



ヒューマンコンピュータインタラクション特論 第5回

情報理工学部門 複合情報工学分野

Human-Computer Interaction (HCI) 研究室

小野哲雄 (8-12室)

tono@ist.hokudai.ac.jp

授業計画 (予定)

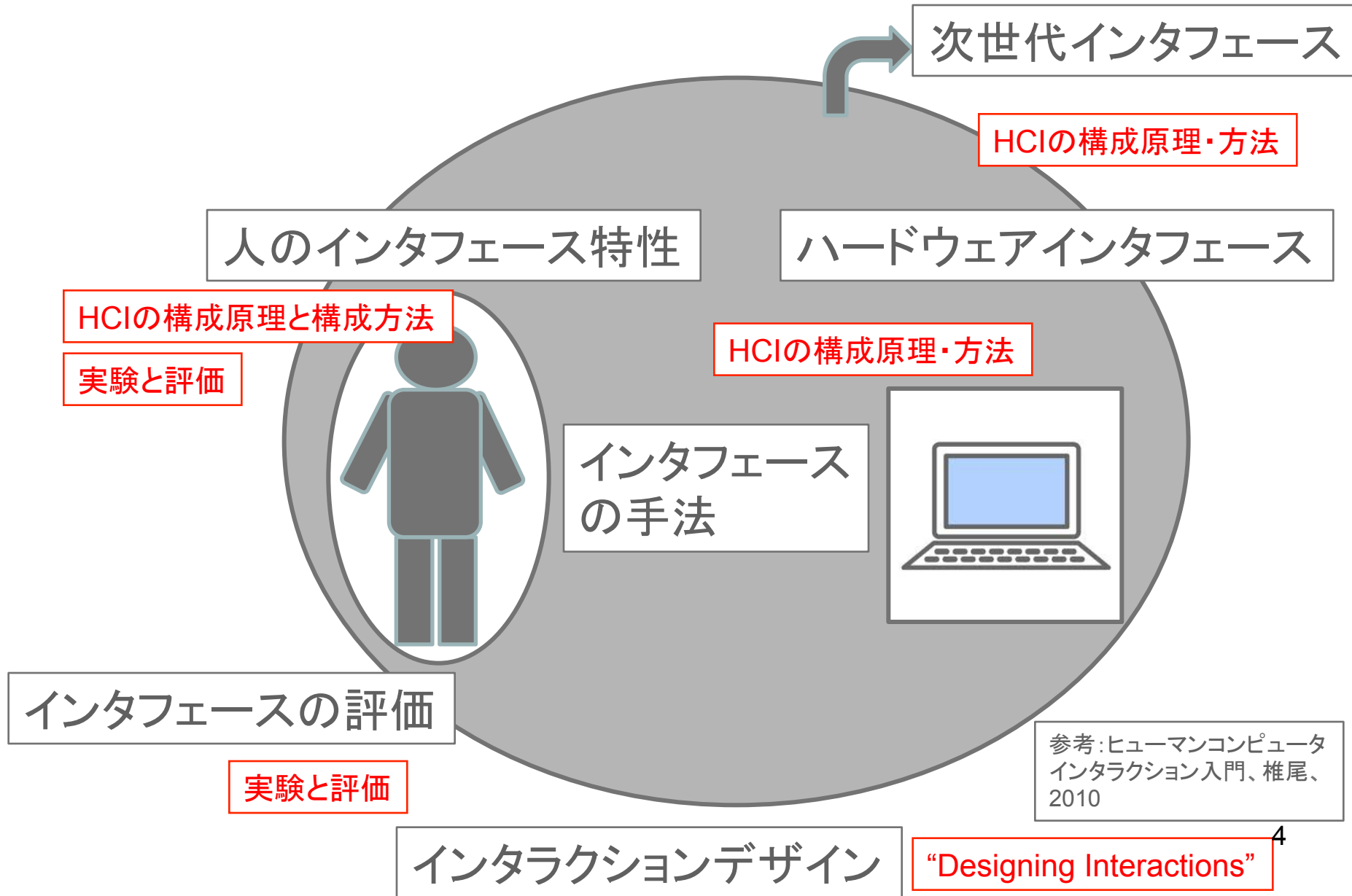
1. 10/1(火) 4限 (小野) HCI入門
2. 10/4(金) 3限 (小野) HCI概論
3. 10/8(火) 4限 : **[休講]**
4. 10/11(金) 3限 (小野) HCI概論(2) # 論文発表の説明
5. 10/15(火) 4限 (小野) HCIの構成原理と構成方法(1) # 登録開始
6. 10/25(金) 3限 (小野) HCIの構成原理と構成方法(2)
7. 10/31(木) 4限 <- 火曜授業 (小野) **[論文発表1(1)]**
8. 11/1(金) 3限 (小野) **[論文発表1(2)]**
9. 11/5(火) 4限 (小野) ヒューマンロボットインタラクション(HRI) # 説明
10. 11/8(金) 3限 (小野) ヒューマンエージェントインタラクション(HAI) #登録
11. 11/12(火) 4限 (小野) Predicting Human Decision-Making (1)
12. 11/15(金) 3限 (小野) Predicting Human Decision-Making (2)
13. 11/19(火) 4限 (坂本) HCIにおける実験と評価(1)
14. 11/22(金) 3限 (坂本) HCIにおける実験と評価(2)
15. 11/26(火) 4限 (小野) **[論文発表2(1)]**
16. 11/29(金) 3限 (小野) **[論文発表2(2)]**

HCIの構成原理と構成方法 (2)

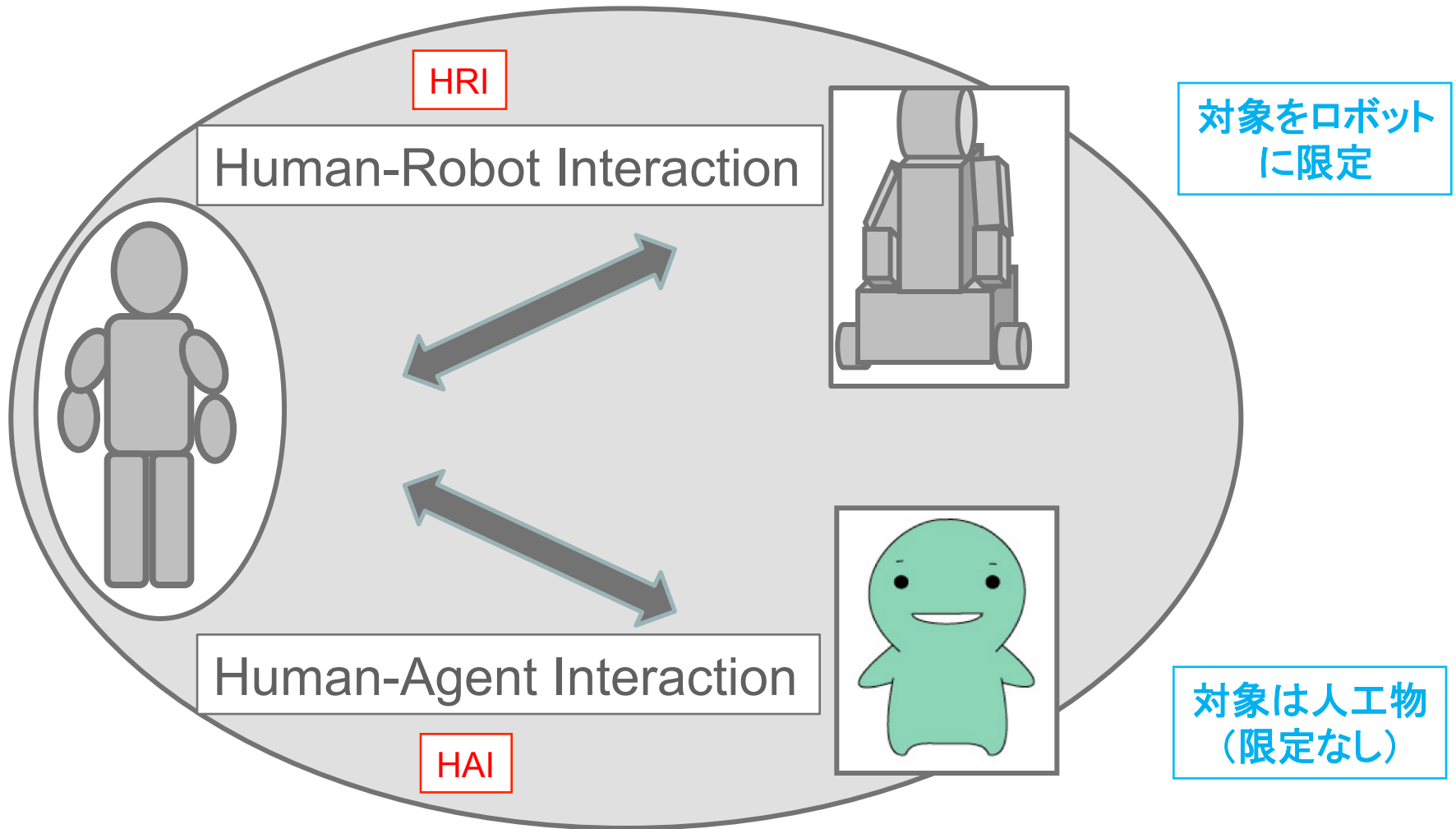
インタフェースの手法：

入力(認識)・出力

講義の概要（1）：HCI全般



講義の概要 (2) : HRIとHAI



HCIとの相違点：人と「エージェント」のインタラクションデザインを研究
HCIとの共通点：人を含む「開放系」を研究の対象とする

前回：HCIの構成原理と構成方法（1）

インタフェースの手法：入力・出力

－ 出力インタフェースの手法

- ・ 従来の出力デバイス、Virtual Reality、Augmented Reality
- ・ 特徴
 - － 理論的なアプローチがしづらい→正解がみつけづらい
 - － 製作（試作）のコストが高い
 - － 応用先は多い（特に、エンタテインメント系）

－ 入力インタフェースの手法

- ・ 深層学習 (CNN)
- ・ 識別(SVM)
- ・ 認識アルゴリズム (DPマッチング、隠れマルコフモデル)
- ・ 部分観測マルコフ決定過程 (POMDP)



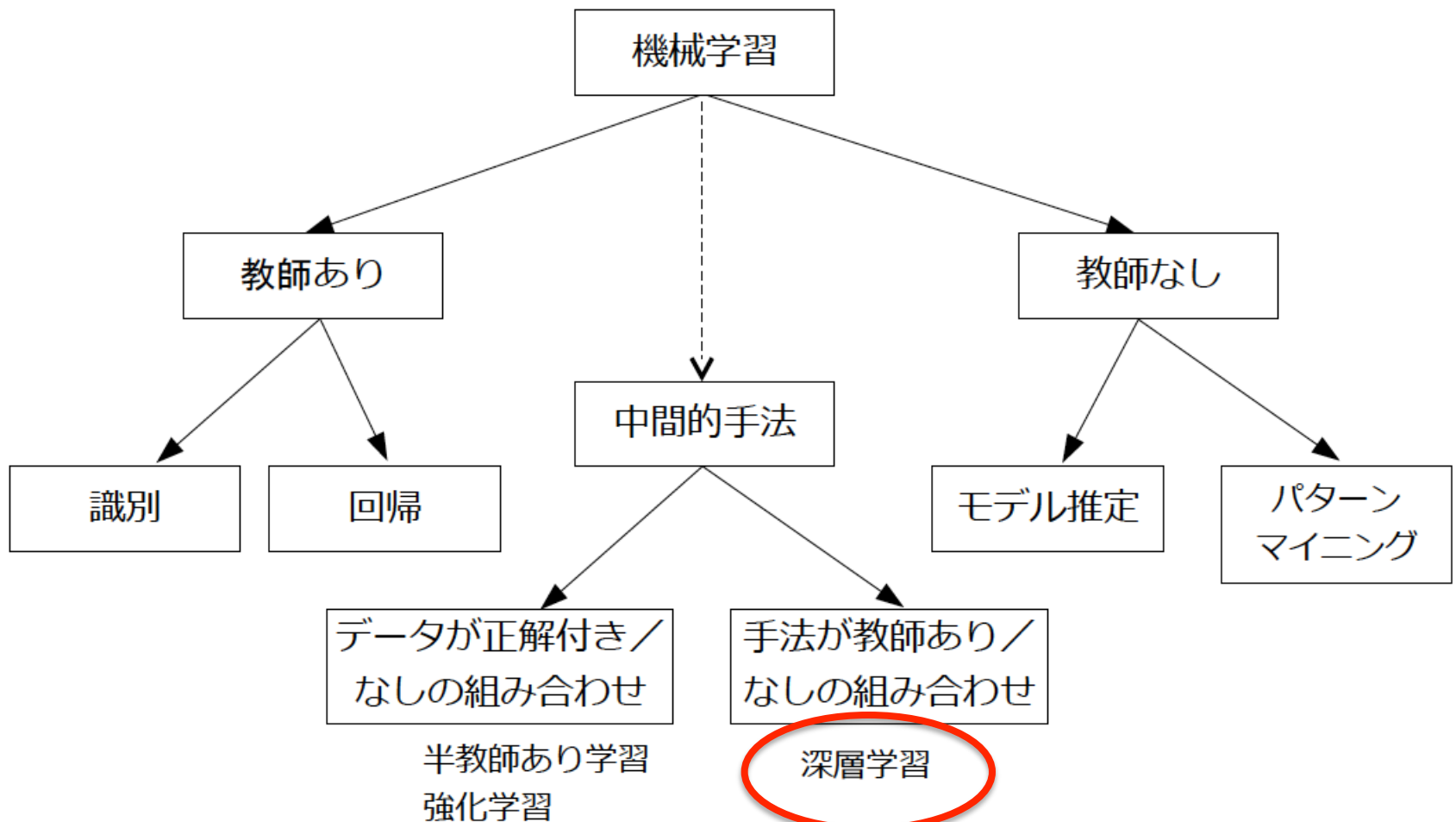
HCIの構成原理と構成方法 (2)

HCIと機械学習

CNN

HCIにおける機械学習とは？(1)

ー 機械学習の分類 (荒木 14)



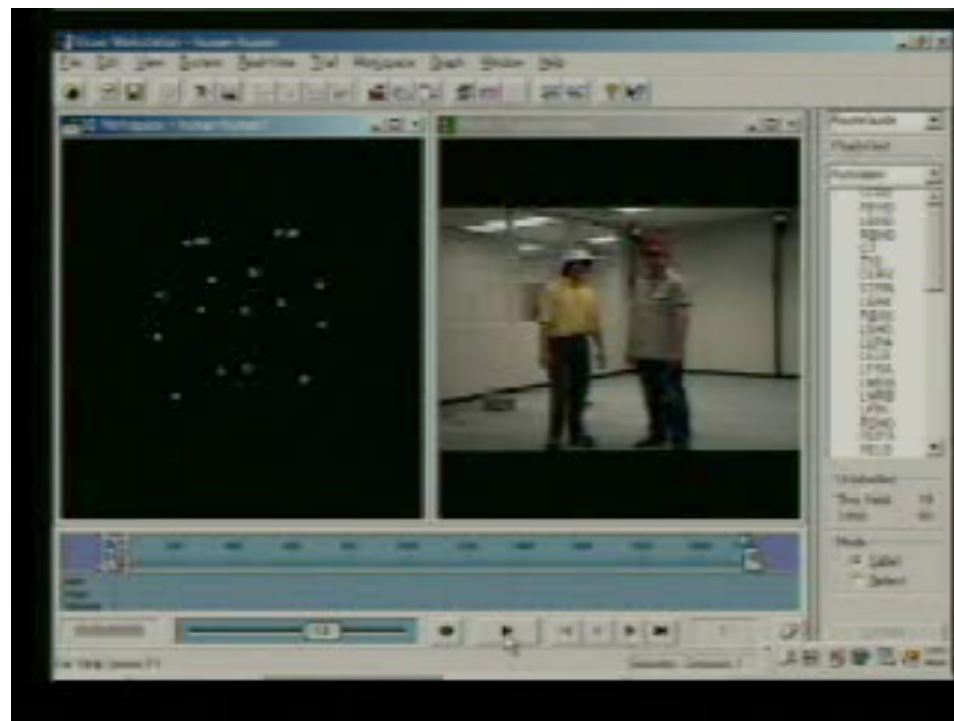
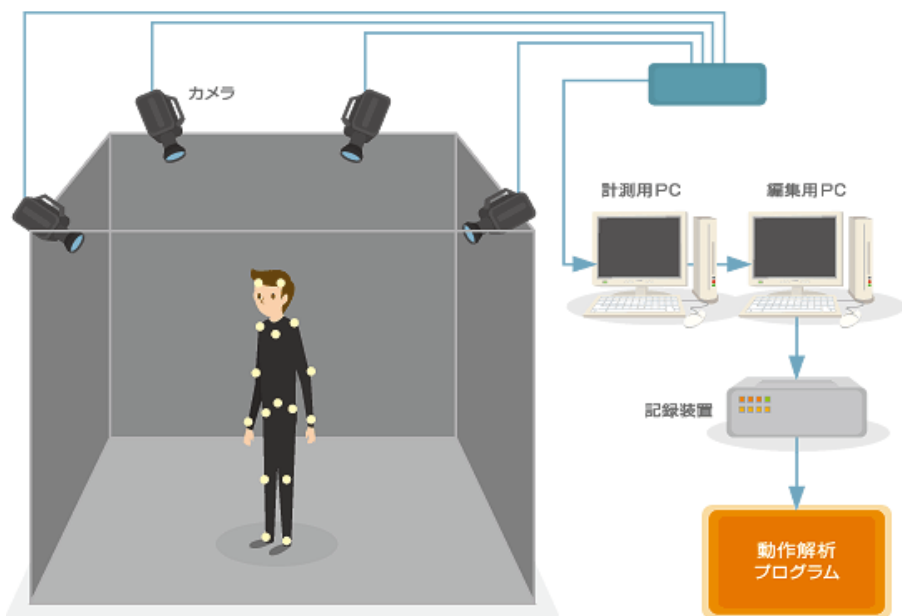


OpenPose

人物の姿勢推定

- モーションキャプチャを用いた動作計測

光学式モーションキャプチャ



OpenPose (2016)



OpenPose (1)

– 論文情報

- “Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields”
- Zhe Cao Tomas Simon Shih-En Wei Yaser Sheikh (CMU)
- CVPR 2017

– 概要

- 身体部位の関係性 (part affinity) を活かした、リアルタイムの2D姿勢推定 (pose estimation)
- CNNの組み合わせで画像内の人物姿勢の推定
- NP-hard問題を緩和法 (relaxation) を用いて解く
- top-downアプローチではなく、bottom-upアプローチ
- 実時間処理可能

OpenPose (2)

– 姿勢推定の難しさ

- 複数人写っている2D画像中の各人の身体部位の位置を検知
- 困難な点
 - 人数、スケールが不定
 - 人同士のインタラクション(occlusionによる干渉)
 - 人数に比例して計算量増加

– 従来手法

- top-downアプローチ：人の検知を行い、その後、個人の姿勢推定
- 問題点
 - ✓ 画像内の人の検知に失敗すると、姿勢推定できない
 - ✓ 人数に比例して計算量増大

OpenPose (3)

– 提案手法

- 画像中の身体部位の位置のみではなく、部位間の関連度(affinity)もエンコードする
- 一度に画像全体の中の複数人に対して姿勢推定を行うbottom-upアプローチ → 推定の精度・速度の改善

– 提案手法のポイント

- エンコーディングを学習するCNNユニットのステージを繰り返す
- 身体部位の位置、部位間の関連度のエンコーディングを共通で学習するアーキテクチャ

– 技術・発想の違い

- 斬新な発想、価値の高いソースを公開 → **破壊的！**

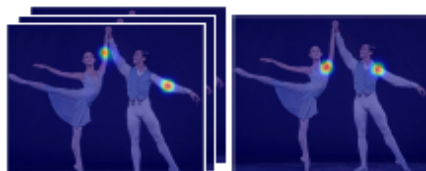
OpenPose (4)

– 処理手順

- 入力画像
- 各チャンネルで部位の位置をエンコード
- 部位間の関連度のエンコード：位置情報+向き情報を含んだ表現=Part Affinity Fields
- 各部位のつながり合わせ部分を2Dベクトルでエンコードできるフィルタを学習
- 姿勢推定完了



(a) Input Image



(b) Part Confidence Maps



(c) Part Affinity Fields



(d) Bipartite Matching

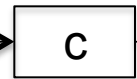


(e) Parsing Results

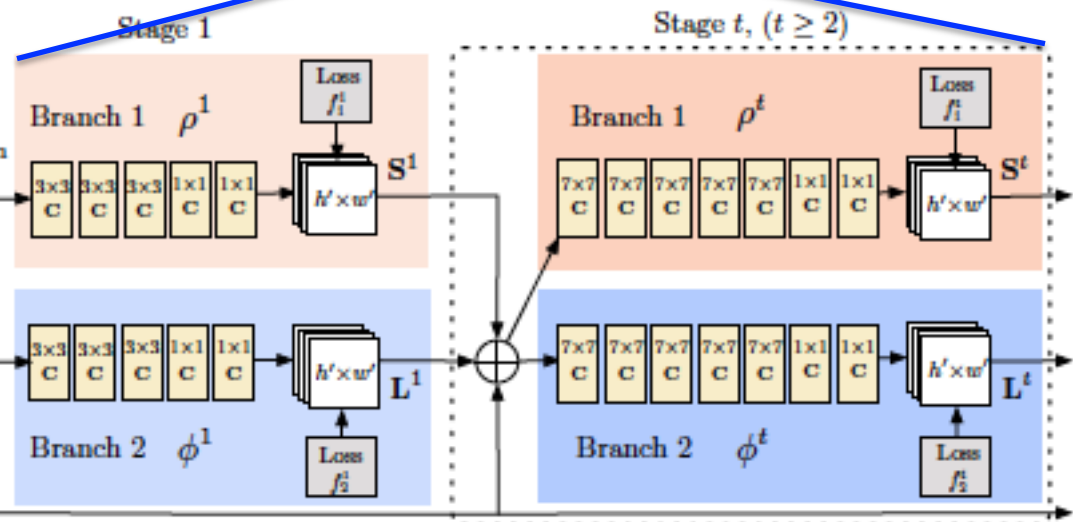
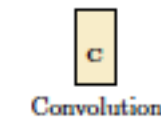
OpenPose (5)

– 提案アーキテクチャ

- Branch1 : (b) confidence mapの予測
- Branch2 : (c) part affinity fieldsの予測
- 同じBranch1, 2を持つユニットを繰り返す(Stage1~6)
- 各Stageの入力は、1つ前のStageにおけるBranch1, 2, Fを連結したもの
- Stageごとに誤差を計算



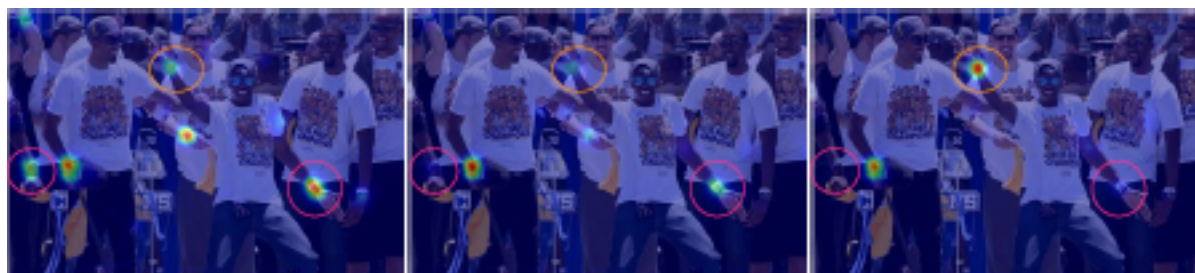
特徴ベクトル



OpenPose (6)

- Stageを重ねるごとに精度が上がる
 - 上：(b) confidence map. 右手首検知のチャンネル. Stageを重ねるごとに右手首のconfidenceが上がる
 - 下：c) part affinity field. 右上腕の流れをエンコードするチャンネル. Stageを重ねるごとに右上腕の確からしさが上がる

右手首検知



右上腕検知



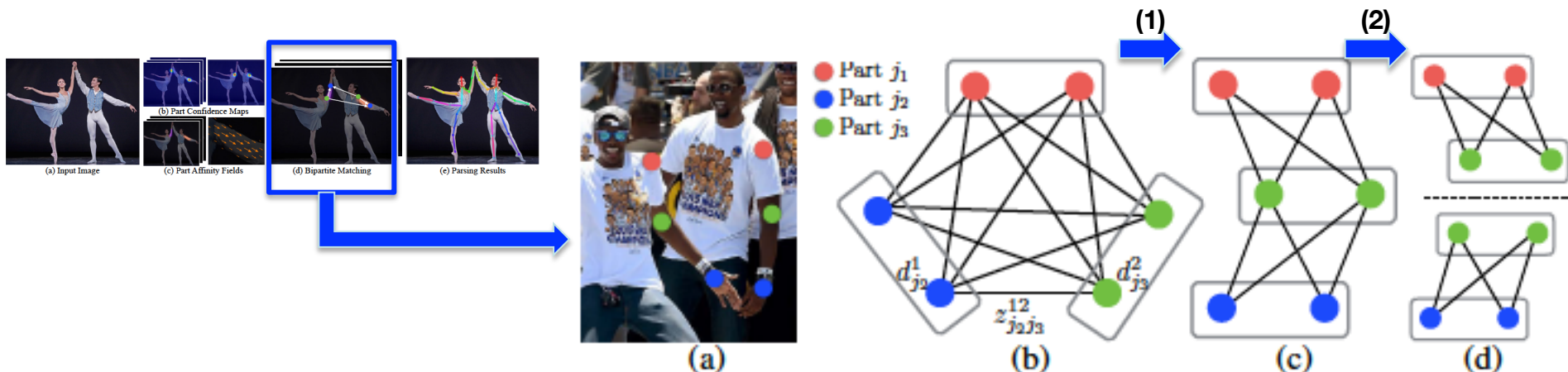
Stage 1

Stage 3

Stage 6

OpenPose (7)

- 各部位のつながりの確からしさの最大化
 - 検知された部位(part confidence map)を、つながり(part affinity fields)の確からしさを最大化するように組み合わせる
 - 組み合わせ問題は、網羅的に計算するとNP-Hardな問題となるため、以下の緩和法を導入
 - (1) 各部位につながる部位の事前情報を与える(例. 右肩とつながるのは、首と右肘のみ)
 - (2) 隣り合う部位のみをみて、part affinity fieldsの確からしさを最大化するように組み合わせる



OpenPose (8)

アーキテクチャの定式化：誤差関数、目的関数

- 誤差関数

各部位 J, 各つながり C

Pでのアノテーションがあるかどうかで 0 or 1

Stage tにおける
Part confidence map S

$$f_S^t = \sum_{j=1}^J \sum_{\mathbf{p}} W(\mathbf{p}) \cdot \|\mathbf{S}_j^t(\mathbf{p}) - \mathbf{S}_j^*(\mathbf{p})\|_2^2,$$

予測値 正解値

Stage tにおける
Part affinity field L

$$f_L^t = \sum_{c=1}^C \sum_{\mathbf{p}} W(\mathbf{p}) \cdot \|\mathbf{L}_c^t(\mathbf{p}) - \mathbf{L}_c^*(\mathbf{p})\|_2^2,$$

各場所 P

- 全体の目的関数：全ステージの足し合わせ

$$f = \sum_{t=1}^T (f_S^t + f_L^t).$$

OpenPose (9)

- アーキテクチャの定式化
 - Part confidence mapの計算手法
 - Part affinity fieldsの計算手法
 - 部位間マッチングの計算手法
(論文参照)

OpenPose (10)

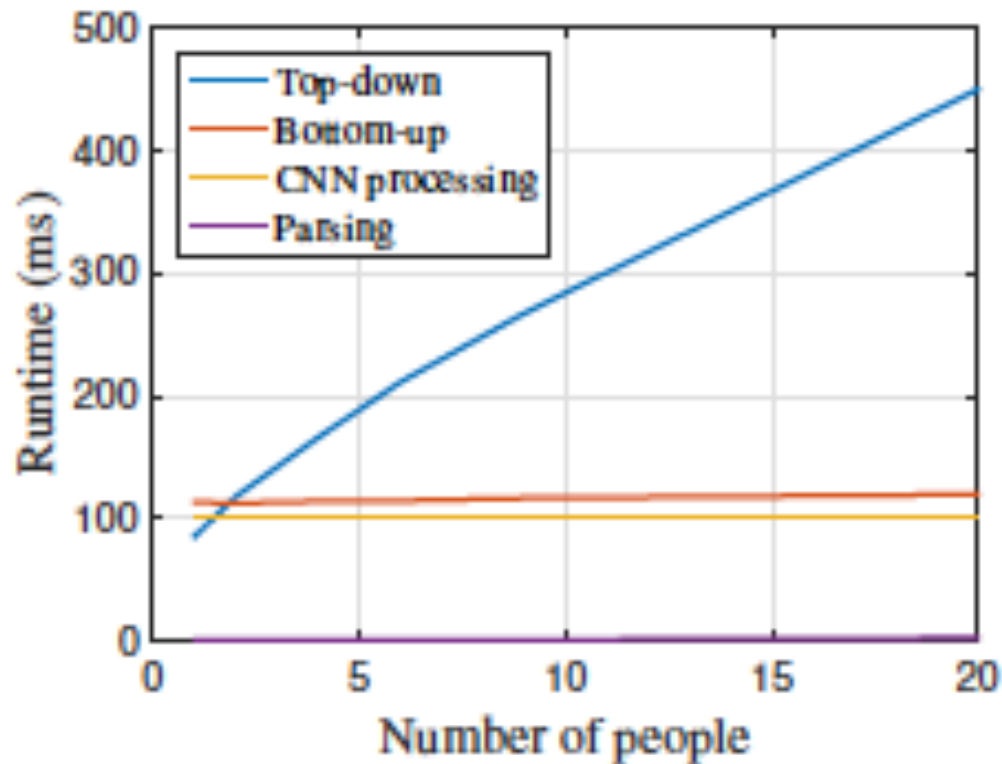
- 実験結果：MPII Multi-Person Dataset
 - 精度(mAP)：従来の手法DeeperCutを8.5%上回る
 - 速度(inference time)：6桁高速化

Method	Hea	Sho	Elb	Wri	Hip	Kne	Ank	mAP	s/image
Subset of 288 images as in [22]									
Deepcut [22]	73.4	71.8	57.9	39.9	56.7	44.0	32.0	54.1	57995
Iqbal et al. [12]	70.0	65.2	56.4	46.1	52.7	47.9	44.5	54.7	10
DeeperCut [11]	87.9	84.0	71.9	63.9	68.8	63.8	58.1	71.2	230
Ours	93.7	91.4	81.4	72.5	77.7	73.0	68.1	79.7	0.005
Full testing set									
DeeperCut [11]	78.4	72.5	60.2	51.0	57.2	52.0	45.4	59.5	485
Iqbal et al. [12]	58.4	53.9	44.5	35.0	42.2	36.7	31.1	43.1	10
Ours (one scale)	89.0	84.9	74.9	64.2	71.0	65.6	58.1	72.5	0.005
Ours	91.2	87.6	77.7	66.8	75.4	68.9	61.7	75.6	0.005

Table 1. Results on the MPII dataset. Top: Comparison result on the testing subset. Middle: Comparison results on the whole testing set. Testing without scale search is denoted as “(one scale)”.

OpenPose (11)

- 実験結果：人数による速度変化
 - top-downアプローチ：人数が増えると急激に増加
 - bottom-upアプローチ：ほとんど変化なし



OpenPose (12)

- 本論文のポイント
 - 部位の位置と方向の両方を符号化して表現
 - 部位検出と関連度を学習するアーキテクチャ
 - 人の数が増えても計算量抑え、十分な精度
- HCIの観点から
 - ある文脈における人物理解において、複数人物の姿勢推定は重要！
 - 3次元は無理なのかな？



DensePose

DesePose (2018)

– 論文

- “DensePose: Dense Human Pose Estimation In TheWild”
- Rıza Alp Guler, Natalia Neverova, Iasonas Kokkinos (INRIA, Facebook), CVPR 2018

– 概要

- 2Dの群衆動画から人間の姿勢を推定し、身体テクチャも貼ってくれるCNNを用いたシステム
 - 提案手法では、Mask-RCNNをベースにしたDensePose-RCNNアーキテクチャを用いて、人体の領域を検出し、パーツ上の3Dサーフェスモデルに対応する点を推定
 - つまり、複数人が密集した2D画像から人体それぞれの3Dモデルを計算し、画像ピクセルを人体のサーフェス座標に関連付けることができる

DensePose



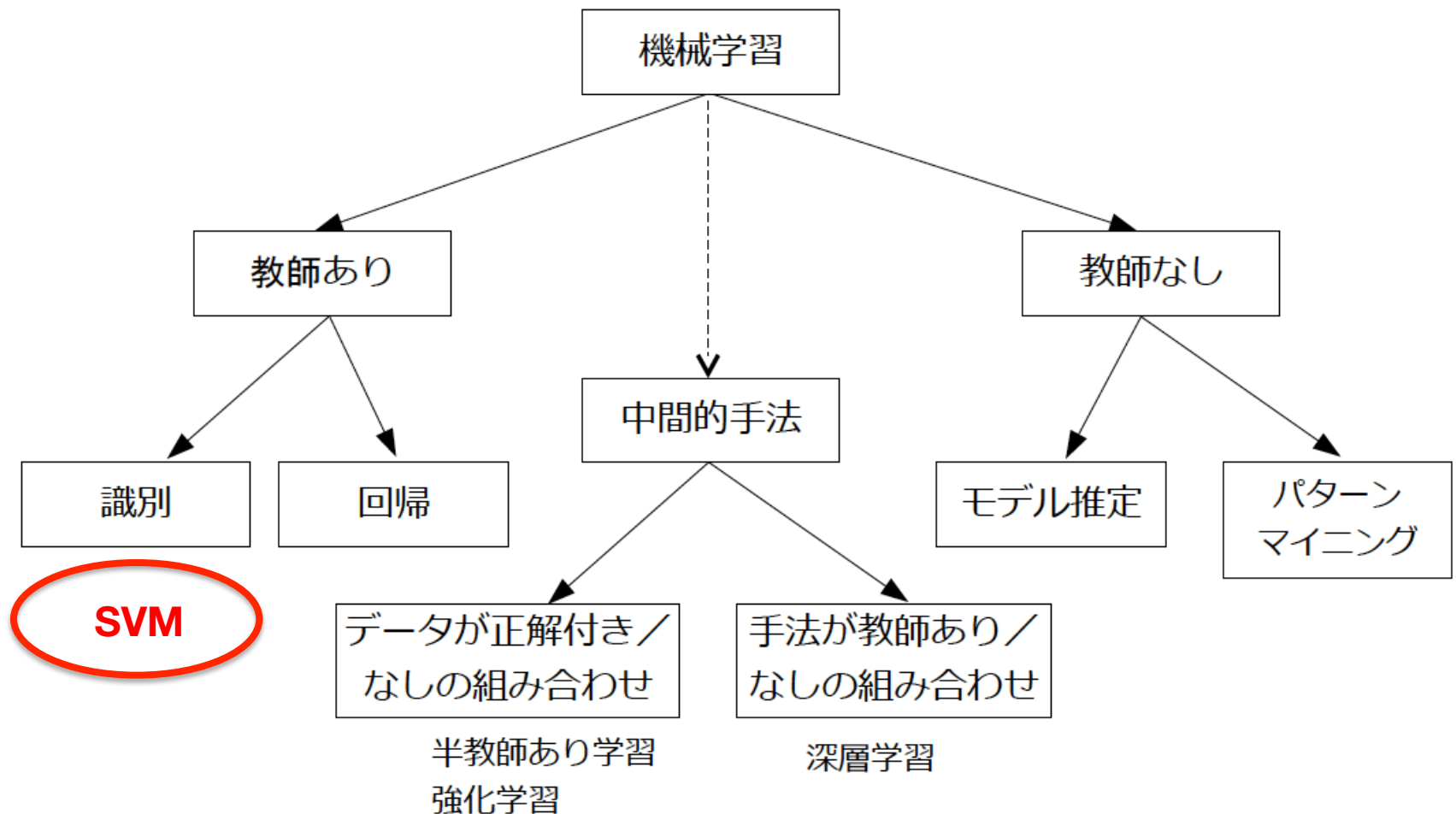


HCIと機械学習

SVM

HCIにおける機械学習とは？(1)

ー 機械学習の分類 (荒木 14)



SVMの応用事例（１）

“Touch & Activate: Adding Interactivity to Existing Objects Using Active Acoustic Sensing”,

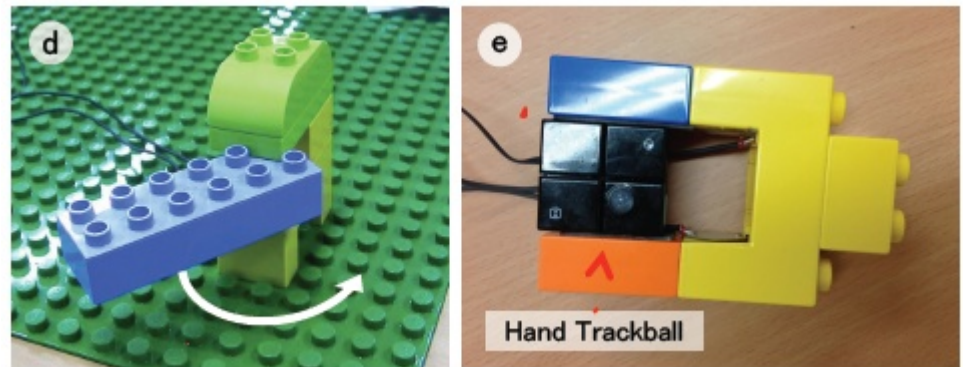
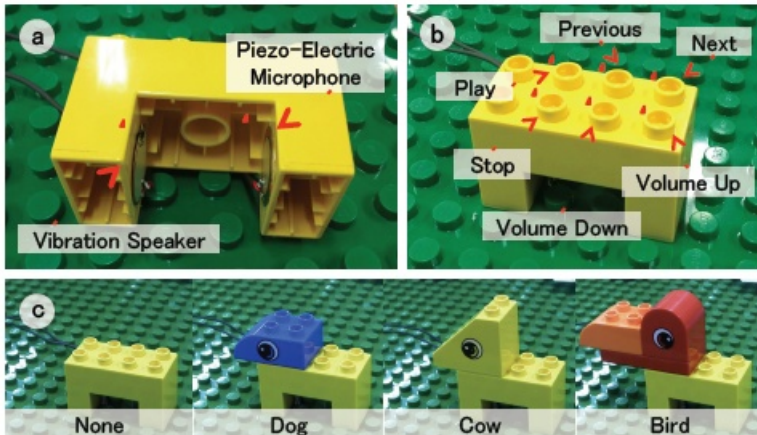
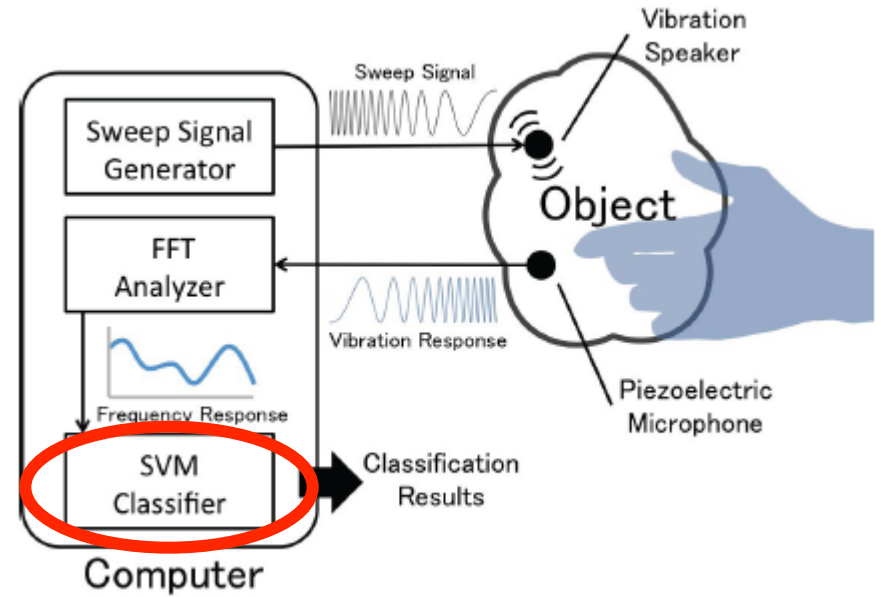
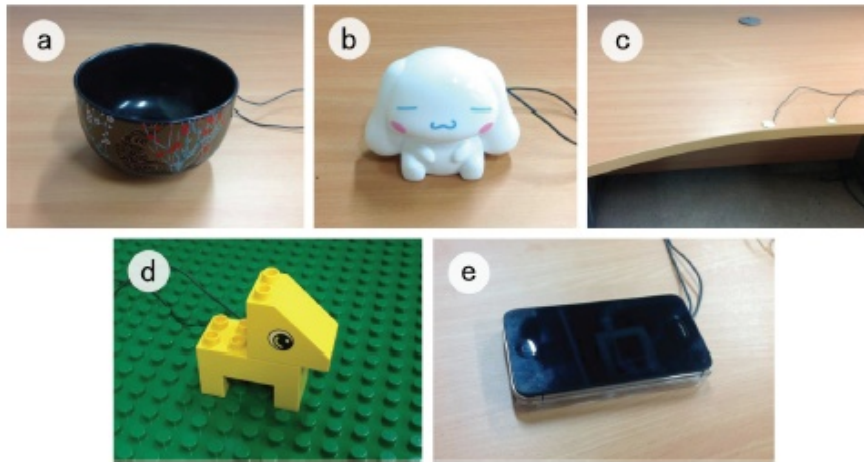
Makoto Ono, Buntarou Shizuki, Jiro Tanaka, UIST2013.

Touch & Activate: Adding Interactivity to Existing Objects
using Active Acoustic Sensing

Makoto Ono, Buntarou shizuki, and Jiro Tanaka
University of Tsukuba

SVMの応用事例（2）

– “Designing Interactions”を確認しよう！

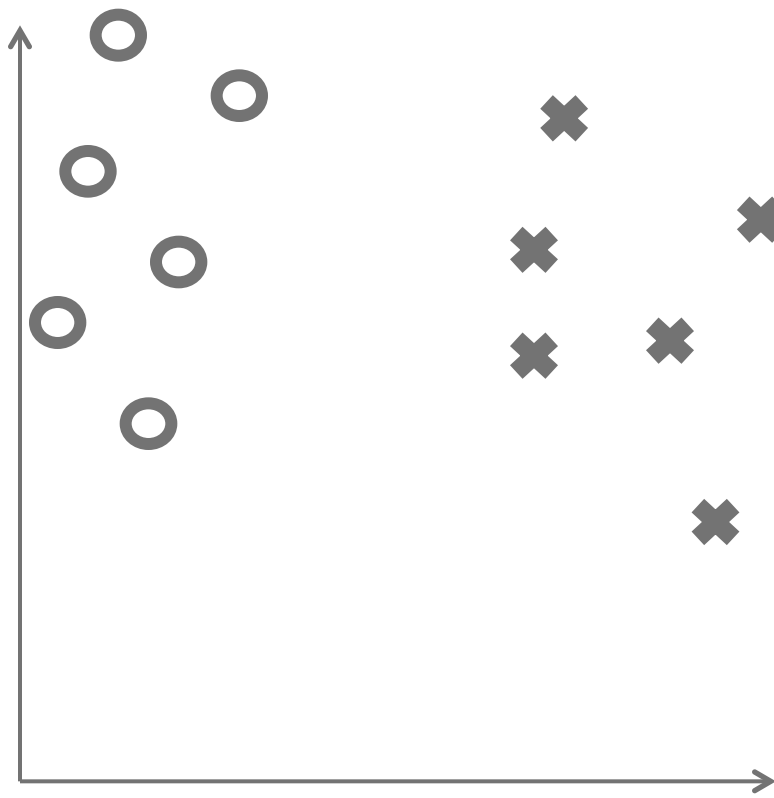
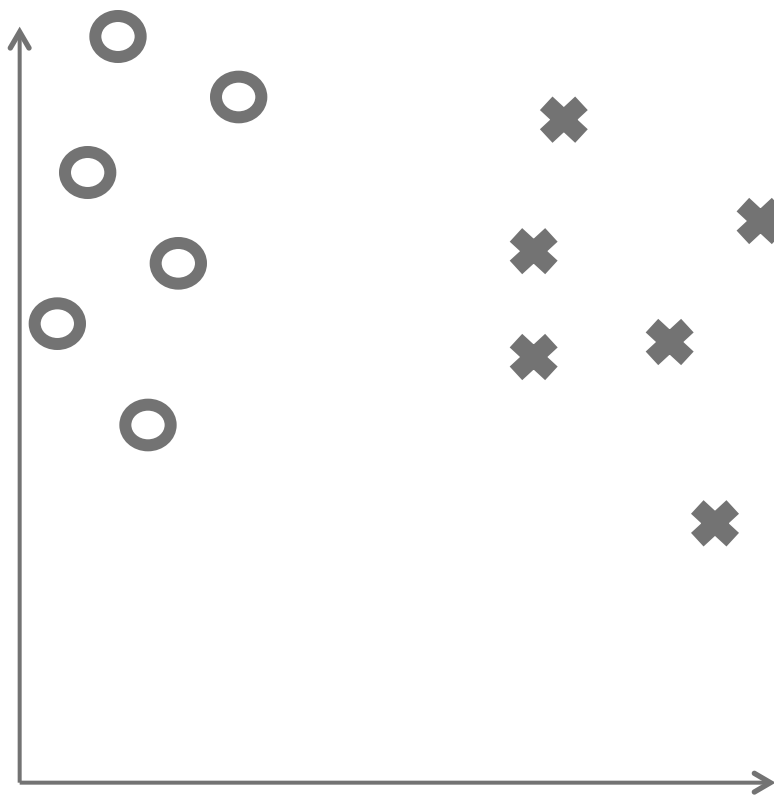


サポートベクターマシンとは？（1）

- サポートベクターマシン (support vector machine: SVM) :
 - 学習の分類: 「教師あり」・「識別」問題
 - 特徴:
 - 線形モデルを使って、学習データに特化し過ぎないような識別面を求める
- Cf. ニューラルネットワークによる学習:
非線形のモデル、過学習・局所最適解に陥る問題あり
- 線形で識別できないデータ:
 - » 誤識別に対するペナルティ
 - » 高次元の空間で線形識別を適用

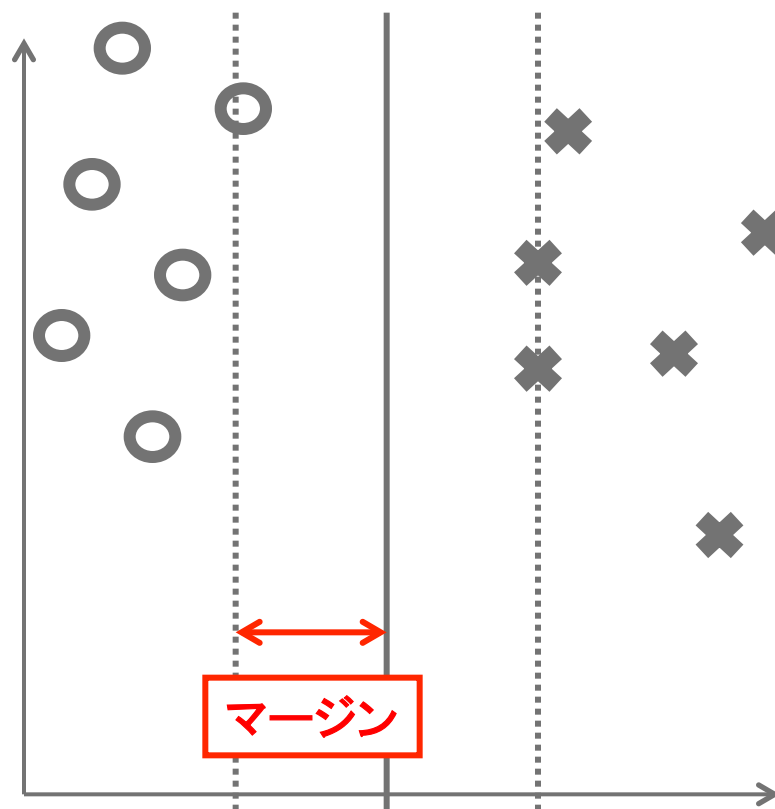
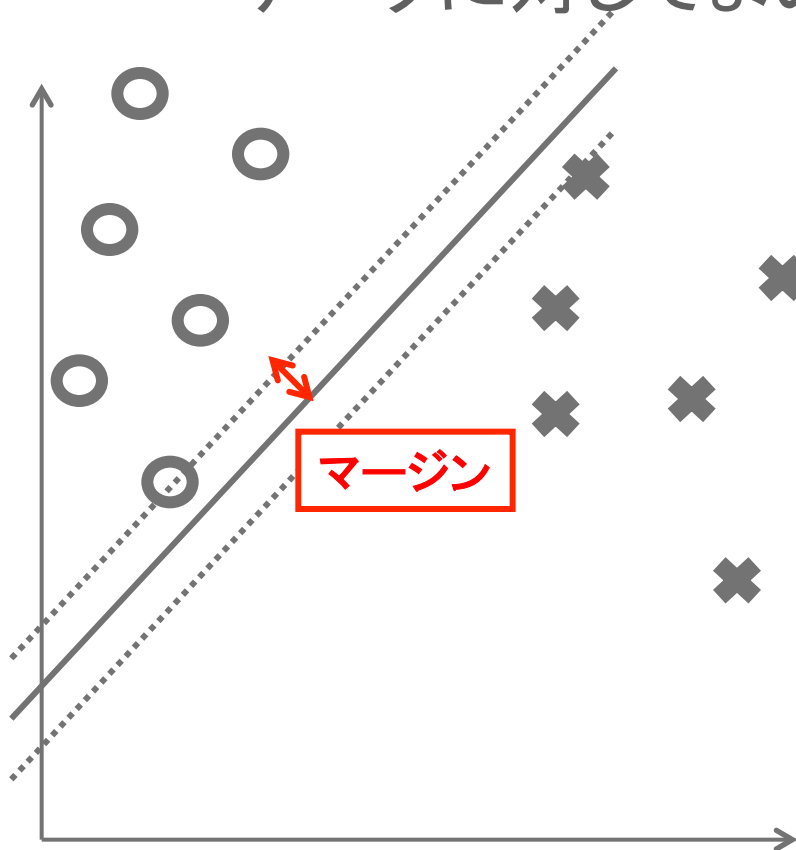
サポートベクターマシンとは？（2）

- データを識別する直線は無数に存在する



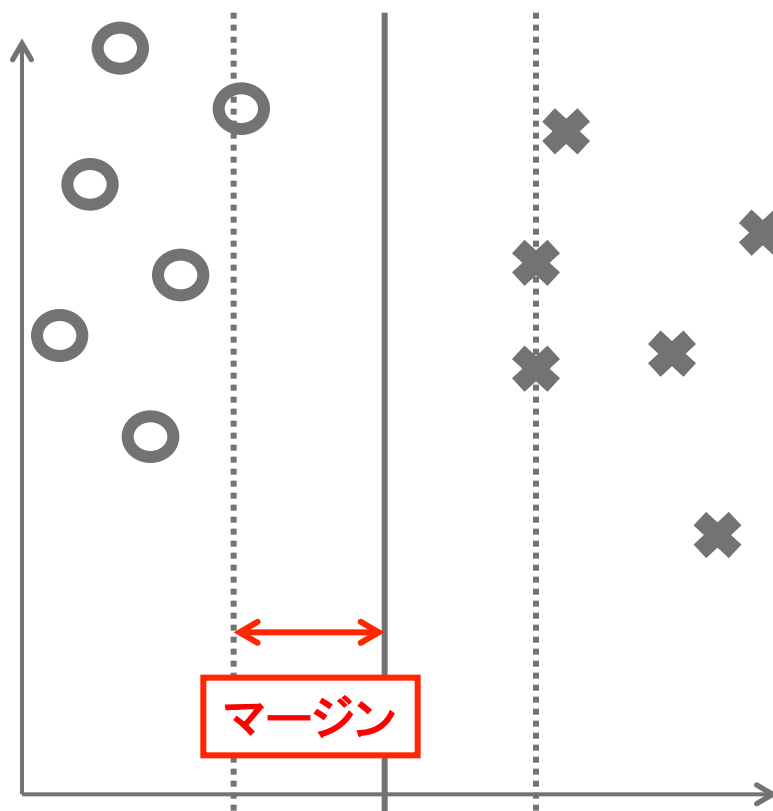
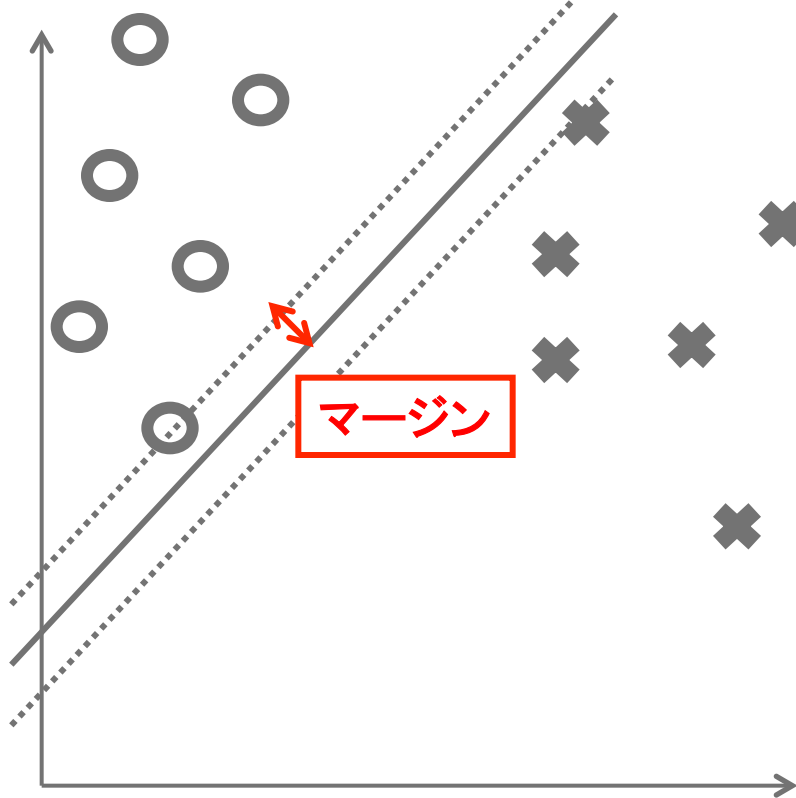
サポートベクターマシンとは？（2）

- 以下の場合、右図の識別境界線の方が、未知データに対してよい結果が得られる？



サポートベクターマシンとは？（2）

- **SVM**: 学習データからのマージンが最大となる識別境界線(識別超平面)を求める手法



マージン最大化のための定式化 (1)

- 学習データが線形分離可能で、マージンが最大となる識別面(2値分類問題に限定. 正解情報の値を正例 1、負例 -1)

$$\{(x_i, y_i)\}, \quad i = 1, \dots, N, \quad y_i = 1 \text{ または } -1$$

- 識別面は平面なので、特徴空間上では式(1)で表現(\mathbf{w}, \mathbf{x} は d 次元のベクトル)

重み(パラメータ)

$$\mathbf{w} \cdot \mathbf{x} + w_0 = 0 \quad (1)$$

- i 番目のデータ x_i とこの識別面の距離 $Dist(x_i)$ は式(2)で表現

$$Dist(x_i) = \frac{|\mathbf{w} \cdot \mathbf{x} + w_0|}{\|\mathbf{w}\|} \quad (2)$$

- 式(1)の右辺は0なので、左辺を定数倍して識別面に最も近いデータを識別面の式に代入し、その絶対値を1になるように重みを調整

$$\min_{i=1, \dots, N} |\mathbf{w} \cdot \mathbf{x} + w_0| = 1 \quad (3)$$

識別面に最も近いデータ

マージン最大化のための定式化 (2)

- 式(2)、(3)より、学習パターンと識別面との最小距離は以下となる

$$\begin{aligned} \min_{i=1,\dots,N} Dist(x_i) &= \min_{i=1,\dots,N} \frac{|\mathbf{w} \cdot \mathbf{x} + w_0|}{\|\mathbf{w}\|} && \text{式(2)より} \\ &= \frac{1}{\|\mathbf{w}\|} && \text{式(3)より} \end{aligned} \quad (4)$$

- 式(4)がマージンを表すので、マージンを最大にする識別面を求める問題は、 $\|\mathbf{w}\|$ (もしくは $\|\mathbf{w}\|^2$) を最小化する問題
- $\mathbf{w} = \mathbf{0}$ は解にならないので、すべての学習データを識別できる式(5)の条件を付加

$$y_i(\mathbf{w} \cdot \mathbf{x} + w_0) \geq 1, \quad i = 1, \dots, N \quad (5)$$

- 式(5)の制約下での $\|\mathbf{w}\|^2$ の最小問題を解く
- 微分を利用して極値を求めて最小解を導くので、乗数1/2を付ける

$$\min \frac{1}{2} \|\mathbf{w}\|^2 \quad (6)$$

$$\text{条件} : y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) \geq 1, \quad i = 1, \dots, N \quad (7)$$

正例、負例の
両方を表す
制約の式

マージンを最大とする識別面の計算 (1)

- ラグランジュの未定乗数法を用いて、上記の定式化の問題を解決
- 具体的には、 $g(x) = 0$ という条件の下で、 $f(x)$ の最小値(あるいは最大値)を求める問題は、 $L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda g(\mathbf{x})$ (ただし、 λ はラグランジュ乗数)というラグランジュ関数を導入し、この関数の極値を求める
- ラグランジュ関数の \mathbf{x} に関する偏微分を0とすると以下の式となる

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = -\lambda \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} \quad (8)$$

最小解で制約と等高線は接する

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = -\lambda \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}}$$

制約がない
ときの最小値

大

制約

$f(x) = \text{定数}$

$g(x) = 0$

マージンを最大とする識別面の計算 (2)

- ラグランジュの未定乗数法を不等式制約条件で用いる
- 式(6)、(7)の制約付きの最小化問題は、ラグランジュ乗数 α_i を導入して、以下の関数 L の最小値を求める問題となる

$$L(\mathbf{w}, w_0, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^N \alpha_i (y_i (\mathbf{w} \cdot \mathbf{x}_i + w_0) - 1) \quad (9)$$

- 最小値では L の勾配が0になるので、以下の式が成り立つ

$$\frac{\partial L}{\partial w_0} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^N \alpha_i y_i = 0 \quad (10)$$

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i = 0 \quad (11)$$

- これを式(9)に代入して以下の式を得る

$$L(\alpha) = -\frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^N \alpha_i \quad (12)$$

識別面を決めるのに
寄与している学習データ

マージンを最大とする識別面の計算 (3)

- 式(12)を(数値計算ソフトウェアにより)解くと、 $\alpha_i \neq 0$ となるのはサポートベクターに対応するもののみで、ほとんどは $\alpha_i = 0$ となる。この α_i を式(11)に代入して \mathbf{w} を得る

$$\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i \quad (13)$$

- また、 w_0 は、 \mathbf{x}_+ 、 \mathbf{x}_- をそれぞれ正例、負例に属するサポートベクターとすると、以下の式で求められる

$$w_0 = -\frac{1}{2} (\mathbf{w} \cdot \mathbf{x}_+ + \mathbf{w} \cdot \mathbf{x}_-) \quad (14)$$

- 以上の方法で、学習データが線形分離可能な場合には、マージンを最大にする決定境界を求めることができる

マージンを最大化する決定境界

$$\left\{ \begin{array}{l} \min \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{条件} : y_i (\mathbf{w} \cdot \mathbf{x}_i + w_0) \geq 1, \quad i = 1, \dots, N \end{array} \right. \quad (6)$$

$$\left. \begin{array}{l} \min \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{条件} : y_i (\mathbf{w} \cdot \mathbf{x}_i + w_0) \geq 1, \quad i = 1, \dots, N \end{array} \right\} \quad (7)$$

ソフトマージンによる誤識別データの吸収

- 学習データが線形分離可能でない場合
 - ➡ 間違ったデータがあってもよいが、それらが識別面からあまり離れていないこと
- 式(7)の制約を緩める変数(スラック変数) $\xi_i (\geq 0)$ を導入して、 i 番目のデータが制約を満たしていない程度を示す

$$y_i(w \cdot x_i + w_0) \geq 1 - \xi_i, \quad i = 1, \dots, N \quad (15)$$

- ξ_i は制約を満たさない程度なので、小さい方が望ましい。この値を、マージン最大化の式(6)(7)に加える

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \quad (16)$$

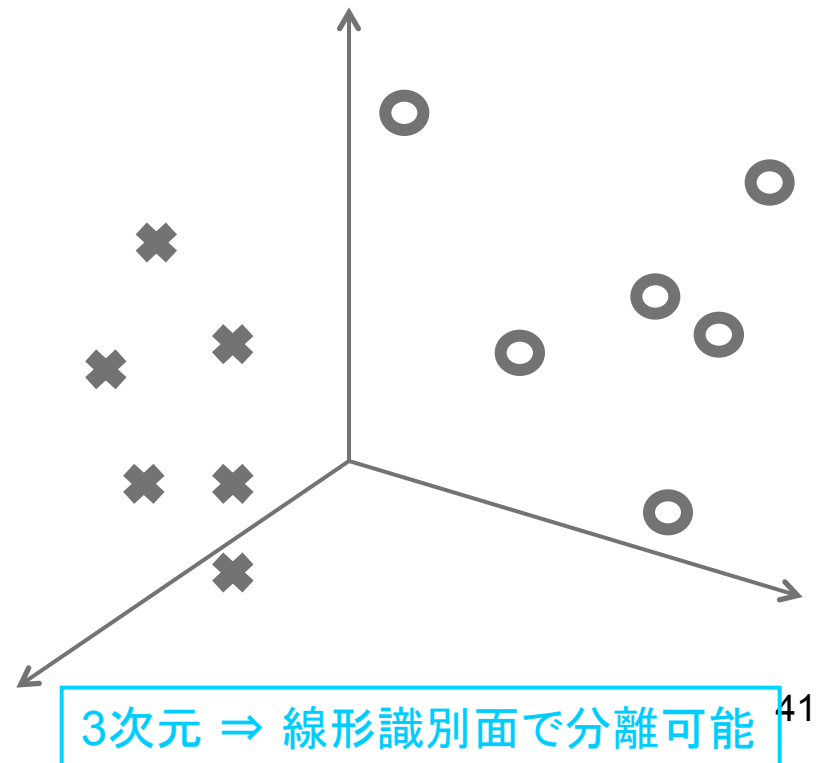
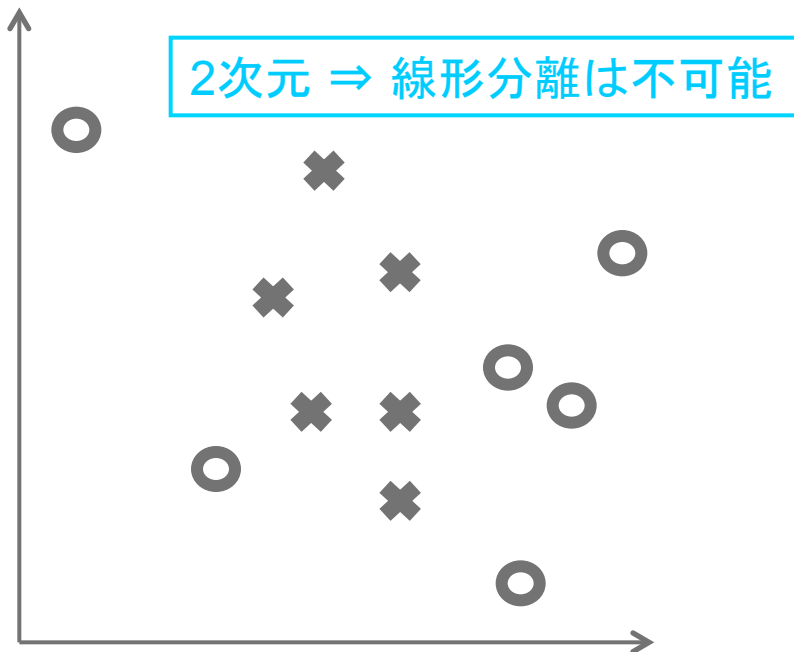
$$\text{条件: } y_i(w \cdot x_i + w_0) \geq 1 - \xi_i, \quad i = 1, \dots, N, \quad \xi_i \geq 0 \quad (17)$$

- ここで、 C は制約を満たさないデータをどの程度の重みで組み込むか？ ➡ 大きければ影響大、小さければ無視する振る舞い

重要!

カーネル関数を用いたSVM

- 特徴空間の次元 d が大きい場合、線形識別面が存在する可能性大
- この性質を使い、低次元の特徴ベクトルを高次元に写像し、線形分類の可能性を高める ➡ 高次元空間上でSVMを使って識別超平面を求める
- ポイント: もとの空間におけるデータ間の距離関係を保存するような非線形写像を見つけること ➡ カーネル関数(カーネルトリック)



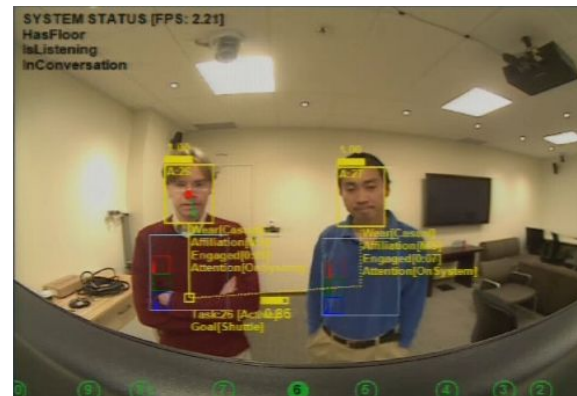


HCIと機械学習

部分観測マルコフ決定過程

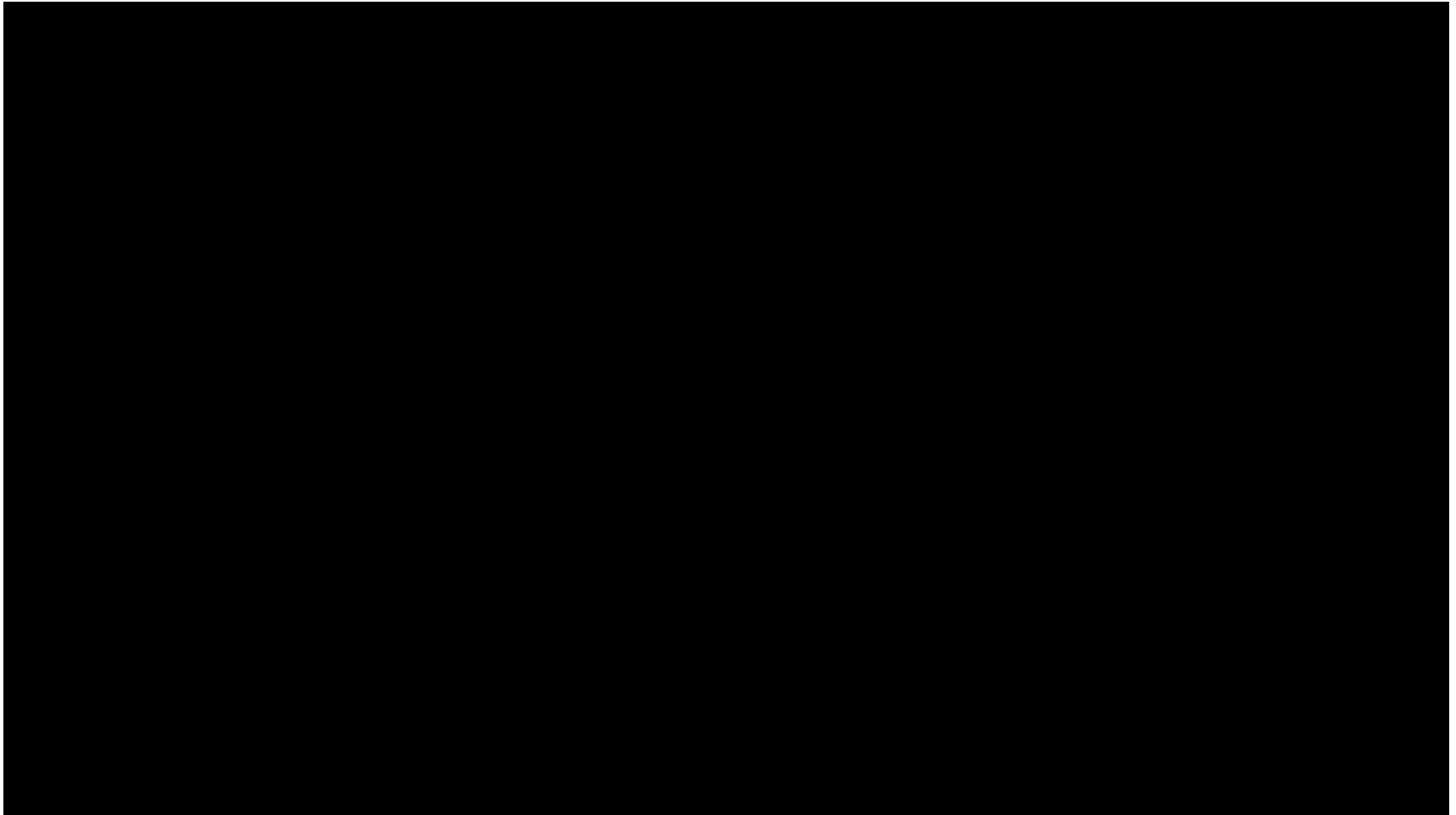
応用事例：近年の音声対話システム

- 対話状況認識（会話シーン分析、マルチモーダルセンシング・フュージョン）
- 自然言語による多人数対話
- 対話制御（部分観測マルコフ決定過程）



Situated Interaction project (MSR)

Situated Interaction (Microsoft Research)



応用事例：Human Computation

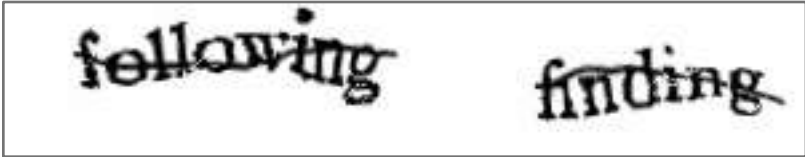
- 現時点での情報技術の限界：
 - ・ 文字認識：崩れた文字は認識できない
 - ・ 機械翻訳：翻訳された文の質が低い
- HCIにおける新しいアプローチ
 - ・ クラウドソーシング（Crowdsourcing）：
 - 不特定多数の人の寄与を募り、必要とするサービス、アイデア、またはコンテンツを取得するプロセス
 - ・ ヒューマンコンピューテーション（Human Computation）：
 - 人間とコンピュータどちらが欠けても解答することが不可能な問題に対し、人間の知能と計算機を組み合わせる手法

みんながHappyに問題を解決する！

Dr. Luis von Ahnの研究のまとめ: “一石二鳥”手法

– reCAPTCHA:

CAPTCHA: ウェブサイトの制限エリアへアクセスを試みるボットの排除 + OCRで認識の難しい文字を課題として紙の本のデジタル化に活かすシステム



following finding

• 具体例:

- ニューヨーク・タイムズの記事アーカイブの電子化
- Googleブックスの書籍電子化

– Duolingo:

無料の言語学習、及びクラウドソーシング翻訳用プラットフォーム

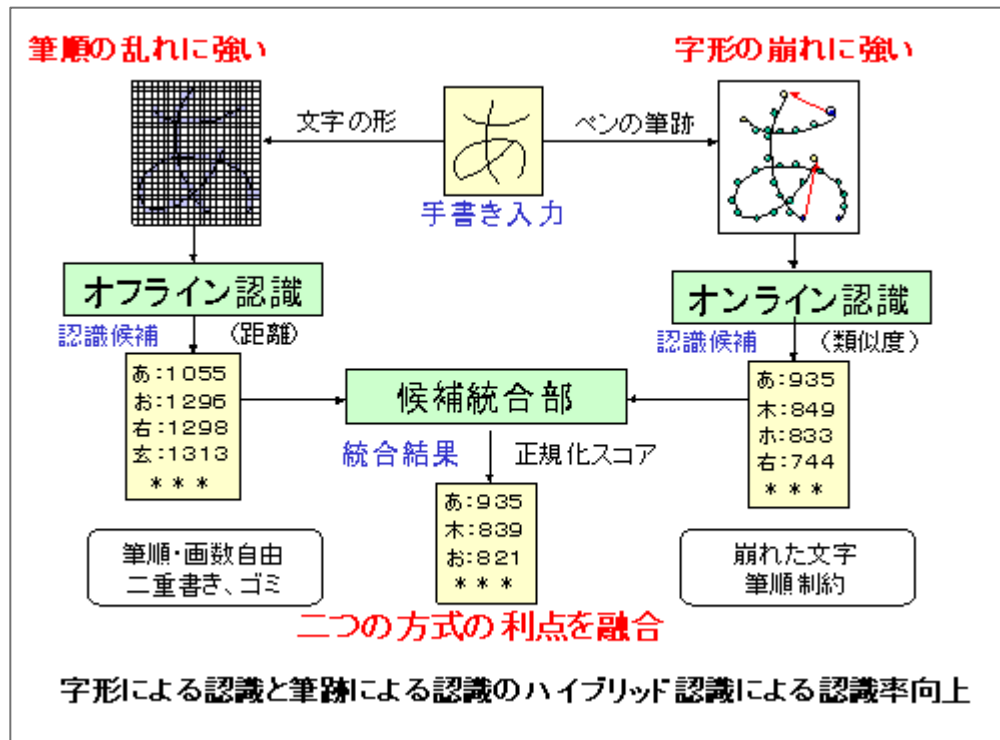
利用者がコースを受講 → その結果をウェブサイトやその他の文書の翻訳に使用



認識アルゴリズム(1)： DPマッチングによる手書き文字認識

手書き文字認識とは？

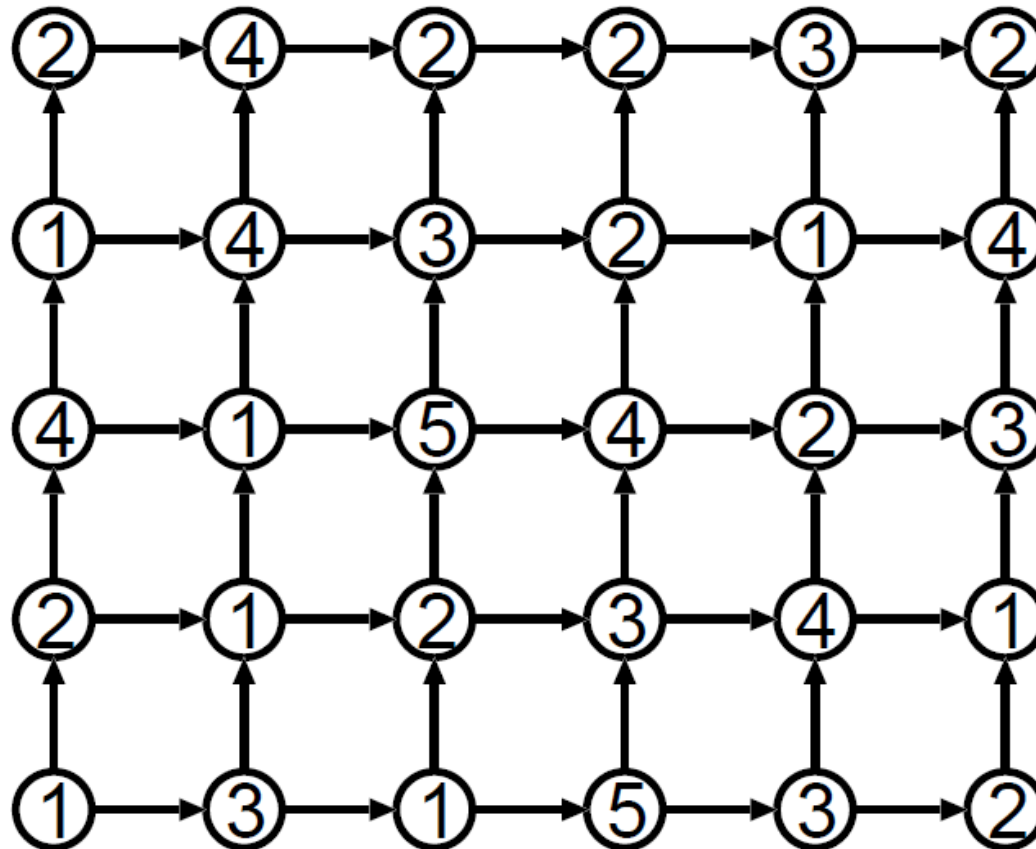
- 認識可能な手書き入力を受け取るコンピュータの機能
 - ・ オフライン手書き文字認識 (例：郵便番号、OCR)
 - ・ オンライン手書き文字認識 (例：タブレット、タッチパネル)



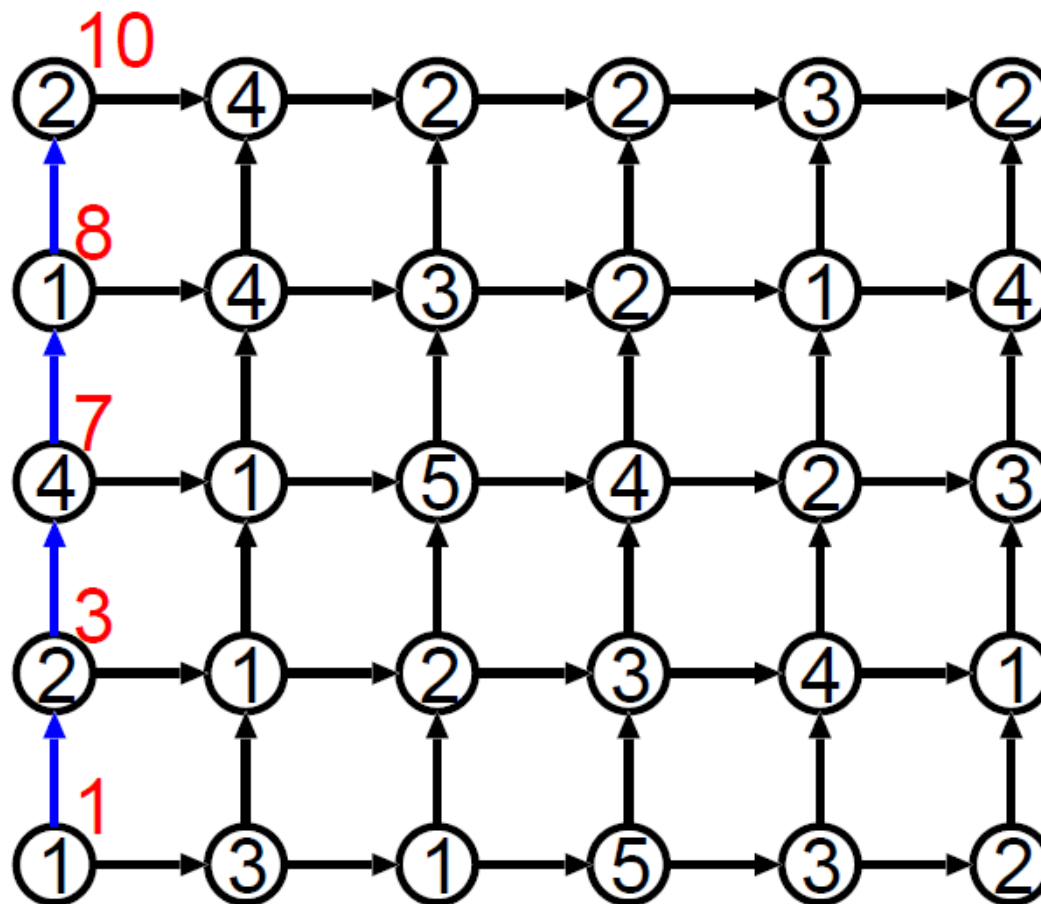
動的計画法

(Dynamic Programming)とは？

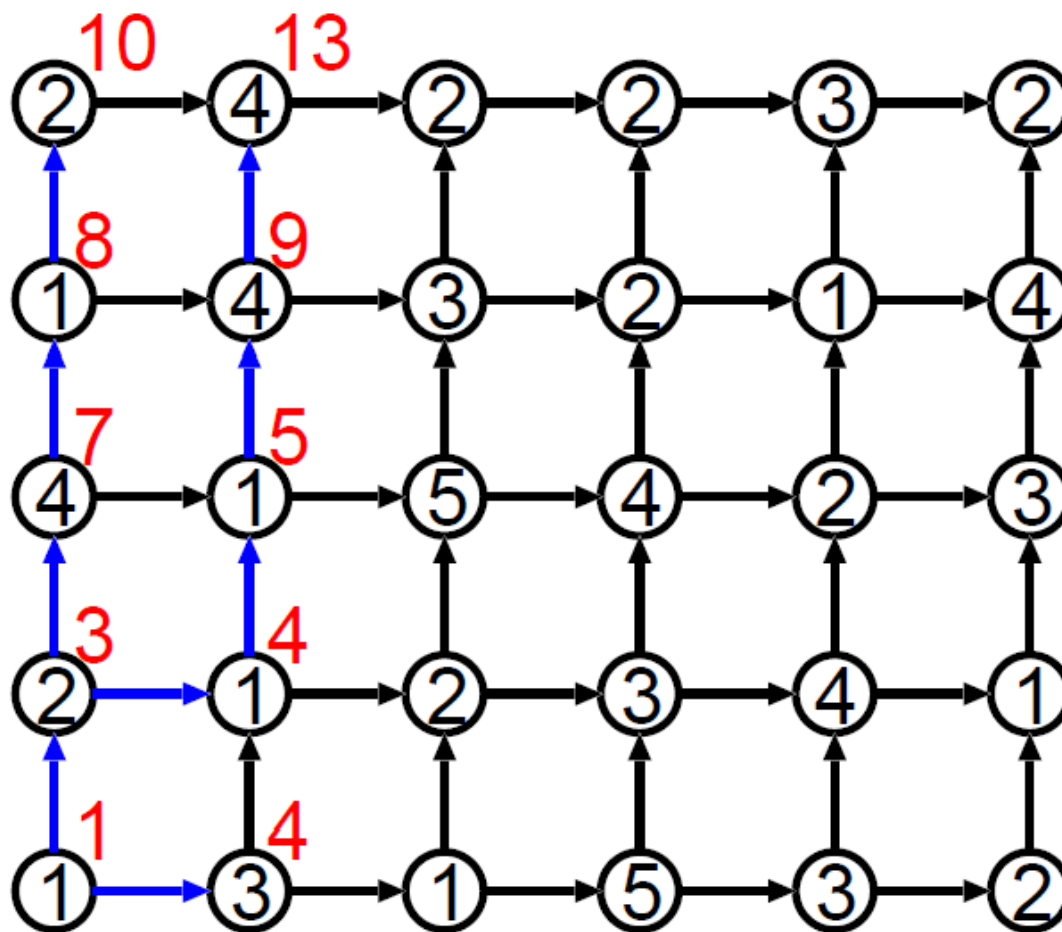
- 単純な動的計画法の例：左下隅から右上隅に至る数値の総和が最小の経路を求める



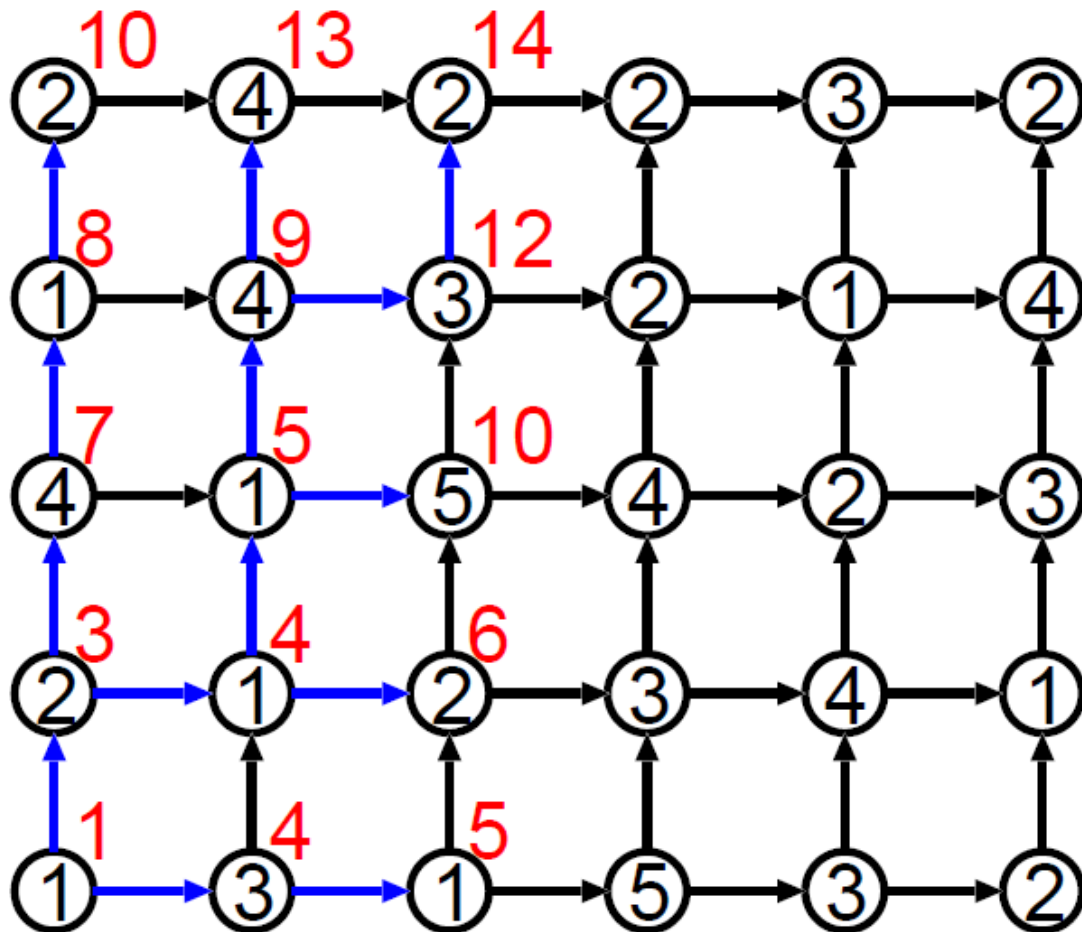
- 単純な動的計画法の例：左下隅から右上隅に至る数値の総和が最小の経路を求める



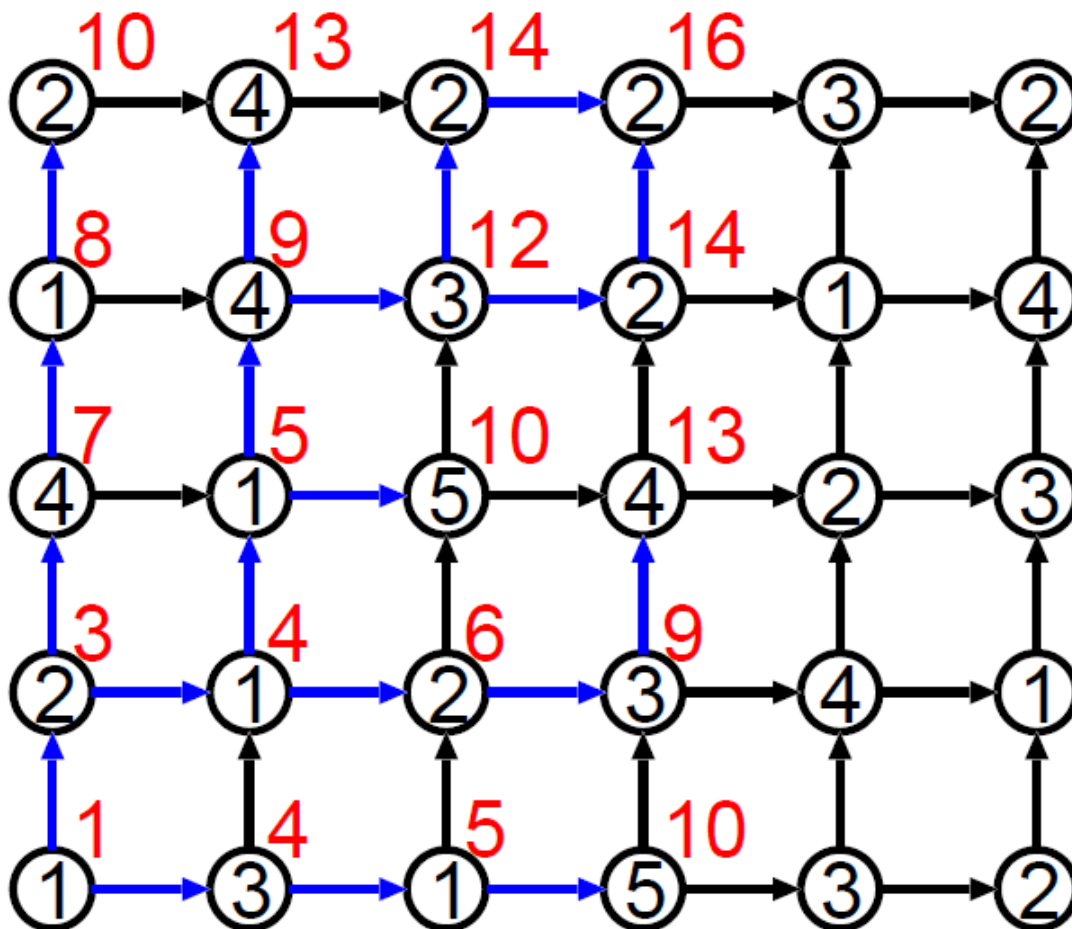
- 単純な動的計画法の例：左下隅から右上隅に至る数値の総和が最小の経路を求める



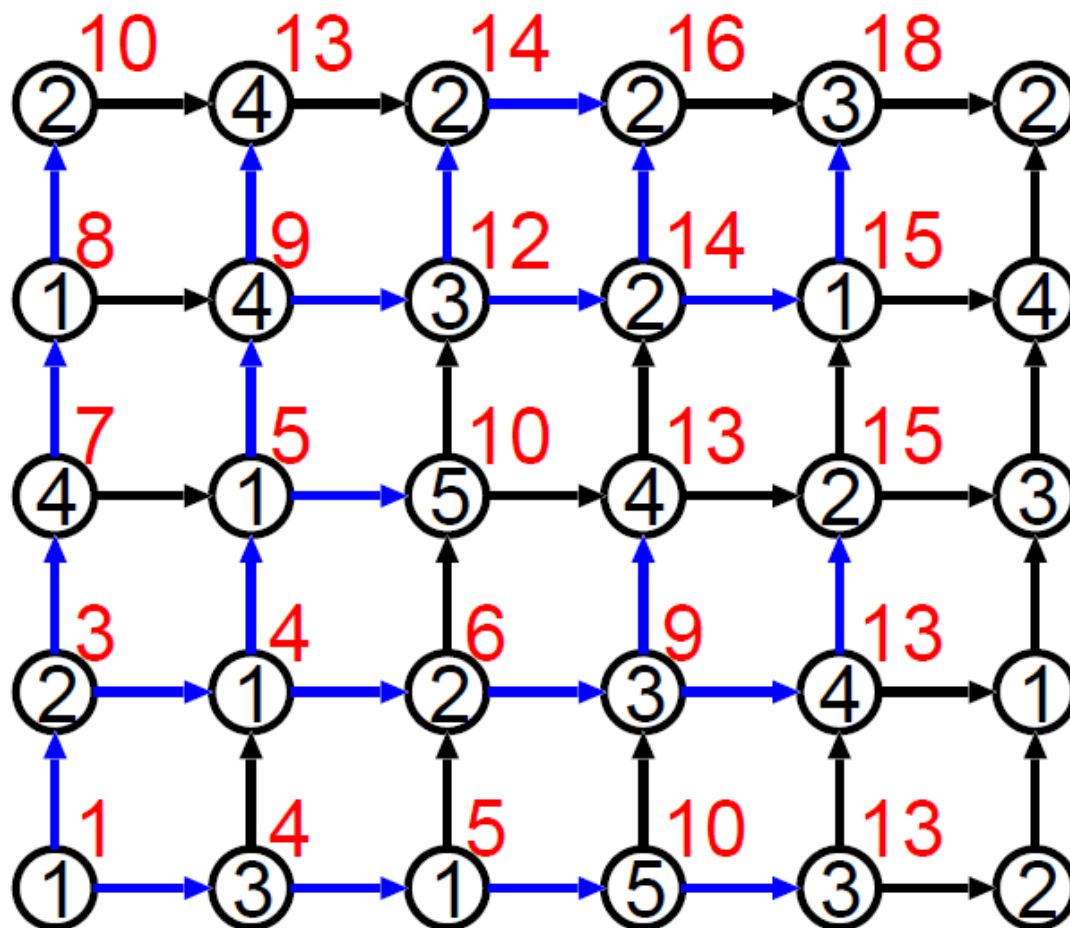
- 単純な動的計画法の例：左下隅から右上隅に至る数値の総和が最小の経路を求める



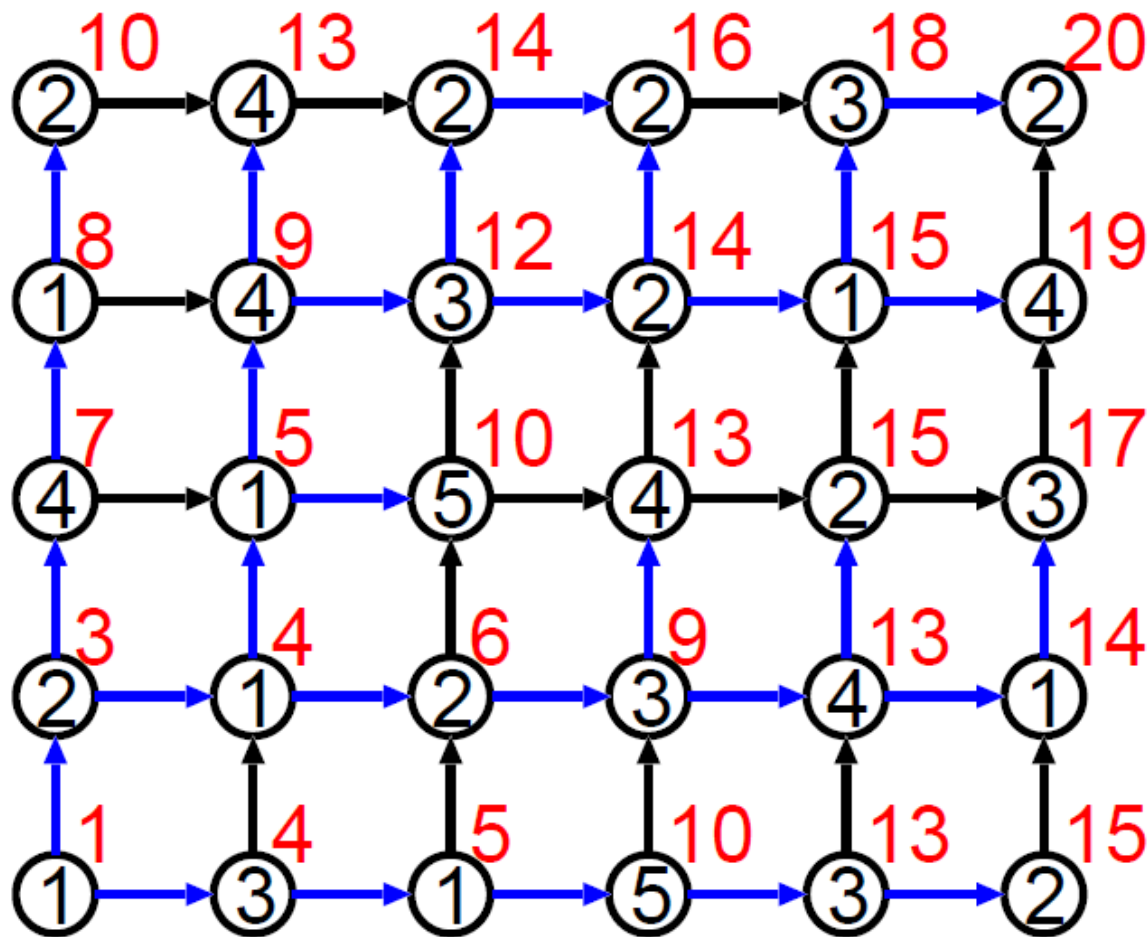
- 単純な動的計画法の例：左下隅から右上隅に至る数値の総和が最小の経路を求める



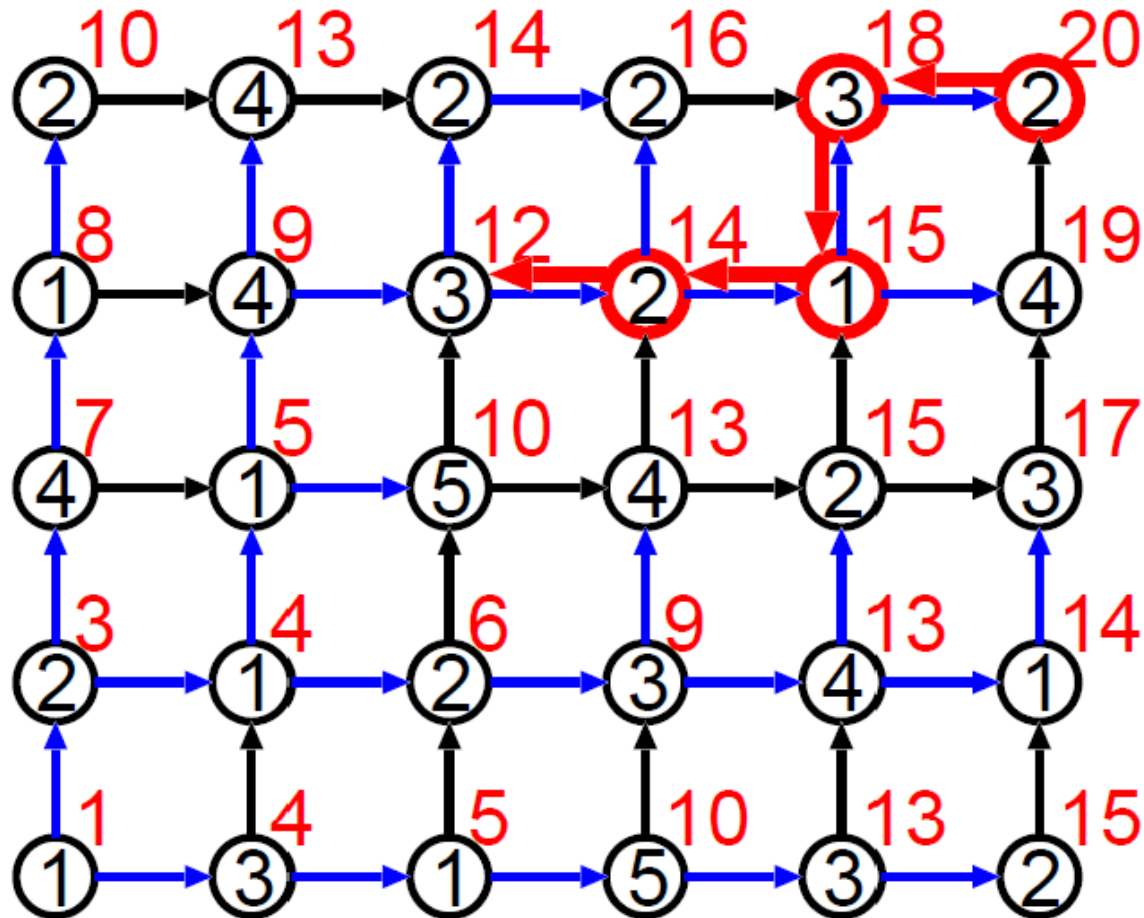
- 単純な動的計画法の例：左下隅から右上隅に至る数値の総和が最小の経路を求める



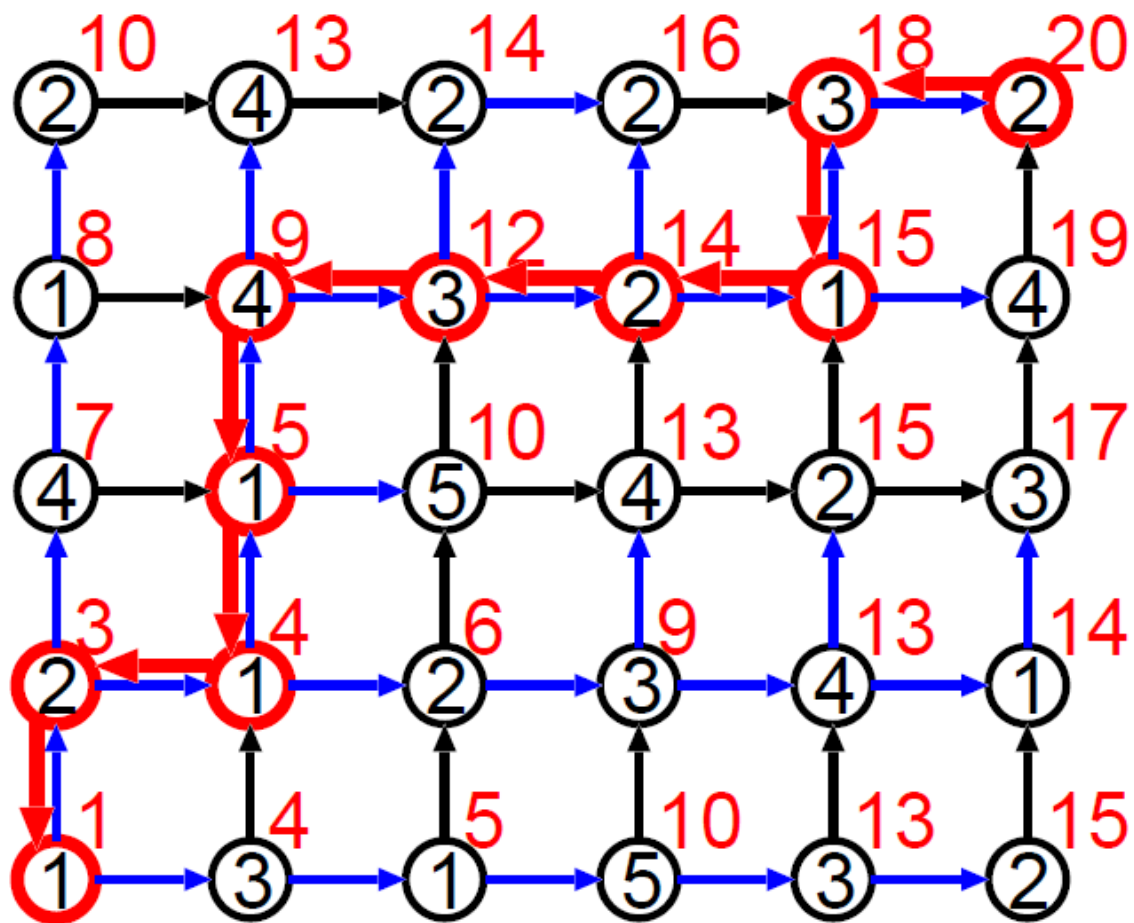
- 単純な動的計画法の例：左下隅から右上隅に至る数値の総和が最小の経路を求める



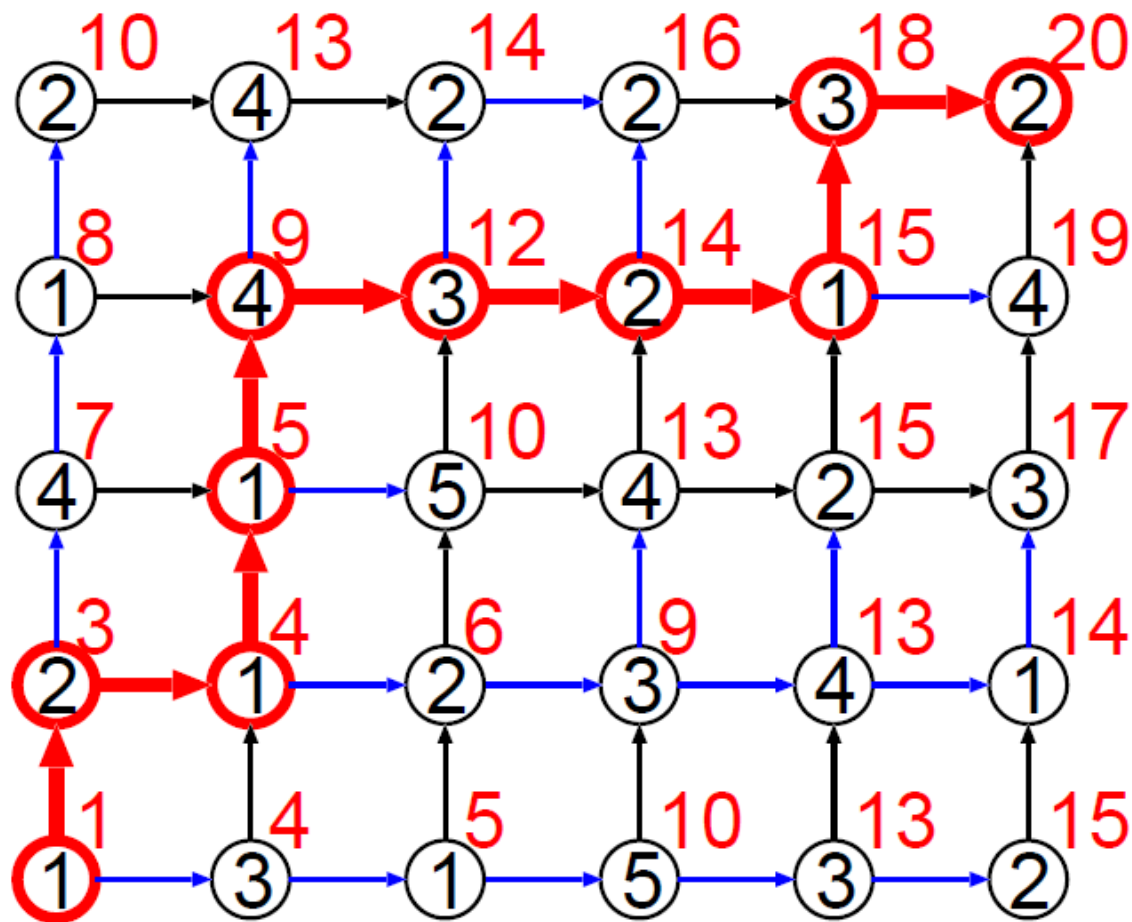
- 最適経路決定：トレースバック (traceback)
 (トレースバックするまで最適経路は決まらない)



- 最適経路の決定！



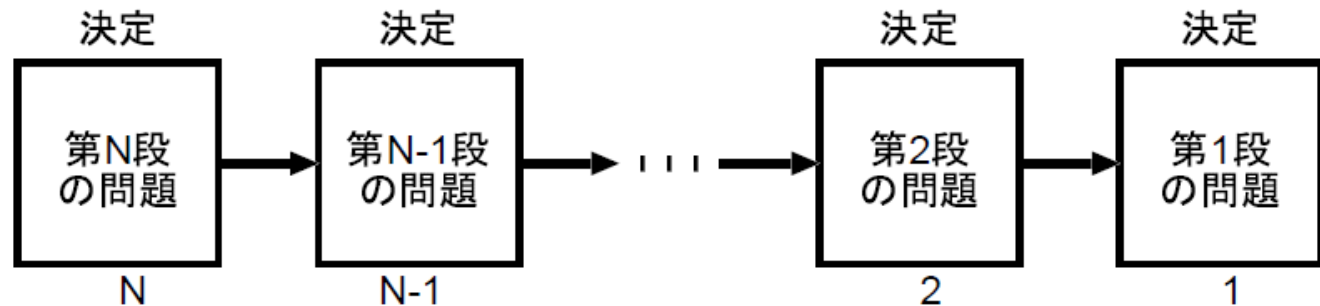
- ・トレースバックするためには、バックポインタをすべて記憶する必要がある ⇒ メモリ消費



動的計画法とは？

動的計画法の考え方：

- 最適性原理に基づき、決定すべきN個の最適決定を順次行うことによって問題全体の最適解を模索

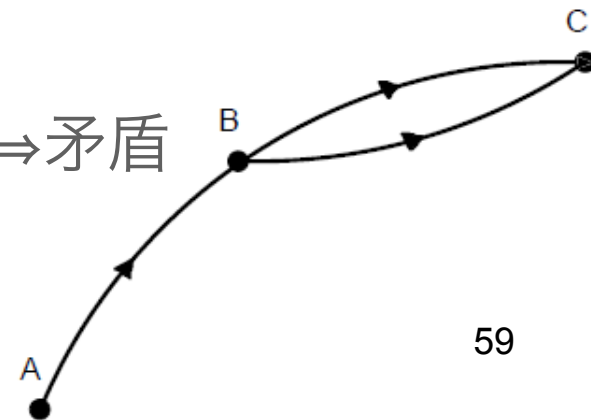


最適性原理 (Principle of Optimality)

- A→C (上): ある問題の最適経路
- B: 最適経路の通過点
- B→C (下): 部分最適経路とすると⇒矛盾

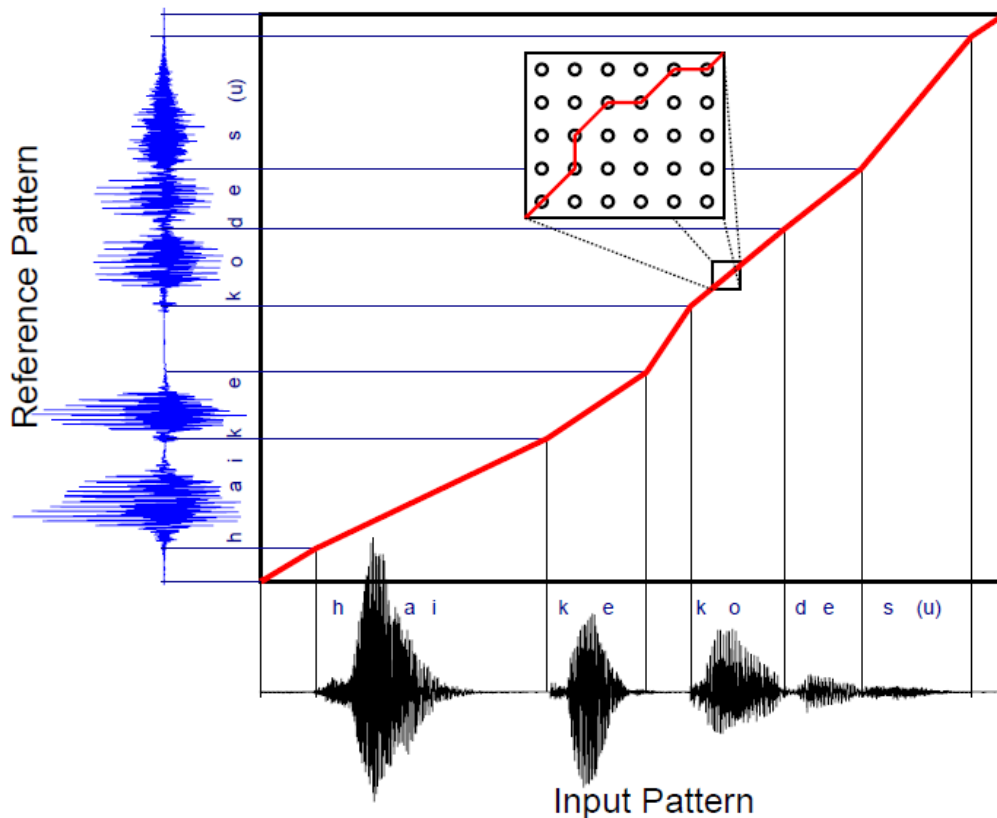
つまり、最適経路は部分最適経路を含まなくてはならない

⇒ 最適性原理



DPマッチング：音声認識

- 動的計画法を利用した、長さが異なるパターン間のマッチング法
- 全体としての距離が最小となるような対応づけをする



入力パターンと標準パターン
を伸縮しながらマッチング



弾性マッチング

DPマッチングの事例(1)

- インタラクティブシステム⇒時間的に変化する入力パターンを認識する必要あり

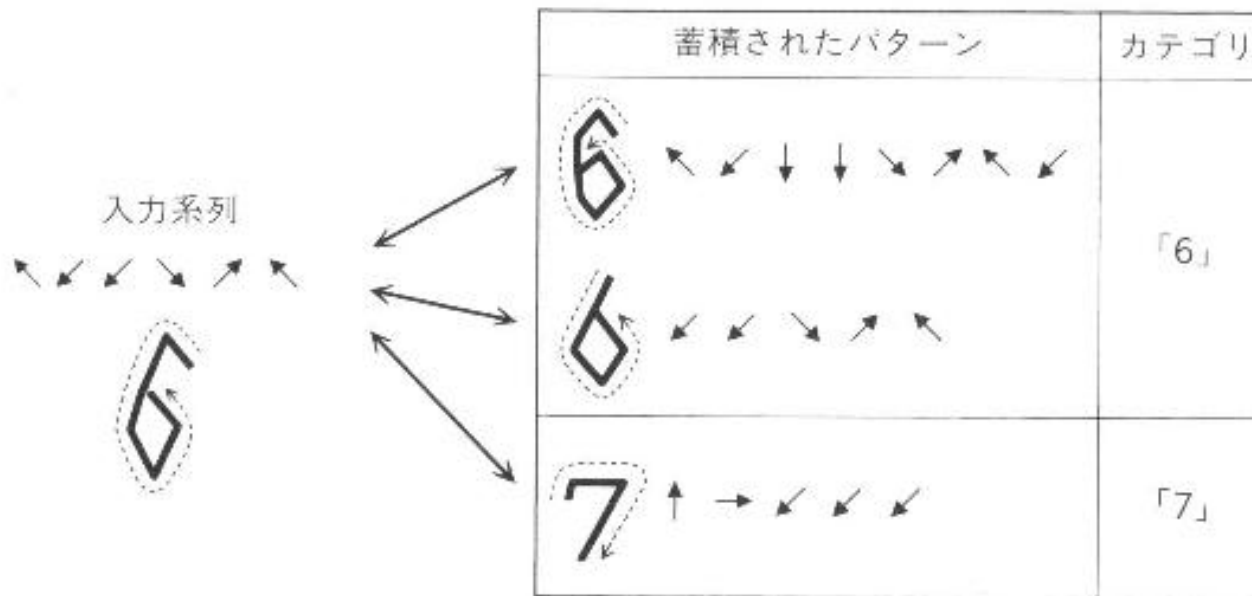
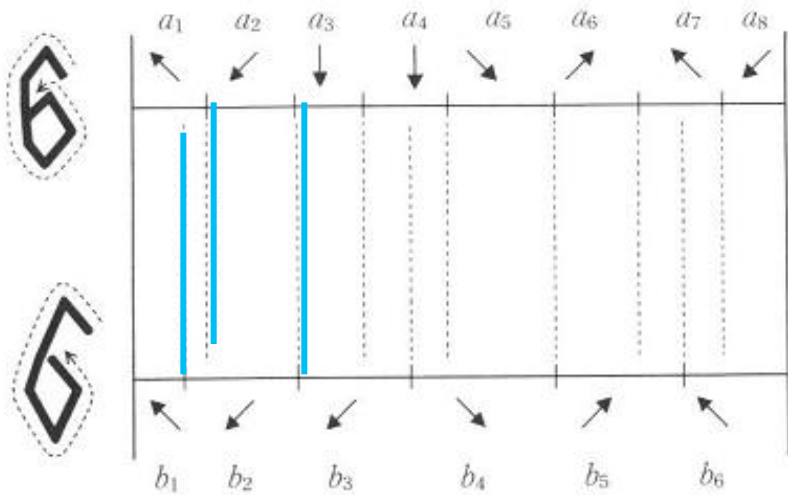


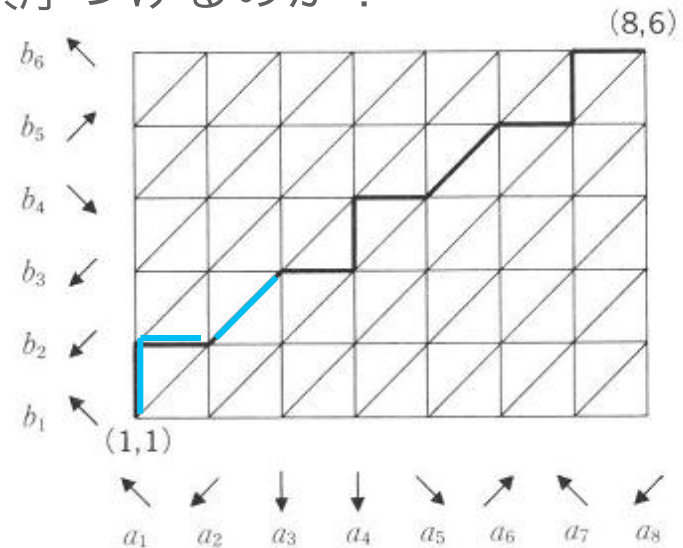
図 5.4 DP マッチングによる時系列パターン認識の考え方. 与えられた入力系列 B を, あらかじめ用意した各カテゴリの典型的なパターン A_1, A_2, \dots, A_p のそれぞれと比較し, もっとも近いパターン A_i を求め, それに対応づけられたカテゴリ $C(A_i)$ を認識結果とする. この図は, ペン入力によるストロークデータからの文字認識を想定したもの.

DPマッチングの事例 (2)

- DPマッチング：セグメントの系列A、Bの間のマッチングの度合いを効率的に計算するアルゴリズム
 - 各セグメントの間の遷移をどう順序づけるのか？
 - ⇒ マッチングの可能性



(a) 各セグメント境界の対応づけ



(b) 格子点を用いたセグメント境界の対応づけの表現

DPマッチングの事例(3)

- DPマッチング：
 - 与えられた系列A、Bに対する格子点(1, 1)から(m, n)までの経路の中の、コスト最少のものを効率よく探すためのアルゴリズム
 - 効率化は、各格子点に至る最少経路のコストを記憶しておけば、いちいちすべての経路のコストを計算する必要がないことに注目
 - DPマッチングでは、マッチングの度合いを計算する関数の選定方法や知識ベースに蓄える系列集合の選定方法は任意 ⇒ この部分をマルコフ確率の理論に基づかせると「隠れマルコフモデル」

DPマッチングの形式化(1)：変数定義

- テンプレート系列: $A(a_i) = a_1, a_2, a_3, \dots, a_I$
- 入力系列: $B(b_i) = b_1, b_2, b_3, \dots, a_J (I \neq J)$
- 対応変換関数: $F = c(1), \dots, c(k), \dots, c(K)$
 $c(k) = (i(k), j(k))$ i, j 平面でとる経路
- 特徴ベクトル間距離: $d(c(k)) = d(a_{i(k)}, b_{j(k)})$
 F 上各点での特徴ベクトル間の距離
- F 上の各点の間の距離: $w(k) = c(k)$ と $c(k+1)$ の間の距離
- A, B の間の距離: $D(A, B)$

DPマッチングの形式化(2)：系列間の距離

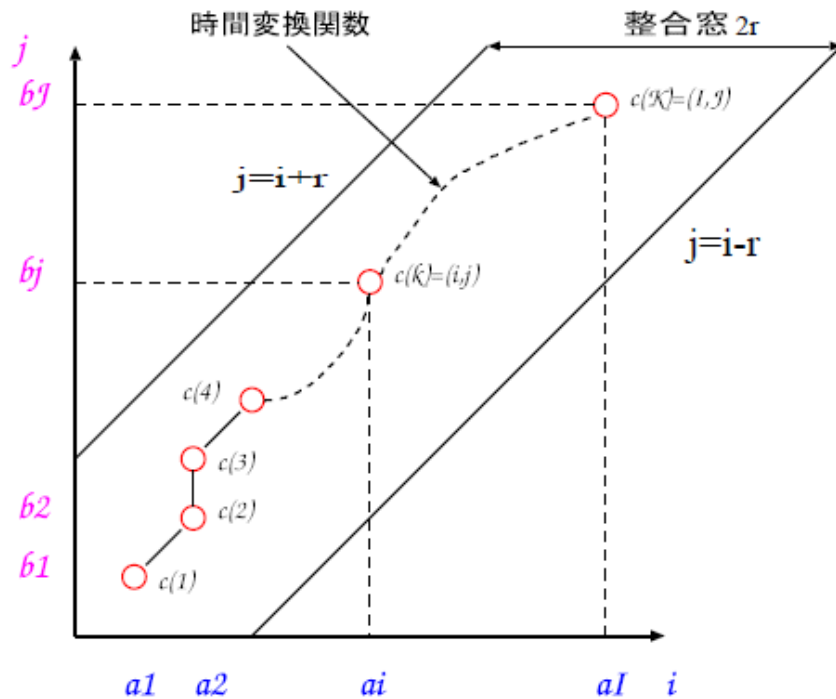


図27. 二系列の対応づけ

経路Fにおける距離の総和は

$$H(F) = \frac{\sum_{k=1}^K d(c_k) \cdot \omega_k}{\sum_{k=1}^K \omega_k}$$

二つのパターンの最小距離は

$$D(A, B) = \min_F \frac{\sum_{k=1}^K d(c_k) \cdot \omega_k}{\sum_{k=1}^K \omega_k}$$

ただし、経路の取り方には制限がかかる。

DPマッチングの形式化(3)：経路制限

- 単調性： $i(k-1) \leq i(k), j(k-1) \leq j(k)$
後戻りしない
- 連続性： $i(k) - i(k-1) \leq 1, j(k) - j(k-1) \leq 1$
格子点を飛び越さない
- 境界条件： $i(1) = 1, j(1) = 1, i(K) = I, j(K) = J$ 入力となる2つの系列の長さ以上に探索空間を広げない
- 整合窓の条件： $|i(k) - j(k)| \leq r$ (極端な伸縮を防ぐため、 r ：定数)

DPマッチングの形式化(4)：荷重係数

二つのパターンの最小距離

$$D(A, B) = \min_F \frac{\sum_{k=1}^K d(c_k) \cdot \omega_k}{\sum_{k=1}^K \omega_k}$$

ここで荷重係数の設定によって $D(A, B)$ の分母を定数とすることでこの最小化問題は簡単化できる。

■ 対称形

$$\omega_k = i_k - i_{k-1} + j_k - j_{k-1}$$

このとき分母は $I + J$ となる

■ 非対称形

$$\omega_k = i_k - i_{k-1} = 1$$
$$0 \leq j_k - j_{k-1} \leq 2$$

このとき分母は I 、分子は $\sum_{i=1}^I d(i, j_i)$ となる

DPマッチングの形式化(5)：応用1

■ 対称型

$$\omega_k = i_k - i_{k-1} + j_k - j_{k-1}$$

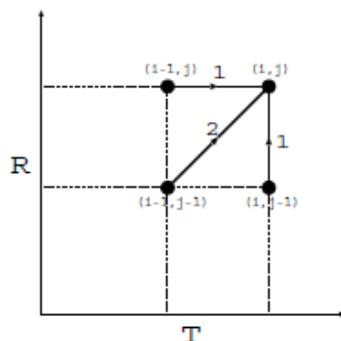


図28. 対称型

■ 非対称型

$$\begin{aligned} \omega_k &= i_k - i_{k-1} = 1 \\ 0 &\leq j_k - j_{k-1} \leq 2 \end{aligned}$$

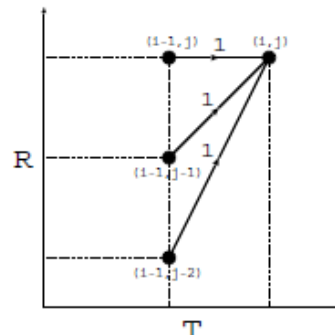


図29. 非対称型

部分和 $g(i, j)$ は,

$$g(c_k) = g(i, j) = \min_{c_1, \dots, c_k} \sum_{t=1}^k d(c_t) \cdot \omega_t$$

ここで、動的計画法が使える。

$$g(c_k) = g(i, j) = \min_{c_{k-1}} [g(c_{k-1}) + d(c_k) \cdot \omega_k]$$

DPマッチングの形式化(5)：応用2

ω_k として、非対称型のものを用いたとすると、

$$g(i, j) = \min \left\{ \begin{array}{l} g(i-1, j) \\ g(i-1, j-1) \\ g(i-1, j-2) \end{array} \right\} + d(i, j)$$

と書ける。

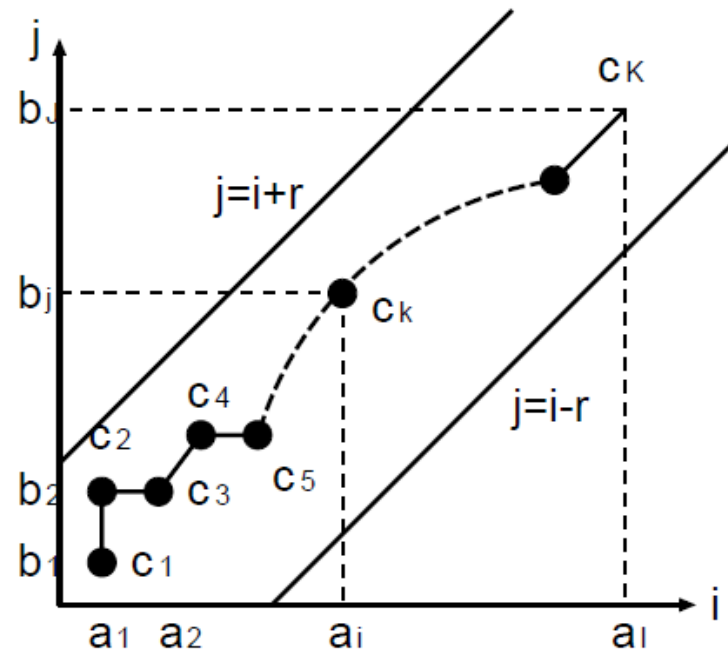
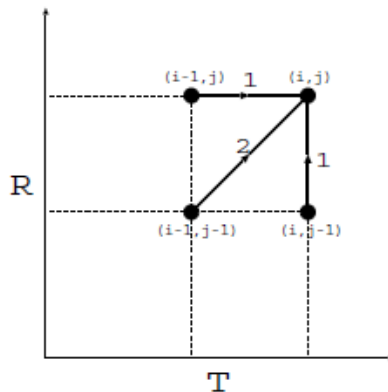


図30. A, B両系列の対応づけ

DPマッチングの形式化(6) : 問題

Sample Pattern	7				/	18	9	9	9			
	6				10	13	7	9	10			
	5			11	8	9	6	9	/			
	1	/	9	10	7	6	10					
	3	5	5	5	4	5						
	6	5	3	2	4							
	5	2	2	3	/							
					3	5	6	4	2	5	8	7
					Input Pattern							

図31. 絶対値差の累積値が最小の経路を求める例 : 横軸 i , 縦軸 j



整合窓は $|i(k) - j(k)| \leq 3$ とした

図32. 用いた荷重係数

DPマッチング：まとめ

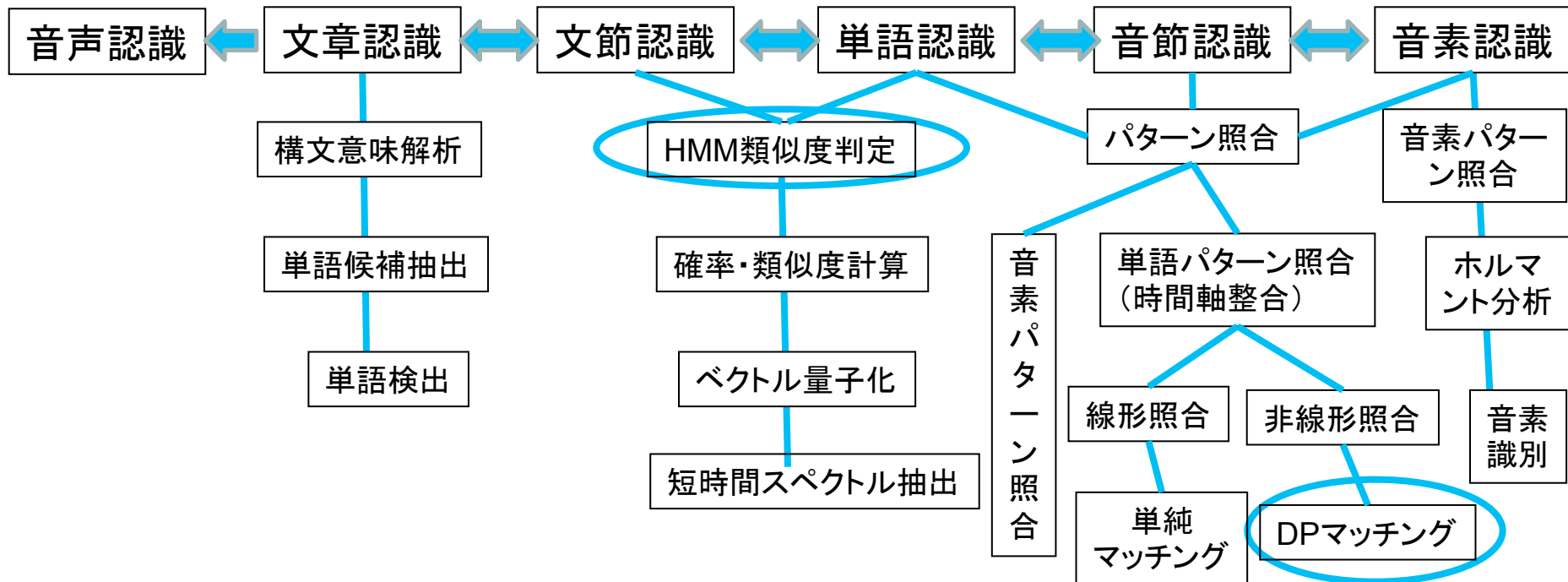
- 要件：
 - 特徴ベクトル間の距離が定義されていること
 - 格子点間の移動コスト：探索空間における格子点間の移動コストが定義されていること
 - 包含性：探索空間において、全体問題の最小解を与える経路が部分問題の最小解を与える経路を含んでいること
- 要点：
 - 再帰的に部分最適化問題へ帰着できる
 - 前向き経路探索で最小累積距離が得られる
 - 後向き経路追跡（トレースバック）により最適経路決定



認識アルゴリズム(2)： 隠れマルコフモデルによる音声認識

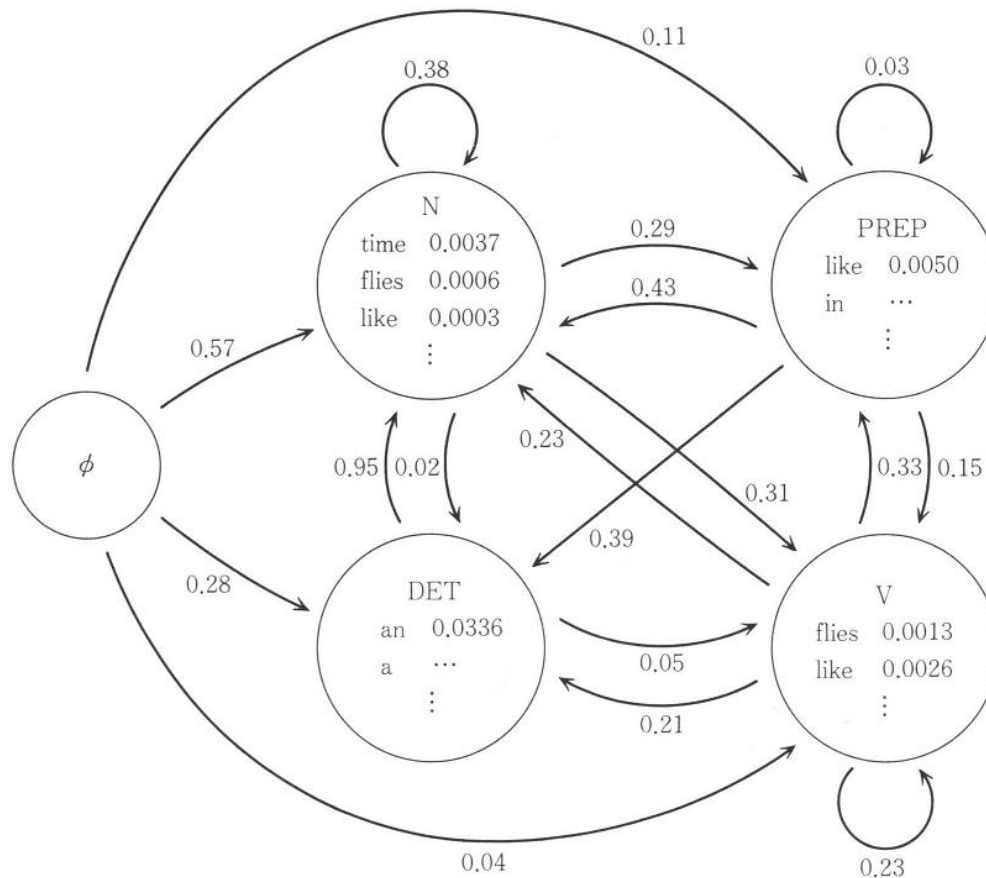
アルゴリズムはどのように使われるのか？

- 音声認識の場合：
 - 隠れマルコフモデル(HMM)
 - (DPマッチングも)



隠れマルコフモデルの応用 (1)

- 単語/品詞の隠れマルコフモデル
 - 例文: time flies like an arrow.



隠れマルコフモデルの応用 (2)

- ・ 単語/品詞の隠れマルコフ・モデル
 - 5通りの構文解釈が可能な文: time flies like an arrow.
 - S[NP/Time VP[VP/flies PP[PREP/like NP[DET/an N/arrow]]]]
 - S[VP/Time NP[NP/flies PP[PREP/like NP[DET/an N/arrow]]]]
 - S[NP[ADJ/Time NP/flies] VP[V/like NP[DET/an N/arrow]]]
 - S[NP[NP/Time NP/flies] VP[V/like NP[DET/an N/arrow]]]
 - S[VP[V/Time NP/flies] PP[PREP/like NP[DET/an N/arrow]]]

隠れマルコフモデルの事例(1)

- ・ 隠れマルコフモデル：
 - いくつかの状態を確率的に遷移するマルコフモデルを用いて系列を生成する情報源モデルを表現し、それを使って系列の認識を行う

$$P([A, D, F]|S_1 \rightarrow S_2 \rightarrow S_3) = 0.6 \times 0.5 \times 0.7 \times 0.7 \times 0.1 \times 0.5$$

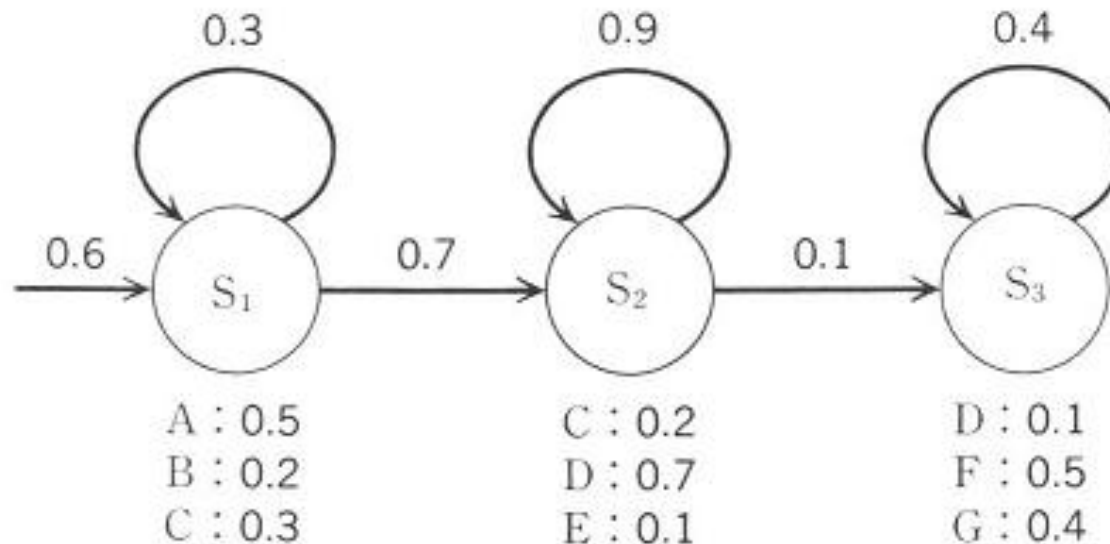
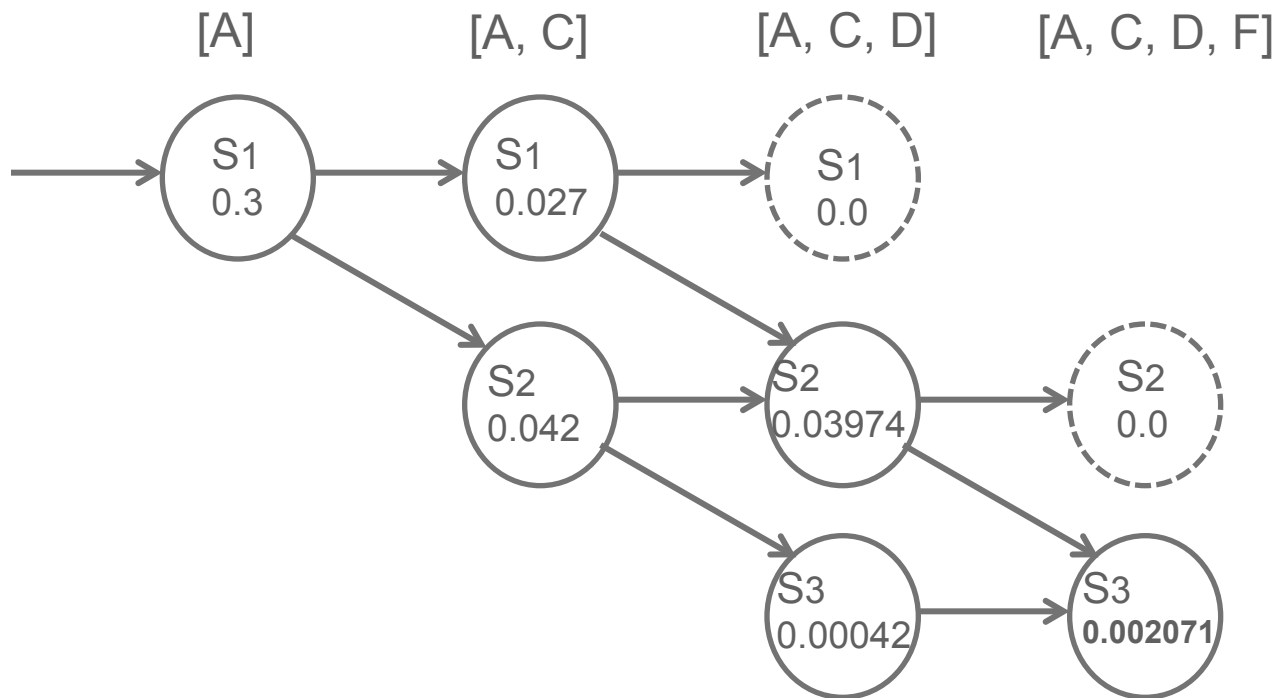


図 5.6 隠れマルコフモデルの例。

隠れマルコフモデルの事例(2)

シンボルの系列が生成される確率を計算するためには、与えられた系列の部分系列を生成しうる状態遷移にそって生起確率を計算し、それを加算すればよい



隠れマルコフモデルの事例(3)

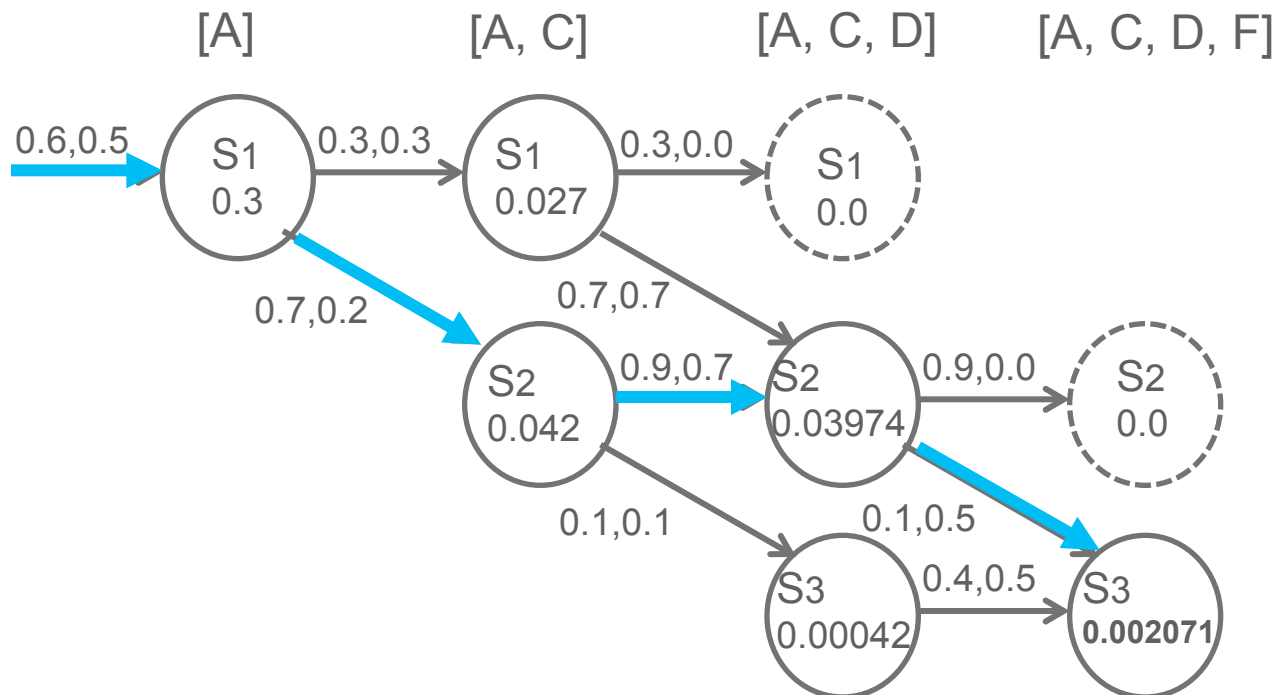
- ・ HMMモデルを使った系列パターン認識
 - 与えられたHMMモデルとシンボル系列から、その系列を生成するもっとも確からしいHMM中の状態列とその確率を計算する
 - 素朴なアルゴリズム(前向きアルゴリズム): $O(M^n)$
 - ・ 第1シンボルを出力するすべての状態列を求め、次に残った出力列を生成する確からしい確率を求め、...
 - マルコフ性を利用 \Rightarrow ビタビアルゴリズム: $O(bMn)$
 - ・ ある状態に至るまでのすべての可能な状態列を求める必要はなく、そこに至るもっとも確からしい状態列のみ記憶しておけばよい

隠れマルコフモデルの事例(4)

ビタビアルゴリズム

- 各状態に至る確率最大値を与える遷移だけを求める

➡ 最適状態遷移系列の計算



隠れマルコフモデルの形式化(1)

- ・ マルコフ連鎖(確率過程)
 - 状態の集合 $S = \{s_0, \dots, s_{m-1}\}$ に値をとる確率過程の状態時系列 $X = X_1, \dots, X_n$ において

$$P(X_t = a_0) = P(X_t = a_0 \mid X_{t-1} = a_1, \dots, X_{t-k} = a_k)$$

と書けるとき、この確率過程を k 次マルコフ連鎖と呼ぶ

- ・ マルコフモデル(確率モデル)
 - マルコフ連鎖の状態時系列そのものを信号列とみなしたとき、この確率情報源をマルコフモデルと呼ぶ

隠れマルコフモデルの形式化(2)

- ・ 遷移確率(行列)

$$a_{ij} = P(X_t = s_j | X_{t-1} = s_i)$$

- ・ 時刻0の各状態の確率(初期状態分布)を与える代わりに、時刻0の初期状態 s_0 を考える
- ・ 状態列 $x = x_1, \dots, x_T$ が観測される確率

$$p(x) = a_{0x_1} \prod_{t=1}^{T-1} a_{x_t x_{t+1}}$$

隠れマルコフモデルの形式化(3)

• $M = (S, \Sigma, A, B)$

– 状態の集合: $S = \{s_0, \dots, s_{m-1}\}$

– 出力記号の集合: $\Sigma = \{c_1, \dots, c_d\}$

– 状態遷移確率: $A = \{a_{ij}\}$

– 出力確率: $E = \{e(s_i, c_k) = e_i(c_k) = e_{ik}\}$

– 状態遷移は1次マルコフ過程

– 状態は直接観測できず、出力記号が観測可能

– 観測記号列 $x = x_1, \dots, x_T$ に対する状態列 $\pi = \pi_1, \dots, \pi_T$

$$p(\pi, x) = a_{0\pi_1} \prod_{t=1}^{T-1} e(\pi_t, x_t) a_{\pi_t \pi_{t+1}}$$

隠れマルコフモデルの形式化(4)

– ビタビアルゴリズム(状態列の推定)

- 出力記号列と状態列の同時確率の最大化

$$\pi^* = \arg \max_{\pi} P(x, \pi)$$

- ビタビ変数 $v_k(t)$

時刻tまでの観測部分列 $x_{(1,t)} = x_1 \dots x_t$ と

時刻tまでの状態部分列 $x_{(1,t)} = \pi_1 \dots \pi_t$ の同時確率で

$\pi_t = s_k$ である場合の最大確率

以下の漸化式が成り立つ

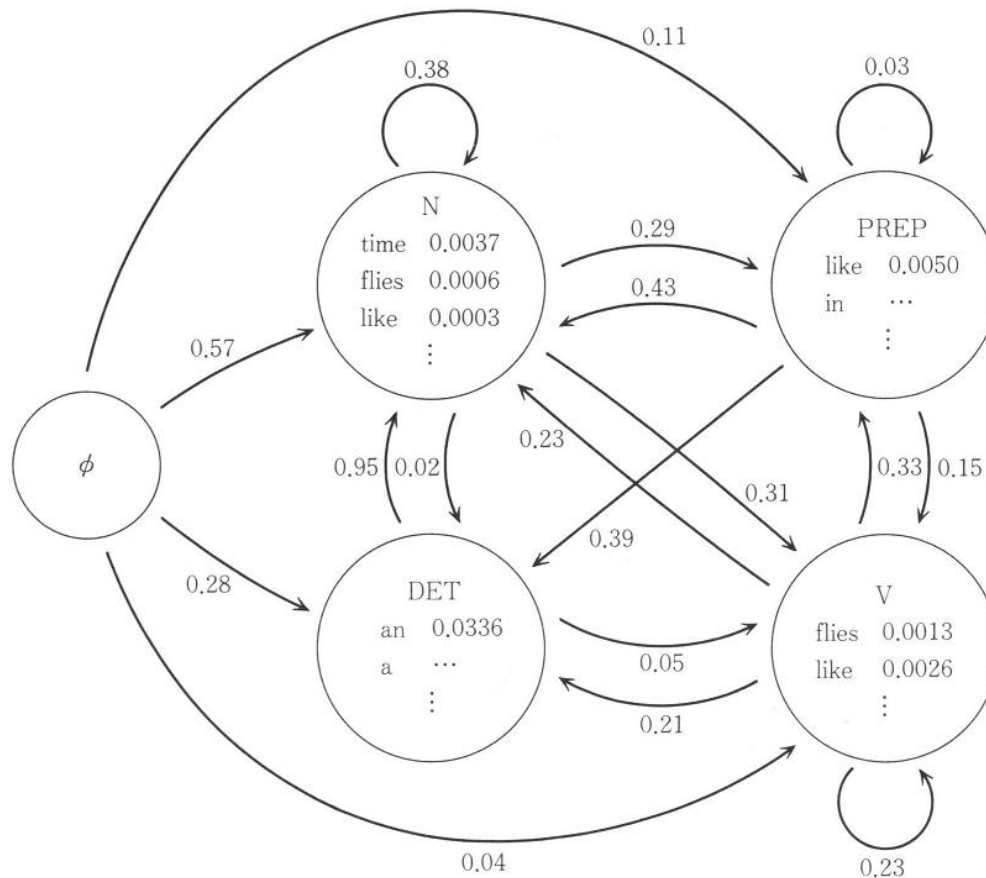
$$v_l(t+1) = e_l(x_{t+1}) \max_k (v_k(t) a_{kl})$$

DPマッチングとの類似点

- ・ DPマッチング：
 - 与えられた系列A、Bに対する格子点(1, 1)から(m, n)までの経路の中の、コスト最少のものを効率よく探すためのアルゴリズム
 - 効率化は、各格子点に至る最少経路のコストを記憶しておけば、いちいちすべての経路のコストを計算する必要がないことに注目
 - DPマッチングでは、マッチングの度合いを計算する関数の選定方法や知識ベースに蓄える系列集合の選定方法は任意 ⇒ この部分をマルコフ確率の理論に基づかせると「隠れマルコフモデル」

隠れマルコフモデルの応用 (1)

- 単語/品詞の隠れマルコフ・モデル
 - 例文: time flies like an arrow.



隠れマルコフモデルの応用 (2)

- ・ 単語/品詞の隠れマルコフ・モデル
 - 5通りの構文解釈が可能な文: time flies like an arrow.
 - S[NP/Time VP[VP/flies PP[PREP/like NP[DET/an N/arrow]]]]
 - S[VP/Time NP[NP/flies PP[PREP/like NP[DET/an N/arrow]]]]
 - S[NP[ADJ/Time NP/flies] VP[V/like NP[DET/an N/arrow]]]
 - S[NP[NP/Time NP/flies] VP[V/like NP[DET/an N/arrow]]]
 - S[VP[V/Time NP/flies] PP[PREP/like NP[DET/an N/arrow]]]

今日の講義のまとめ

– HCIの構成原理と構成方法 (2)

HCIと機械学習

- CNN
 - OpenPose
 - DensePose
- SVM
 - Touch & Activate
- 認識アルゴリズム
 - DPマッチング、隠れマルコフモデル
- 部分観測マルコフ決定過程
 - Situated Interaction (対話エージェント)



おわり