいノードへマッチングする

ステップ 3

すべての測位点について計算すれば終了、そうでなければステップ 1 へ

しかしながら、この手法は道路ネットワークの形状に大きく影響を受ける. 例えば図 3.26 をみると、距離の観点からだと、$\boldsymbol{P}$ はリンク $\boldsymbol{A}$ 上へマッチングされることが適切であるにも関わらず、一番近いノードは $\boldsymbol{B_2}$ なので、リンク $\boldsymbol{B}$ 上にマッチングされる. したがって **point to point** 手法では、図 3.27 のように多くのノードがあるネットワークの方が正確にマッチングされることになる. 別の幾何解析アプローチに **point to curve** といわれるマッチング手法がある [6],[7]. これは先ほどの **point to point** マッチングと類似しているが、測位点をネットワーク上の最も近いリンクにマッチングする手法である. 図 3.28 を見ると、$\boldsymbol{P}$ とリンク $\boldsymbol{A}$、$\boldsymbol{B}$ との距離は $\boldsymbol{d_A}$、$\boldsymbol{d_B}$ で表され、今 $\boldsymbol{d_A} > \boldsymbol{d_B}$ なので、測位データ $\boldsymbol{P}$ はリンク $\boldsymbol{A}$ にマッチングされる. 以下にステップを示す.

ステップ 1

測位点とネットワーク上のすべてのリンク (道路) の距離を計算する

ステップ 2

測位点を最も距離が小さいリンクへマッチングする

ステップ 3

すべての測位点について計算すれば終了、そうでなければステップ 1 へ

point to curve は **point to point** マッチングよりも正確にマッチングが可能であるが、図 3.29 において、移動者はリンク $\boldsymbol{A}$ を利用している可能性が大きいが、$\boldsymbol{P_2}$ はリンク $\boldsymbol{B}$ の方がその距離が近いのでリンク $\boldsymbol{A}$ ではなくリンク $\boldsymbol{B}$ にマッチングされてしまう. これは実際の適用を考えた場合、密度の高いネットワークでは誤ったマッチングが増えてしまう可能性があることを示している. これら二つのマッチング手法に近いが、それより精度がよい手法として **curve to curve** マッチングがある [6],[7],[8]. 以下にステップを示す.

ステップ 1

point to point マッチングにより、測位点をマッチングする候補となるノードを探索する

ステップ 2

候補ノードをつなぎ候補リンクを作成する

ステップ 3

候補ノードを結び候補経路を作成する

ステップ 4

測位点を結びその軌跡との距離の和を計算する

ステップ 5

すべての候補経路のうち最も距離が小さい経路を真の経路に決定する

図 3.33 では水色の経路にマッチングされる. 最後に RRFroad reduction filter) といわれる手法を紹介する [9]. この手法は、まず候補となる道路をすべて列挙して、間違っている道路を機械的に取り除いていく手法である. まず **curve to curve** マッチングによって候補経路を探索する. その後測位位置

データと比較することでマッチングを行う. ここでは差分 GPS(VDGPS) が用いられる. これは正確に位置の分かっている場所を参照点として受信機をおき、そこでの観測データから未知点と参照点に共通な誤差から位置の補正を行うための手段である. ステップを以下に示す.

ステップ 1

測位点と道路ネットワークの上のリンクとの距離を計算し最も近い道路ネットワーク上に投影する

ステップ 2

VDGPS を用いて位置を修正する

ステップ 3

次の時点の測位点とステップ 2 で生成した修正点によって新たな点を生成する

ステップ 4

ステップ 3 で生成した点を道路ネットワーク上に投影する

ステップ 5

道路ネットワークが 1 つに定まれば終了、そうでなければステップ 1 からステップ 4 を繰り返す

2): 位相幾何解析マップマッチング手法

位相解析マップマッチング手法は幾何的な情報に加え、リンクの接続性、連続性をもとにマッチングを行う [10]. このマッチングでは、連続する測位位置データが道路ネットワーク上のどのリンク上にマッピングされるかを、距離だけでなく、リンクデータと位置データの交差角度、相対角度距離などから決定する (リンク決定フェイズとする). その後、速度と進行方向角度によってリンク上のどの位置にあるのかを決定する (位置決定フェイズとする). リンクとリンク上の位置を決定したのち、測位データが現在のリンク上にあるのか、つまり交差点に入ったかどうかを確認する (リンク確認フェイズとする).

リンク決定フェイズ

リンクはいくつかの評価指標の重みつき総和で決定される. 利用する評

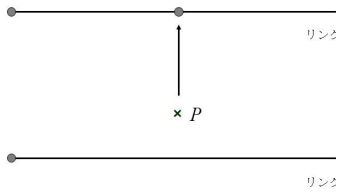


図 3.26　point to point: 誤ったマッチング例

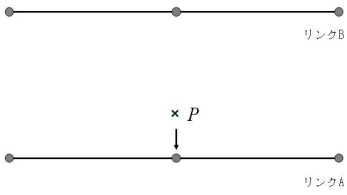


図 3.27　point to point: 正しいマッチング例

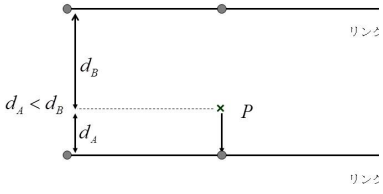


図 3.28　point to curve マッチング

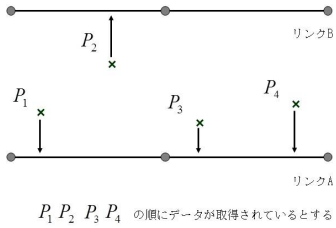


図 3.29　point to curve: 誤ったマッチング例

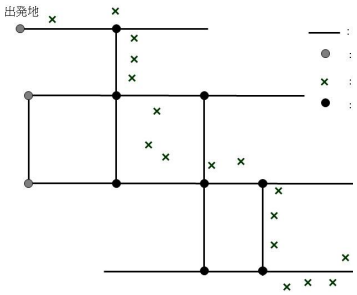


図 3.30 curve to curve: ステップ 1 候補ノード探索

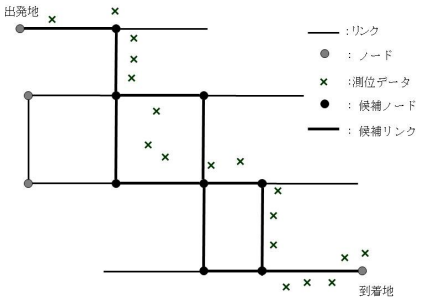


図 3.31 curve to curve: ステップ 2 候補リンク探索

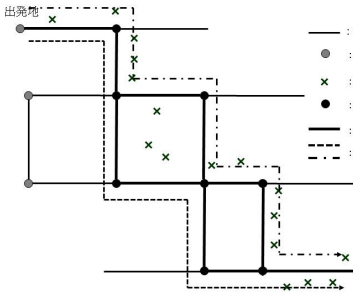


図 3.32 curve to curve: ステップ 3 候補ルート探索

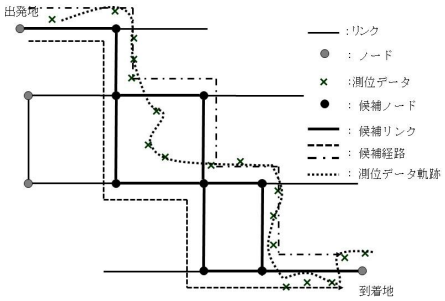


図 3.33 curve to curve: ステップ 4 位置データ軌跡と候補ルートの距離計算

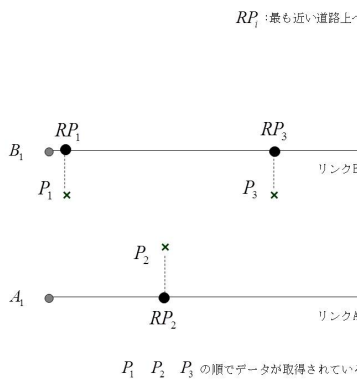


図 3.34　RRF: ステップ 1 測位データを道路ネットワークに投影

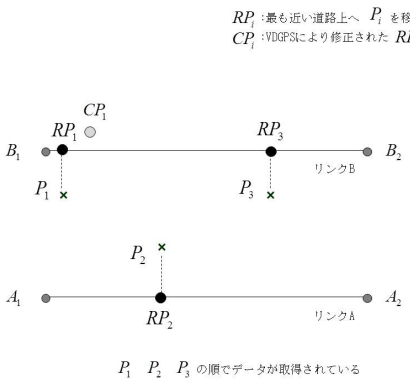


図 3.35　RRF: ステップ 2 VDGPS を用いて位置を修正

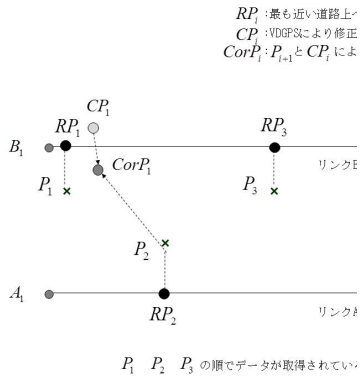


図 3.36　RRF: ステップ 3 次の時点での測位データと修正データで新たな位置を生成

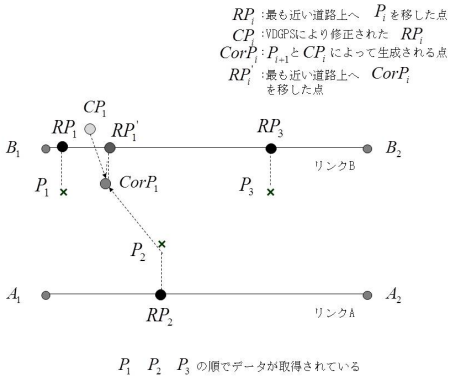


図 3.37　RRF: ステップ 4 再修正したデータを道路ネットワークに投影

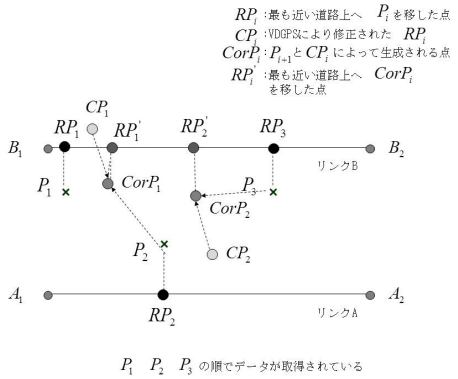


図 3.38 RRF: ステップ 5 繰り返し

価指標は 1) 測位位置データの進行方向とリンク方位角の差、2) 測位位置データとリンクの近接性、3) 測位位置データのリンクに対する相対位置、の 3 つである.

1) 進行方向とリンク方位角の差

図 3.39 において $\boldsymbol{P_1}$〜$\boldsymbol{P_3}$ がリンク 1 にあると決定されているとき、$\boldsymbol{P_4}$ がどのリンクにあるかを決定する. このとき候補となるのはリンク 2,3,4 である. 図中北 (上) 方向と進行方向がなす角度を $\boldsymbol{\beta}$、北 (上) 方向と各リンクがなす角度を $\boldsymbol{\beta_i(i=2,3,4)}$ とする. このとき $\boldsymbol{\Delta\beta=\beta-\beta_i}$ とし、$\boldsymbol{\beta'}$ を式 3.83 と定義する. このとき式 3.84 で表わされる $\boldsymbol{WS_H}$ を進行方向とリンク方位角差の重みつきスコアとする. ただし、$\boldsymbol{A_H}$ は正のパラメータである.$\boldsymbol{\Delta\beta'}$ が小さくなれば、つまり、進行方向とリンク方位角の差が小さければ小さいほどこのスコアは大きくなり、そのリンクにマッチングされる確率が大きくなる. ここで $\boldsymbol{WS_H}$ は進行方向とリンクと方位角の差の重みスコア、$\boldsymbol{A_H(>0)}$ は重みパラメータである.

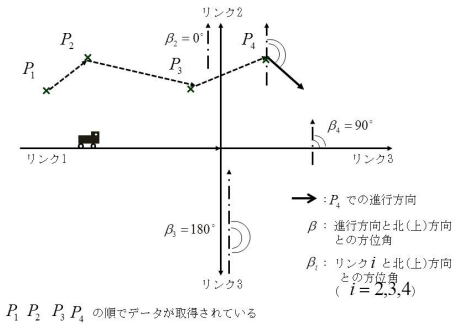


図 3.39　測位位置データの進行方向とリンク方位角差

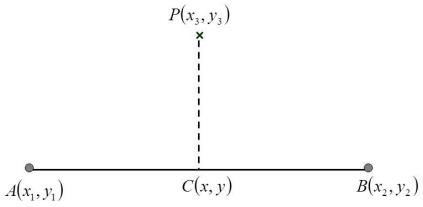


図 3.40　リンクとの近接性: 最短距離

$$\Delta\beta' = \begin{cases} \Delta\beta, \ if \ -180^\circ \leq \Delta\beta \leq 180^\circ \\ 360^\circ - \Delta\beta, \ if \Delta\beta > 180^\circ \\ 360^\circ + \Delta\beta, \ if \Delta\beta < -180^\circ \end{cases} \tag{3.83}$$

$$WS_H = A_H \cos(\Delta\beta') \tag{3.84}$$

2) 測位位置データとリンクの近接性

測位位置データのリンクに対する相対位置には二つの評価指標がある.1つは測位位置データと候補リンクの最短距離 (位置データから候補リンクへの垂線の長さ) で、もう 1 つが連続する 2 つの位置データを結んだときの線と、候補リンクが交差する場合の角度である.

a) 最短距離

図 3.40 において $\boldsymbol{P}$ は測位位置データ、$\boldsymbol{AB}$ は道路ネットワーク上のリンク、$\boldsymbol{C}$ は $\boldsymbol{P}$ からリンク $\boldsymbol{AB}$ に下ろした垂線の足を表わす. このとき、$\boldsymbol{PC}$ の大きさは式 (3.85) のようになる.

$$\boldsymbol{D = \frac{x_3\left(y_1 - y_2\right) - y_3\left(x_1 - x_2\right) + \left(x_1 y_2 - x_2 y_1\right)}{\sqrt{\left(x_1 - x_2\right)^2 + \left(y_1 - y_2\right)^2}}} \tag{3.85}$$

この距離 $\boldsymbol{D}$ が小さくなれば測位データとリンクの近接性は大きくなるので、最短距離についての評価指標は式 (3.86) のようになる.

$$\boldsymbol{WS_{PD} = {A_P}/{D}} \tag{3.86}$$

ここで、$\boldsymbol{WS_{PD}}$ は測位データとリンクの近接性の最短距離についての評価指標、$\boldsymbol{D(>0)}$ は測位データから道路ネットワーク上のリンクへの垂直距離、$\boldsymbol{A_P}$ は重みづけパラメータを表わす.

b) 連続する 2 点を結ぶ線と候補リンクが交差する角度

連続する測位データ、$\boldsymbol{P(x_{i-1}, y_{i-1})}$ と $\boldsymbol{P(x_i, y_i)}$ と道路ネットワーク上の候補リンク $\boldsymbol{l}$ が交差しているときその角度が $\boldsymbol{\theta}$(鋭角) のとき、連続する 2 点の測位データと候補リンクの角度に関する評価指標は式 (3.87) のようになる.

$$\boldsymbol{WS_{PI} = A_P \cos\left(\theta\right)} \tag{3.87}$$

ここで $\boldsymbol{WS_{PI}}$ は交差角度に関する評価指標を表わし、もし連続する 2 点を結ぶ線と候補リンクが交差していなければ 0 とする.

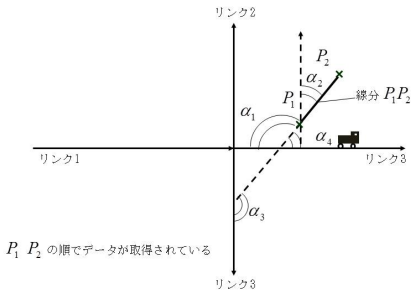


図 3.41　候補リンクに対する相対位置

3) 測位位置データのリンクに対する相対位置

図 3.41 において $\boldsymbol{P(x_i, y_i)}$ は測位データ、$\boldsymbol{\alpha_1}$〜$\boldsymbol{\alpha_4}$ はリンク 1〜リンク 4 からの測位データの相対位置を表わす角度とする. 図 3.41 において測位データはリンク 2 かリンク 3 にマッピングされるのが適当であると考えられるので、角度が小さければ小さいほどそのリンクにマッチングされる確率が高くなる. したがって候補リンクからの相対位置についての評価指標は式 (3.88) のようになる.

$$\boldsymbol{WS_{RP} = A_{RP} \cos(\alpha)} \tag{3.88}$$

$\boldsymbol{WS_{RP}}$ は候補リンクに対する相対位置に関する評価指標、$\boldsymbol{\alpha \leq 180}$ は測位データと最も近いノードを結んだ線と候補リンクがなす角度、$\boldsymbol{A_{RP}}$ は重みづけパラメータである.

すべてのの重み付きの評価指標を足し合わせたものを $\boldsymbol{TWS}$ とすると式 (3.89) のようになる.

$$\boldsymbol{TWS = WS_H + (WS_{PD} + WS_{PI}) + WS_{RP}} \tag{3.89}$$

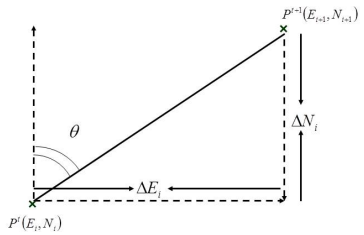


図 3.42 位置決定フェイズ: 速度と進行方向から推定

この $\boldsymbol{TWS}$ によって道路ネットワークのどのリンク上にあるのかを決定する.

位置決定フェイズ

リンク上のどの位置にあるのかを進行方向と速度 $\boldsymbol{v}$ によって決定する. 図 3.42 において $\boldsymbol{P^t}$ と $\boldsymbol{P^{t+1}}$ はあるリンク $\boldsymbol{A}$ 上の時刻 $\boldsymbol{t}$、$\boldsymbol{t+1}$ の自動車の位置を表すとする.$\boldsymbol{P^t}$ における速度が $\boldsymbol{v}$ であるから式 (3.90) が成り立つ.

$$\begin{aligned} \boldsymbol{\Delta E_i} &= \boldsymbol{(v * 1) \sin \theta} \\ \boldsymbol{\Delta N_i} &= \boldsymbol{(v * 1) \cos \theta} \end{aligned} \tag{3.90}$$

したがって $\boldsymbol{P^{t+1}}$ における自動車の位置は式 (**??**) にようになる.

$$\begin{aligned} \boldsymbol{E_{i+1}} &= \boldsymbol{E_i + \Delta E_i} \\ \boldsymbol{N_{i+1}} &= \boldsymbol{N_i + \Delta N_i} \end{aligned} \tag{3.91}$$

このようにリンク上の位置を決定する. 測位位置を決定するのに GPS/DR のセンサデータとデジタルロードマップ上の位置データを用いる手法もある. 図 3.43 において GPS/DR によって測位された点を $\boldsymbol{P^S}$、測位された点を道路ネットワーク上のリンクに投影したものを

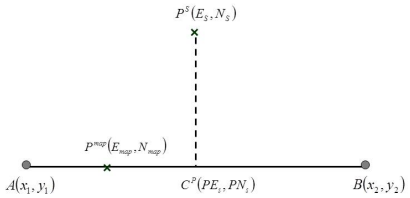


図 3.43　位置決定フェイズ: デジタルマップと測位データ平均から推定

$\boldsymbol{C^P}$、デジタルマップからの点を $\boldsymbol{P^{map}}$ とする．リンクの両端のノード $\boldsymbol{A}$、$\boldsymbol{B}$ の座標と測位データの座標を用いると $\boldsymbol{C^P}$ の座標は式 (3.92) のように表わされる．

$$\begin{aligned} \boldsymbol{PE_s} &= \frac{(x_2-x_1)[E_s(x_2-x_1)+N_s(y_2-y_1)]+(y_2-y_1)(x_1y_2-x_2y_1)}{(x_2-x_1)^2+(y_2-y_1)^2} \\ \boldsymbol{PN_s} &= \frac{(y_2-y_1)[E_s(x_2-x_1)+N_s(y_2-y_1)]-(x_2-x_1)(x_1y_2-x_2y_1)}{(x_2-x_1)^2+(y_2-y_1)^2} \end{aligned} \tag{3.92}$$

今 $\boldsymbol{P^{map}}$ と $\boldsymbol{C^P}$ は独立であるので、自動車の位置は式 (**??**) のようにそれぞれの重みづけ平均で推定される．

$$\begin{aligned} \hat{\boldsymbol{E}} &= \left(\frac{\sigma^2_{E_S}}{\sigma^2_{E_S}+\sigma^2_{map}}\right)\boldsymbol{E_{map}} + \left(\frac{\sigma^2_{map}}{\sigma^2_{E_S}+\sigma^2_{map}}\right)\boldsymbol{PE_S} \\ \hat{\boldsymbol{N}} &= \left(\frac{\sigma^2_{N_S}}{\sigma^2_{N_S}+\sigma^2_{map}}\right)\boldsymbol{N_{map}} + \left(\frac{\sigma^2_{map}}{\sigma^2_{N_S}+\sigma^2_{map}}\right)\boldsymbol{PN_S} \end{aligned} \tag{3.93}$$

$\hat{\boldsymbol{E}}$、$\hat{\boldsymbol{N}}$ は推定された位置、$\boldsymbol{\sigma^2_{map}}$ は $\boldsymbol{E_{map}}$ に関する誤差分散、$\boldsymbol{\sigma^2_{E_S}}$ は $\boldsymbol{E_S}$ に関する誤差分散、$\boldsymbol{\sigma^2_{N_S}}$ は $\boldsymbol{N_S}$ に関する誤差分散を表わす．

リンク確認フェイズ

正しいリンクにマッチングしたのち、測位データが現在のリンクにあるのかどうかを確認する．基準は以下の 2 点である．

1. 二つの連続する線 (つまり $\boldsymbol{P(x_i, y_i)}$ と $\boldsymbol{P(x_{i+1}, y_{i+1})}$ を結んだ線と $\boldsymbol{P(x_{i+1}, y_{i+1})}$ と $\boldsymbol{P(x_{i+2}, y_{i+2})}$ を結んだ線) の進行方向角度が 45° より大きいか
2. 二つの連続する測位点の進行方向角度が 45° より大きいか、また $\boldsymbol{\alpha}$(測位店と最も近いノードを結んだ線とリンクのなす角度) が 90° より大きいか

もしこれらの条件が満たされれば、自動車は交差点に入ったとみなし、再びリンク決定フェイズに戻る.

以上のマップマッチングアルゴリズムを以下に示す.

ステップ 1

最初の測位点と最も近いノードを探索する

ステップ 2

ステップ 1 で探索した最も近いノードを通るすべてのリンクを選択する

ステップ 3

式 (3.89) で表わされる重みづけ式を用いて正しいリンクを決定する (このとき最初の測位点と次の測位点はこのリンクにマッチングされる)

ステップ 4

式 (3.93) を用いてリンク上の位置を決定する

ステップ 5

$\boldsymbol{\Delta\beta' < 45°}$、$\boldsymbol{\alpha \leq 90°}$ を確認しもし満たされていれば式 (3.93) によりリンク上の位置を決定することを継続し、もし満たされていなければステップ 1 へ戻る

ステップ 6

すべての測位点がマッチングされるまでステップ 5 を継続する

3): 確率的マップマッチング手法

確率的手法マップマッチング手法ではセンサーから取得された測位データ周辺に楕円や円形の信頼域を定義することが必要となる. この技術は DR セン

サからの位置をマッチングさせるのに初めて導入された [11]. また GPS センサから取得されたデータに対しても利用され、エラー領域は GPS 測位データの誤差分散から導くことが提案された [12]. ここではより精度の高い確率的マップマッチング手法を紹介する [13]. このマップマッチング手法においても位相幾何マップマッチング手法と同様にマッチングするリンクを決定したのちにリンク上の場所を決定する. 確率的マップマッチングではリンクを決定する段階で確率的手法が用いられている. 図 3.44 において測位データが $\boldsymbol{P}$ だとすると、その周囲の楕円形がエラー域として定義され、このエラー域内にあるリンクが候補リンクとなる. 図 3.44 の場合候補リンクは”リンク 3”となる. エラー域は図 3.45 のように長方形を用いることもできる. このとき楕円の軸の長さ $\boldsymbol{a}$、$\boldsymbol{b}$ は式 (3.94)、(3.95) のように表わされる.

$$\boldsymbol{a = \hat{\sigma}_0 \sqrt{\frac{1}{2}\left(\sigma_x^2 + \sigma_y^2 + \sqrt{\left(\sigma_x^2 - \sigma_y^2\right)^2 + 4\sigma_{xy}^2}\right)}} \tag{3.94}$$

$$\boldsymbol{b = \hat{\sigma}_0 \sqrt{\frac{1}{2}\left(\sigma_x^2 + \sigma_y^2 - \sqrt{\left(\sigma_x^2 - \sigma_y^2\right)^2 + 4\sigma_{xy}^2}\right)}} \tag{3.95}$$

また楕円形が図 3.44 中の上矢印からどれだけ傾いているかを表わす $\boldsymbol{\phi}$ は式 (3.96) のように表わされる.

$$\boldsymbol{\phi = \frac{\pi}{2} - \frac{1}{2}\arctan\left(\frac{2\sigma_{xy}}{\sigma_x^2 - \sigma_y^2}\right)} \tag{3.96}$$

ここで $\boldsymbol{\sigma_x\ 2}$ と $\boldsymbol{\sigma_y\ 2}$ は測位位置の分散、$\boldsymbol{\sigma_x y}$ は共分散、$\boldsymbol{\hat{\sigma}_0}$ は拡張係数を表わす. 拡張係数は GPS の軌道が安定していないこと、大気伝搬、様々な伝搬経路を通ること、受信者側のノイズなどを補うための定数で 3.03 がよいとされている [12]. 図 3.46 にこのリンク決定アルゴリズムのフローを示す. リンクが決定した後は位相幾何マップマッチングのときと同様にリンク上の位置を決定する.

4): 高度な技術を利用したマップマッチング手法

この節ではカルマンフィルタ、パーティクルフィルタ [15]、DDR などより高

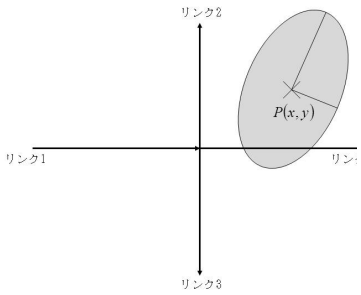


図 3.44 信頼区域: 楕円形

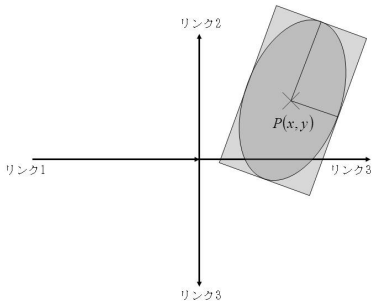


図 3.45 信頼区域: 長方形

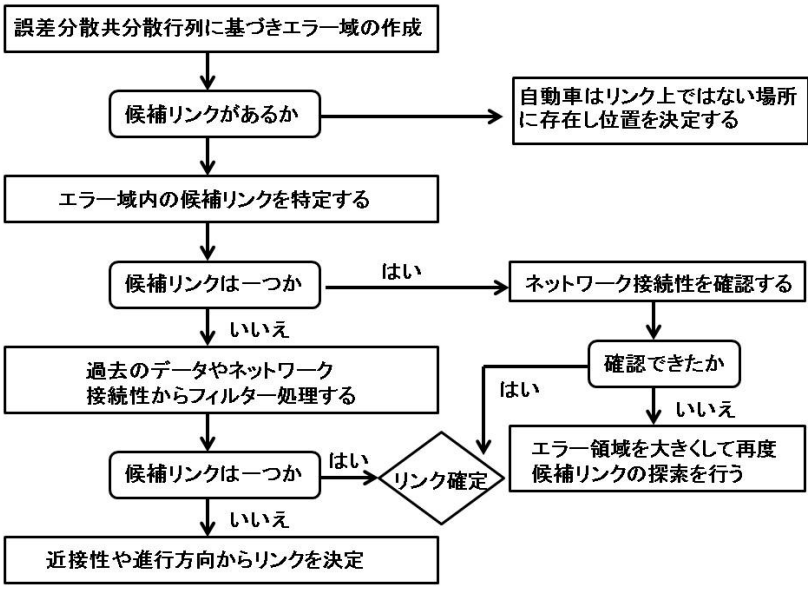


図 3.46 リンク決定アルゴリズム フローチャート

度な概念を用いたマップマッチング手法を紹介する. **カルマンフィルタ** ここではカルマンフィルタを用いたマップマッチング手法について説明する [16]. カルマンフィルタとはシステムや観測値に様々なランダムなノイズがのっているときにそれぞれのノイズの大きさに応じて適切な重みづけを行い、変化していくシステムの状態を精度よく推定するための手法である. 今、自立航法 ($\boldsymbol{DeadReckoning}$) によって取得したデータを $\boldsymbol{j(i)}$、実際の走行軌跡の候補データを $\boldsymbol{m(i)}$ としたとき、$\boldsymbol{m}$ と $\boldsymbol{j}$ を観測モデル式 (3.97) と状態モデル式 (3.98) でモデル化する.

$$\boldsymbol{j\,(i) = M\,(i)\,X\,(i) + v\,(i)} \tag{3.97}$$

$$\boldsymbol{X\,(i+1) = X\,(i)} \tag{3.98}$$

ただし $\boldsymbol{M(i)}$、$\boldsymbol{X(i)}$ はそれぞれ式 (3.99) で表わされるとする.

$$\begin{aligned}&\boldsymbol{M\,(i) = [m\,(i-2)\ \ m\,(i-1)\ \ m\,(i)\ \ m\,(i+1)\ \ m\,(i+2)]}\\&\boldsymbol{X\,(i) = [x_1\ x_2\ x_3\ x_4]^T}\end{aligned} \tag{3.99}$$

観測モデル、状態モデルでカルマンフィルタを用いて $\boldsymbol{X(i)}$ の推定値 $\boldsymbol{\hat{X}\,(i)}$ を求める. ここではその推定法については割愛する. 推定値と実測値の差であるイノベーション $\boldsymbol{v(i)}$ を式 (3.100) によって計算する.

$$\boldsymbol{v\,(i) = j\,(i) - \hat{X}\,(i)} \tag{3.100}$$

カルマンフィルタのイノベーションは平均値 0 のガウス性白色雑音であるという性質がある. したがって複数の走行軌跡の候補が与えられた場合、それぞれの候補に対してイノベーションを求め、その白色度が最も高いものが真の走行軌跡であるとする. イノベーションの白色度を評価するものとして式 (3.101) を用いる.

$$\boldsymbol{\rho_v = \frac{1}{(N-1)\,\sigma_v^2} \sum_{k=0}^{N-2} \{v\,(k) - m\}\,\{v\,(k+1) - m\}} \tag{3.101}$$

ただし $\boldsymbol{m}$ は $\boldsymbol{v(i)}$ の平均値である. 今状態モデル式 (3.98) は自動車が急に進行方向を変化させないという仮定を置いている. したがって、急激に進行方向が変わった場合すなわち式 (3.102) が成り立つとき

$$|m(i) - m(i-k)| > \varepsilon, \quad k = -2, -1, 1, 2 \tag{3.102}$$

$m(i-k) = m(i)$ として $X(i)$ の推定を行う.

パーティクルフィルタ

パーティクルフィルタは、観測モデルとして非線形の任意のモデルを適用できるため、画像などの観測過程が複雑なコンピュータビジョン等で多く用いられている. 前節のカルマンフィルタは誤差が正規分布に従う場合用いることができないのでパーティクルフィルタを利用することができる. 自動車に関する位置データで分析を行う場合、まずマップマッチングを行う. しかし速度が低下している交差点近傍で位置補正を行う際は、移動時よりも誤差によるばらつきが有意になることがあるため、誤ったマップマッチングが発生しやすい. そこでパーティクルフィルタを用いた位置補正をあらかじめ行うことで、交差点近傍でのマップマッチングの精度を向上させる. パーティクルフィルタは、時刻 t における観測値 θ_t から状態ベクトル x_t の事後確率である確率密度関数 $P(x_t|\theta_t)$ を推定する. この事後確率 $P(x_t|\theta_t)$ は、ベイズの定理により以下の式 (3.103) に置換される.

$$P(x_t|\theta_t) = \frac{P(y_t|x_t) \cdot P(x_t|\theta_t)}{P(y_t|\theta_{t-1})} \tag{3.103}$$

$$P(x_t|\theta_{t-1}) = \int P(x_t|x_{t-1}) \cdot P(x_{t-1}|\theta_{t-1})\, dx_{t-1} \tag{3.104}$$

まず $t-1$ の状態をもとに抽出されたパーティクル群 $P(x_{t-1}|\theta_{t-1})$ を選択し、対象領域に拡散する. これを事前分布とする. つぎに、運動モデル $P(x_{t-1}|\theta_{t-1})$ を用い、パーティクル群を一定の規則に基づいて誤差を加えた形で移動させる (拡散:　運動モデルを用いる (後述)). さらに、各パーティクルに対しその尤度を計算する (重み付け (後述)). 最後にリサンプリングを実施し尤度の高いパーティクルについては複数個サンプリングされ、低いものに関しては消滅する. このとき最も尤度の高いパーティクルが集中する位置を推定値として位置を確定する.

運動モデル

パーティクルフィルタでは、推定対象の動作時性をモデル化したもので、モデルの精度を左右するものである. ここでは速度が低下している交差点近傍で位置補正を行う際、誤差項が小さくなるように定式化を行う. 現時刻 $\boldsymbol{t}$ における $\boldsymbol{i}$ 番目のパーティクルの状態ベクトル $\boldsymbol{x_t}$ の運動モデル式は式 (3.105) のようになる.

$$x_t^i = \begin{bmatrix} \boldsymbol{a_t} \\ \boldsymbol{b_t} \\ \boldsymbol{Va_t} \\ \boldsymbol{Vb_t} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a_{t-1}} \\ \boldsymbol{b_{t-1}} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\alpha \cdot Va_{t-1} \cdot \varepsilon} \\ \boldsymbol{\alpha \cdot Vb_{t-1} \cdot \varepsilon} \\ \boldsymbol{\beta \cdot \xi} \\ \boldsymbol{\beta \cdot \xi} \end{bmatrix} \tag{3.105}$$

$\boldsymbol{(a, b)}$ は 2 次元平面座標、$\boldsymbol{(V_a, V_b)}$ は速度を表わす.$\boldsymbol{\alpha}$、$\boldsymbol{\beta}$ は分散の大きさを表わすパラメータであり、$\boldsymbol{\varepsilon \sim N\left(1, \sigma^2\right)}$ 、$\boldsymbol{\xi \sim N\left(0, \sigma^2\right)}$ は正規乱数を表わしている.

重み付け

重み付けは、観測モデルを用い、パーティクルの重みを評価する. 観測モデルは、あらかじめ状態空間内で観測される値の分布状況を測定し、これを学習データとしてモデル化したものである. 観測モデルでは、現時刻における観測値とパーティクルが示す状態量を入力する. 入力した状態量において入力した観測値が観測される確率密度を求める. この確率密度から、そのパーティクルが示す状態量における移動体の存在尤度を決定し、重みに反映する. ここでは各パーティクルから現時刻 $\boldsymbol{t}$ における観測値 $\boldsymbol{y_i}$ までの距離の逆数を尤度とする.

図 3.47 にパーティクルフィルタの概念図を示す. 以上のような流れでパーティクルフィルタを用いて測位位置データを補正することでマップマッチングの精度を向上させる.

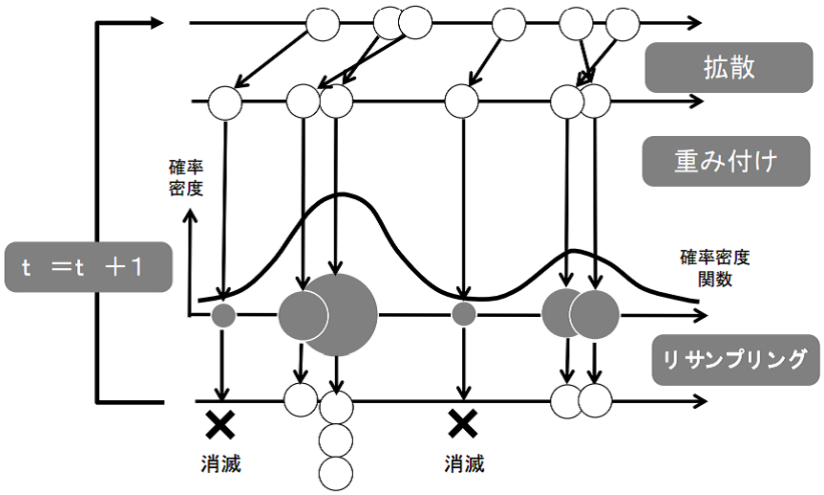


図 3.47　パーティクルフィルタ概念図 (三谷, 羽藤 (2009)[15] から引用)

参考文献

[1] 高橋厚年, 朝倉康夫, 羽藤英二: 移動体通信システムによる交通行動データ特性に関する基礎的分析, 土木計画学研究 講演集, Vol.22, pp.413-416, 1999

[2] 大村朋之: 位置-加速度情報を用いた交通機関識別と移動負荷モデルの開発, 修士課程学位論文, 東京大学, 2011

[3] 村上仁一, 確率的言語モデルによる自由発話認識に関する研究, 博士課程学位論文, 豊橋技術科学大学, 2003

[4] Eiji Hato: Development of behavioral context addressable loggers in the shell for travel-activity analysis, Transportation Research PartC, Vol. 18, pp. 55-67, 2010

[5] C.M. ビショップ：パターン認識と機械学習 下 - ベイズ理論による統計的予測,

[6] Bernstein, D., Kornhauser, A. : An introduction to map-matching for personal navigation assistants,?,1996

[7] White, C.E., Bernstein, D., Kornhauser, A.L.: Some map-matching algorithms for personal navigation assistants. Transportation Research Part C, Vol. 8, pp.91-108,200

[8] Phuyal, B.: Method and use of aggragated dead reckoning sensor and GPS data for map-matching, In: proceedings of the Institute of Navigation (ION) annual conference, Portland, OR (20-27 September), 2002

[9] Taylor, G., Blewitt, G., Steup, D., Corbett, S., Car, A.: Road reduction filtering for GPS-GIS navigation, Transactions in GIS, Vol. 5, pp. 193-207, 2001

[10] Quddus, M.A., Ochieng, W.Y., Zhao, L., Noland, R.B.: A general map-matching algorithm for transport telematics applications, GPS Solutions, Vol.7 , pp. 157-167, 2003

[11] Honey, S.K., Zavoli, W.B., Milnes, K.A., Phillips, A.C., White, M.S., Loughmiller, G.E.: Vehicle navigational system and method, United States Patent No., 4796191, 1989

[12] Zhao, Y.: Vehicle Location and Navigation System. Artech House, Inc., MA., 1997

[13] Ochieng, W.Y., Quddus, M.A., Noland, R.B.: Map-matching in complex urban road networks, Brazilian Journal of Cartography (Revista Brasileira de Cartografia), Vol. 55(2), pp. 1-18.

[14] Najjar, M.E., Bonnifait, P.: A roadmap matching method for precise vehicle localization using belief theory and Kalman filtering. In: The 11th International Conference in Advanced Robotics, Coimbra, Portugal, 2003

[15] 三谷卓摩, 羽藤英二:パーティクルフィルタを用いた空間データの自動作成法, 土木計画学研究・講演集, Vol.40, CD-ROM, 2009.

[16] Jo, T., Haseyamai, M., Kitajima, H.: A map-matching method with the innovation of the Kalman filtering, IEICE Trans. Fund. Electron. Comm. Comput. Sci. E79-A, pp. 1853-1855, 1996

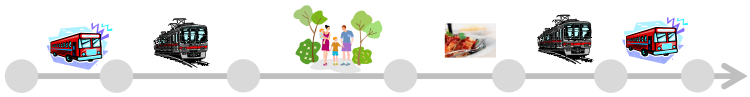
3.1.2 行動データの分析

本節では，実際の行動データに対する応用例を示す.

(a) 活動パターン分析

sequential aliment method

活動パターンの分類は行動分析における一つの重要なトピックである．活動パターンの分析は主に空間情報や社会経済変数を用いて行われ，活動パターン生成シミュレータを用いた需要予測や交通シミュレーションの評価などに用いられる．従来より用いられてきた方法として活動パターン同士のユークリッド距離を用いて相互の関係を評価する方法があるが，この方法では活動の連続性を評価することができない．遺伝子配列の分析や言語処理では活動の連続性（系列）を考慮した類似度の指標が用いられているが，一次元の配列パターンの評価であることが多く，交通分野において対象とする空間や時間，社会経済変数など（図 3.48）多次元の情報とその相互作用を扱うには不十分で



活動パターン	移動	移動	娯楽	食事	移動	移動
活動場所			公園A	レストランA		
開始・終了時刻	9:00-9:30	9:30-10:00	10:00-13:45	13:45-15:20	15:20-15:50	15:50-16:25
交通機関	バス	鉄道	徒歩	-	鉄道	バス

図 3.48 活動パターン例

ある．Chang-Hyeon et al(2002) では，従来の一次元の sequential aliment method を多次元の変数に拡張した分析手法を示した．

問題の定義

2 つの多次元アクティビティパターン（ソースパターン，ターゲットパターン）を比較する問題を考える．ここで，それぞれのアクティビティパターンは，$\boldsymbol{K}$ 次元の属性を持ち，それぞれ $\boldsymbol{m}$,$\boldsymbol{n}$ 次元のシーケンスを持つとする．このとき，ソースパターン $\boldsymbol{s}$ とターゲットパターン $\boldsymbol{g}$ を比較するとは，$\boldsymbol{K}\times\boldsymbol{m}$ のマトリクスと $\boldsymbol{K}\times\boldsymbol{n}$ のマトリクスを比較することになる．

$$\boldsymbol{s} = \boldsymbol{s}[\boldsymbol{s_1}\cdots\boldsymbol{s_k}\cdots\boldsymbol{s_K}]' \tag{3.106}$$

ただし，

$$\boldsymbol{s_k} = \boldsymbol{s_k}[\boldsymbol{s_{k0}}\cdots\boldsymbol{s_{ki}}\cdots\boldsymbol{s_{km}}] \tag{3.107}$$

$$\boldsymbol{g} = \boldsymbol{g}[\boldsymbol{g_1}\cdots\boldsymbol{g_k}\cdots\boldsymbol{g_K}]' \tag{3.108}$$

ただし，

$$\boldsymbol{g_k} = \boldsymbol{g_k}[\boldsymbol{g_{k0}}\cdots\boldsymbol{g_{ki}}\cdots\boldsymbol{g_{kn}}] \tag{3.109}$$

ここで，$\boldsymbol{s_k}$,$\boldsymbol{g_k}$ はそれぞれソースパターン，ターゲットパターンの $\boldsymbol{k}$ 番目配列の属性ベクトルである．

類似度の定義と計算例

Moves in the table	Operations meant	Resulting s	Cumulative cost
arrow (①) (1st diagonal move)	*identity* of A: A = A	ACB	0
arrow (②) (horizontal move)	*insertion* of B after A	ABCB	1
arrow (③) (2nd diagonal move)	*identity* of C: C = C	ABCB	1
arrow (④) (vertical move)	*deletion* of B at the last	ABC	2

図 3.50 類似度コスト途中計算例（Chang-Hyeon et al(2002)）

簡単のため，まず一次元の属性におけるアクティビティパターンの比較を考える．ここで，2 つのアクティビティの類似度を，2 つのパターンが同じ配列になるために必要な最小の配列要素の操作（削除，挿入，置換，同一）コストによって 2 つのアクティビティ間の類似度を定義する．この手法を sequential aliment method と呼ぶ.

図 3.49 には, 類似度コスト計算の例を示した．ここでは, $\boldsymbol{s} = \boldsymbol{s}[\boldsymbol{ACB}]$,$\boldsymbol{g} = \boldsymbol{g}[\boldsymbol{ABC}]$ のパターンを考え，$\boldsymbol{s}$ が $\boldsymbol{g}$ と等しくなるための最小コストをもつ操作パターンを求める．ただし，削除，挿入，置換，同一のコストをそれぞれ 1,1,2,0 とする．

具体的な計算は，動的計画法を用いて直前の操作と最小コストを記憶しながら図 3.49 の表を埋めていく．図 3.50 に途中計算例を示した．この例では，最適な操作パターンは（同一，挿入，同一，削除）となり，そのときのコストは 2 となる．

多次元への拡張

前節で述べた手法は，2 つのパターンを同一にするための最小オペレーションコストであり，それぞれのパタンの連続性（順序）を考慮した類似度計算手法といえる．ただし，単一の属性における系列類似度の評価であり多次元の属性間の相互依存性を考慮した系列類似度評価は行えない．　Chang-Hyeon

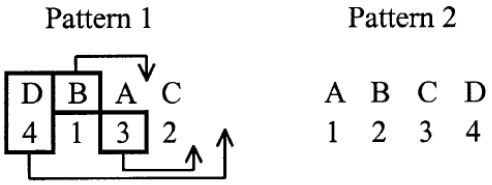


図 3.51　2 次元の類似度概念（Chang-Hyeon et al(2002)）

et al(2002) は，この手法を多次元に拡張し，多次元の属性の各操作の set を一つの操作としてみなす方法を提案している．多次元の属性に関する操作の組み合わせを一つの操作とみなし評価する方法といえる．具体的な例を図 3.51 に示した．一次元の操作であれば，D の削除，挿入と 4 の削除，挿入は別々の操作となり，図 3.51 の例では合計 8 回の操作が必要となるが，提案手法では D,4 の削除，挿入をセットで一つの操作としてみなすため，全体で 6 回の操作となる．

このように集合での操作を認める場合，そのコストをどう設定するかが問題になる．従来法では，集合内の 2 つの操作コストの平均，中央値，最大値をとるなどの方法が存在するが，ここでは最大値を用いて以下のように定義する．

$$\boldsymbol{max(\beta_1, \beta_2) \times w_d} \tag{3.110}$$

ただし，$\boldsymbol{w_d}$ は削除コスト，$\boldsymbol{\beta_k}$ は $\boldsymbol{k}$ 番目の属性の重みとする．このように置くことで，多次元の属性が同時に生じることを表現することができる．現実的には，すべての組み合わせを考慮することは非常に計算コストが高いため，次に示すようなヒューリスティックな工夫によって計算コストを低減する．

多次元パターンの計算手法

具体的な多次元での計算手法を以下に述べる．

ステップ 1

各属性における一次元での最小コスト（操作パターンを求める）

以下のように，各属性での操作とその位置を記憶する．

$$\boldsymbol{O_{k,l} = \{p | p = d(i,k) \vee i(j,k) \vee s(i,j,k)\}} \tag{3.111}$$

ただし，$\boldsymbol{i \in \{1,...,m\}}$,$\boldsymbol{j \in \{1,...,n\}}$,$\boldsymbol{1 \leq k \leq K}$
ここで，$\boldsymbol{O_{k,l}}$ は $\boldsymbol{k}$ 番目の属性系列の $\boldsymbol{l}$ 番目の一次元オペレーションを表す．$\boldsymbol{d(i,k)}$ は $\boldsymbol{i}$ 番目の要素の削除，$\boldsymbol{i(j,k)}$ は $\boldsymbol{j}$ 番目への要素の挿入，$\boldsymbol{s(i,j,k)}$ は $\boldsymbol{i}$ 番目から $\boldsymbol{j}$ 番目の要素への置換を表す．

ステップ 2

ステップ 1 で求めた一次元の属性での最小コスト操作パターンを多次元の属性間のアライメント操作へ統合する
ここで，$\boldsymbol{T_k}$ を $\boldsymbol{k}$ 番目の一次元のオペレーション集合とすると，その組み合わせ集合は $\boldsymbol{T}$ は以下のように与えられる．

$$\boldsymbol{T = \prod_{k=1}^{K} T_k} \tag{3.112}$$

また，$\boldsymbol{O^t}$ を $\boldsymbol{t}$ 次元のオペレーションセットとすると，以下のように書ける．

$$\boldsymbol{O^t = \{p | p = d(i,k) \vee i(j,k) \vee s(i,j,k)\}} \tag{3.113}$$

【具体例】
今，次のような 3 つの属性に関する一次元オペレーションセットがあるとする．

$$\boldsymbol{O^t_{1,l'} = \{p | p = d(2,1), i(4,1), s(5,6,1)\}} \tag{3.114}$$

$$\boldsymbol{O^t_{2,l} = \{p | p = d(3,2), i(4,2), s(5,5,2)\}} \tag{3.115}$$

$$\boldsymbol{O^t_{3,l''} = \{p | p = d(2,3), i(4,3), s(5,5,3)\}} \tag{3.116}$$

このとき、$\boldsymbol{t}$ 番目の多次元オペレーションセットは以下のようにかける．

$$\boldsymbol{O^t = \{p | p = d(2,\{1,3\}), d(3,\{2\}), i(4,\{1,2,3\}), s(5,5,\{2,3\}), s(5,6,\{1\})\}} \tag{3.117}$$

Source pattern				Target pattern						
A	D	B	C	A	B	C	D	E	→	activity-type sequence
1	6	2	3	1	2	3	4	5	→	activity-location sequence
a	b	c	f	a	b	c	d	e	→	transport-mode sequence
α	δ	φ	γ	α	φ	γ	δ	ε	→	accompanying-person sequence

図 3.52　アクティビティパターンへの適用例（Chang-Hyeon et al(2002)）

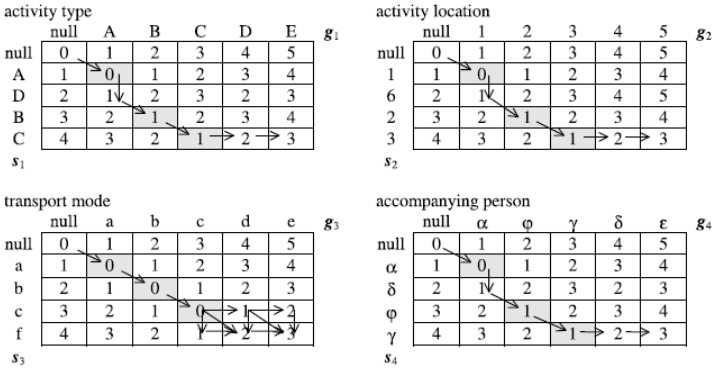


図 3.53　アクティビティパターンでの計算例（Chang-Hyeon et al(2002)）

ステップ 3

統合されたアライメント操作の総コストを求める.

$$C^t = \sum_{p \in O_t} c_p \tag{3.118}$$

$$c_p = \begin{cases} w_d \beta_k^{max} & if p = d(i,\mathrm{k}) \\ w_i \beta_k^{max} & if p = i(j,k) \\ w_s \beta_k^{max} & if p = s(i,j,k) \end{cases} \tag{3.119}$$

アクティビティパターンへの適用例

ここでは，前節までの手法を用いて実際のアクティビティパターンに適用した例を示す.

まとめ

このように，異なる次元のアクティビティパターンを多次元に拡張された

sequential aliment method によって統一的に評価することが可能となる．このような手法は，アクティビティパターンの生成過程を記述したりシミュレータのアクティビティパタン生成モデルに利用できる可能性がある．アクティビティを符号化して操作することで本来連続的な属性も扱いやすくすることが可能である．個々の属性の重みや各操作のコスト設定に関してはさまざまな設定が可能であり，より記述力の高い重み設定を工夫する余地もあるといえる．

3.2　行動モデル

3.2.1　離散選択モデル

(a)　効用と選択確率の定式化

効用の定式化

離散選択モデルでは，選択肢ごとに効用をある関数により与えた上で，選択肢間の効用の比較によって選択結果が導出される．選択肢の効用は直接観測することはできないが，効用は連続変数であり，かつ説明変数の線形和であると仮定する．効用関数は次となる．

$$\boldsymbol{V_{in} = \beta_1 x_{1in} + \beta_2 x_{2in} + \cdots + \beta_K x_{Kin}} \tag{3.120}$$

ここで，$\boldsymbol{V_{in}}$ は個人 $\boldsymbol{n}$ の選択肢 $\boldsymbol{i}$ に対する効用を示す．$\boldsymbol{x_{kin}}$ は個人 $\boldsymbol{n}$ の選択肢 $\boldsymbol{i}$ に対する $\boldsymbol{k}$ 番目の説明変数，$\boldsymbol{\beta_k}$ は $\boldsymbol{k}$ 番目の未知パラメータを示す．交通手段選択モデルの場合には，説明変数には所要時間・費用・乗り換え回数・交通手段までのアクセス時間等の交通手段別特性と，移動目的・性別・年齢等の個人ごとの移動・個人特性が用いられる．

説明変数 ($\boldsymbol{x_{kin}}$) の観測誤差や個人個人の重み付けパラメータ ($\boldsymbol{\beta_k}$) の差異が実際は存在しており，式 (3.120) の効用関数は測定可能な内容による近似に過ぎないと考えられる．そこで，真の効用関数は測定可能な効用と不可能な効用による和で表す．測定不可能な効用を確率変数を用いて表現し，真の効用関数は次となる．