学位論文公聴会
February 20, 2012

Studies on Computational Learning via Discretization

（離散化に基づく計算論的学習に関する研究）

杉山 麿人

知能情報学専攻 知能情報基礎論分野

調査委員：山本 章博，阿久津 達也，田中 利幸（敬称略）

主要な結果

連続的な対象の離散化とそこからの学習という
2 つのプロセスを融合し，理論から実践までに渡って
学習の計算論的側面を解析した

主要な結果

> 連続的な対象の離散化とそこからの学習という
> 2 つのプロセスを融合し，理論から実践までに渡って
> 学習の計算論的側面を解析した

1. Gold 型学習モデル（極限同定）に基づく理論的解析
   - 図形（ユークリッド空間上のコンパクト集合）の学習
2. 離散化によるクラス分類
   - 新規尺度符号化ダイバージェンス
3. 高速かつ柔軟なクラスタリング
   - 新規アルゴリズム COOL と BOOL
4. 形式概念解析を用いた半教師あり学習と順序学習
   - 新規アルゴリズム SELF と LIFT

目次

- 第一部：理論
  - フラクタルとハウスドルフ距離を用いた図形の学習
    — 計算可能な 2 値分類を目指して
- 第二部：理論から実践
  - 符号化ダイバージェンス
  - MCL とグレイコードを用いたクラスタリング
  - 2 進符号化を活用した高速クラスタリング
- 第三部：形式概念と共に
  - 離散値・連続値混在データからの半教師あり学習
  - 半教師ありのマルチラベルクラス分類によるリガンド発見

目次

- 第一部：理論
  - フラクタルとハウスドルフ距離を用いた図形の学習 — 計算可能な 2 値分類を目指して
- 第二部：理論から実践
  - 符号化ダイバージェンス
  - MCL とグレイコードを用いたクラスタリング
  - 2 進符号化を活用した高速クラスタリング
- 第三部：形式概念と共に
  - 離散値・連続値混在データからの半教師あり学習
  - 半教師ありのマルチラベルクラス分類によるリガンド発見

フラクタルとハウスドルフ距離を用いた図形の学習
— 計算可能な 2 値分類を目指して

- Sugiyama, M., Hirowatari, E., Tsuiki, H., Yamamoto, A.: Learning Figures with the Hausdorff Metric by Fractals, ALT 2010, *LNAI* 6331
- Sugiyama, M., Hirowatari, E., Tsuiki, H., Yamamoto, A.: Learning Figures with the Hausdorff Metric by Fractals — Towards Computable Binary Classification, *Machine Learning* (submitted)

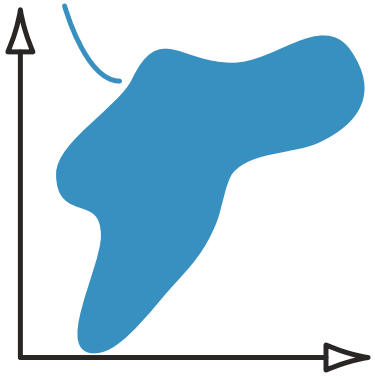
概要

- 計算論的な立場から2 値分類を再定式化し，解析する
  - 離散化された実数値データからの学習モデルを構築
  1. Gold 型学習モデルを基礎とする
  2. 計算可能性解析の発想を学習に適用する
  3. 学習機械が用いる表現形としてフラクタルを使う

概要

- 計算論的な立場から2 値分類を再定式化し，解析する
  - 離散化された実数値データからの学習モデルを構築
  1. Gold 型学習モデルを基礎とする
  2. 計算可能性解析の発想を学習に適用する
  3. 学習機械が用いる表現形としてフラクタルを使う

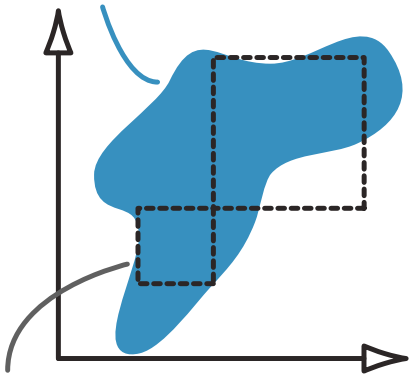
学習対象：図形（空でない $\mathbb{R}^d$ のコンパクト集合）



概要

- 計算論的な立場から2 値分類を再定式化し，解析する
  - 離散化された実数値データからの学習モデルを構築
  1. Gold 型学習モデルを基礎とする
  2. 計算可能性解析の発想を学習に適用する
  3. 学習機械が用いる表現形としてフラクタルを使う

学習対象：図形（空でない $\mathbb{R}^d$ のコンパクト集合）

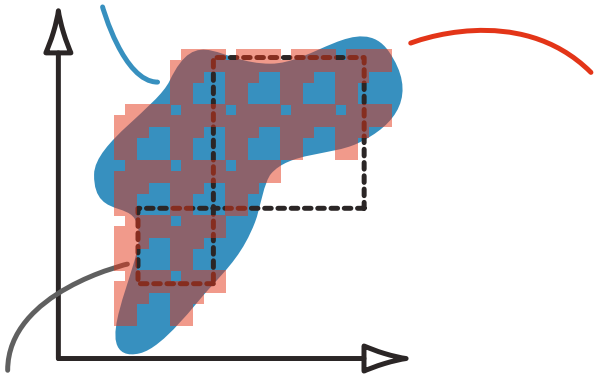


学習に使う（訓練）データ：超立方体（データのコードの接頭辞）

概要

- 計算論的な立場から2 値分類を再定式化し，解析する
  - 離散化された実数値データからの学習モデルを構築
  1. Gold 型学習モデルを基礎とする
  2. 計算可能性解析の発想を学習に適用する
  3. 学習機械が用いる表現形としてフラクタルを使う

学習対象：図形（空でない $\mathbb{R}^d$ のコンパクト集合）



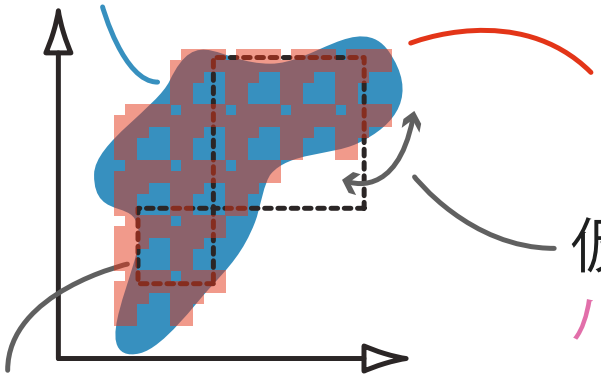
仮説：言語（プログラム）
- フラクタルを表現している

学習に使う（訓練）データ：超立方体（データのコードの接頭辞）

概要

- 計算論的な立場から2 値分類を再定式化し，解析する
  - 離散化された実数値データからの学習モデルを構築
  1. Gold 型学習モデルを基礎とする
  2. 計算可能性解析の発想を学習に適用する
  3. 学習機械が用いる表現形としてフラクタルを使う

学習対象：図形（空でない $\mathbb{R}^d$ のコンパクト集合）



仮説：言語（プログラム）
- フラクタルを表現している

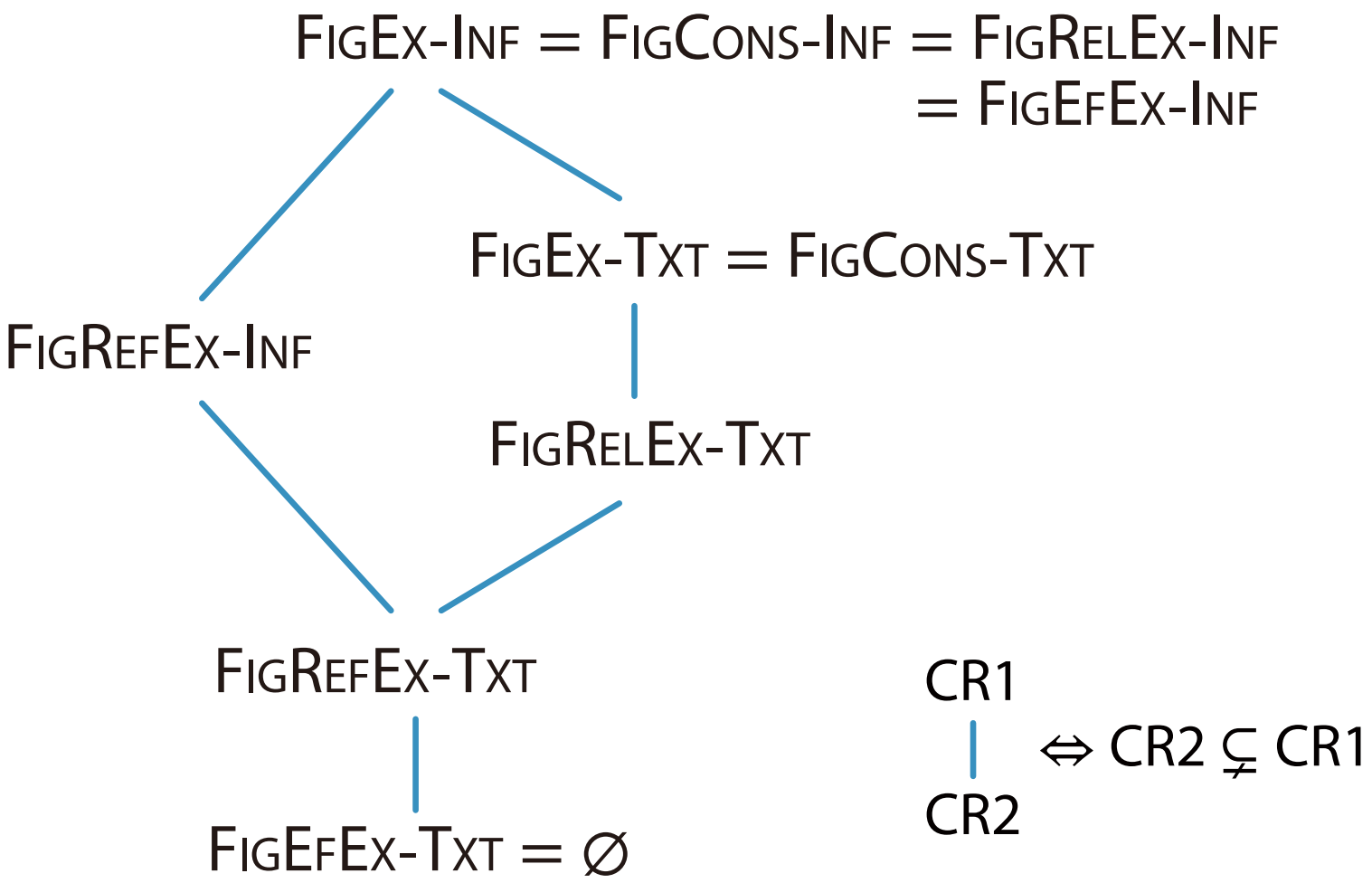
仮説の評価：
ハウスドルフ距離（汎化誤差）

学習に使う（訓練）データ：超立方体（データのコードの接頭辞）

結果

1. Gold 型学習モデルに基づき，自己相似集合（フラクタルの一族）を用いた図形の学習を定式化した
   - Collage Theorem によって自己相似集合の表現力を保証
2. 様々な学習基準のもとでの学習可能性の階層を明らかにした
3. 計算論的学習とフラクタル幾何との関係を示した
   - 学習の困難さをハウスドルフ次元とVC 次元で測る
4. 計算可能性解析におけるTTE（Type-2 Theory of Effectivity）の枠組を用いて，学習と計算の関係を示した

学習可能性の階層


FigEx-Inf = FigCons-Inf = FigRelEx-Inf
= FigEfEx-Inf
FigEx-Txt = FigCons-Txt
FigRefEx-Inf
FigRelEx-Txt
FigRefEx-Txt
FigEfEx-Txt = ∅
CR1
⇔ CR2 ⊊ CR1
CR2

離散化と学習

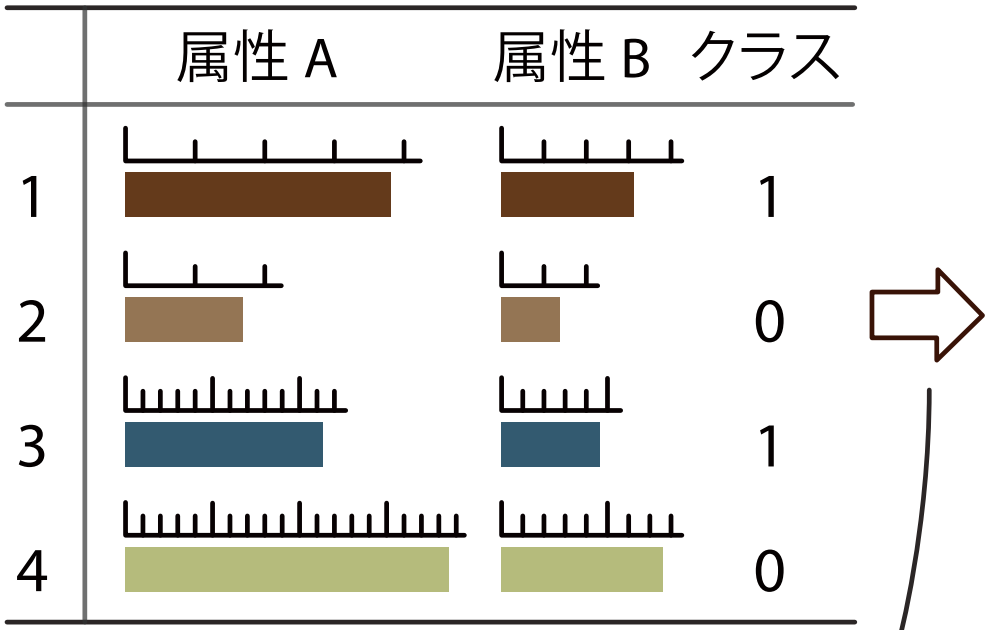
アナログデータ（実数）

	属性 A	属性 B	クラス
1			1
2			0
3			1
4			0

目的: 分類規則の学習

離散化と学習

アナログデータ（実数）



目的: 分類規則の学習

離散化と学習

アナログデータ（実数）

	属性 A	属性 B	クラス
1			1
2			0
3			1
4			0

測定による離散化

デジタルデータ（有理数）

属性 A	属性 B	クラス
1.5	0.6	1
0.6	0.2	0
1.1	0.4	1
1.8	0.7	0

目的: 分類規則の学習

離散化と学習

アナログデータ（実数）

	属性 A	属性 B	クラス
1	1.539582…	0.6069…	1
2	0.676711…	0.2655…	0
3	1.111577…	0.4998…	1
4	1.871501…	0.7569…	0

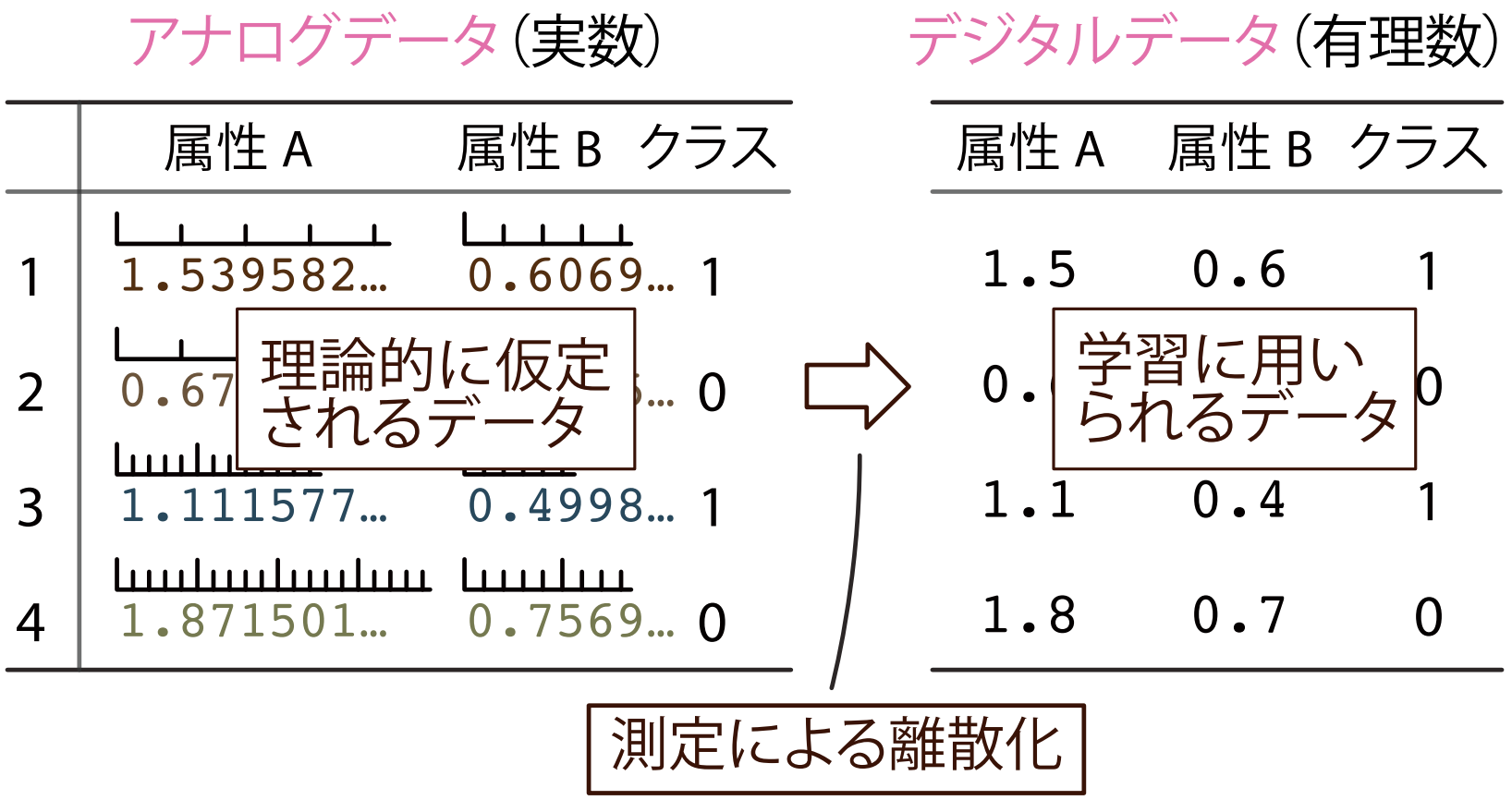
測定による離散化

デジタルデータ（有理数）

属性 A	属性 B	クラス
1.5	0.6	1
0.6	0.2	0
1.1	0.4	1
1.8	0.7	0

目的: 分類規則の学習

離散化と学習



目的: 分類規則の学習

離散化に伴う危うさ

- 以下の連立方程式を解く［Schröder, 2002］

$$40157959.0\,x + 67108865.0\,y = 1$$
$$67108864.5\,x + 112147127.0\,y = 0$$

  - 以下の解の公式で解ける

$$x = \frac{b_1 a_{22} - b_2 a_{12}}{a_{11} a_{22} - a_{21} a_{12}},\quad y = \frac{b_2 a_{11} - b_1 a_{21}}{a_{11} a_{22} - a_{21} a_{12}}$$

- 倍精度浮動小数点（IEEE 754）による計算：
  $x = 112147127,\ y = -67108864.5$
- 正しい答え：
  $x = 224294254,\ y = -134217729$

データを区間として扱う

アナログデータ（実数）

	属性 A	属性 B	クラス
1			1
2			0
3			1
4			0

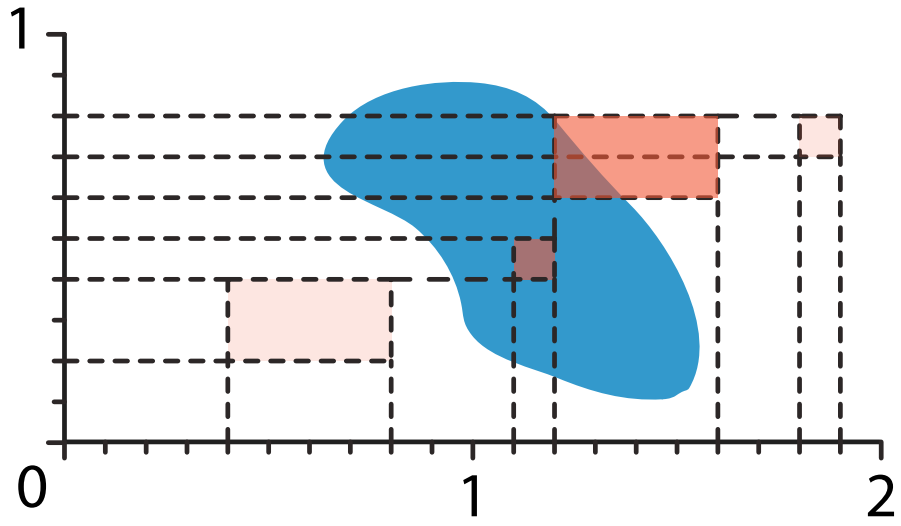
測定による離散化

デジタルデータ（有理数）

属性 A	属性 B	クラス
1.2~1.6	0.6~0.8	1
0.4~0.8	0.2~0.4	0
1.1~1.2	0.4~0.5	1
1.8~1.9	0.7~0.8	0

目的: 分類規則の学習

幾何学的な解釈



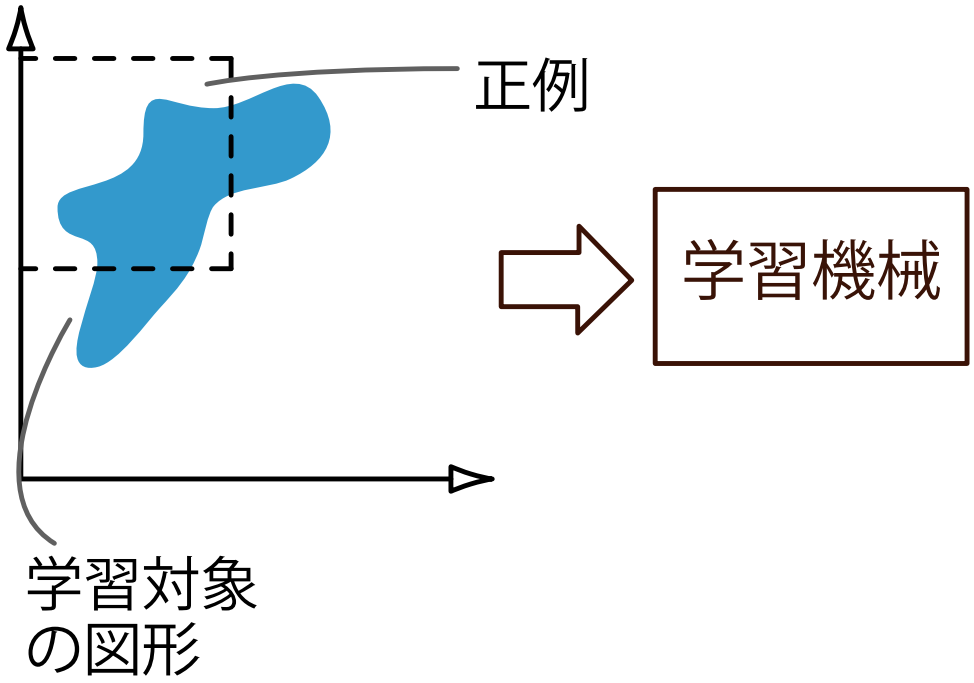
デジタルデータ（有理数）

属性 A	属性 B	クラス
1.2~1.6	0.6~0.8	1
0.4~0.8	0.2~0.4	0
1.1~1.2	0.4~0.5	1
1.8~1.9	0.7~0.8	0

- 離散化されたデータは $\mathbb{R}^d$ の区間
  - 区間の幅がデータの誤差に対応
- 学習機械はクラスが 1 の区間と交わる図形を学習する

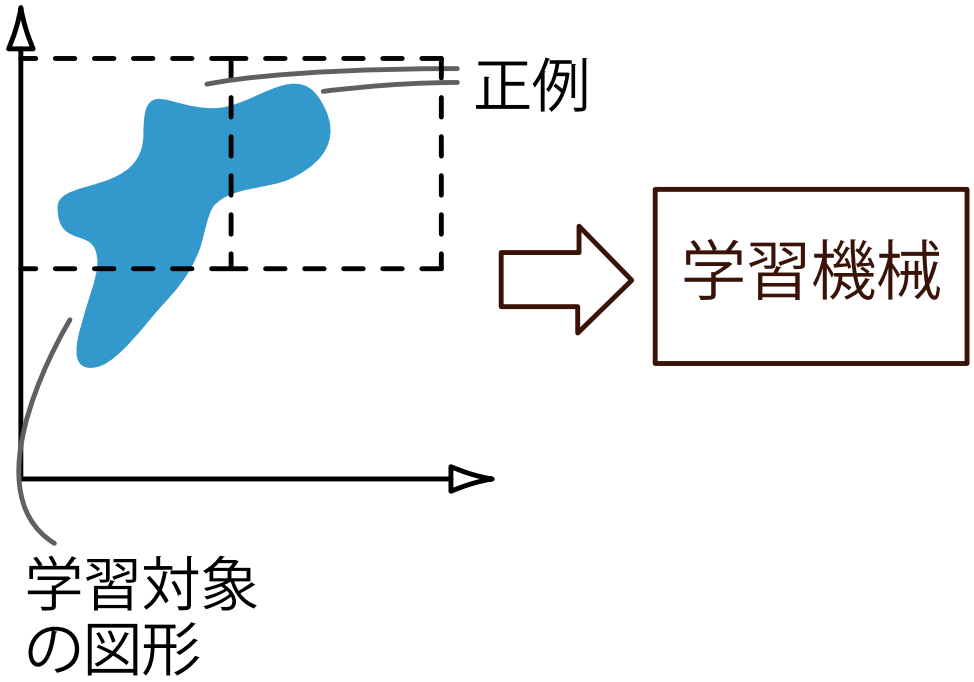
学習の概要

正例：学習対象と交わる閉区間
負例：学習対象と互いに疎な閉区間



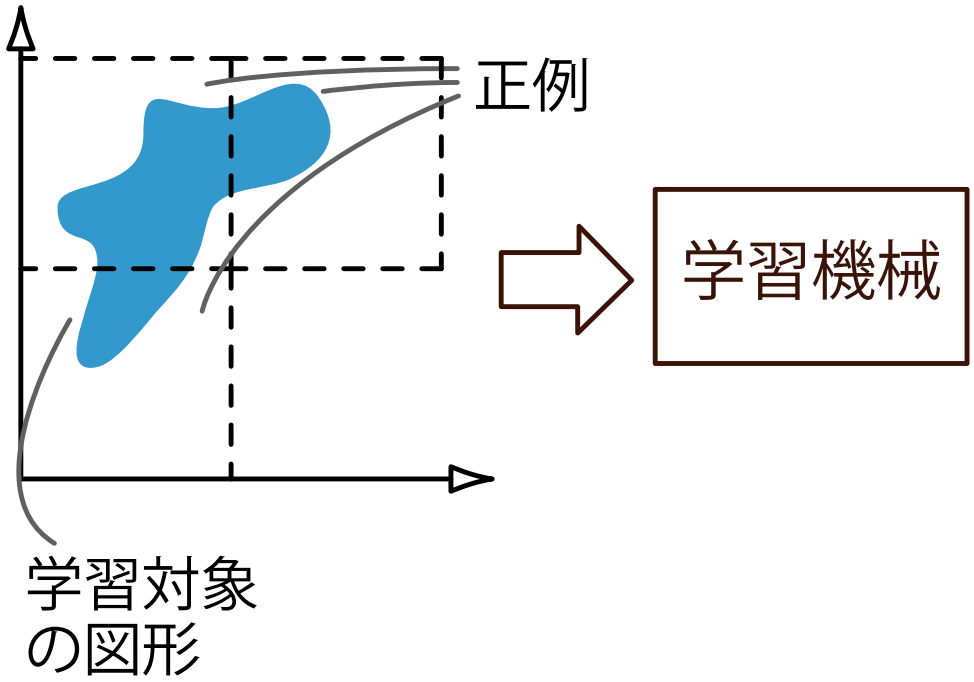
学習の概要

正例：学習対象と交わる閉区間
負例：学習対象と互いに疎な閉区間



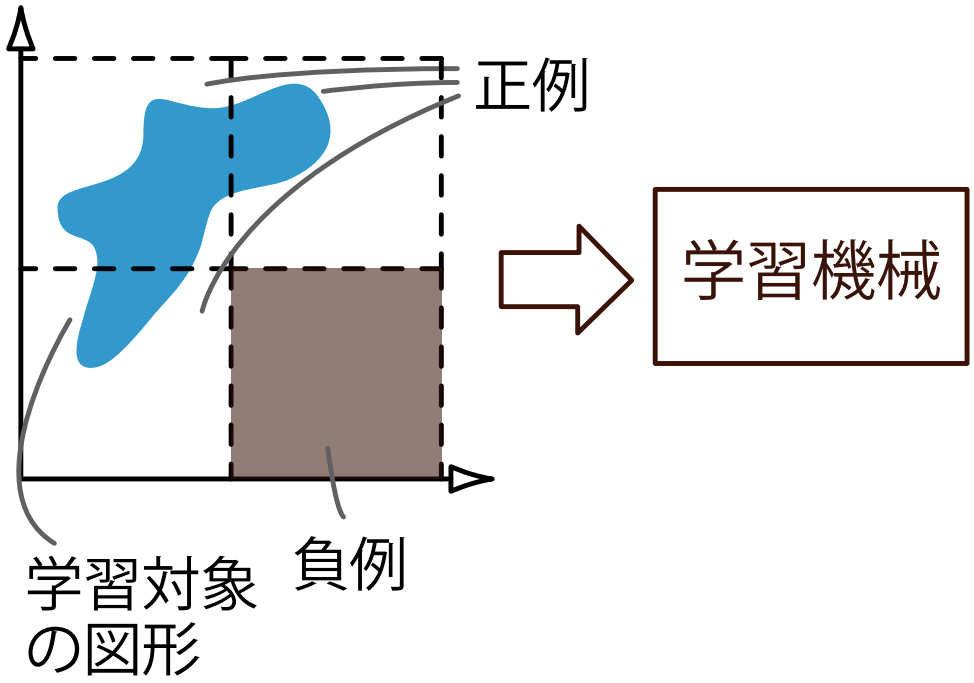
学習の概要

正例：学習対象と交わる閉区間
負例：学習対象と互いに疎な閉区間



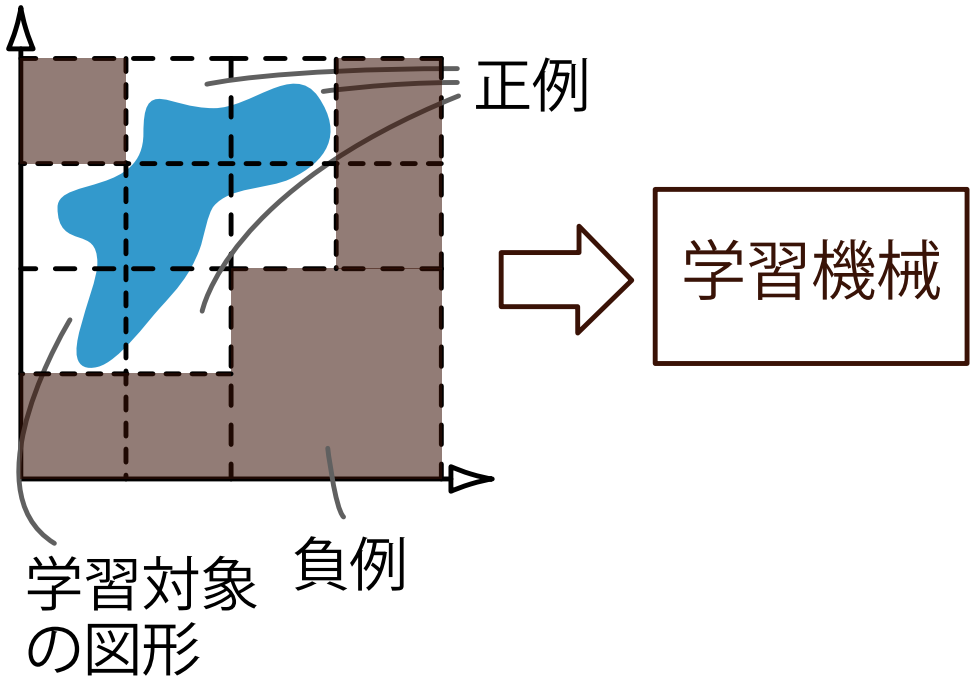
学習の概要

正例：学習対象と交わる閉区間
負例：学習対象と互いに疎な閉区間



学習の概要

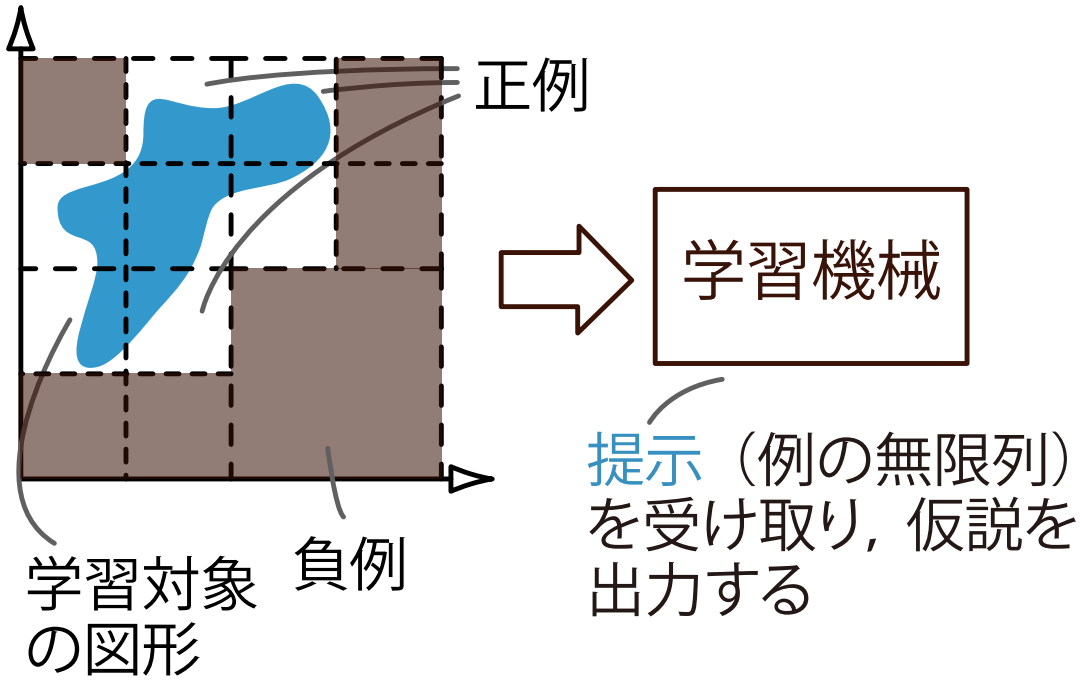
正例：学習対象と交わる閉区間
負例：学習対象と互いに疎な閉区間



例の誤差は小さくなっていく

学習の概要

正例：学習対象と交わる閉区間
負例：学習対象と互いに疎な閉区間

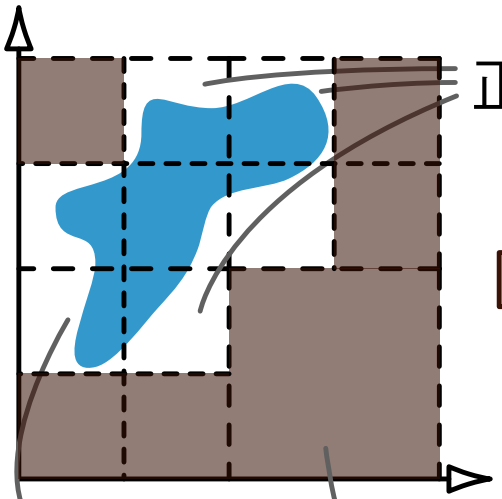


例の誤差は小さくなっていく

学習の概要

正例：学習対象と交わる閉区間
負例：学習対象と互いに疎な閉区間

仮説：自己相似集合を
表現する言語（プログラム）



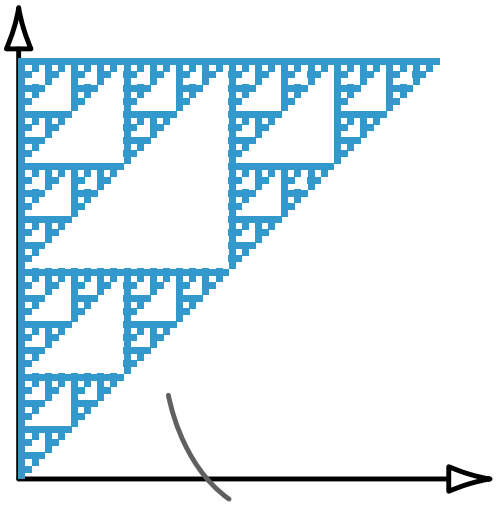
正例

学習対象
の図形

負例

学習機械

提示（例の無限列）
を受け取り，仮説を
出力する



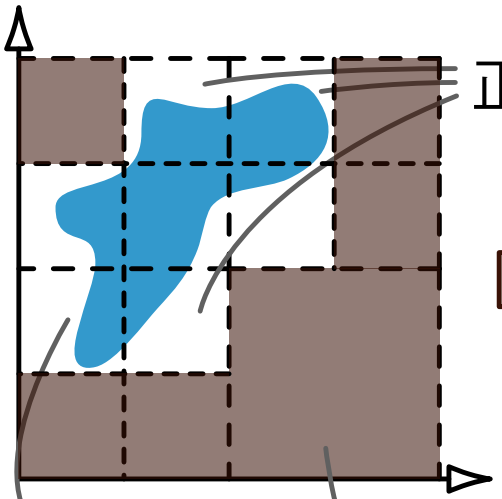
仮説によって
表現されている
自己相似集合

例の誤差は小さくなっていく

学習の概要

正例：学習対象と交わる閉区間
負例：学習対象と互いに疎な閉区間

仮説：自己相似集合を
表現する言語（プログラム）



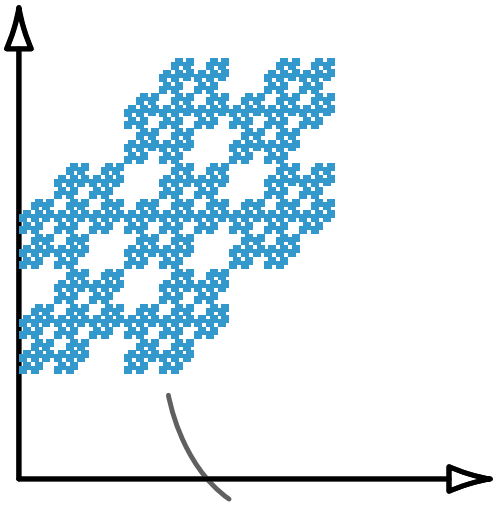
正例

負例

学習対象
の図形

学習機械

提示（例の無限列）
を受け取り，仮説を
出力する



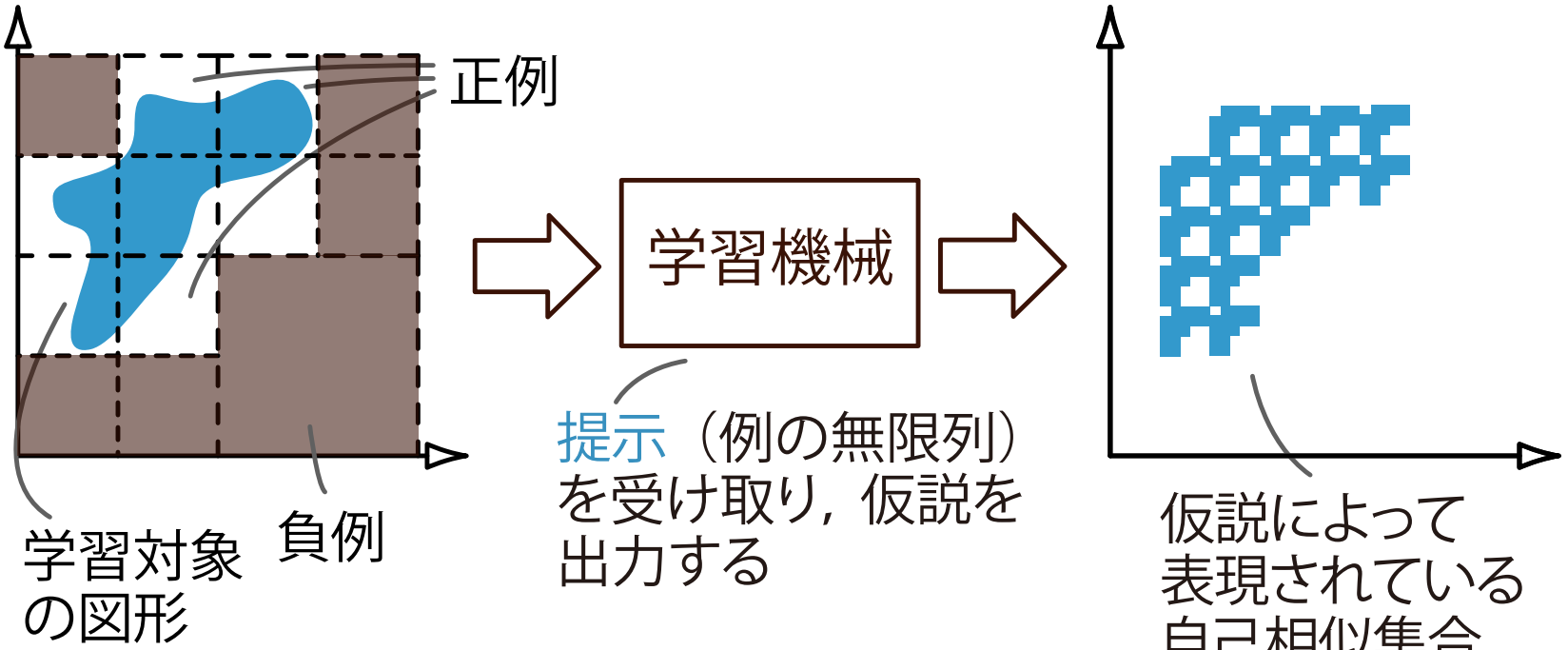
仮説によって
表現されている
自己相似集合

例の誤差は小さくなっていく

学習の概要

正例：学習対象と交わる閉区間
負例：学習対象と互いに疎な閉区間

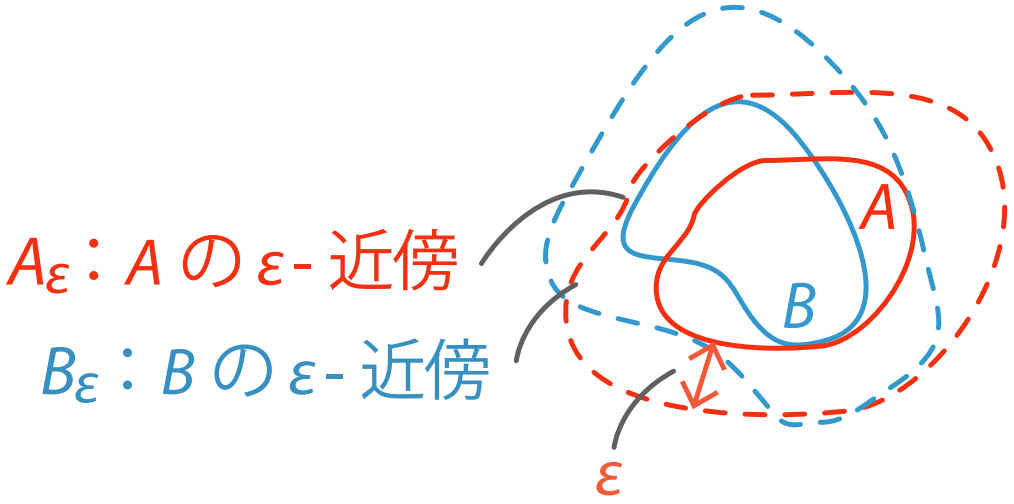
仮説：自己相似集合を
表現する言語（プログラム）



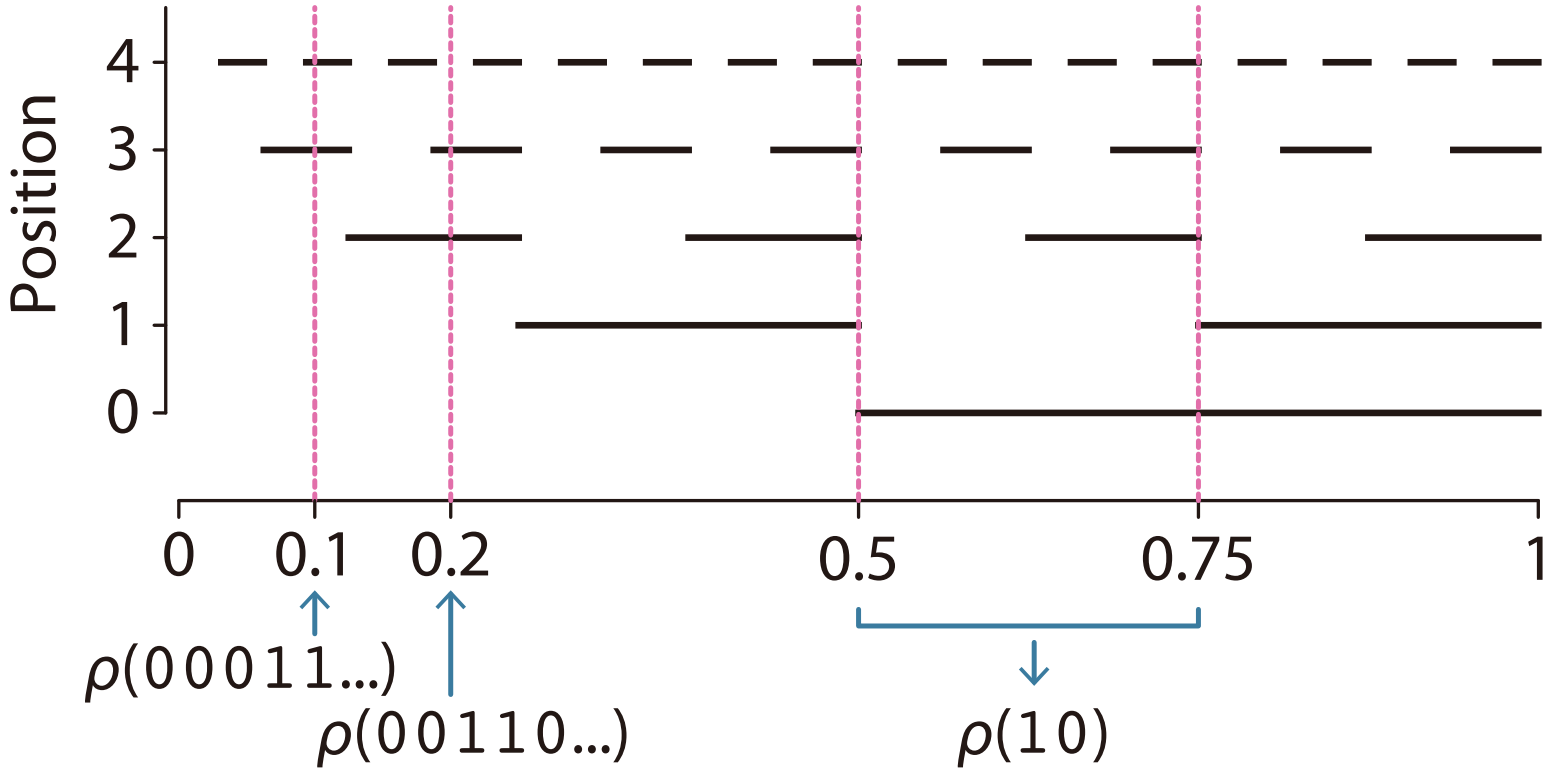
例の誤差は小さくなっていく

汎化誤差

- 仮説の「良さ」を汎化誤差で測る
  - ハウスドルフ距離を用いる
- 図形 A と B との間のハウスドルフ距離 $d_H(A, B)$ とは，$A \subset B_\varepsilon$ と $B \subset A_\varepsilon$ を満たす最小の ε



2 進表現



$$\rho(w) := \left[\sum w_i \cdot 2^{-(i+1)},\ \sum w_i \cdot 2^{-(i+1)} + 2^{|w|} \right]$$

仮説：言語で自己相似集合を表現する

- 言語 H に対する表記を以下のように定める

$$\begin{cases} H^0 := \{ \lambda \}, \\ H^k := \left\{ \langle w^1 u^1, \ldots, w^d u^d \rangle \;\middle|\; \begin{array}{l} \langle w^1, \ldots, w^d \rangle \in H^{k-1}, \\ \langle u^1, \ldots, u^d \rangle \in H \end{array} \right\} \end{cases}$$

- 言語 H が表す自己相似集合 $\kappa(H)$ は

$$\kappa(H) := \bigcap_{k=0}^{\infty} \bigcup \rho(H^k)$$

- 例：仮説 $H = \{\langle 0,0 \rangle, \langle 0,1 \rangle, \langle 1,1 \rangle\}$ とすると，

$$H^0 = \varnothing,\ H^1 = \{\langle 0,0 \rangle, \langle 0,1 \rangle, \langle 1,1 \rangle\},$$

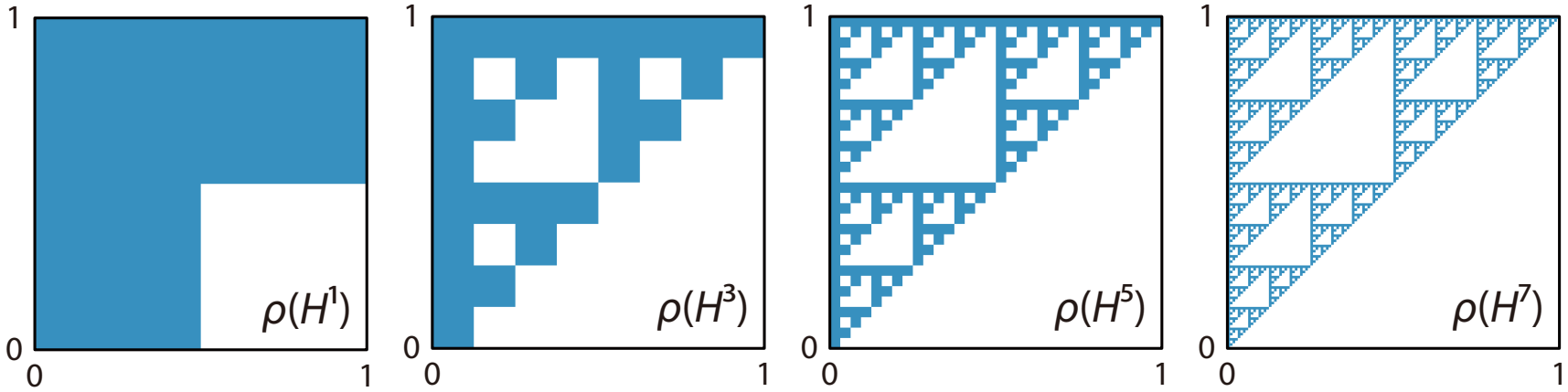
$$H^2 = \left\{ \begin{array}{l} \langle 00,00 \rangle, \langle 00,01 \rangle, \langle 01,01 \rangle, \langle 00,10 \rangle, \langle 00,11 \rangle, \\ \langle 01,11 \rangle, \langle 10,10 \rangle, \langle 10,11 \rangle, \langle 11,11 \rangle \end{array} \right\}$$

例：シェルピンスキー三角形

- このとき $\kappa(H)$ はシェルピンスキー三角形になる
  - H^1 は以下の写像でマップされる矩形に対応する

$$\varphi_1 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \varphi_2 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1/2 \end{bmatrix},$$

$$\varphi_3 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}$$



仮説の性質

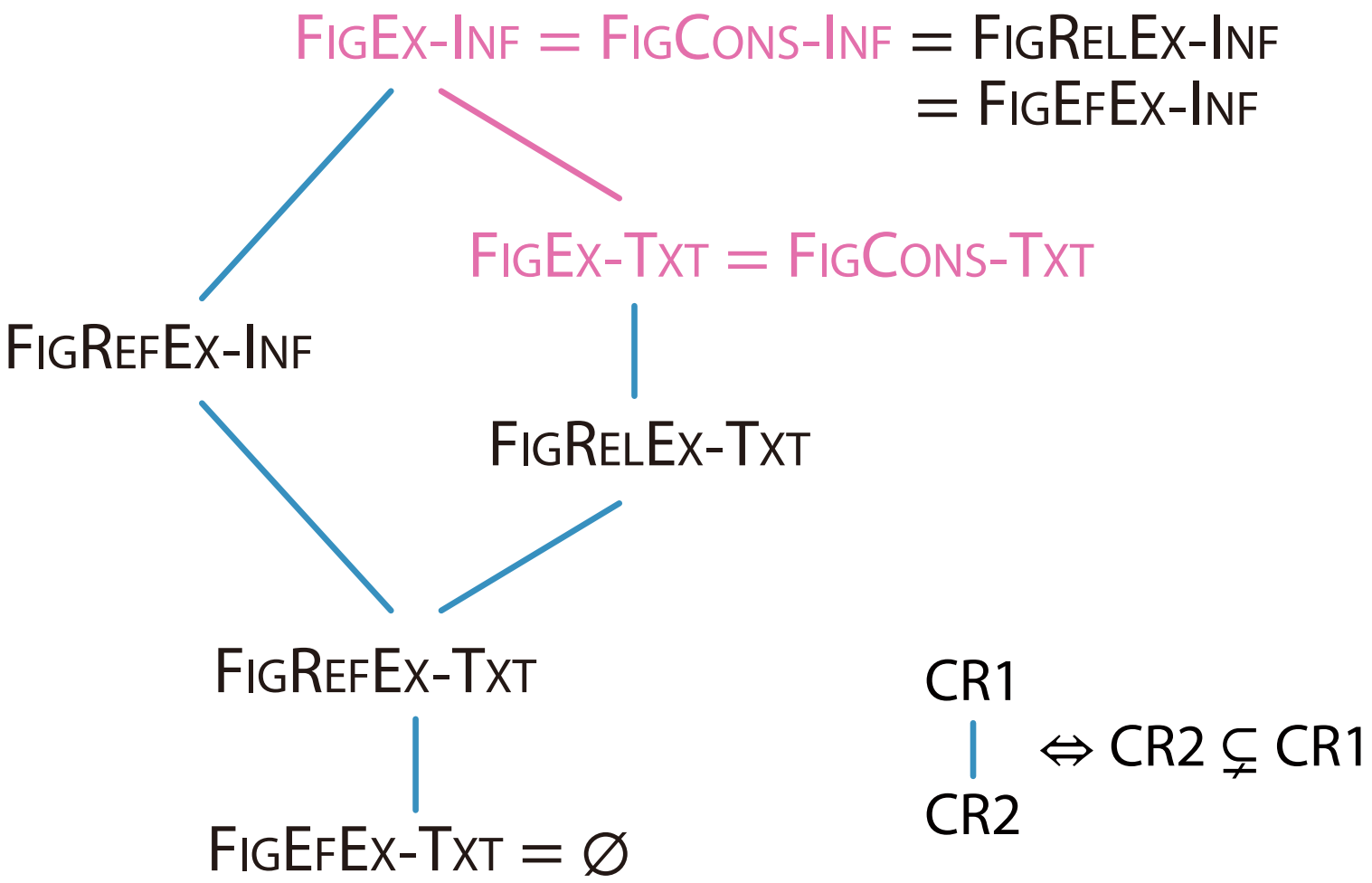
- 仮説の健全性：
  任意の仮説 $H \in \mathscr{H}$ に対して，集合 $\kappa(H)$ は自己相似集合
- 仮説の完全性（表現力）：
  任意の $\delta \in \mathbb{R}$ と任意の図形 K に対して，$d_{\mathrm{H}}(K, \kappa(H)) < \delta$ を満たす仮説 H が必ず存在する
  - Collage Theorem を用いた補題 [Falconer, 2003] からいえる
- 仮説の計算可能性：
  任意の仮説 $H \in \mathscr{H}$ に対して，以下のように定める分類器 h は計算可能

$$h(w) = \begin{cases} 1 & \text{if } \rho(w) \cap \kappa(H) \neq \emptyset, \\ 0 & \text{otherwise} \end{cases}$$

極限での図形の学習

- Gold 型学習モデルを基礎として図形の学習可能性を解析
  - まず，学習対象を記述する仮説が必ず存在する場合を考える
- 学習機械が図形の集合 $\mathscr{F} \subseteq \mathscr{K}^*$ を**FigEx-Inf** 学習（**FigEx-Txt** 学習）する $\iff$
  任意の $K \in \mathscr{F}$ と完全提示（正提示）に対して，
  出力が仮説 H に収束し，$\mathrm{GE}(K, P) = 0$
  - $\mathscr{K}^*$：図形全体の集合，汎化誤差 $\mathrm{GE}(K, H) := d_{\mathrm{H}}(K, \kappa(H))$
  - $\mathscr{F}$ は **CR** 学習可能 $\iff$ $\mathscr{F} \in$ **CR**
- 無矛盾学習（**FigCons-Inf** 学習 及び**FigCons-Txt** 学習）では，
  学習機械は必ずそのときまでに受け取った例に無矛盾な仮説を出力するという制約を課す

学習可能性の階層


FigEx-Inf = FigCons-Inf = FigRelEx-Inf
= FigEfEx-Inf
FigEx-Txt = FigCons-Txt
FigRefEx-Inf
FigRelEx-Txt
FigRefEx-Txt
FigEfEx-Txt = ∅

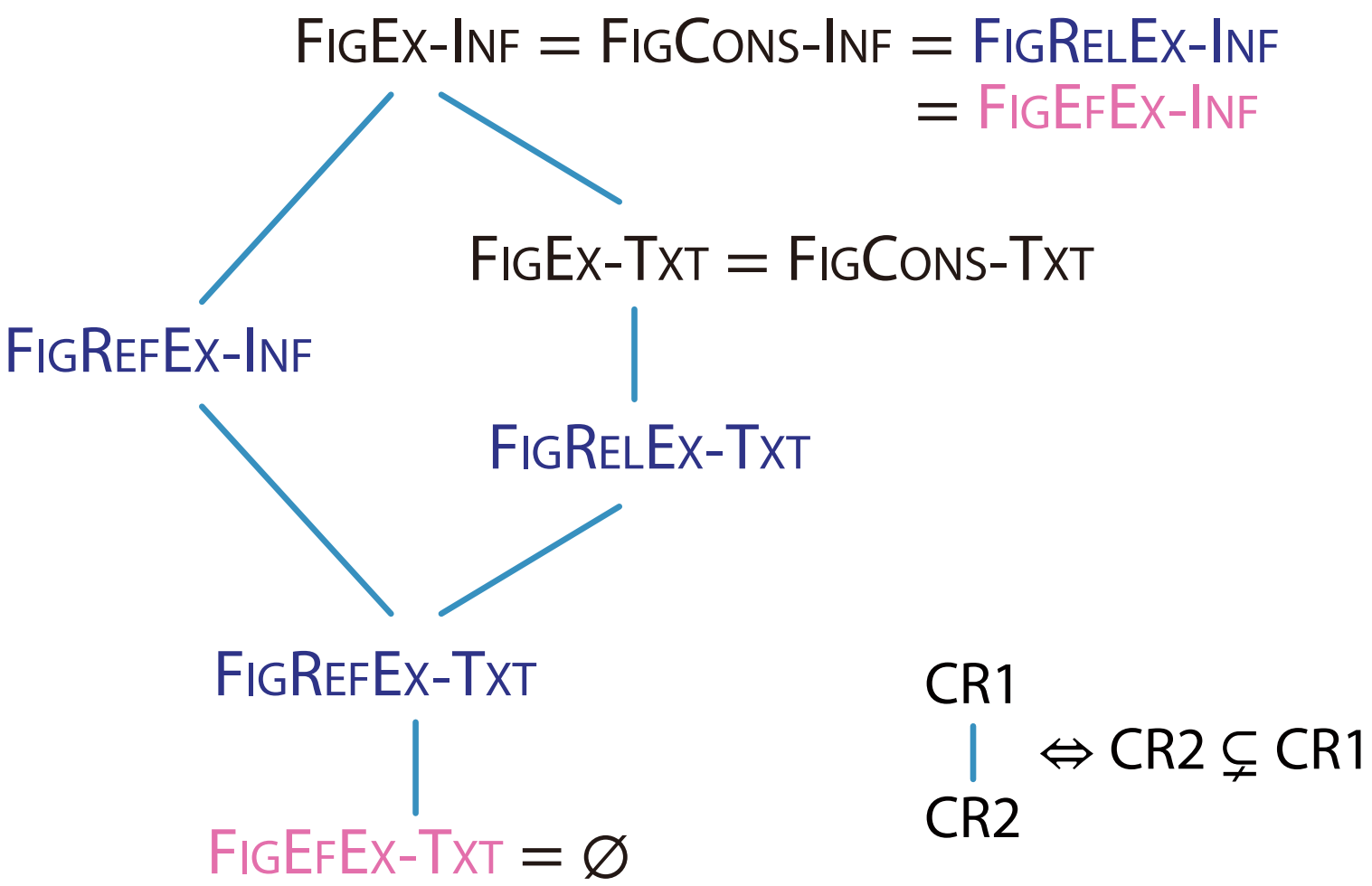
CR1
⇔ CR2 ⊊ CR1
CR2

任意の図形の学習

- FIGEX-INF 学習（FIGEX-TXT 学習）では，学習対象 $K \notin \mathscr{F}$ のときは何も保証されない
- そのような場合も含めて考察する
  - 自己相似集合だけでなく任意の図形を学習対象として扱う

1. 反駁可能学習：もし $K \notin \mathscr{F}$ なら，学習機械は停止する
2. 信頼可能学習：もし $K \notin \mathscr{F}$ なら，仮説は収束しない
   - これらは言語の学習でも同様の考察 [Mukouchi and Arikawa, 1995]
3. 実効的学習：汎化誤差は単調に 0 に収束

学習可能性の階層


FigEx-Inf = FigCons-Inf = FigRelEx-Inf
= FigEfEx-Inf
FigEx-Txt = FigCons-Txt
FigRefEx-Inf
FigRelEx-Txt
FigRefEx-Txt
FigEfEx-Txt = ∅
CR1
CR2
⇔ CR2 ⊊ CR1

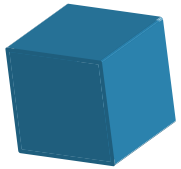
ハウスドルフ次元 $\dim_H$

- ハウスドルフ次元はフラクタル幾何における中心的概念
  - どのくらい空間を占めているかを表す
  - ハウスドルフ測度で定義される
- 通常の（位相）次元の拡張になっている

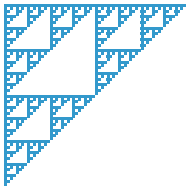
Point  = 0

Line = 1

Plane  = 2

Cube  = 3

Sierpiński triangle



= log3 / log2
= 1.584...

$\dim_H$ で学習の複雑さを測る

- 一般の場合：任意の図形 $K \in \mathscr{K}^*$ と $s < \dim_H K$ に対して，
  離散化レベル k が十分大きいとき，

  レベル k での正例の数 $\geqslant 2^{ks}$

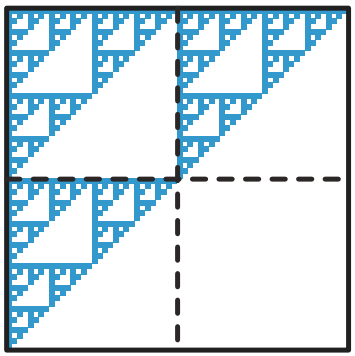
- 特殊な場合：任意の図形 $K \in \kappa(\mathscr{H})$ に対して，
  （$K = \kappa(H)$ となる仮説 H が存在する）
  離散化レベル k が十分大きいとき，

  レベル k での正例の数 $\geqslant 2^{k \dim_H K}$

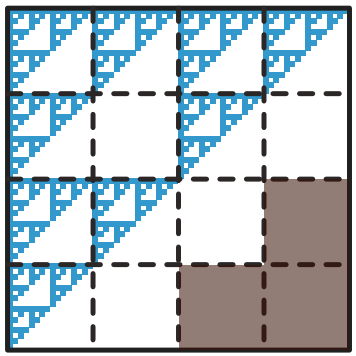
$\dim_H$ で学習の複雑さを測る（例）

- 学習対象の図形 K をシェルピンスキー三角形とする ($\dim_H K = 1.584\ldots$)
- 離散化レベル 1 では：
  正例の数 $\geqslant 3$ ($2^{\dim_H K} = 3$)
- 離散化レベル 2 では：
  正例の数 $\geqslant 9$ ($4^{\dim_H K} = 9$)

レベル 1



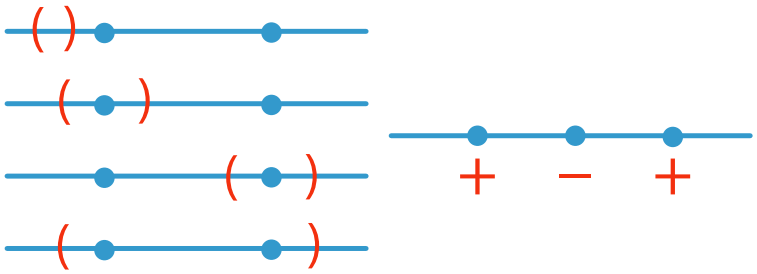
レベル 2



VC 次元 $\dim_{VC}$

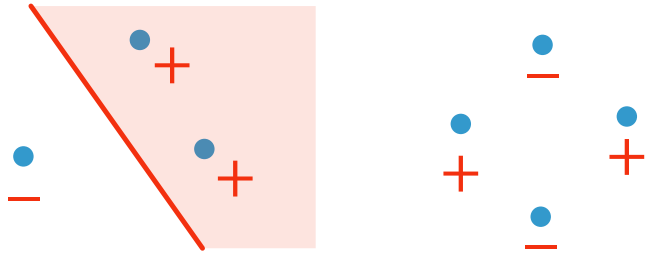
- VC 次元はクラスの分離可能性（複雑性）を表す値
  - いくつの点が分離可能か？
  - Valiant 型（PAC）学習モデルでは，学習に必要なサンプルサイズが VC 次元で特徴付けられる

I: 数直線 $\mathbb{R}$ 上の開区間
全体からなるクラス



$\dim_{VC} \boldsymbol{I} = 2$

H: 平面 $\mathbb{R}^2$ 上の半空間
全体からなるクラス



$\dim_{VC} \boldsymbol{H} = 3$

$\dim_H$ と $\dim_{VC}$ で学習の複雑さを測る

- 離散化レベル k において，$\dim_{VC} \mathscr{H}^k = 2^{kd}$
  - $\mathscr{H}^k := \{H \in \mathscr{H} \mid |w| = k \text{ for all } w \in H\}$
- 一般の場合：任意の図形 $K \in \mathscr{K}^*$ と $s < \dim_H K$ に対して，
  離散化レベル k が十分大きいとき，

  レベル k での正例の数 $\geqslant (\dim_{VC} \mathscr{H}^k)^{s/d}$
- 特殊な場合：任意の図形 $K \in \kappa(\mathscr{H})$ に対して，
  （$K = \kappa(H)$ となる仮説 H が存在する）
  離散化レベル k が十分大きいとき，

  レベル k での正例の数 $\geqslant (\dim_{VC} \mathscr{H}^k)^{\dim_H K/d}$

まとめ

- 2 値分類を図形の学習として計算論的な立場から定式化した
  - Gold 型学習モデルを用いた
  - 実数値データの離散化プロセスを，計算可能性解析における実効的計算モデルを用いて適切に扱った
  - 仮説（分類器）の表現形としてフラクタル（自己相似集合）を用いた
- 学習可能性の階層を明らかにした
- ハウスドルフ次元とVC 次元を用いて，
  学習とフラクタル幾何の関係を示した
- TTE（Type-2 Thoery of Effectivity）の枠組を用いて，
  学習と計算の関係を示した

目次

- 第一部：理論
  - フラクタルとハウスドルフ距離を用いた図形の学習
    — 計算可能な 2 値分類を目指して
- 第二部：理論から実践
  - 符号化ダイバージェンス
  - MCL とグレイコードを用いたクラスタリング
  - 2 進符号化を活用した高速クラスタリング
- 第三部：形式概念と共に
  - 離散値・連続値混在データからの半教師あり学習
  - 半教師ありのマルチラベルクラス分類によるリガンド発見

符号化ダイバージェンス

- 小林茂夫, 杉山麿人.:
  生命科学研究に成功するための統計法ノート, 講談社, 2009.
- Sugiyama, M., Yamamoto, A.:
  The Coding Divergence for Measuring the Complexity
  of Separating Two Sets, ACML 2010
- Sugiyama, M., Yoshioka, T., Yamamoto, A.:
  High-throughput Data Stream Classification on Trees, ALSIP 2011

概要・結果

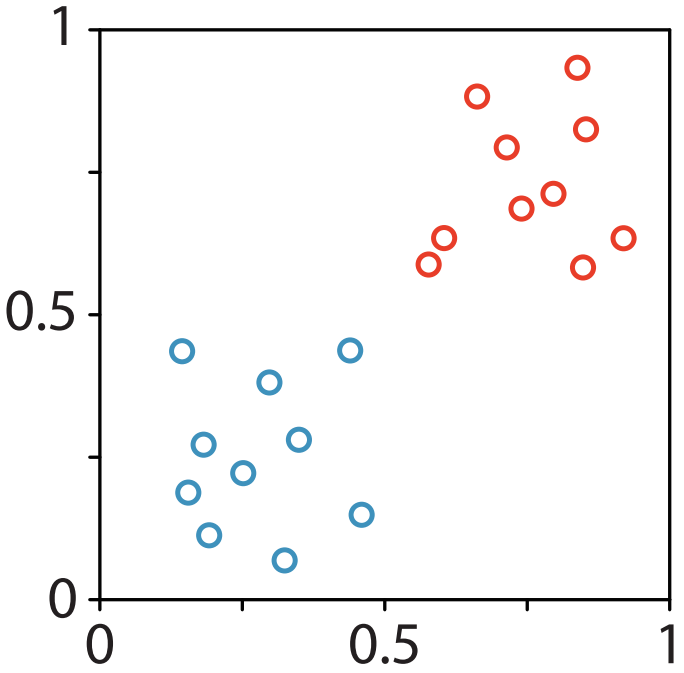
- **主な結果:**
  1. 符号化ダイバージェンスという 2 つの実数値データ集合の間の異なり具合を測る新規の尺度を提案した
     - 2 つの集合を分離するときの困難さを定量化する
  2. 怠惰学習器を構築し，クラス分類において既存手法と遜色ない性能を持つことを示した

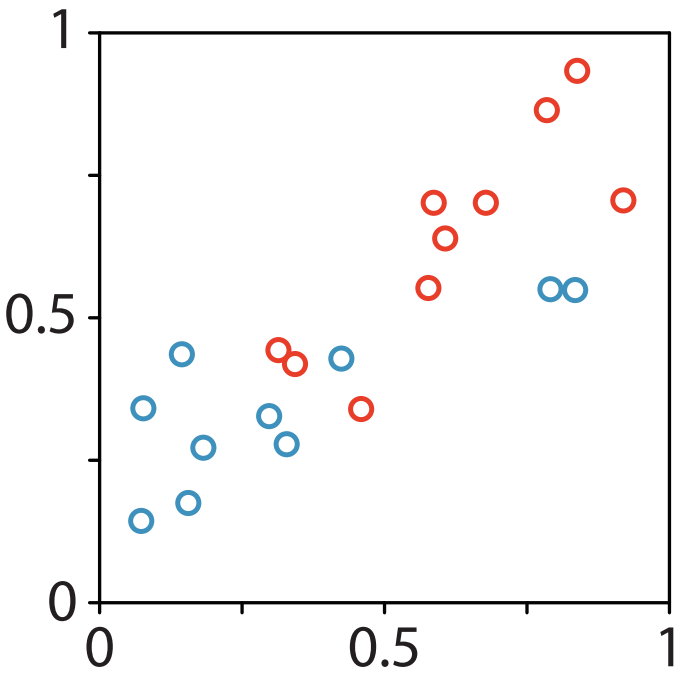
概要・結果

- **主な結果:**
  1. 符号化ダイバージェンスという 2 つの実数値データ集合の間の異なり具合を測る新規の尺度を提案した
     - 2 つの集合を分離するときの困難さを定量化する
  2. 怠惰学習器を構築し，クラス分類において既存手法と遜色ない性能を持つことを示した

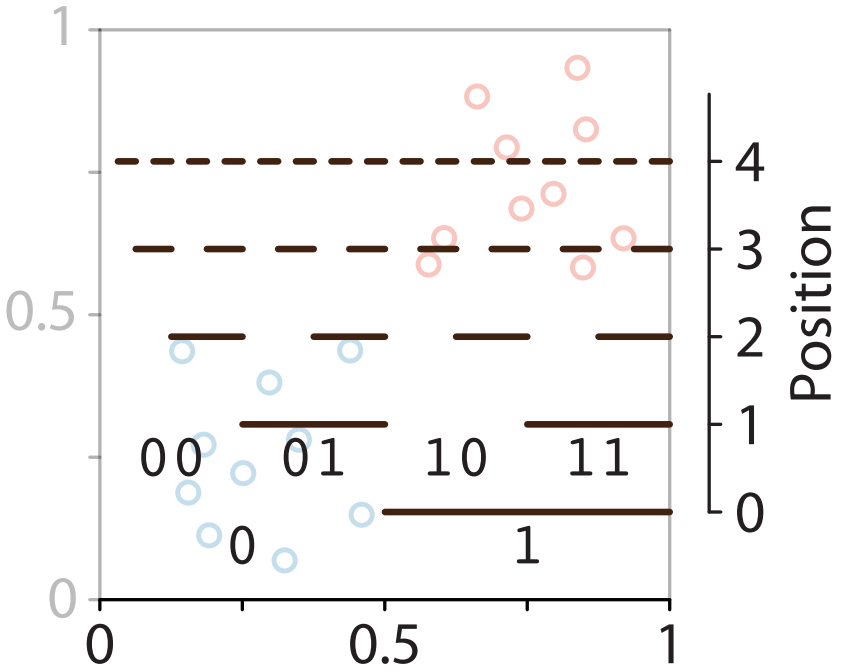
- **主な手法:**
  1. ユークリッド空間 $\mathbb{R}^d$ 上の実数値データをカントール空間 Σ^ω へ位相的に埋め込む（離散化）
  2. Σ^ω の中でデータに無矛盾なモデル（開集合）を学習する
  3. 学習された開集合を表現するコードの長さで定量化

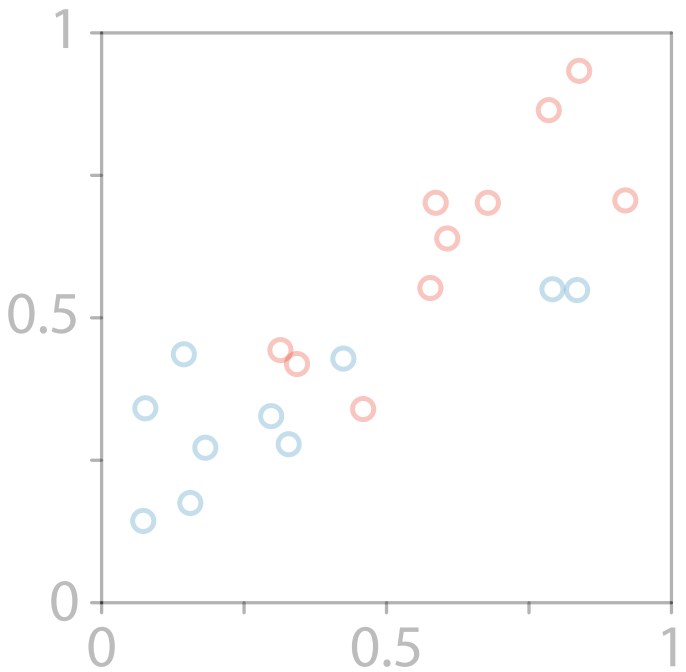
符号化ダイバージェンスの例





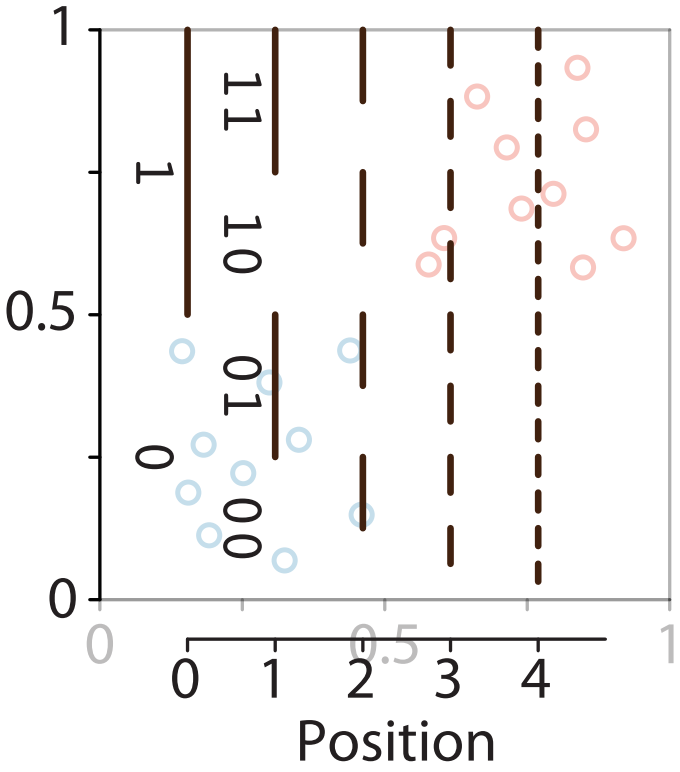
符号化ダイバージェンスの例



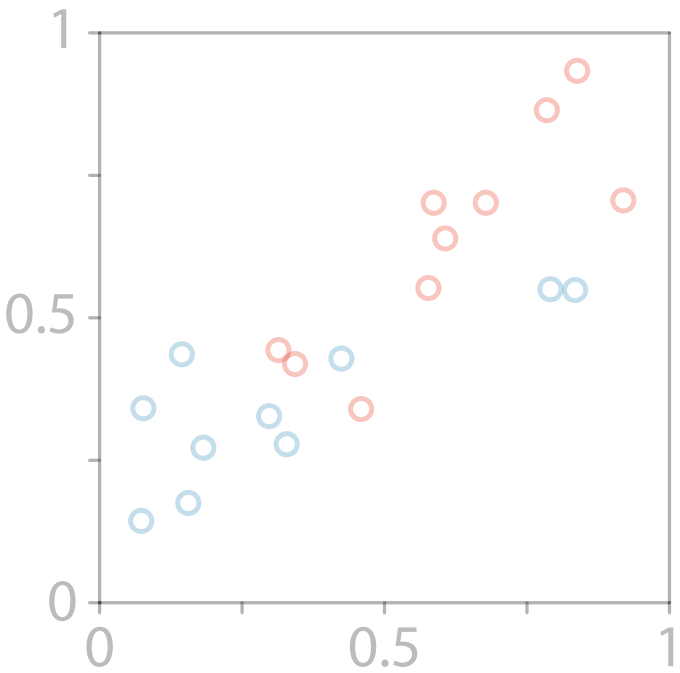


実数の 2 進符号化

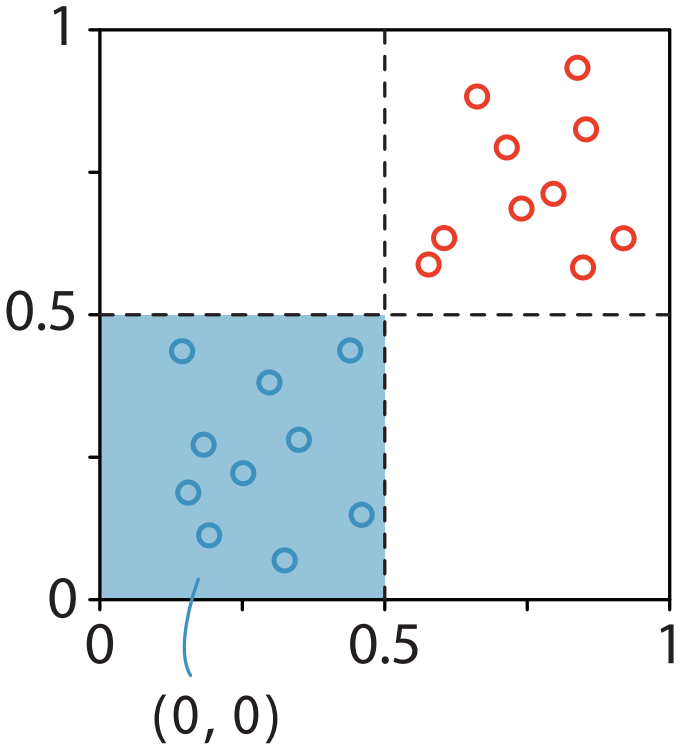
符号化ダイバージェンスの例

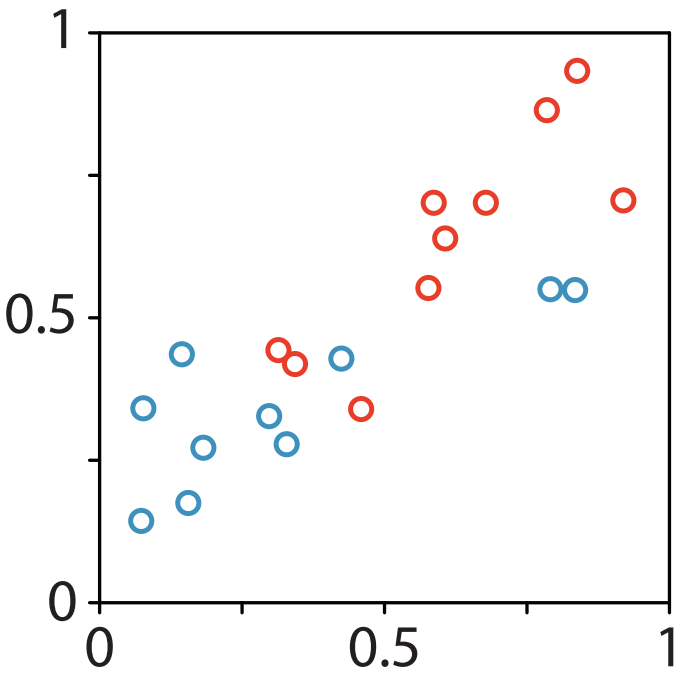


実数の 2 進符号化

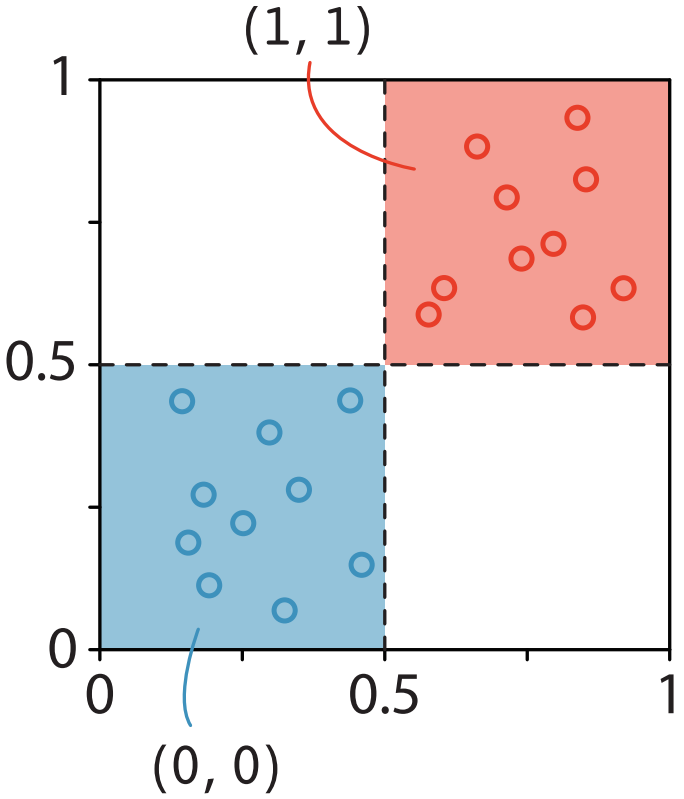


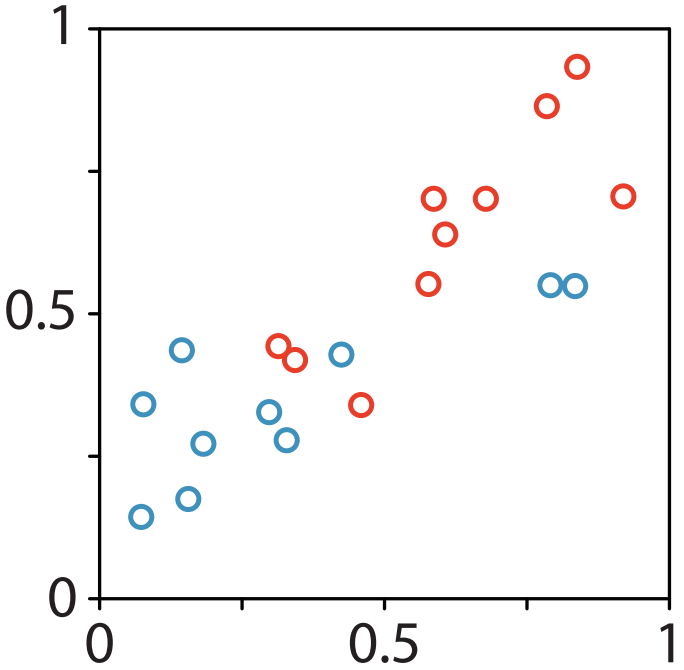
符号化ダイバージェンスの例



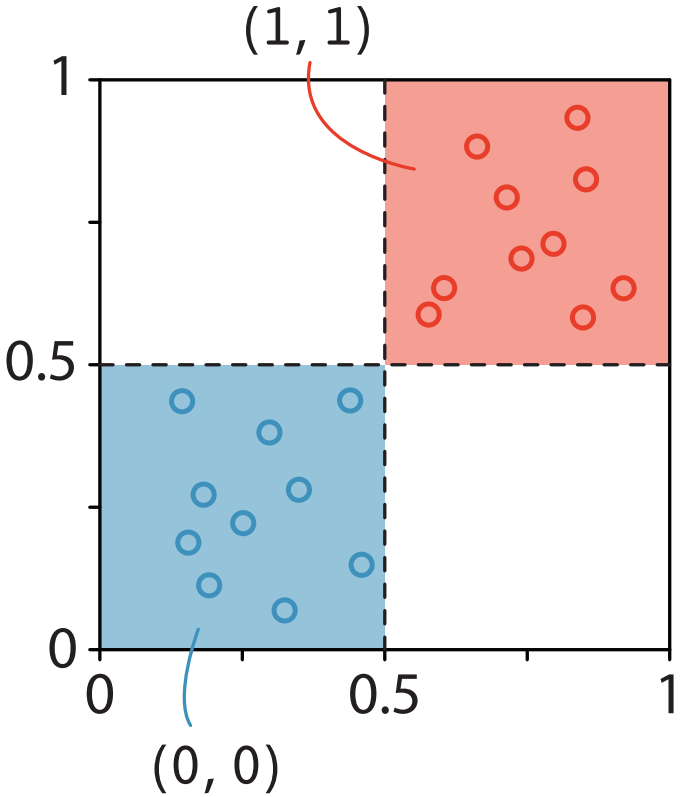


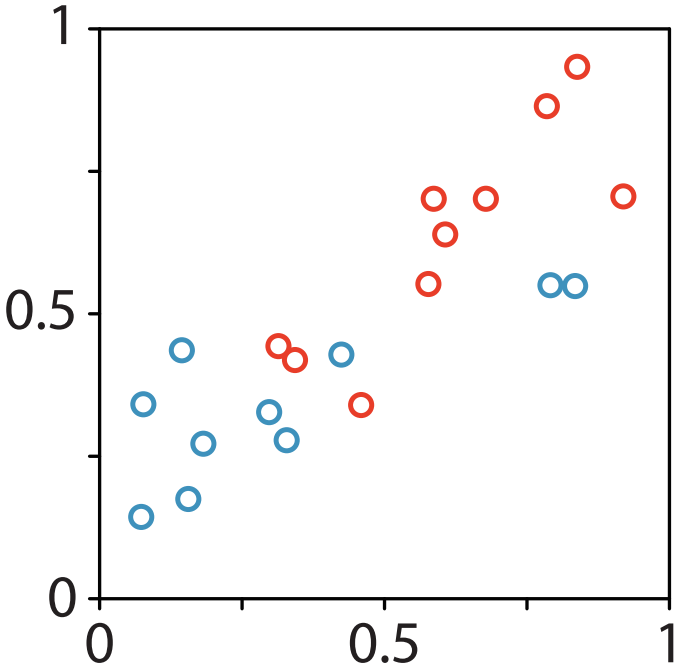
符号化ダイバージェンスの例





符号化ダイバージェンスの例

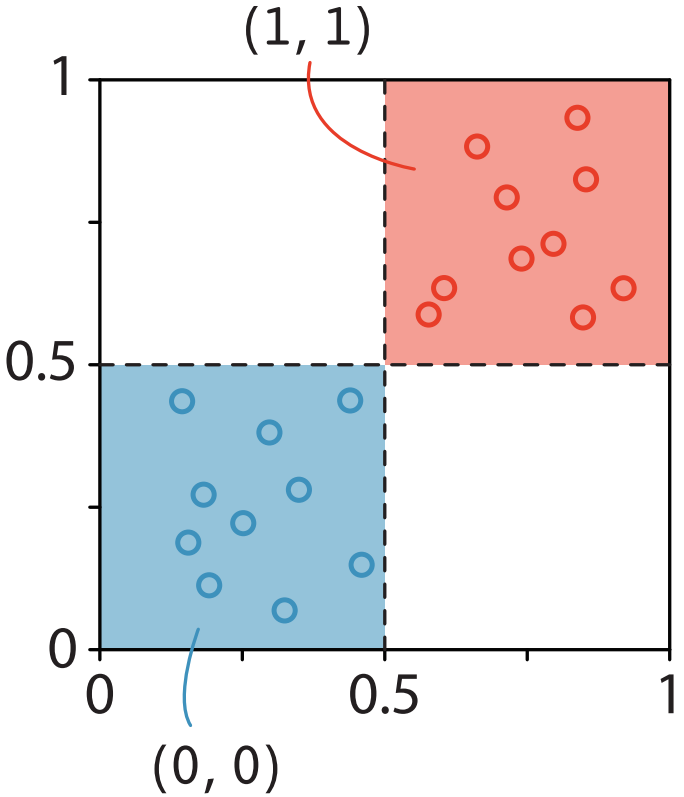


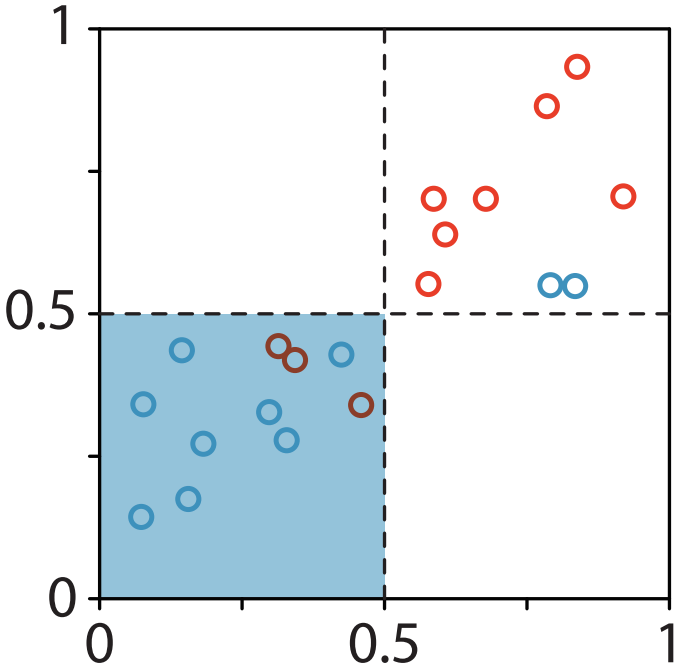


符号化ダイバージェンス：

2/10 + 2/10 = 0.4

符号化ダイバージェンスの例

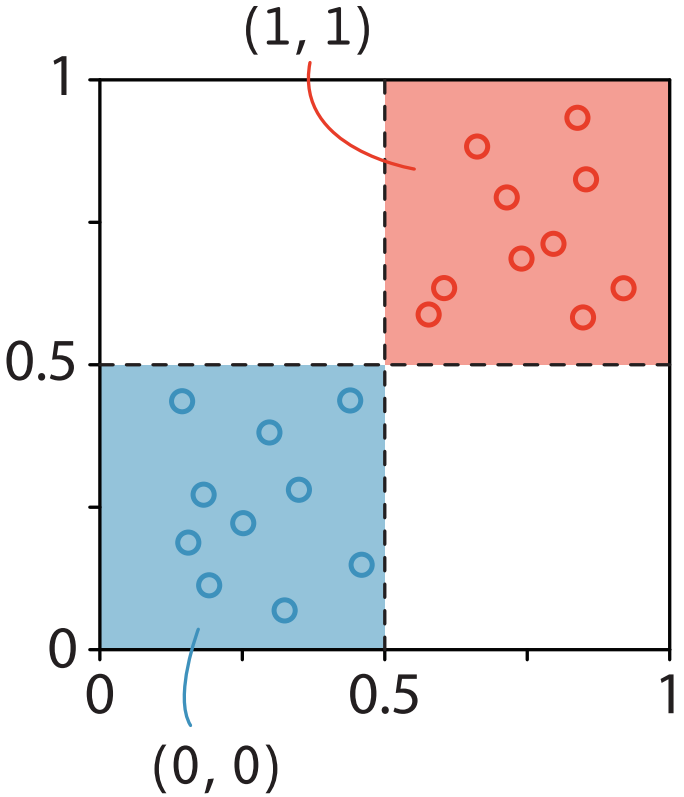


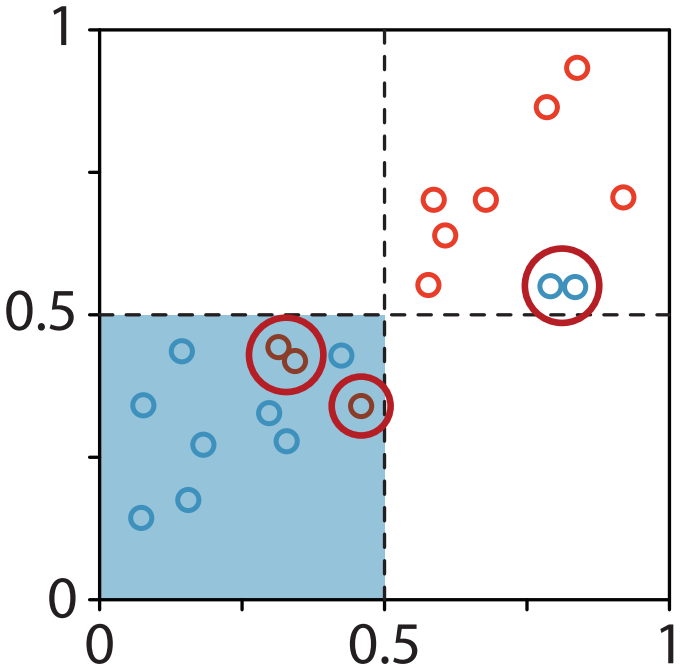


符号化ダイバージェンス：

2/10 + 2/10 = 0.4

符号化ダイバージェンスの例

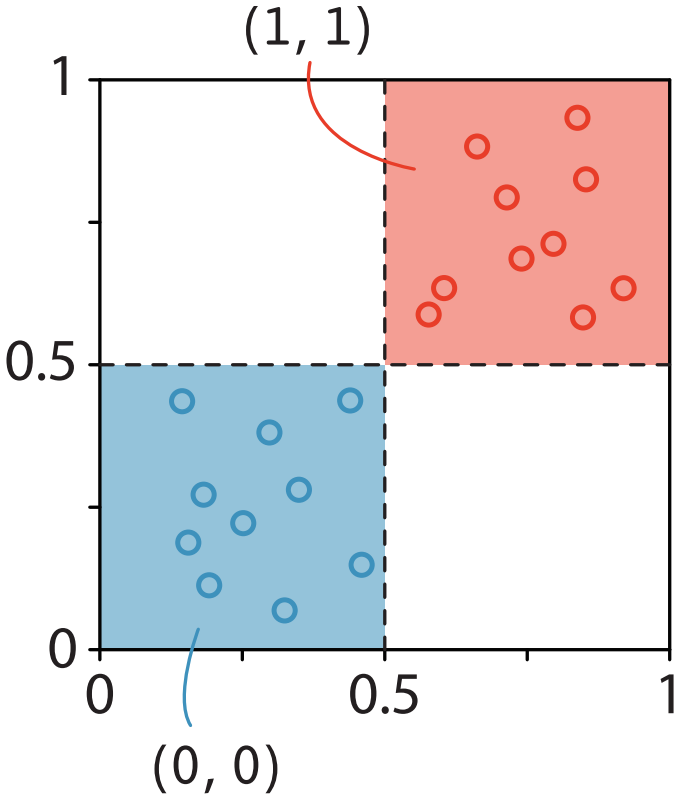




符号化ダイバージェンス：

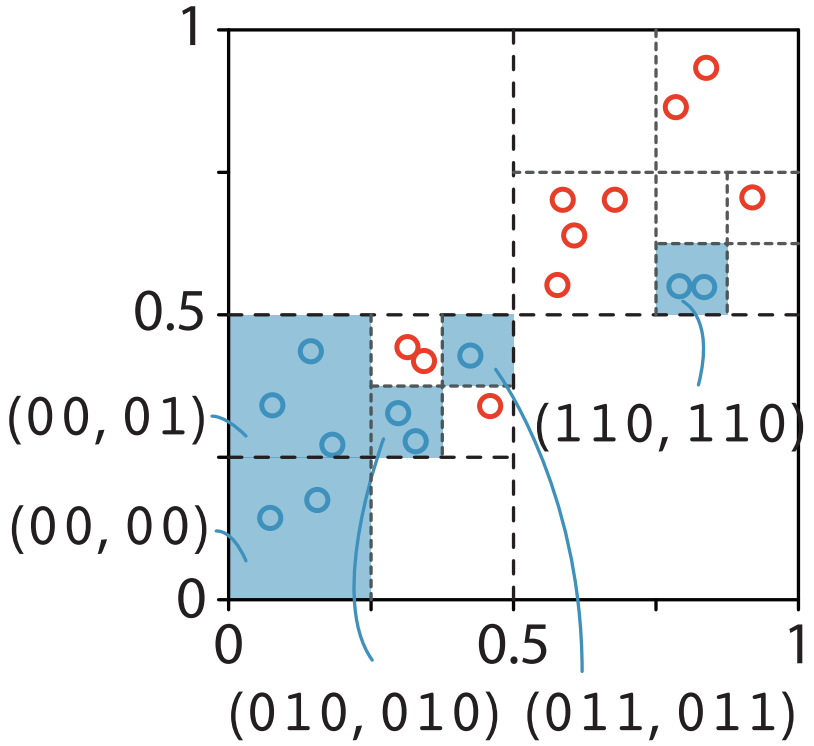
2/10 + 2/10 = 0.4

符号化ダイバージェンスの例

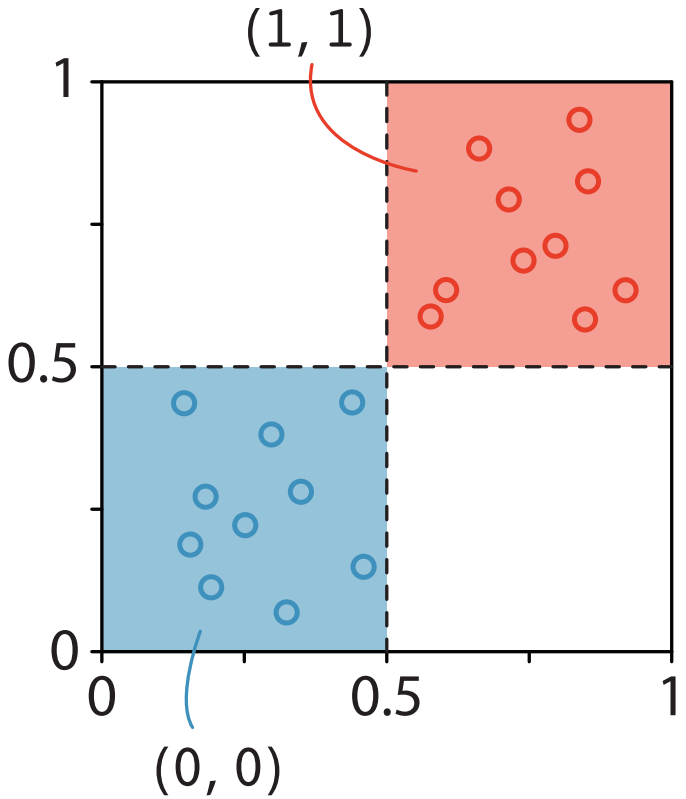


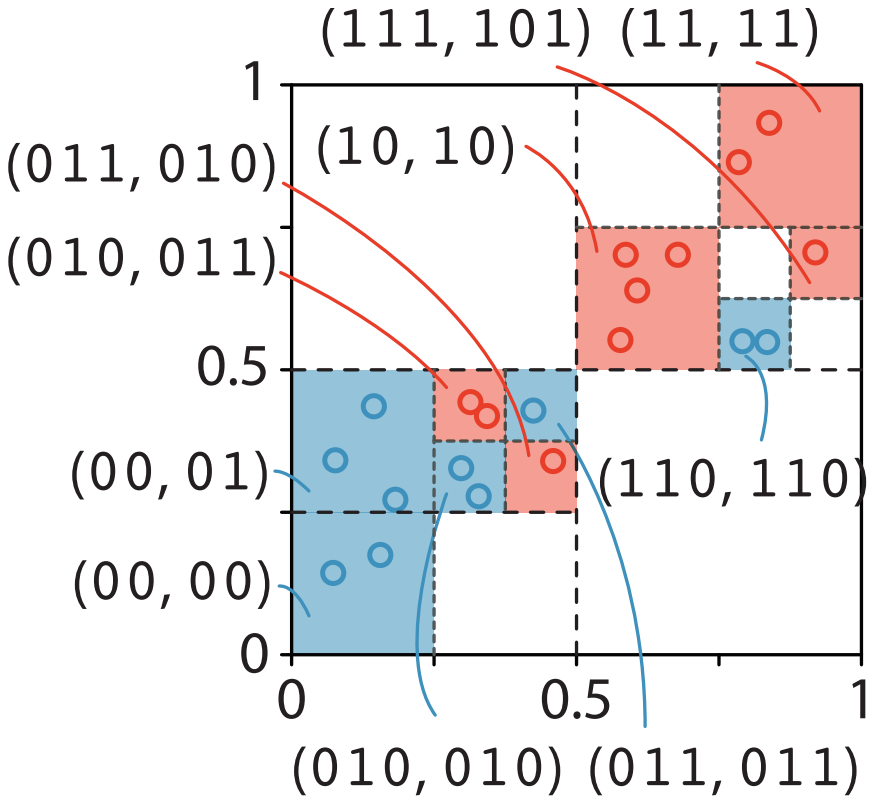
符号化ダイバージェンス：

2/10 + 2/10 = 0.4



符号化ダイバージェンスの例

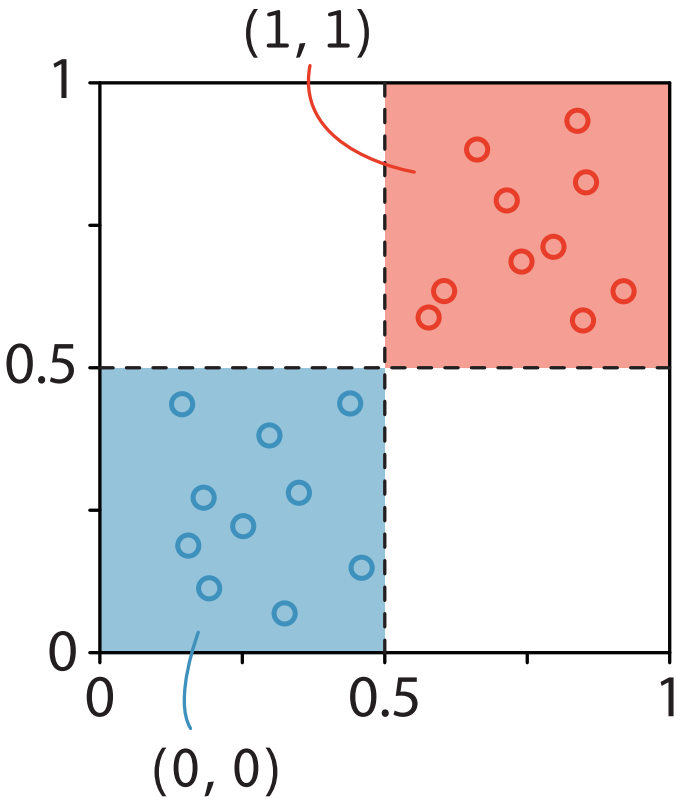




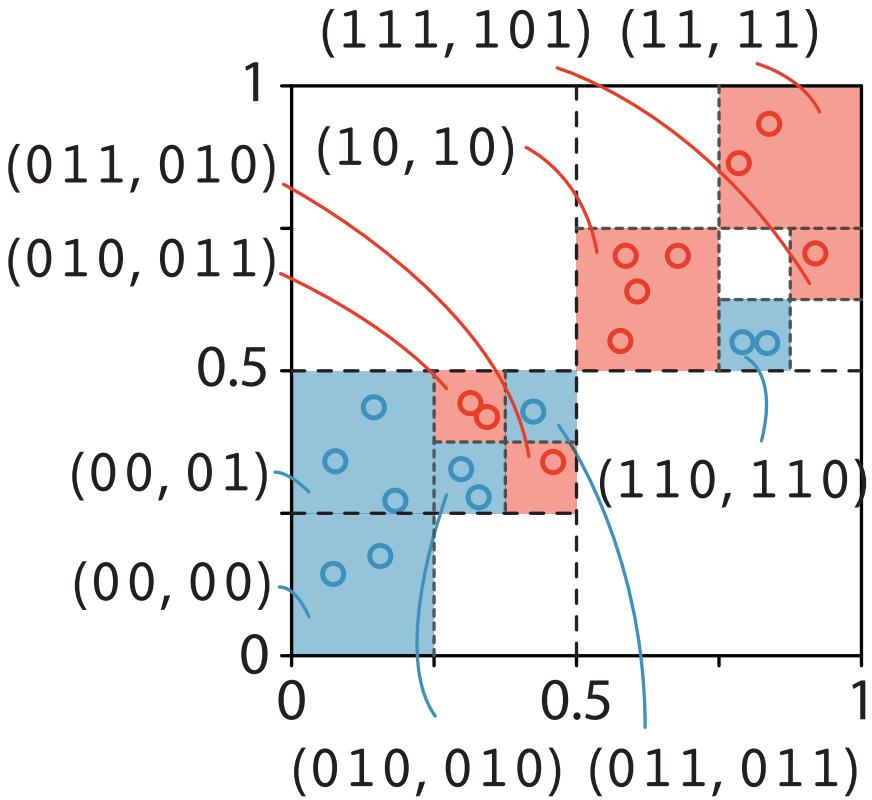
符号化ダイバージェンス：

$2/10 + 2/10 = 0.4$

符号化ダイバージェンスの例



符号化ダイバージェンス：
2/10 + 2/10 = 0.4



符号化ダイバージェンス：
26/10 + 26/10 = 5.2

符号化ダイバージェンス

- 空でない有限集合 $X, Y \subset \mathscr{I}$（$\mathscr{I}$ は単位区間）に対して，埋め込み $\varphi : \mathscr{I} \to \Sigma^\omega$ に関する符号化ダイバージェンスを以下のように定義

$$C_\varphi(X, Y) := \begin{cases} \infty & \text{if } X \cap Y \neq \emptyset, \\ D_\varphi(X; Y) + D_\varphi(Y; X) & \text{otherwise,} \end{cases}$$

  - ここで，D_φ は有向符号化ダイバージェンス

$$D_\varphi(X; Y) := \frac{1}{\#X} \min \left\{ |O| \,\middle|\, \begin{array}{l} O \text{ は開，かつ} \\ (\varphi(X), \varphi(Y)) \text{ に無矛盾} \end{array} \right\}$$

    - $\#X$ は X の要素数
    - $|O| := \sum_{w \in W} |w|$ （$O = \uparrow W$）
    - O は無矛盾 $\iff$ $O \supseteq \varphi(X)$ かつ $O \cap \varphi(Y) = \emptyset$

符号化ダイバージェンス

- 空でない有限集合 $X, Y \subset \mathcal{I}$（$\mathcal{I}$ は単位区間）に対して，埋め込み $\varphi : \mathcal{I} \to \Sigma^{\omega}$ に関する符号化ダイバージェンスを以下のように定義

$$C_{\varphi}(X, Y) := \begin{cases} \infty & \text{if } X \cap Y \neq \emptyset, \\ D_{\varphi}(X; Y) + D_{\varphi}(Y; X) & \text{otherwise,} \end{cases}$$

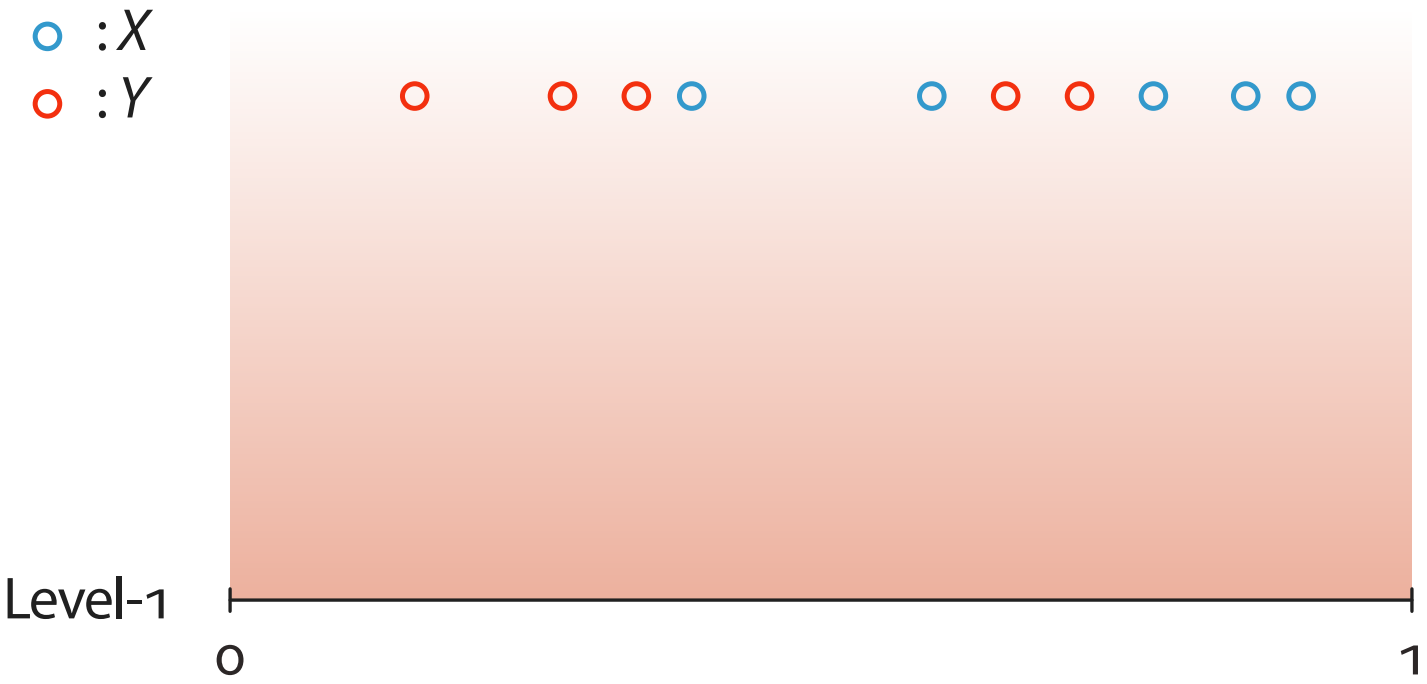
- 符号化ダイバージェンスはカントール空間の位相的構造のみに依存
  - 確率分布や統計的パラメータはまったく必要ない
    - 現在機械学習で主流の統計的アプローチと異なる

符号化ダイバージェンスの学習

$D_2(X; Y) \rightarrow \{$ $\}$ $C_2(X, Y) =$

$D_2(Y; X) \rightarrow \{$ $\}$

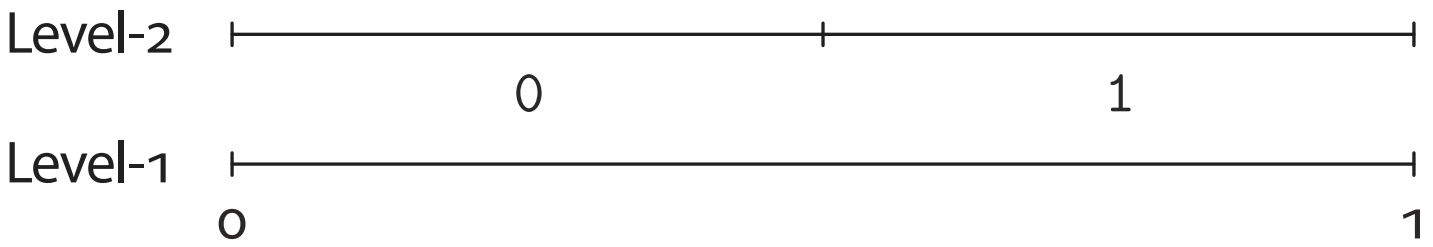
符号化ダイバージェンスの学習



$D_2(X; Y) \rightarrow \{$ $\}$ $C_2(X, Y) =$

$D_2(Y; X) \rightarrow \{$ $\}$

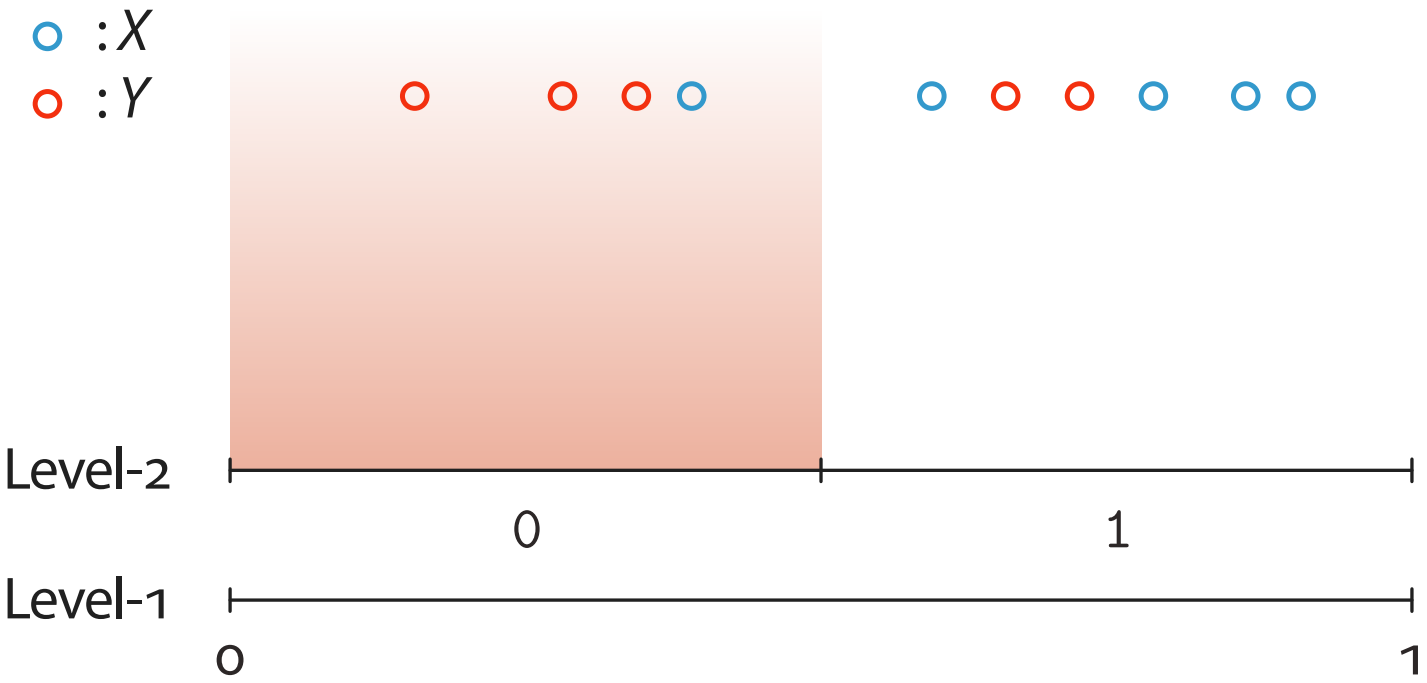
符号化ダイバージェンスの学習



$D_2(X; Y) \rightarrow \{$ $\}$ $C_2(X, Y) =$

$D_2(Y; X) \rightarrow \{$ $\}$

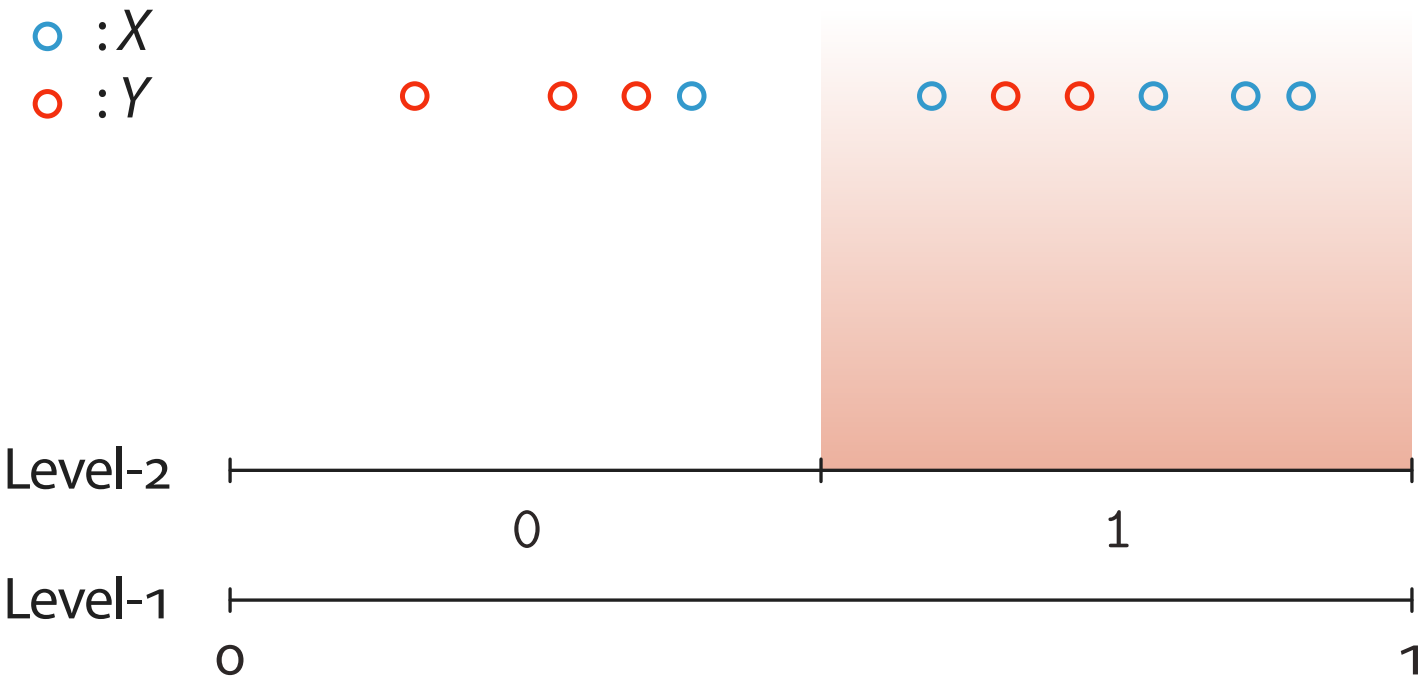
符号化ダイバージェンスの学習



$D_2(X; Y) \rightarrow \{$ $\}$ $C_2(X, Y) =$

$D_2(Y; X) \rightarrow \{$ $\}$

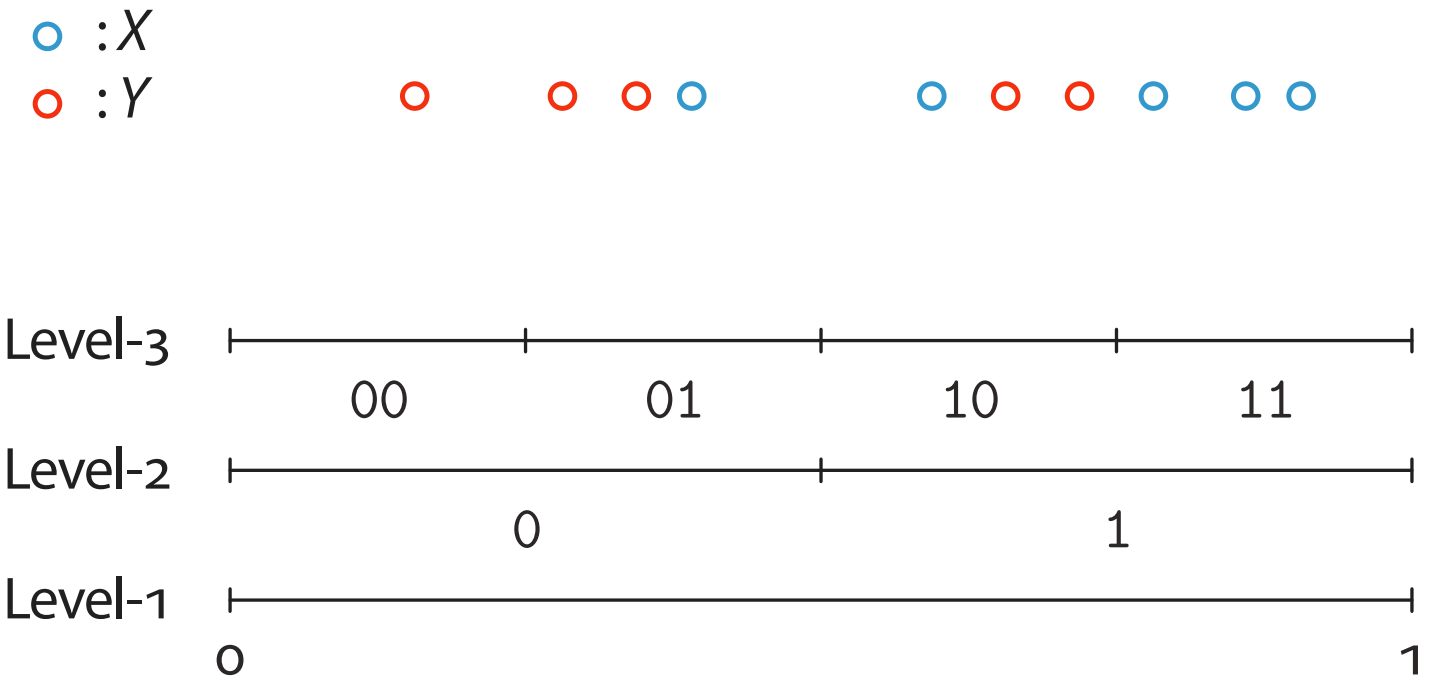
符号化ダイバージェンスの学習



$D_2(X; Y) \rightarrow \{$ $\}$ $C_2(X, Y) =$

$D_2(Y; X) \rightarrow \{$ $\}$

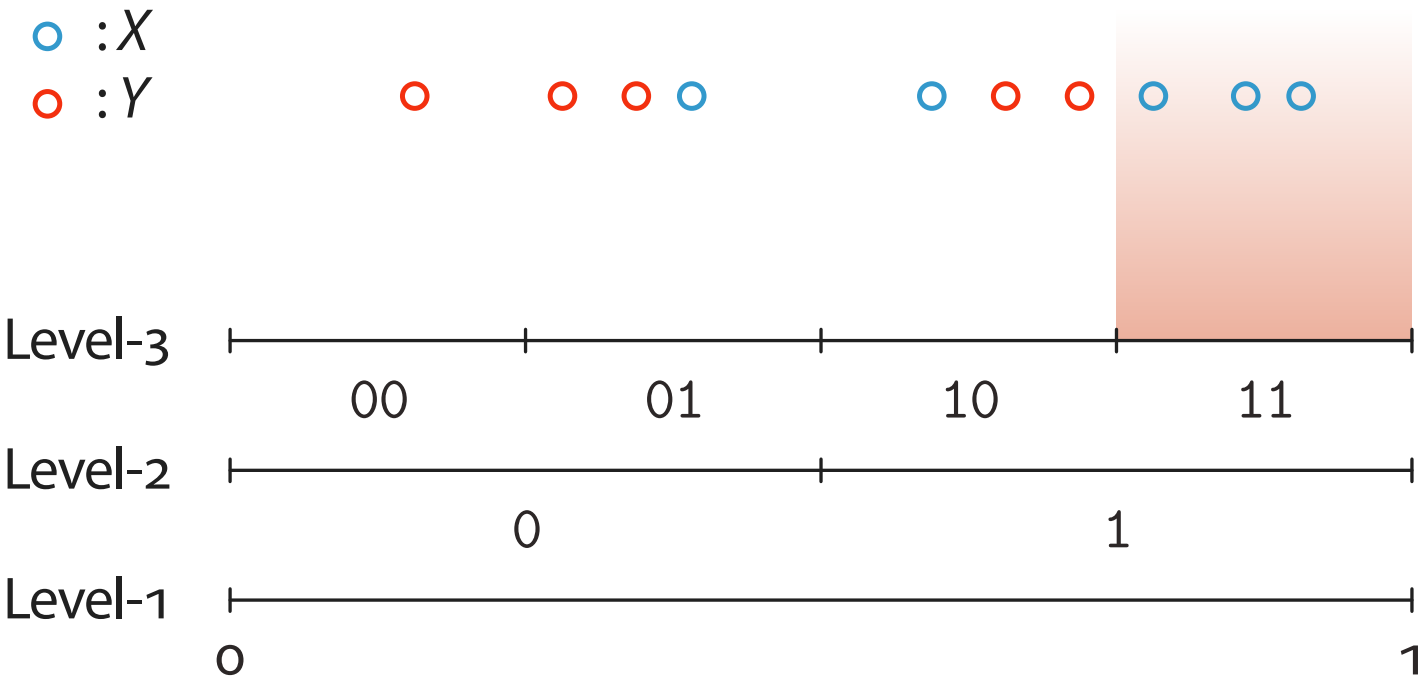
符号化ダイバージェンスの学習



$D_2(X; Y) \rightarrow \{$ $\}$ $C_2(X, Y) =$

$D_2(Y; X) \rightarrow \{$ $\}$

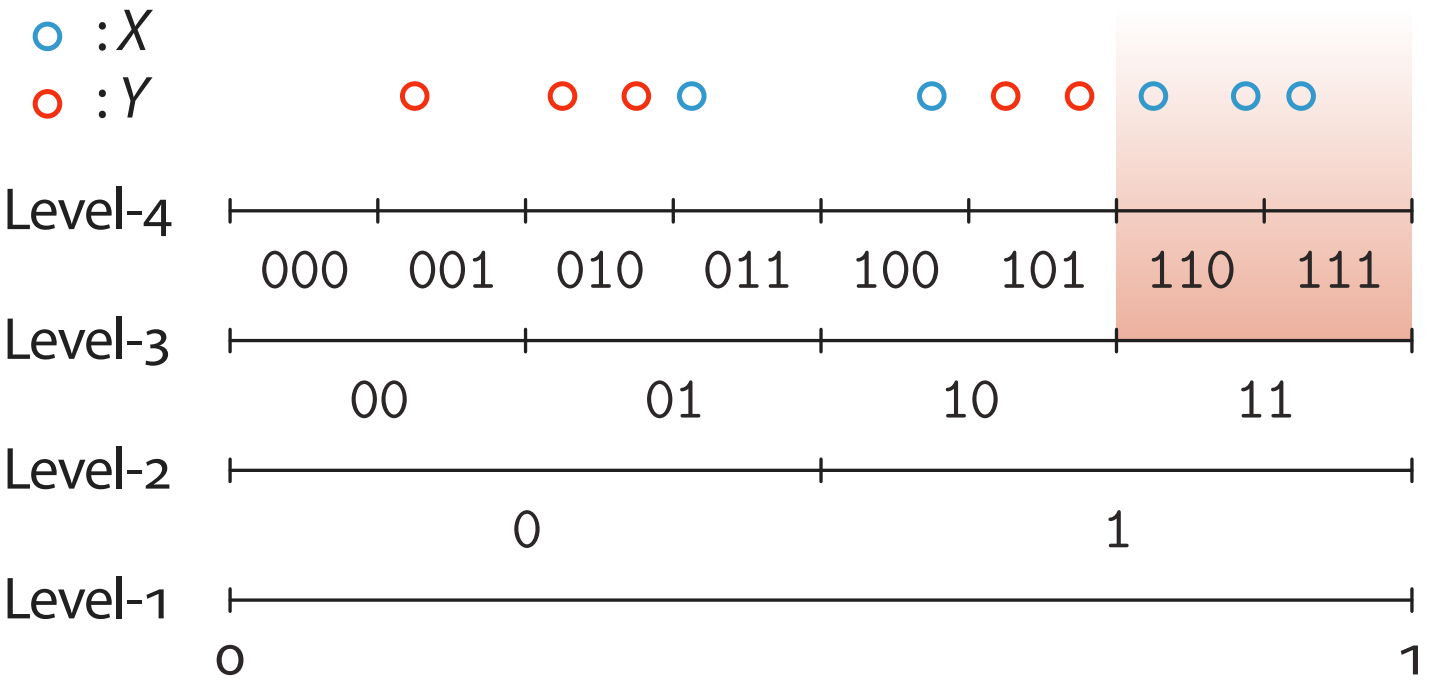
符号化ダイバージェンスの学習



$D_2(X; Y) \rightarrow \{11 \quad \}$ $C_2(X, Y) =$

$D_2(Y; X) \rightarrow \{ \quad \}$

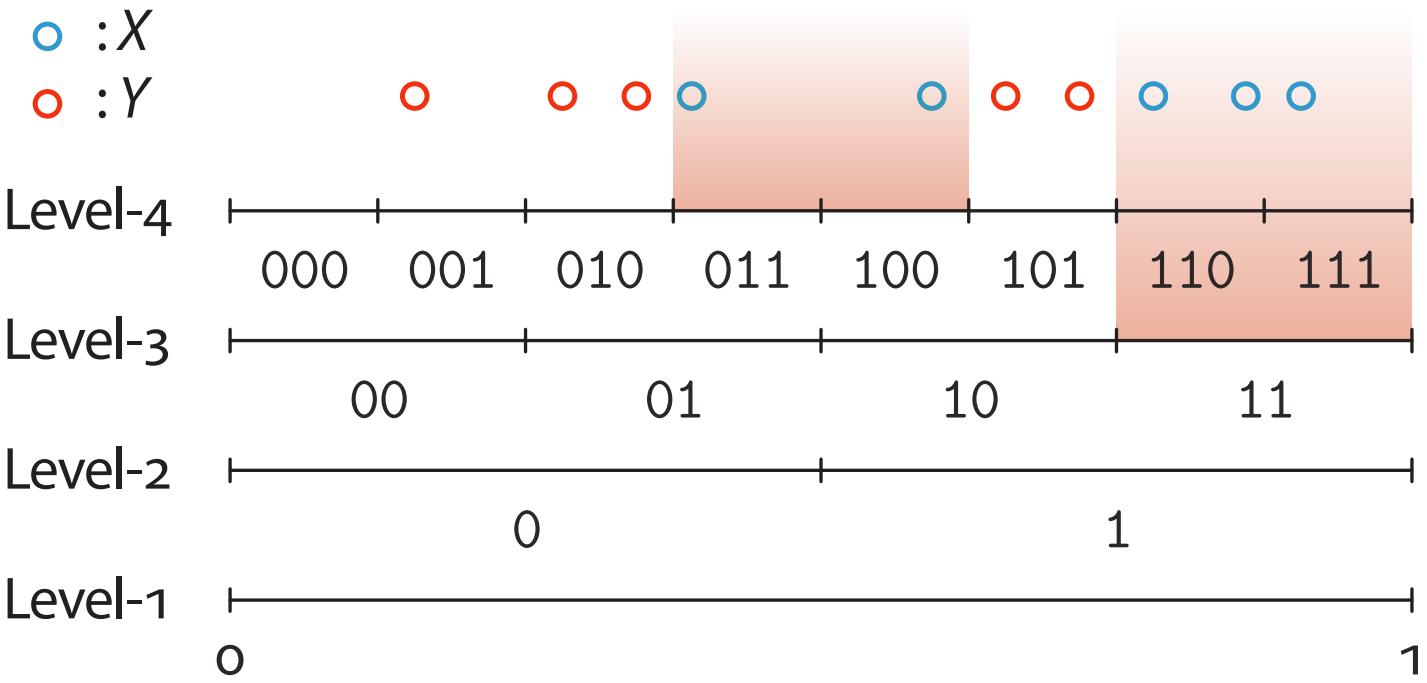
符号化ダイバージェンスの学習



$D_2(X; Y) \rightarrow \{11 \quad \}$ $C_2(X, Y) =$

$D_2(Y; X) \rightarrow \{ \quad \}$

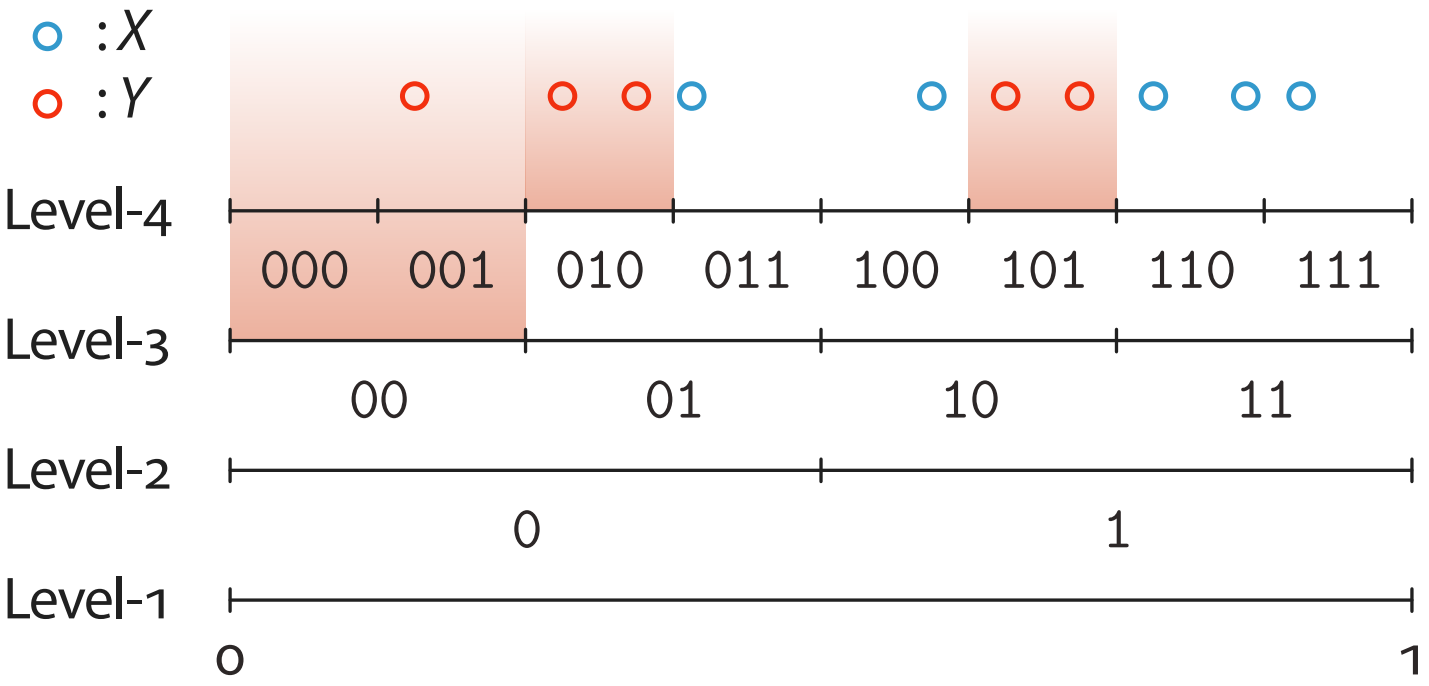
符号化ダイバージェンスの学習



$D_2(X; Y) \rightarrow \{11, 011, 100\}$ $C_2(X, Y) =$

$D_2(Y; X) \rightarrow \{$ $\}$

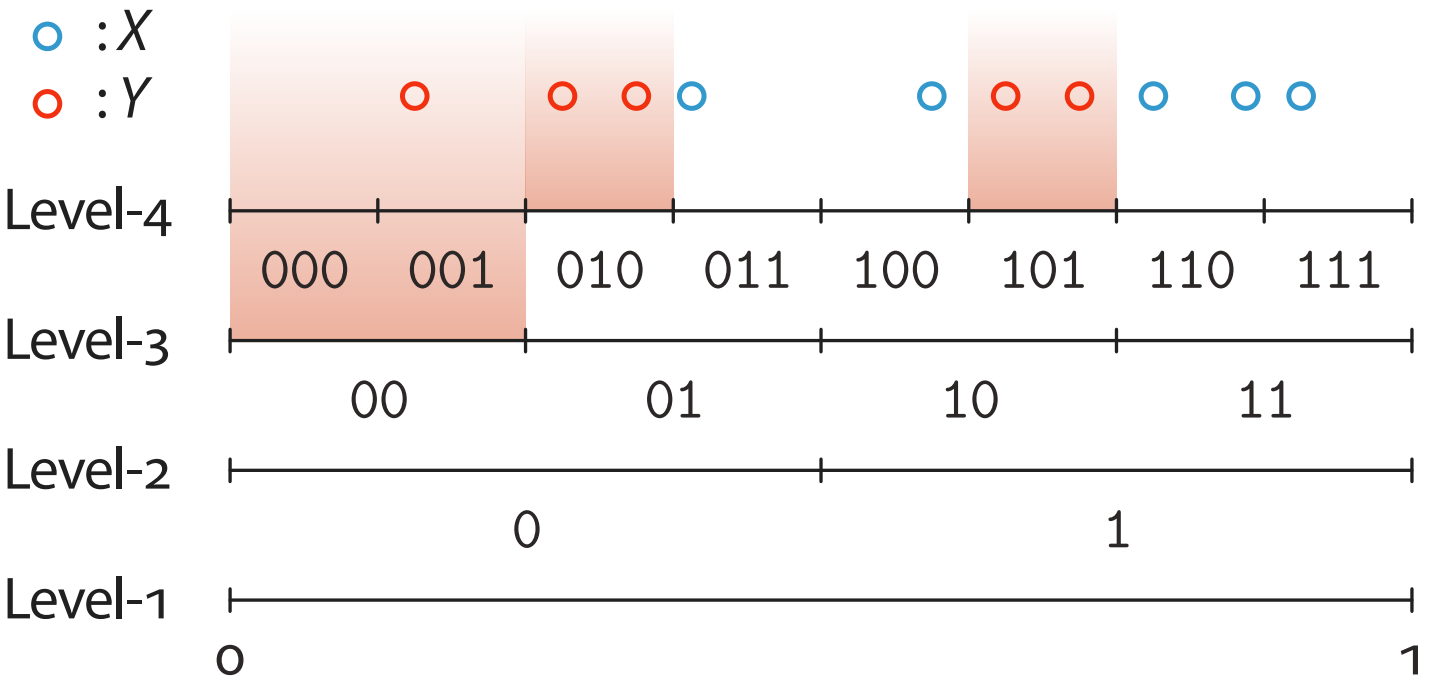
符号化ダイバージェンスの学習



$D_2(X; Y) \rightarrow \{11, 011, 100\}$ $C_2(X, Y) = 8/5 + 8/5 = 3.2$

$D_2(Y; X) \rightarrow \{00, 010, 101\}$

符号化ダイバージェンスの学習



$D_2(X; Y) \rightarrow \{11, 011, 100\}$

$D_2(Y; X) \rightarrow \{00, 010, 101\}$

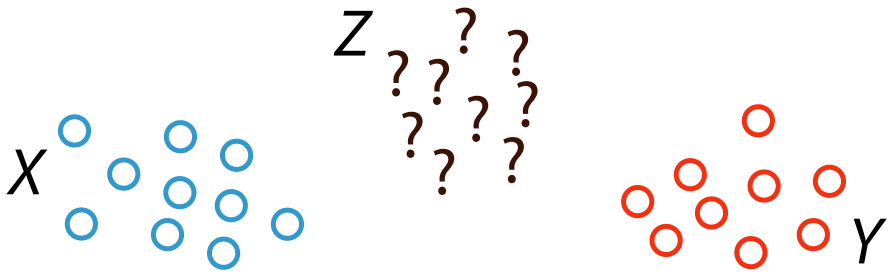
$C_2(X, Y) = 8/5 + 8/5 = 3.2$

The computational complexity:
$O(mn)$ ($m = \|X\|, n = \|Y\|$)

符号化ダイバージェンスを用いた分類

- 怠惰学習でクラス分類をおこなう
- この分類器は，訓練データ X と Y（それぞれラベルは A と B とする）を受け取り，テストデータ Z を A か B へ分類する
  - 仮定：Z 中のデータのラベルは全て同じ

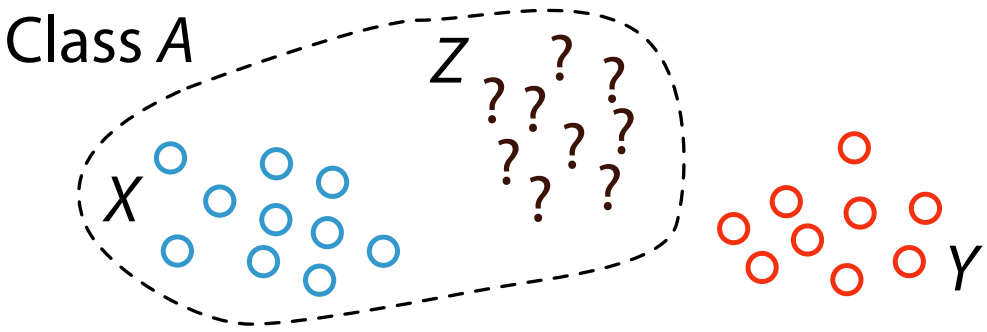
$$Z \text{ は } \begin{cases} A \text{ に属する} & \text{if } C_\varphi(X, Z) > C_\varphi(Y, Z), \\ B \text{ に属する} & \text{otherwise.} \end{cases}$$



符号化ダイバージェンスを用いた分類

- 怠惰学習でクラス分類をおこなう
- この分類器は，訓練データ X と Y（それぞれラベルは A と B とする）を受け取り，テストデータ Z を A か B へ分類する
  - 仮定：Z 中のデータのラベルは全て同じ

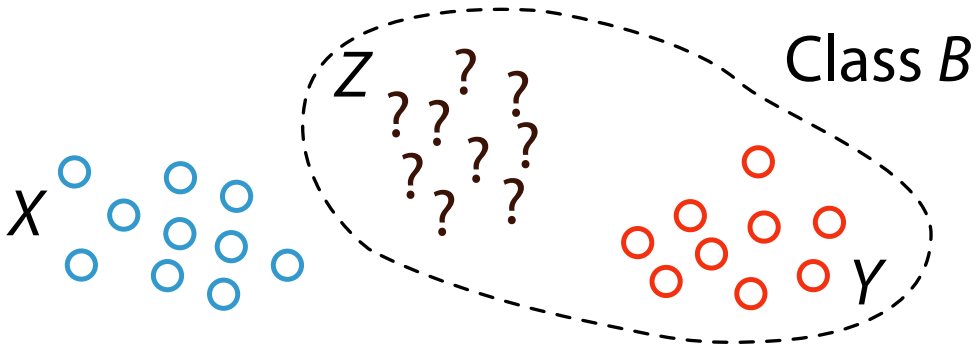
$$Z \text{ は } \begin{cases} A \text{ に属する} & \text{if } C_\varphi(X, Z) > C_\varphi(Y, Z), \\ B \text{ に属する} & \text{otherwise.} \end{cases}$$



符号化ダイバージェンスを用いた分類

- 怠惰学習でクラス分類をおこなう
- この分類器は，訓練データ X と Y（それぞれラベルは A と B とする）を受け取り，テストデータ Z を A か B へ分類する
  - 仮定：Z 中のデータのラベルは全て同じ

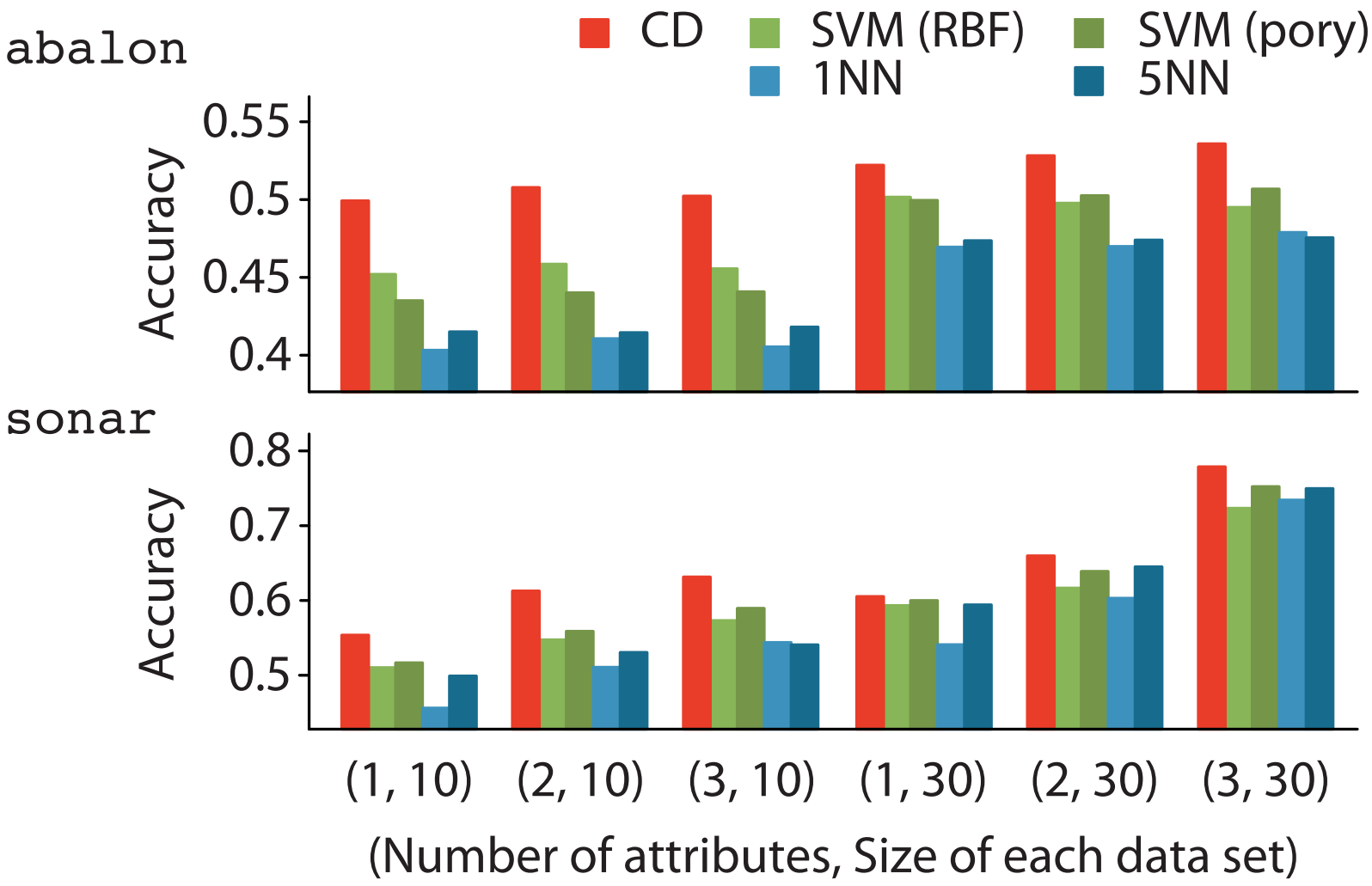
$$Z \text{ は } \begin{cases} A \text{ に属する} & \text{if } C_\varphi(X, Z) > C_\varphi(Y, Z), \\ B \text{ に属する} & \text{otherwise.} \end{cases}$$



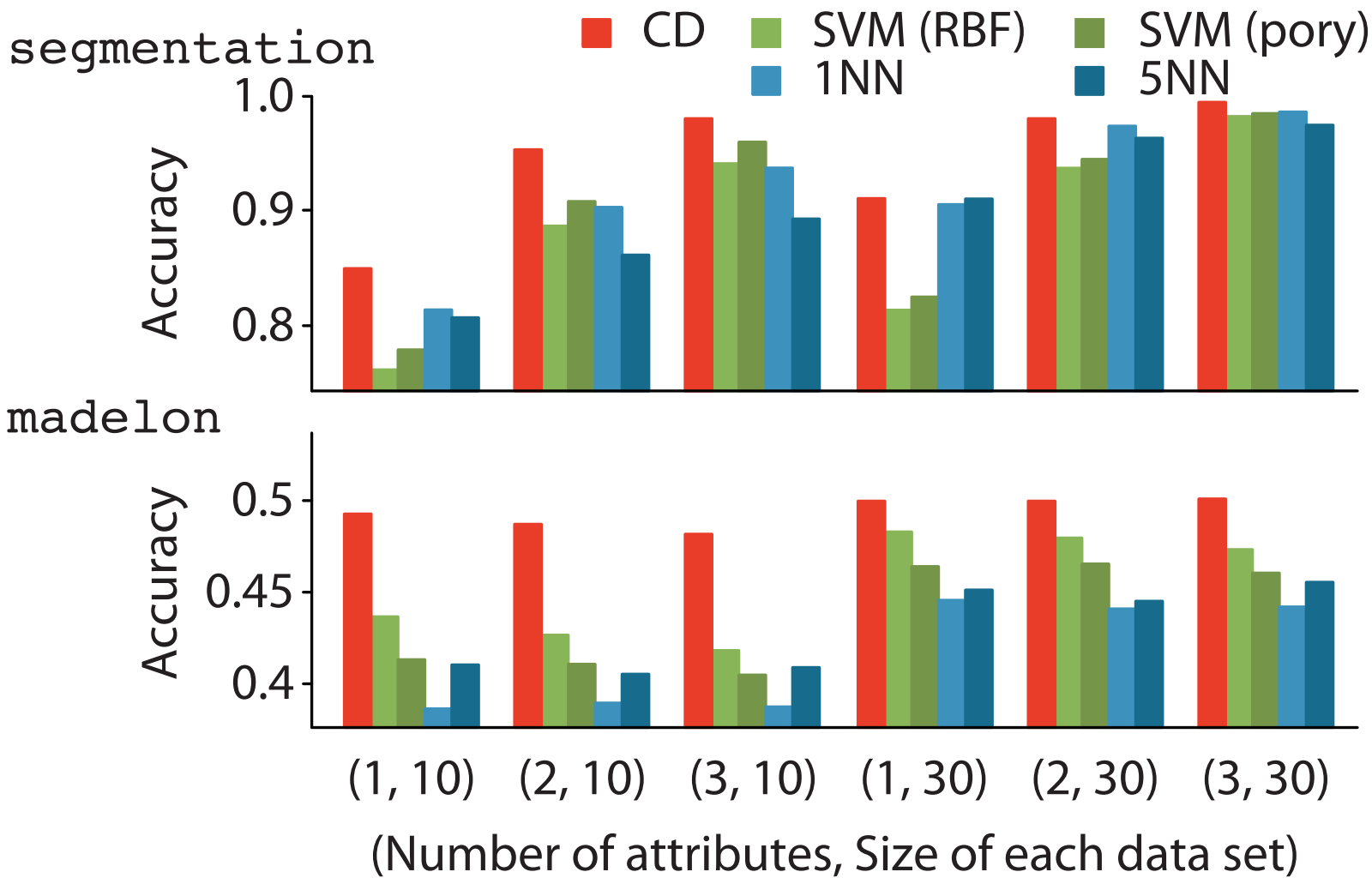
実験手法

- R 言語 (2.10.1) で実装
- UCI のデータセット（`abalon`, `sonar`, ...）を用いる
- 以下を 10,000 回繰り返して，sensitivity と specificity から accuracy を求める
  - 識別に用いる属性をランダムに決める
  - 双方のラベルからそれぞれランダムに（非復元で）n 個サンプリングを 2 回（X, T_+ と Y, T_-）
    - X, Y は訓練データ，T_+, T_- はテストデータ
  - データを正規化（min-max normalization）
  - 符号化ダイバージェンス（と他の手法）を用いて T_+ が X と Y のどちらに近いかを判定して分類，T_- も同様に分類
- 得られた真陽性の数を t_{pos}，真偽性の数を t_{neg} として，$(t_{\mathrm{pos}} + t_{\mathrm{neg}})/20000$ で accuracy を求める

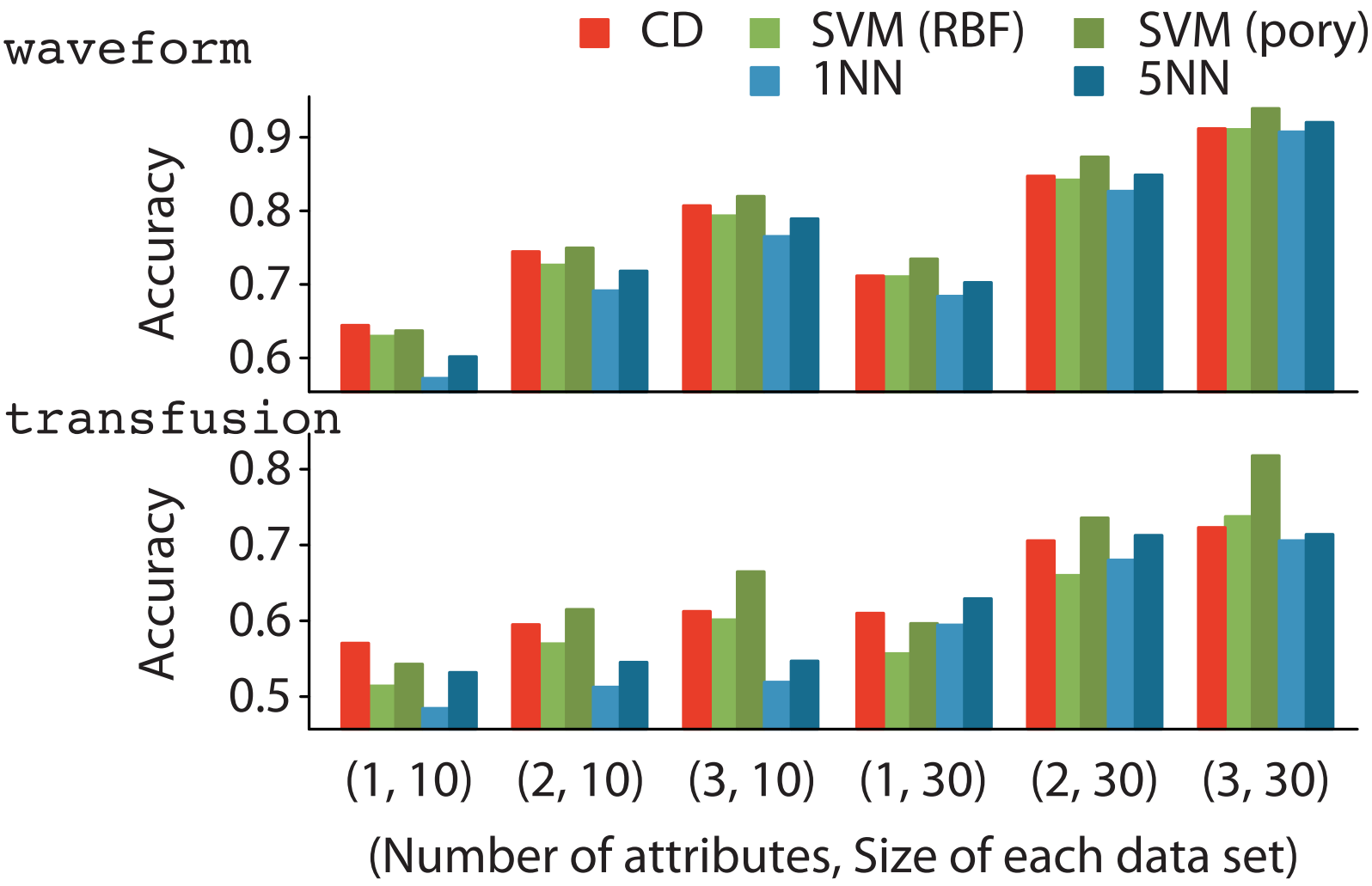
実験結果


abalon
CD
SVM (RBF)
SVM (pory)
1NN
5NN
Accuracy
0.55
0.5
0.45
0.4
sonar
Accuracy
0.8
0.7
0.6
0.5
(1, 10)
(2, 10)
(3, 10)
(1, 30)
(2, 30)
(3, 30)
(Number of attributes, Size of each data set)

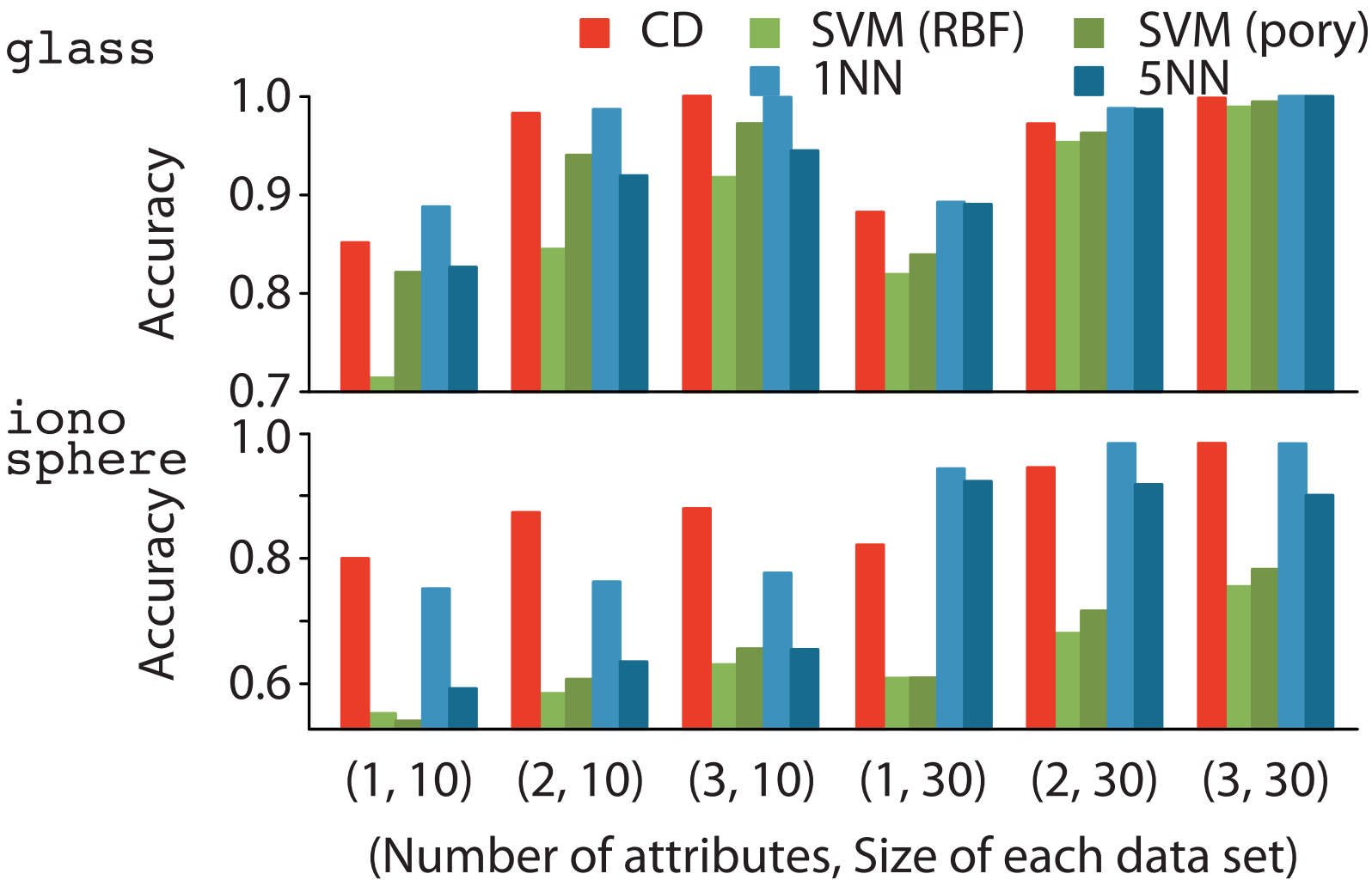
実験結果


segmentation
CD
SVM (RBF)
SVM (pory)
1NN
5NN
Accuracy
1.0
0.9
0.8
madelon
Accuracy
0.5
0.45
0.4
(1, 10)
(2, 10)
(3, 10)
(1, 30)
(2, 30)
(3, 30)
(Number of attributes, Size of each data set)

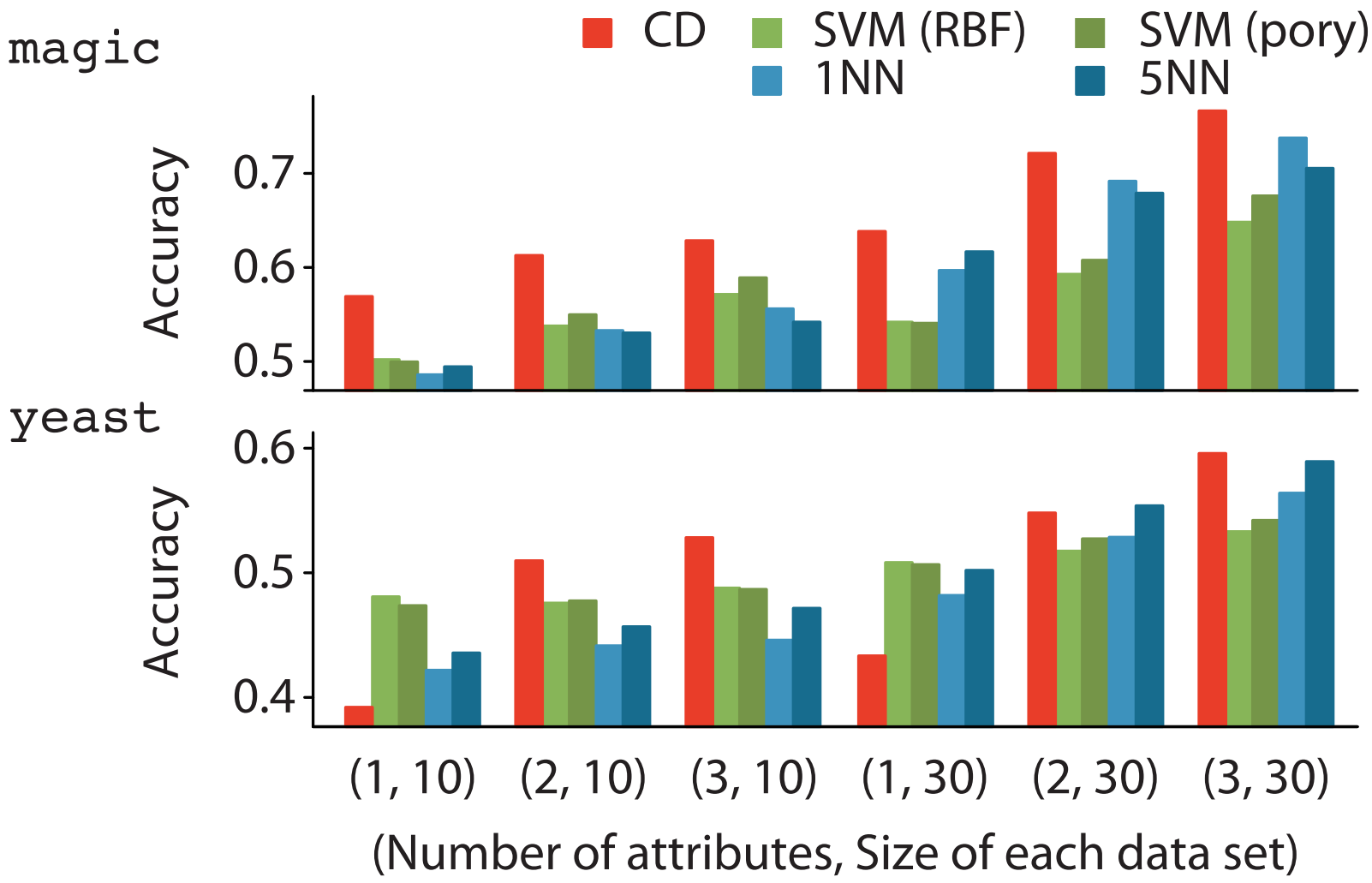
実験結果


waveform
CD
SVM (RBF)
SVM (pory)
1NN
5NN
Accuracy
0.9
0.8
0.7
0.6
transfusion
Accuracy
0.8
0.7
0.6
0.5
(1, 10)
(2, 10)
(3, 10)
(1, 30)
(2, 30)
(3, 30)
(Number of attributes, Size of each data set)

実験結果


glass
CD
SVM (RBF)
SVM (pory)
1NN
5NN
Accuracy
1.0
0.9
0.8
0.7
iono
sphere
Accuracy
1.0
0.8
0.6
(1, 10)
(2, 10)
(3, 10)
(1, 30)
(2, 30)
(3, 30)
(Number of attributes, Size of each data set)

実験結果


magic
CD
SVM (RBF)
SVM (pory)
1NN
5NN
Accuracy
0.7
0.6
0.5
yeast
Accuracy
0.6
0.5
0.4
(1, 10)
(2, 10)
(3, 10)
(1, 30)
(2, 30)
(3, 30)
(Number of attributes, Size of each data set)

まとめ

- 符号化ダイバージェンスという実数値データ集合間の類似度を測る新規の尺度を提案
  - ユークリッド空間 $\mathbb{R}^d$ 上の実数値データをカントール空間 Σ^ω へ埋め込む（離散化）
  - 最も単純，かつ無矛盾なモデル（カントール空間の開集合）を学習する
  - モデルを表現するコードの長さで定量化
- 怠惰学習をおこなう分類器を構築
  - クラス分類の性能を実データを用いた実験で検証
  - SVM や k 近傍法に比べて遜色ない性能を持つことを確認

目次

- 第一部：理論
  - フラクタルとハウスドルフ距離を用いた図形の学習
    — 計算可能な 2 値分類を目指して
- 第二部：理論から実践
  - 符号化ダイバージェンス
  - MCL とグレイコードを用いたクラスタリング
  - 2 進符号化を活用した高速クラスタリング
- 第三部：形式概念と共に
  - 離散値・連続値混在データからの半教師あり学習
  - 半教師ありのマルチラベルクラス分類によるリガンド発見

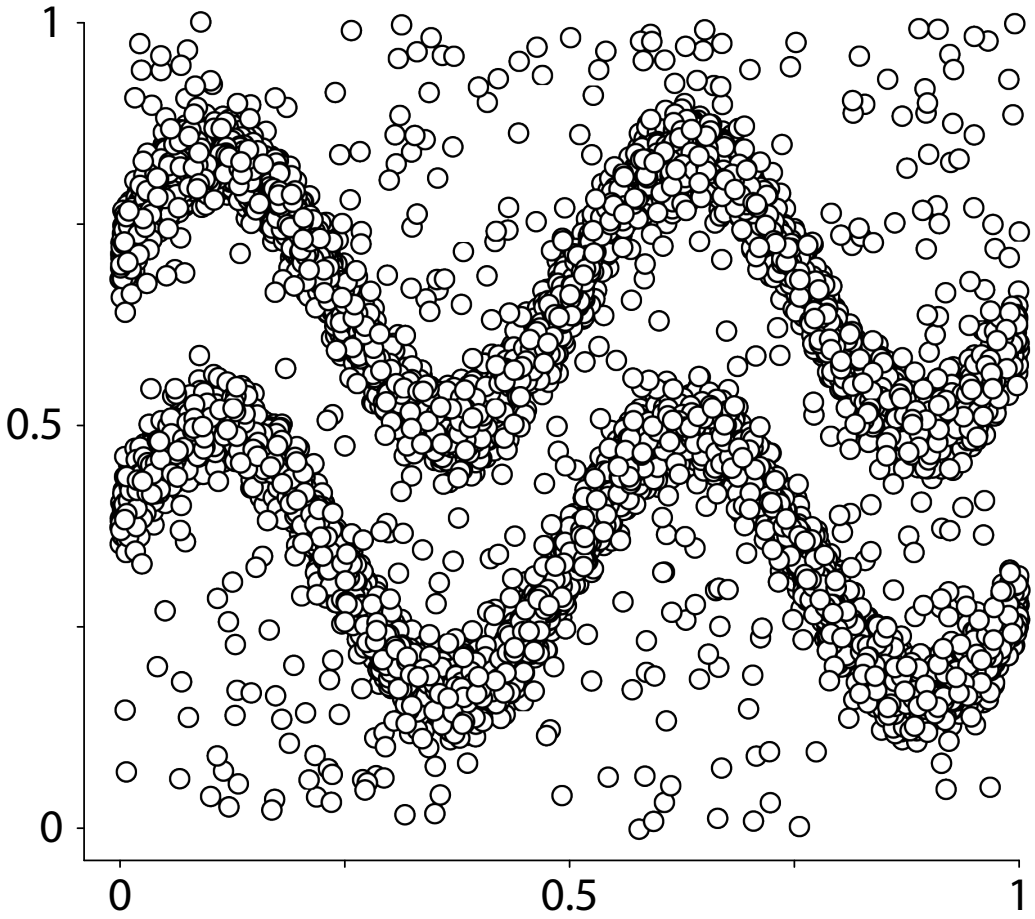
MCL とグレイコードを用いたクラスタリング

- Sugiyama, M., Yamamoto, A.:
  The Minimum Code Length for Clustering Using the Gray Code,
  ECML PKDD 2011, *LNAI* 6913

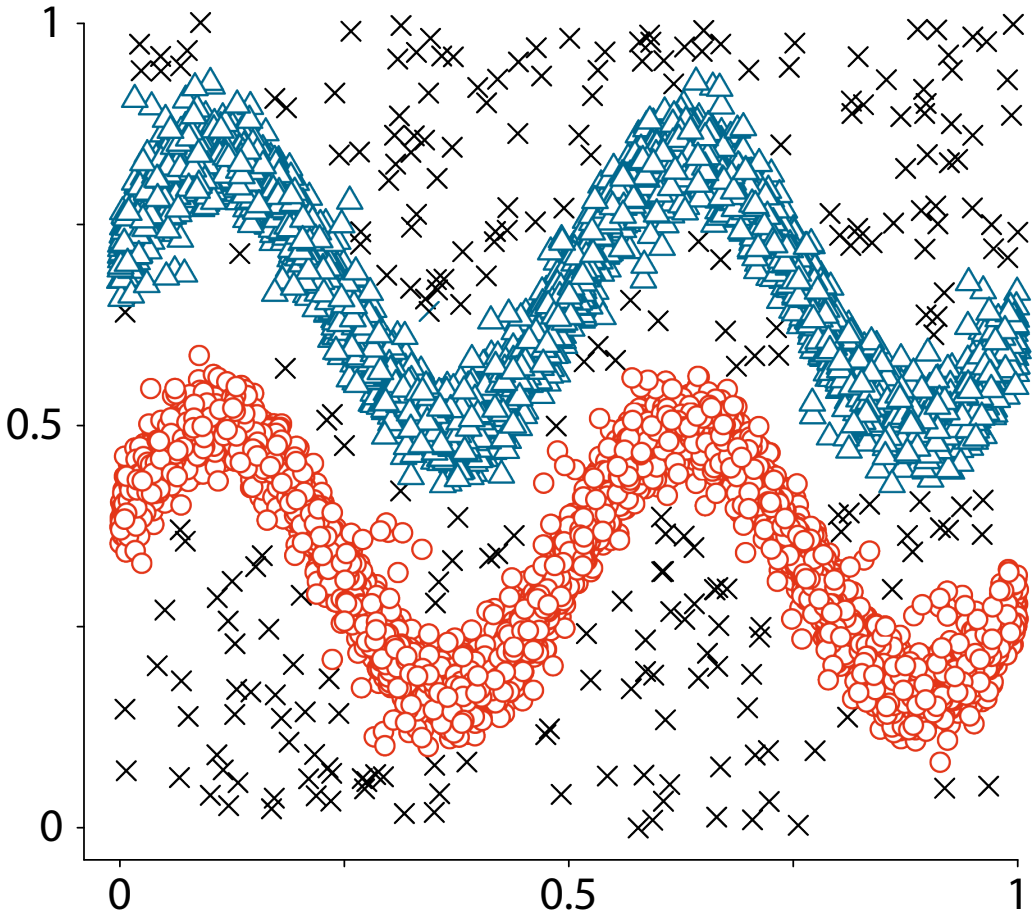
概要・結果

1. ***The MCL (Minimum Code Length)***
   - クラスタリングの結果を評価する新たな評価基準
   - 実数値データに対する固定された符号化方式のもとで
     各クラスタを分離するために必要となるコード長
     - 符号化ダイバージェンスと同じ発想，その発展
2. ***COOL (COding-Oriented cLustering)***
   - クラスタリング手法
   - 大域的最適解（MCL を最小化する）を$O(nd)$で必ず出力
   - パラメータの調整はほとんど必要ない
3. ***G-COOL (COOL with the Gray code)***
   - 内的結合と外的分離を達成
   - 任意形状のクラスタを発見可能

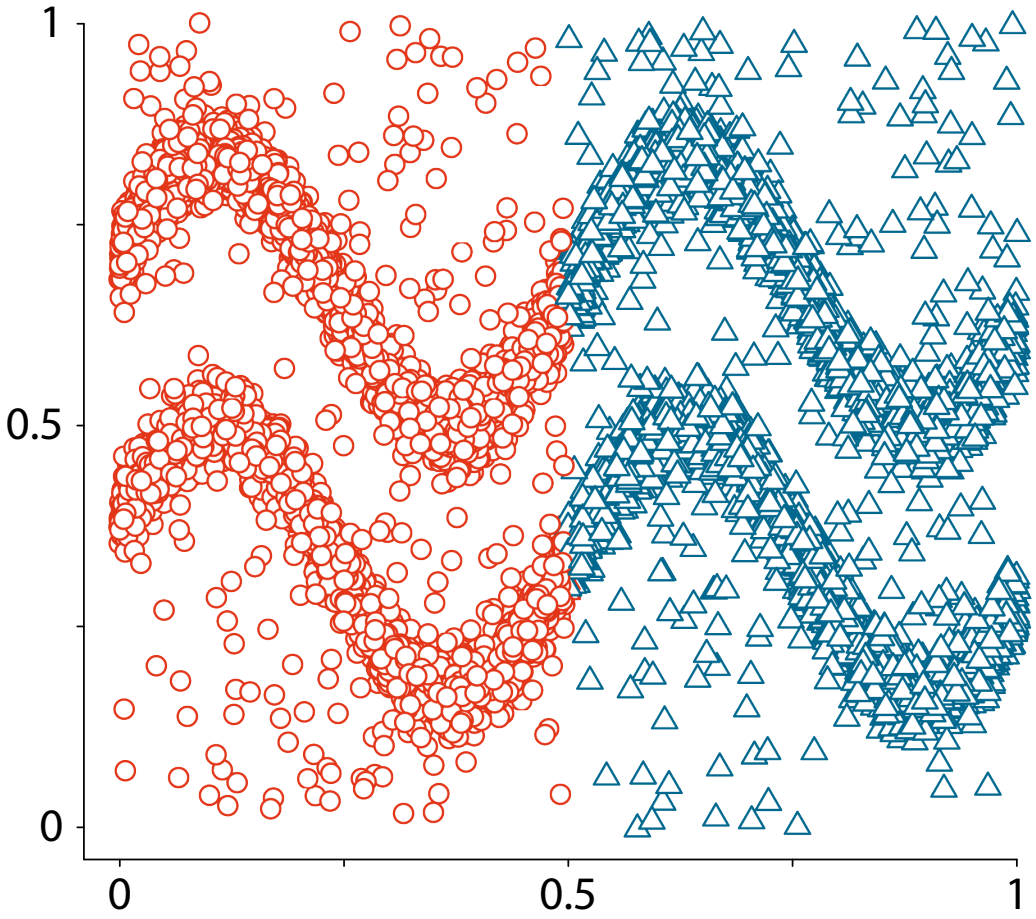
合成データへの適用例



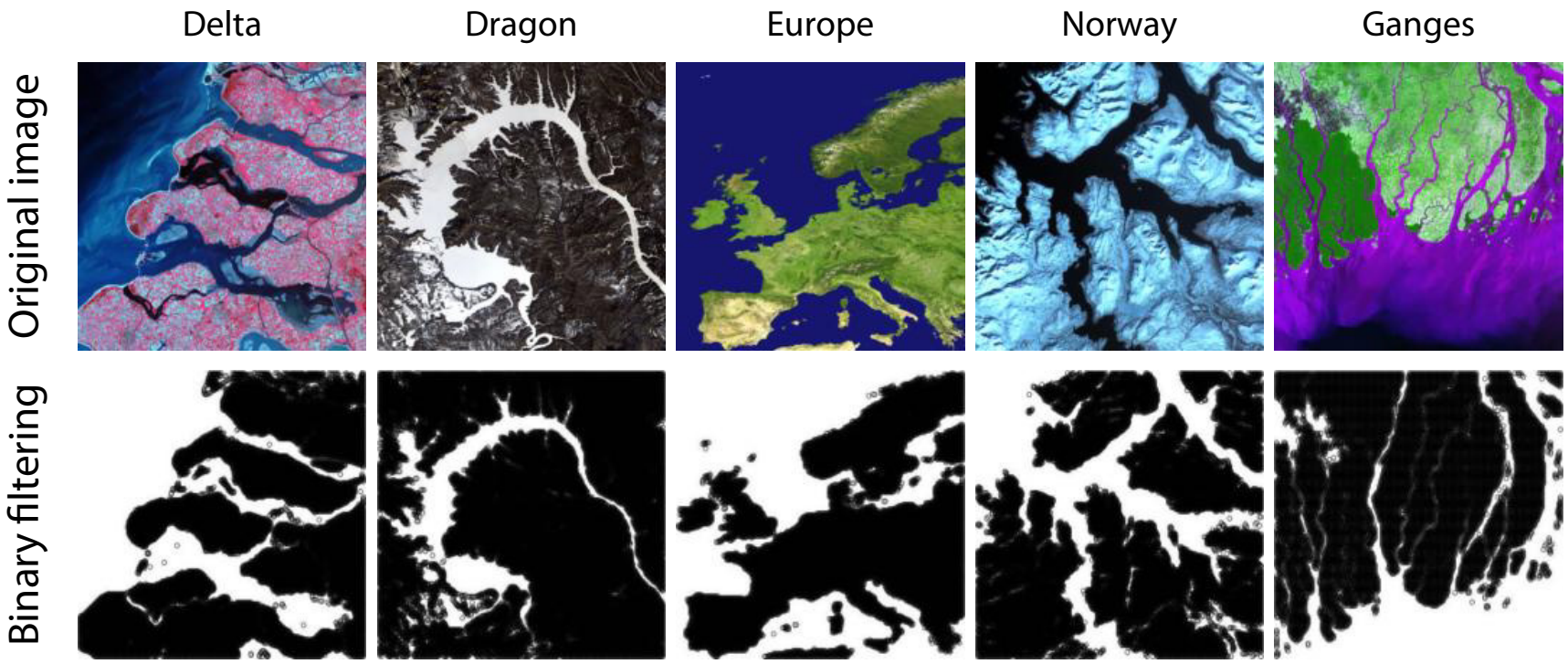
G-COOL



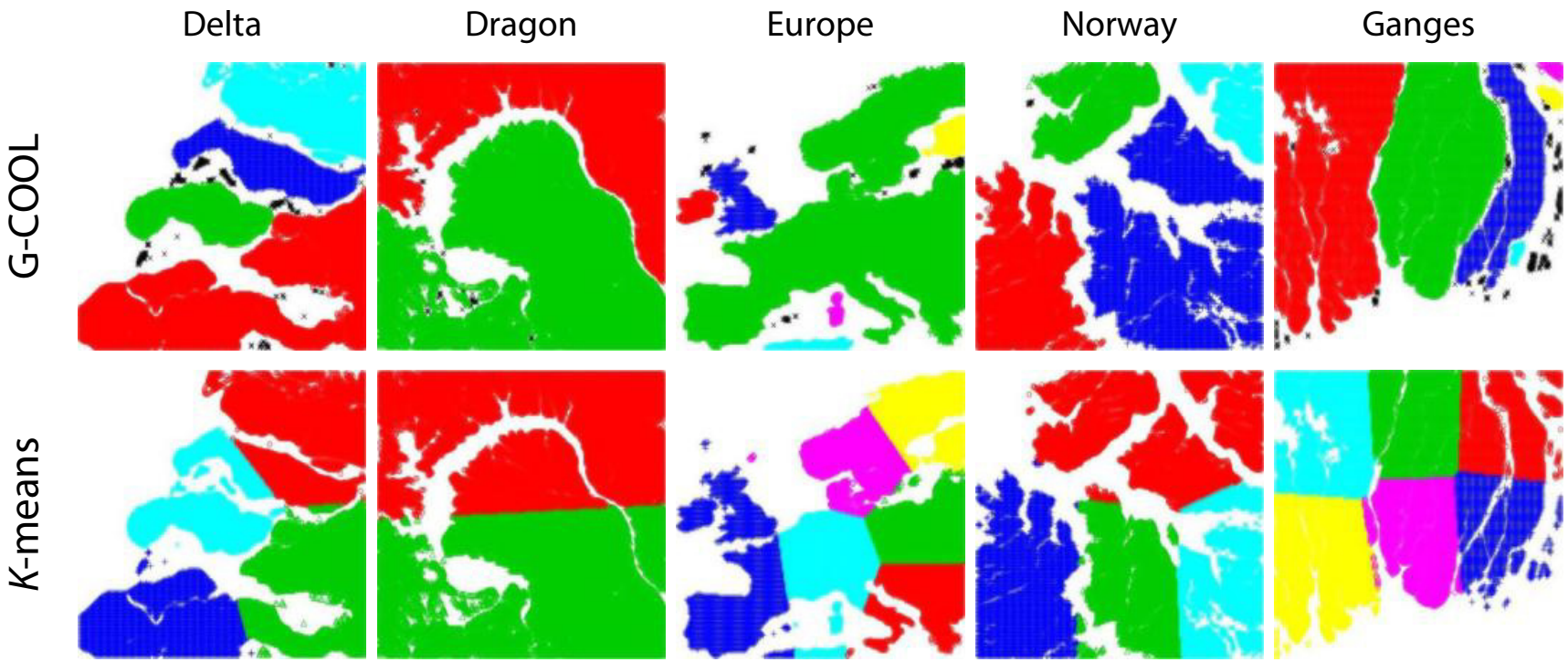
K-means



実データでの結果


Delta
Dragon
Europe
Norway
Ganges
Original image
Binary filtering

実データでの結果



背景：圧縮に着目したクラスタリング

- MDL を用いたアプローチ [Kontkanen *et al.*, 2005]
  - データの符号化を最適化する
    - 全ての符号化方式を（暗に）扱う
    - 計算量は $O(n^2)$ 以上
- コルモゴロフ複雑性を用いたアプローチ [Cilibrasi, 2005]
  - データ間の距離を有限列の圧縮率に基づいて測る
    - 実数値データへの適用は困難
  - 実際のクラスタリングは普通の凝集型階層クラスタリング
    - 計算量は $O(n^2)$ 以上
- これらのアプローチはともに，大規模データへの適用には向いていない

背景：既存手法の問題点

- *K*-means アルゴリズムは広く使われている
  - 単純かつ効果的
  - 主な欠点は，クラスタリング能力：
    超球状でない形状のクラスタは発見できない
- 多くの形状に基づくアルゴリズムが提案されている
  - 任意形状のクラスタが発見可能
  - 欠点：
    1. スケーラブルでない
       （計算量は 2 乗から 3 乗のオーダー）
    2. 入力パラメータに対して頑健でない

アプローチ

- 新たなクラスタリング手法への要求：
  1. 高速，特にデータサイズに対して線形時間
  2. 入力パラメータに対して頑健
  3. 任意形状のクラスタを発見可能

アプローチ

- 新たなクラスタリング手法への要求：
  1. 高速，特にデータサイズに対して線形時間
  2. 入力パラメータに対して頑健
  3. 任意形状のクラスタを発見可能
- 解決策：
  1. 実数値データの符号化方式を固定する
     - 計算可能性解析からのアイデア［Weihrauch, 2000］
  2. クラスタリングをデータの離散化と同一視する
     - MCL に関して必ず最良の解を出力する
  3. 実数のグレイコードを用いる［Tsuiki, 2002］
     - 離散化されたデータが重なりあうため，隣接するクラスタが併合される

MCL (Minimum Code Length)

- MCL は固定された符号化方式で
  最大限圧縮されたクラスタのコード長
- MCL の計算量は $O(nd)$（基数ソートを使う）
  - n と d はそれぞれデータ数と次元数

MCL (Minimum Code Length)

- MCL は固定された符号化方式で
  最大限圧縮されたクラスタのコード長
- MCL の計算量は $O(nd)$（基数ソートを使う）
  - n と d はそれぞれデータ数と次元数

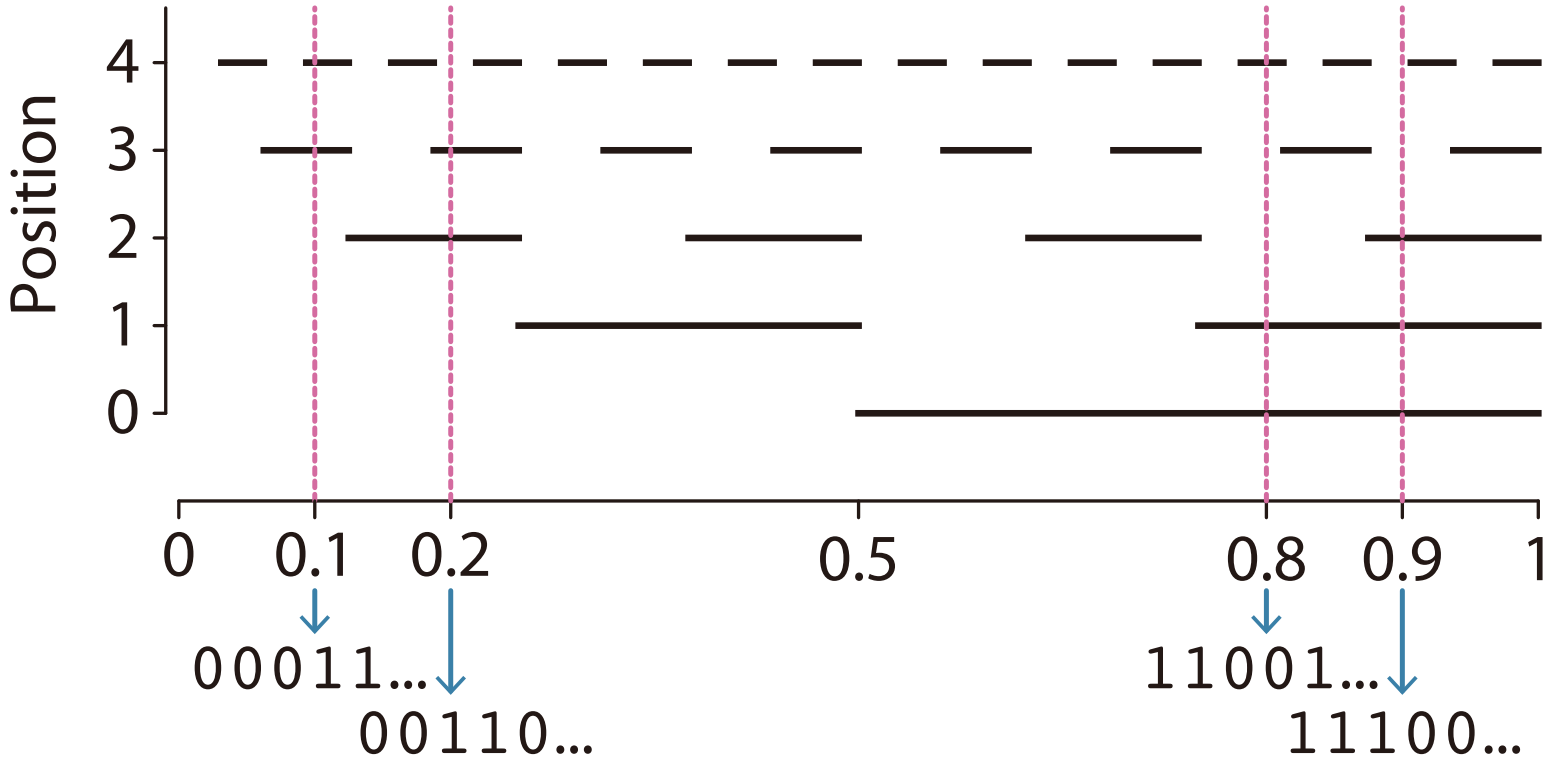
例: $X = \{0.1, 0.2, 0.8, 0.9\}$,

$$\mathscr{C}_1 = \{\{0.1, 0.2\}, \{0.8, 0.9\}\}$$

$$\mathscr{C}_2 = \{\{0.1\}, \{0.2, 0.8\}, \{0.9\}\}$$

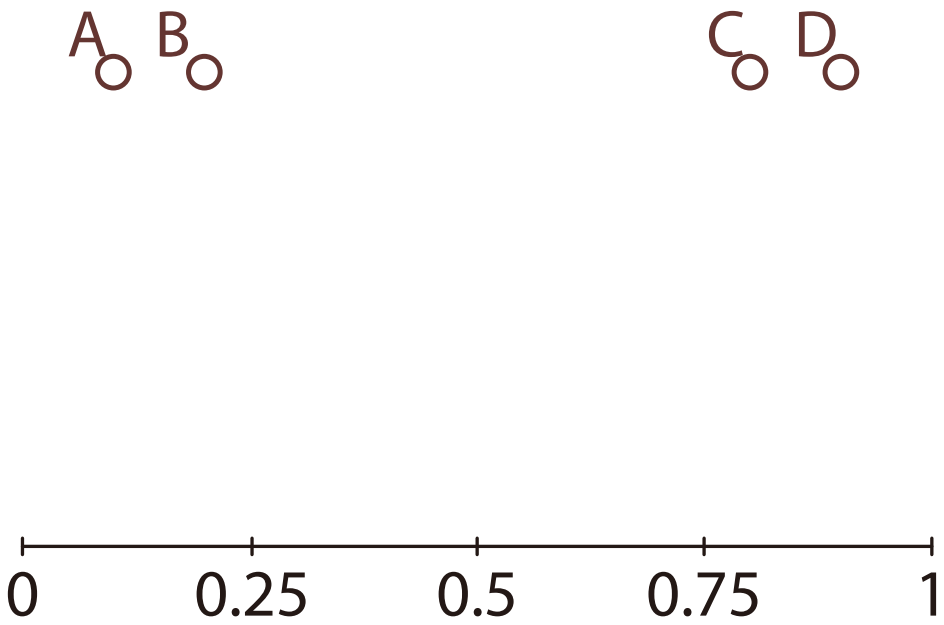
- 2 進符号化を使う
- どちらが良い？

2 進符号化



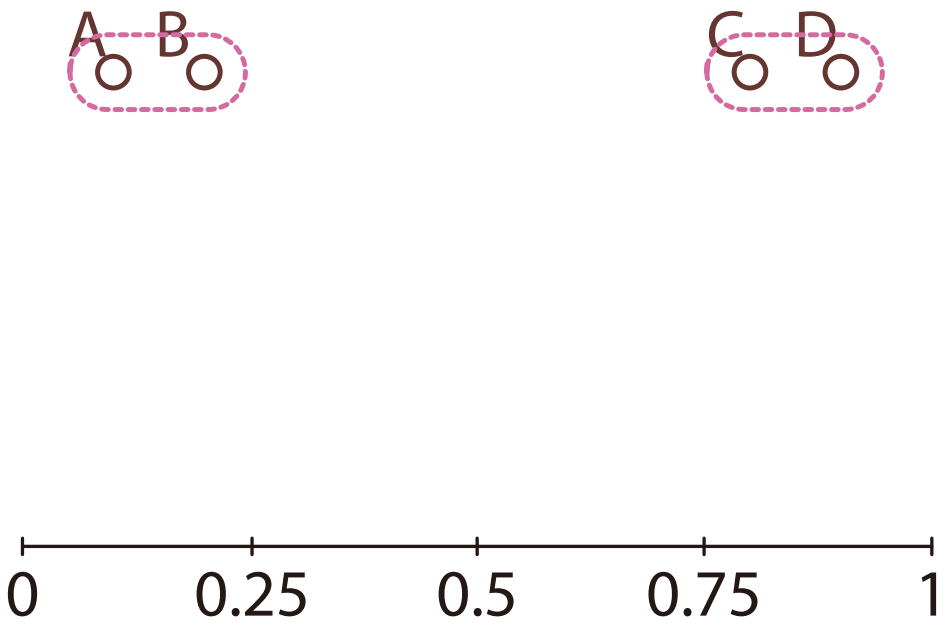
2 進符号化での MCL

id	value
A	0.1
B	0.2
C	0.8
D	0.9



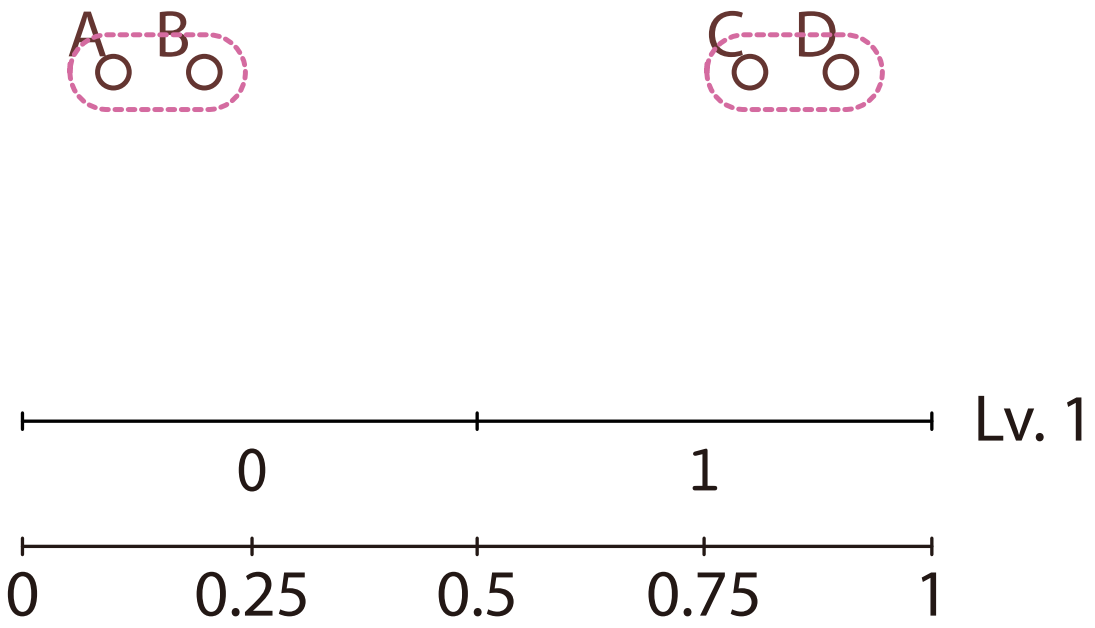
2 進符号化での MCL

id	value
A	0.1
B	0.2
C	0.8
D	0.9



2 進符号化での MCL

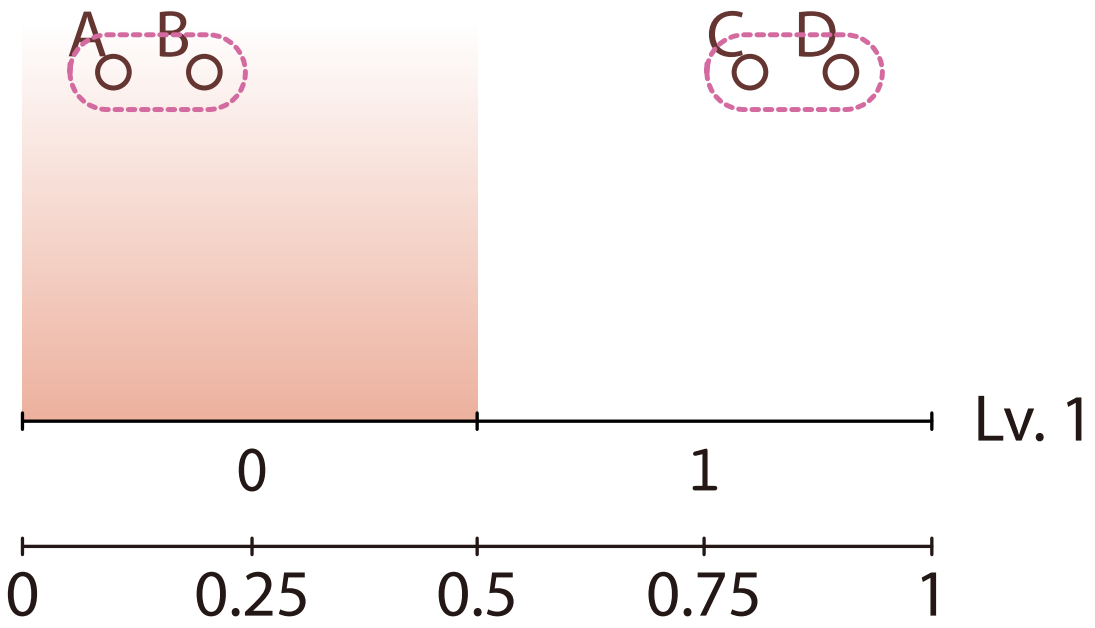
id	value
A	0.1
B	0.2
C	0.8
D	0.9



MCL = 1 + 1 = 2

2 進符号化での MCL

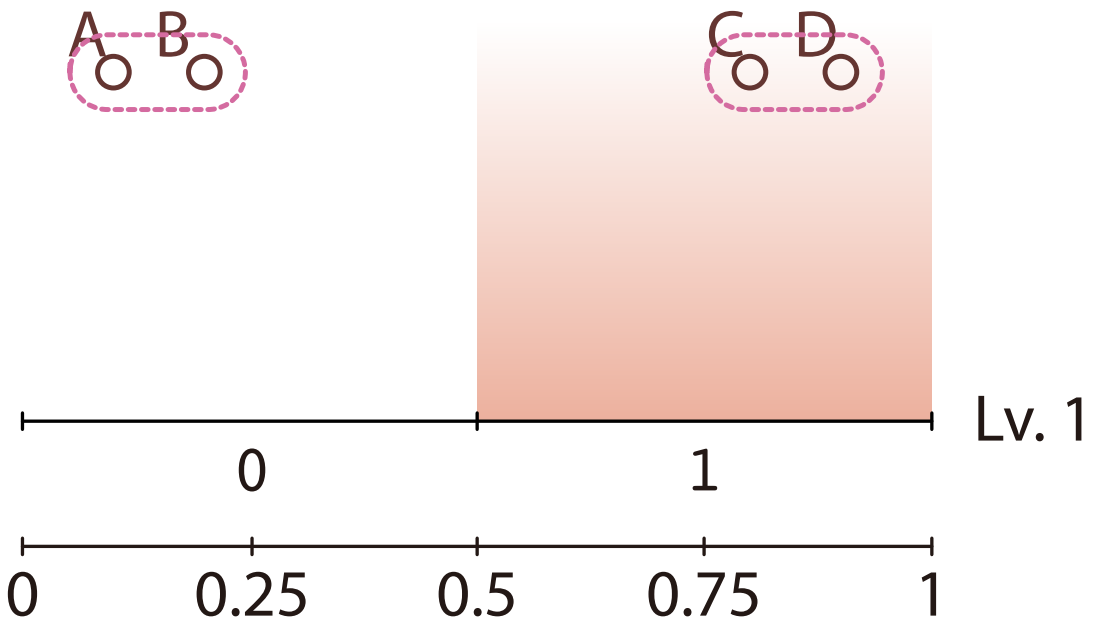
id	value
A	0.1
B	0.2
C	0.8
D	0.9



MCL = 1 + 1 = 2

2 進符号化での MCL

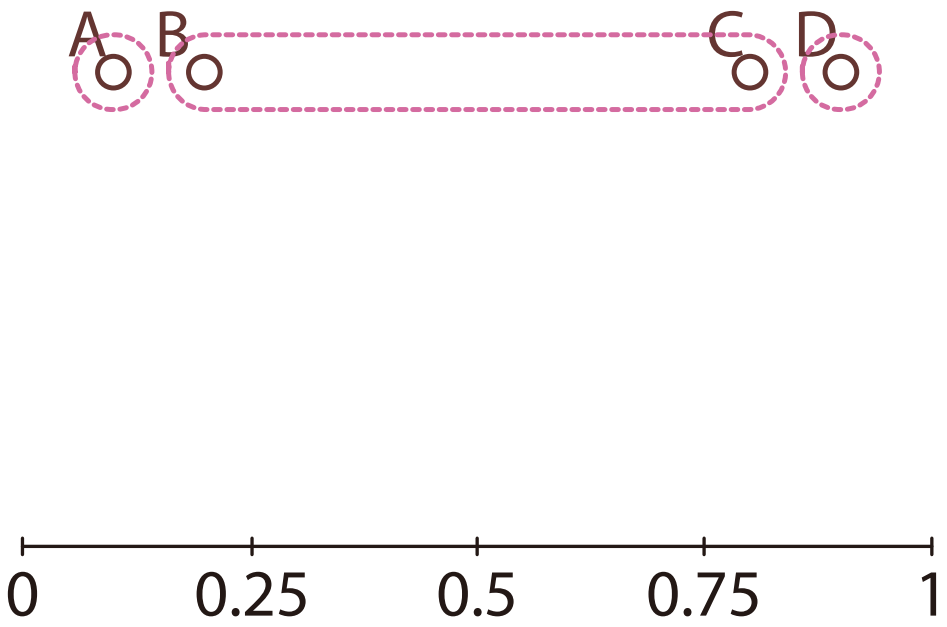
id	value
A	0.1
B	0.2
C	0.8
D	0.9



$MCL = 1 + 1 = 2$

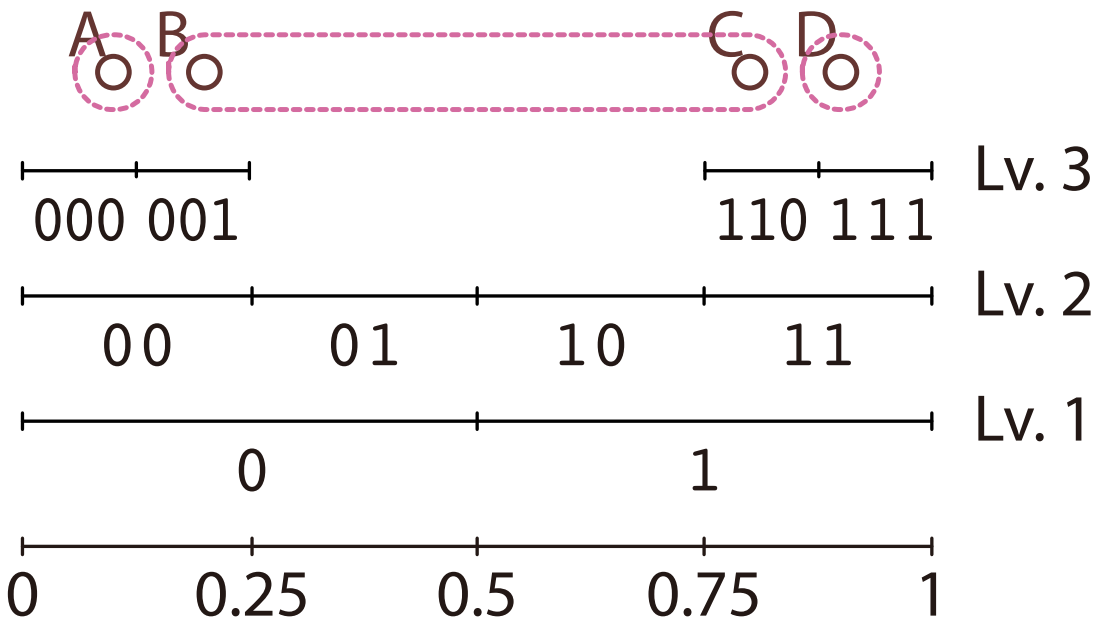
2 進符号化での MCL

id	value
A	0.1
B	0.2
C	0.8
D	0.9



2 進符号化での MCL

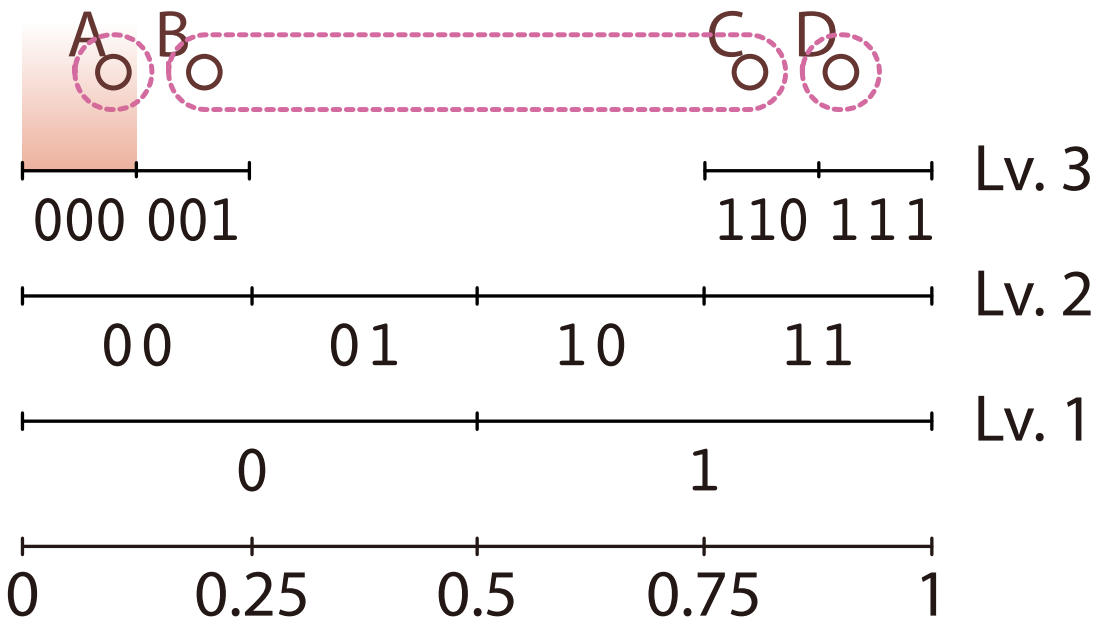
id	value
A	0.1
B	0.2
C	0.8
D	0.9



$$\mathrm{MCL} = 3 \cdot 4 = 12$$

2 進符号化での MCL

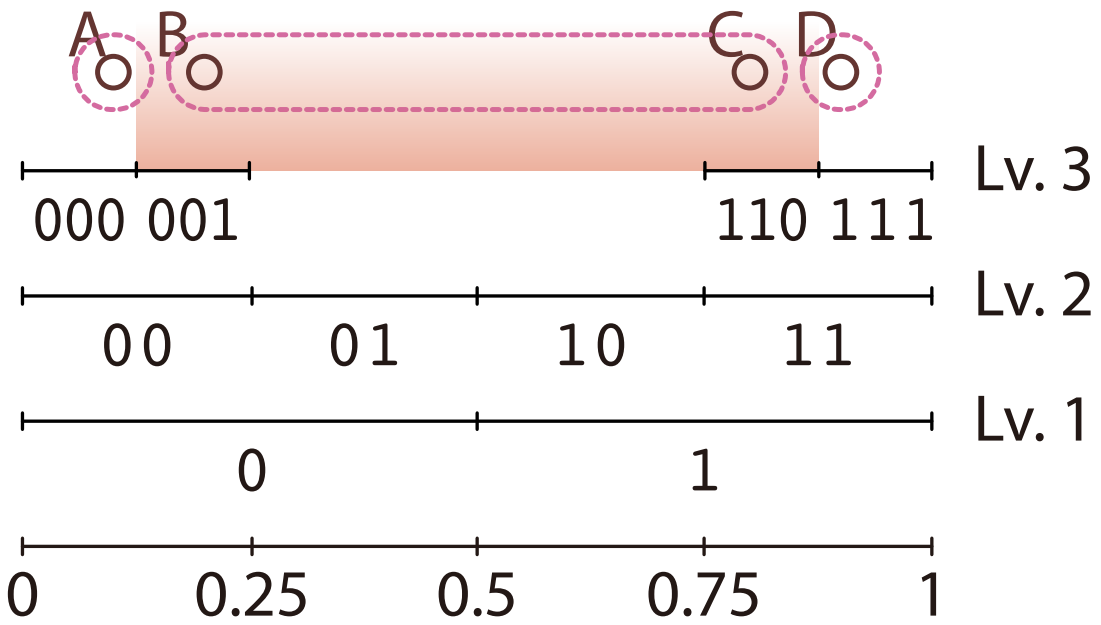
id	value
A	0.1
B	0.2
C	0.8
D	0.9



$$\mathrm{MCL} = 3 \cdot 4 = 12$$

2 進符号化での MCL

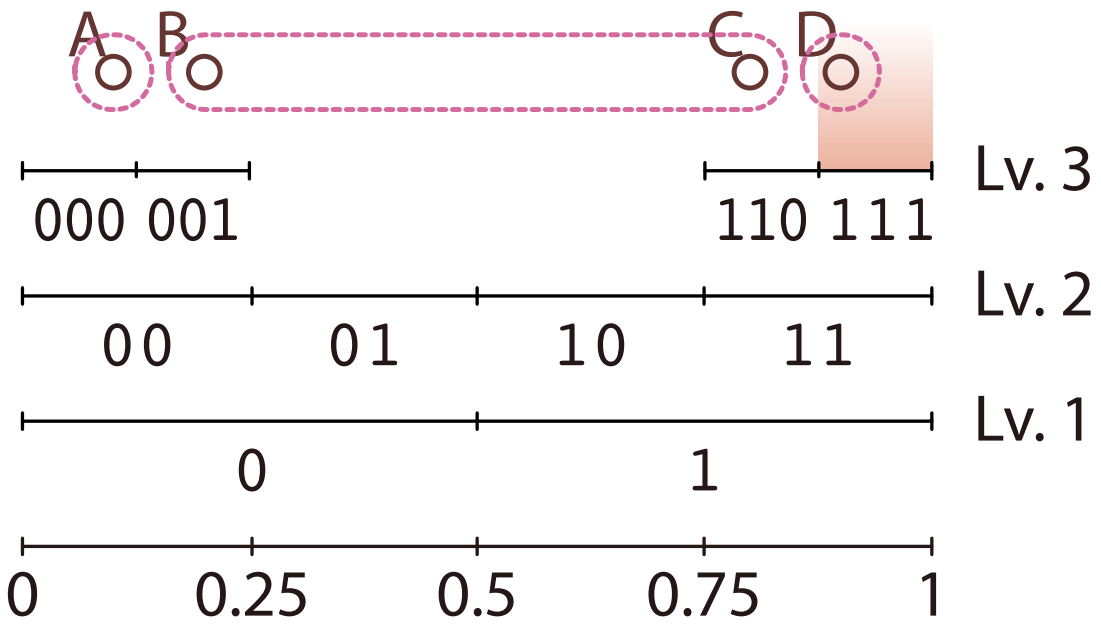
id	value
A	0.1
B	0.2
C	0.8
D	0.9



$$\mathrm{MCL} = 3 \cdot 4 = 12$$

2 進符号化での MCL

id	value
A	0.1
B	0.2
C	0.8
D	0.9



$$\mathrm{MCL} = 3 \cdot 4 = 12$$

MCL の定義

- 埋め込み $\varphi : \mathbb{R}^d \to \Sigma^\omega$ を固定（通常 $\Sigma = \{0, 1\}$）
- $p \in \mathrm{range}(\varphi)$ と $P \subset \mathrm{range}(\varphi)$ に対して，以下を定義

$$\Phi(p|P) := \left\{ w \in \Sigma^* \;\middle|\; \begin{array}{l} p \in \uparrow w \text{ and } P \cap \uparrow v = \varnothing \text{ for all } v \\ \text{such that } |v| = |w| \text{ and } p \in \uparrow v \end{array} \right\}$$

  - $\Phi(p|P)$ の各要素は p を P から分離する接頭辞

データセット X の分割 $\mathscr{C} = \{C_1, \ldots, C_K\}$ に対して，

$$\mathrm{MCL}(\mathscr{C}) := \sum\nolimits_{i \in \{1,\ldots,K\}} L_i(\mathscr{C}), \qquad \text{where}$$

$$L_i(\mathscr{C}) := \min \left\{ |W| \;\middle|\; \begin{array}{l} \varphi(C_i) \subseteq \uparrow W \text{ and} \\ W \subseteq \bigcup\nolimits_{x \in C_i} \Phi(\varphi(x) \mid \varphi(X \setminus C_i)) \end{array} \right\}$$

MCL の最小化とクラスタリング

MCL 基準でのクラスタリングとは，MCL を最小化する大域的最適解を見つけること

– 以下をみたす $\mathscr{C}_{\mathrm{op}}$ を見つける

$$\mathscr{C}_{\mathrm{op}} \in \operatorname*{argmin}_{\mathscr{C} \in \mathscr{C}(X)_{\geqslant K}} \mathrm{MCL}(\mathscr{C}),$$

where $\mathscr{C}(X)_{\geqslant K} = \{\, \mathscr{C} \text{ is a partition of } X \mid \#C \geqslant K \,\}$

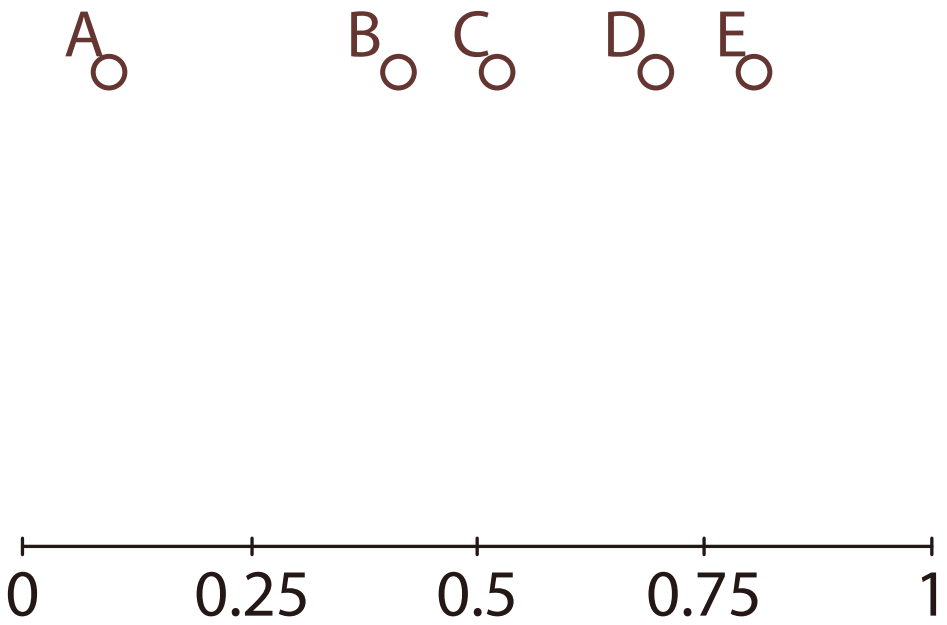
- クラスタ数 K の下限をパラメータとして与える
  – この仮定がないと，$\mathscr{C}_{\mathrm{op}}$ は $\{X\}$ になってしまう

COOL による最適化

- COOL は最適化問題を$O(nd)$で解く
  - n と d はそれぞれデータ数と次元数
    - naïve な方法だと指数時間かかる
  - 離散化による MCL の計算そのものがクラスタリングとなる
- COOL はレベルワイズなアルゴリズムであり，
  以下を満たすレベル k 分割 $\mathscr{C}^k$ を $k = 1, 2, \ldots$ で求める
  - 任意の $x, y \in X$ に対して，それらが同じクラスタに入る $\iff$
    ある $v \sqsubset \varphi(x)$ と $w \sqsubset \varphi(y)$（$|v| = |w| = k$）があって $v = w$
    - レベル k 分割は階層構造をなす
    - $C \in \mathscr{C}^k$ に対して，$\mathscr{D} \subseteq \mathscr{C}^{k+1}$ が存在して $\bigcup \mathscr{D} = C$
- すべての $C \in \mathscr{C}_{\mathrm{op}}$ に対して，ある k があって $C \in \mathscr{C}^k$

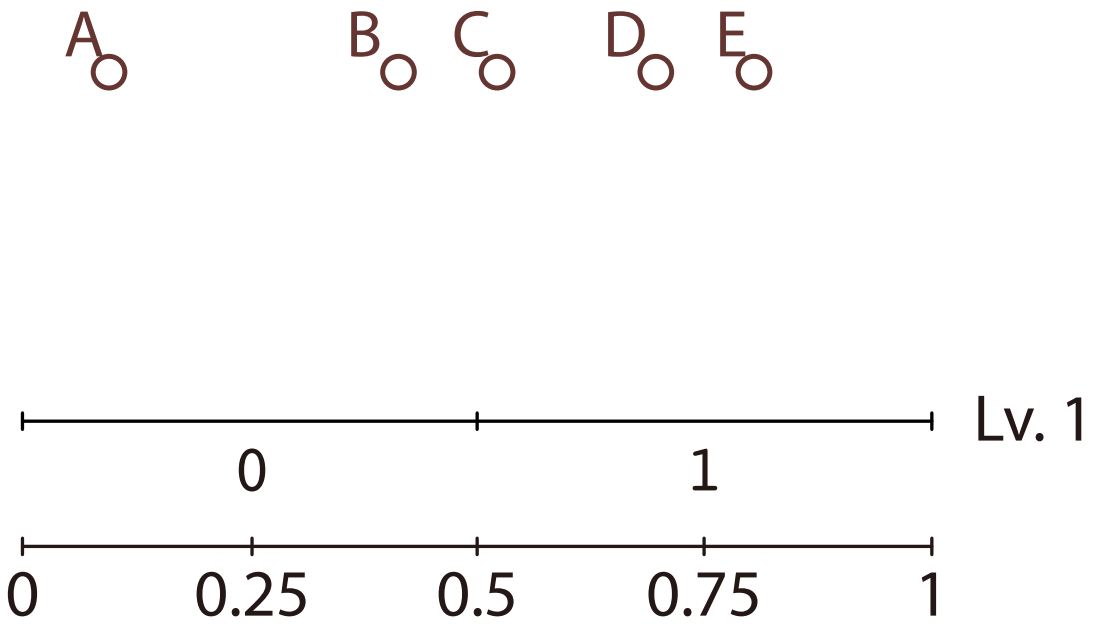
2 進符号化での COOL

id	value
A	0.113
B	0.398
C	0.526
D	0.701
E	0.796



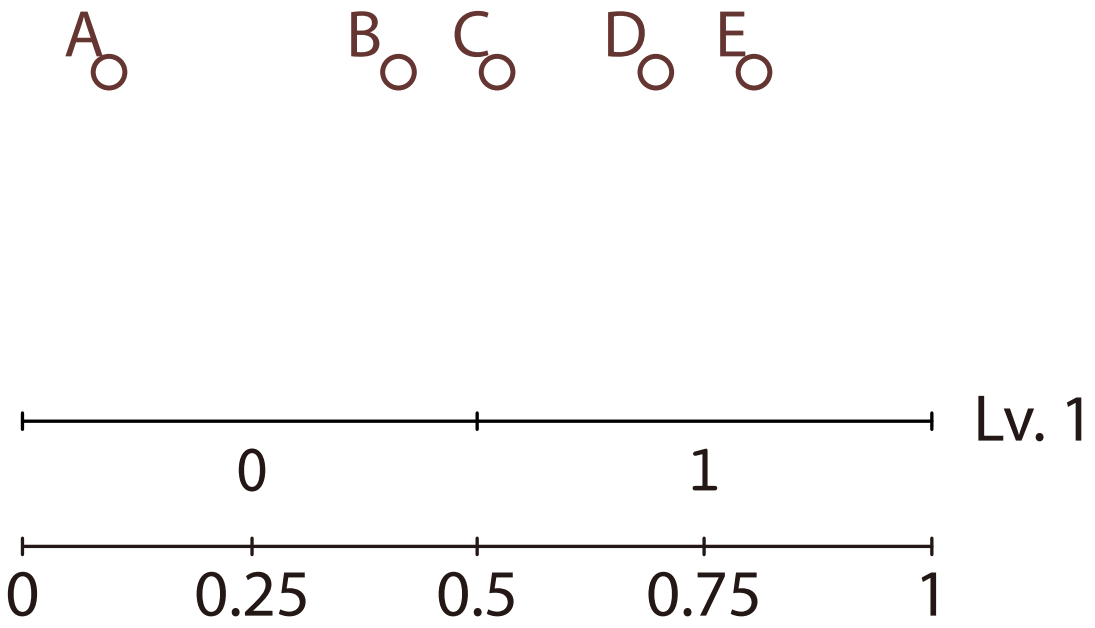
2 進符号化での COOL

id	value
A	0.113
B	0.398
C	0.526
D	0.701
E	0.796



2 進符号化での COOL

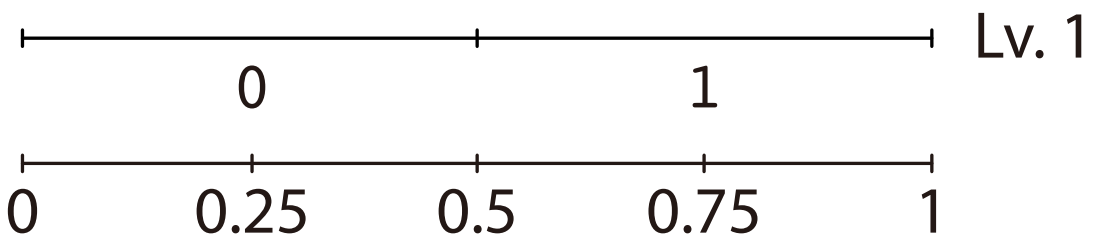
id	value
A	0
B	0
C	1
D	1
E	1



2 進符号化での COOL

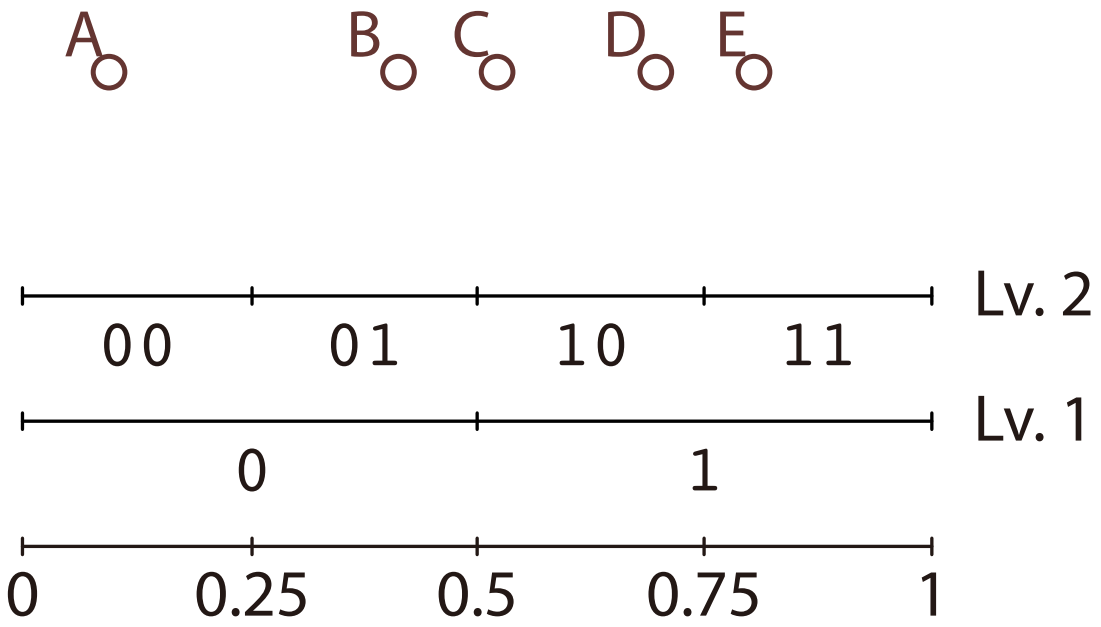
id	value
A	0
B	0
C	1
D	1
E	1

MCL = 1 + 1 = 2



2 進符号化での COOL

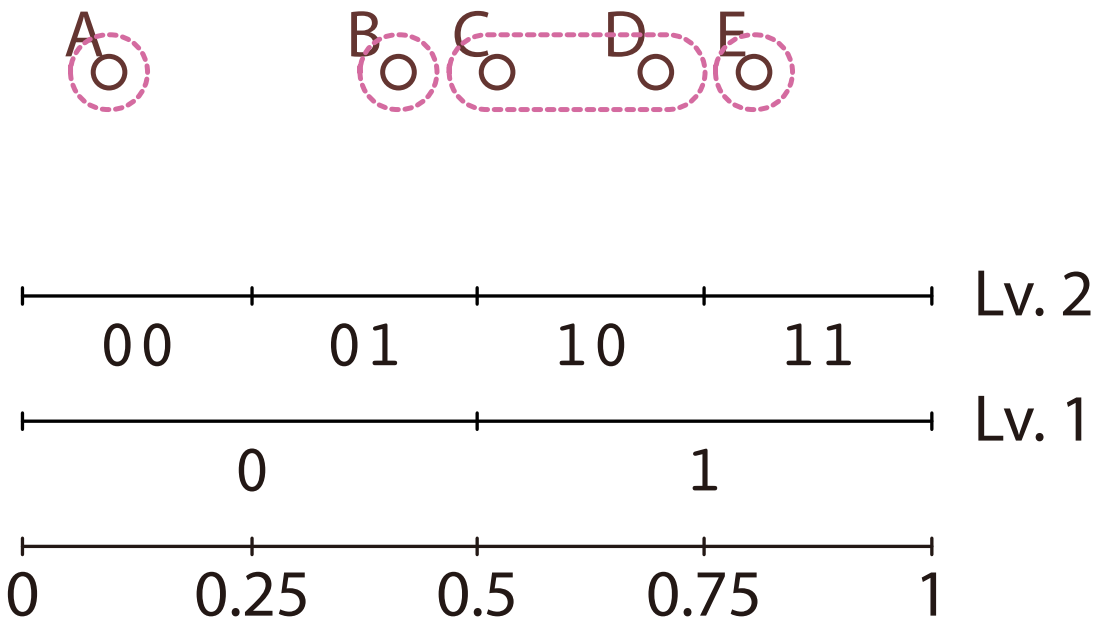
id	value
A	00
B	01
C	10
D	10
E	11



2 進符号化での COOL

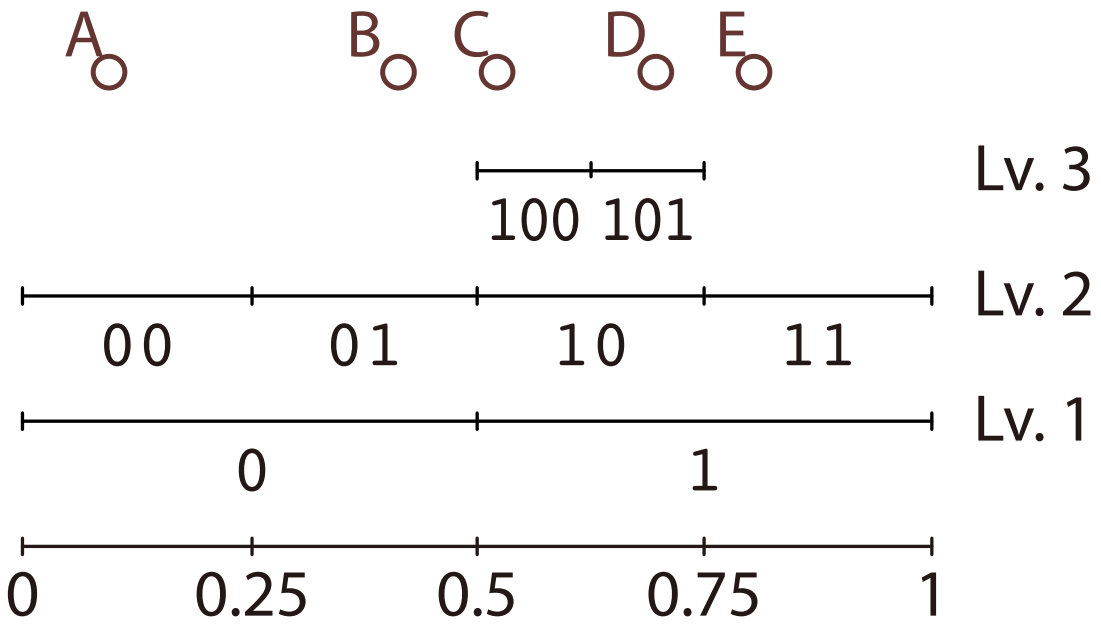
id	value
A	00
B	01
C	10
D	10
E	11

$MCL = 2 \cdot 4 = 8$



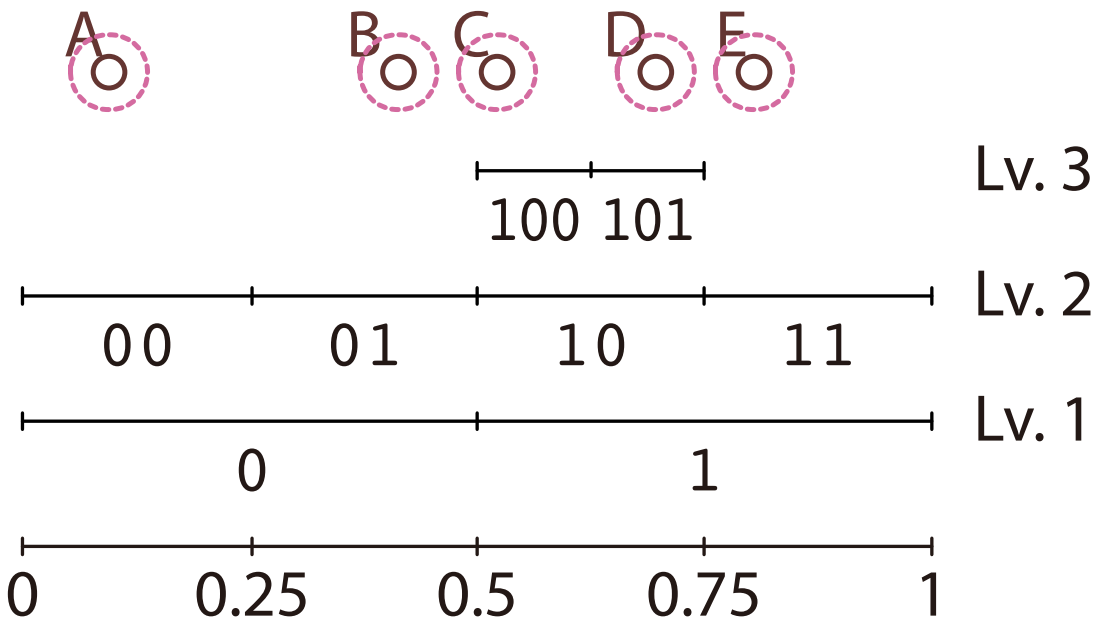
2 進符号化での COOL

id	value
A	00
B	01
C	100
D	101
E	11



2 進符号化での COOL

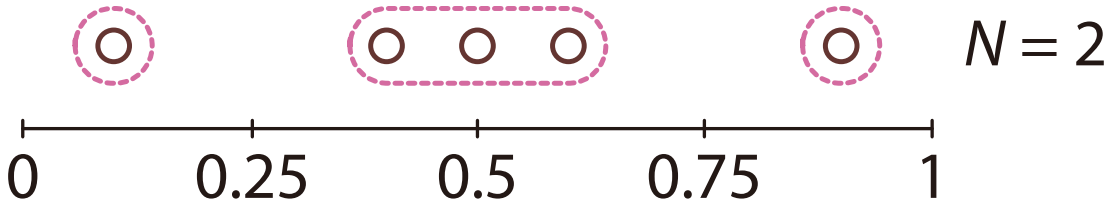
id	value
A	00
B	01
C	100
D	101
E	11



MCL = 6 + 6 = 12

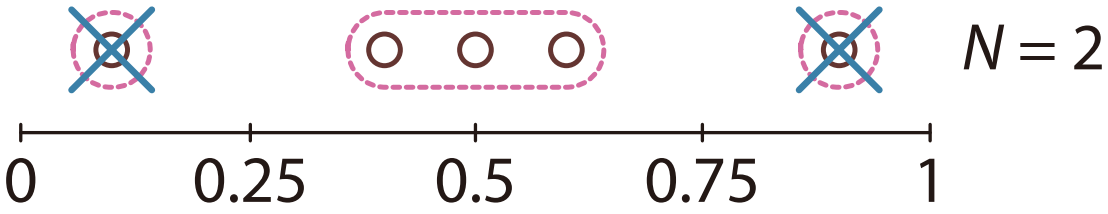
COOL によるノイズ除去

- COOL ではノイズ除去を容易に実現できる
- 分割 $\mathscr{C}$ に対して，$\mathscr{C}_{\geqslant N} := \{C \in \mathscr{C} \mid \#C \geqslant N\}$ と定義する
  - クラスタ C を，$\#C < N$ のときノイズとみなす
- 例：$\mathscr{C} = \{\{0.1\}, \{0.4, 0.5, 0.6\}, \{0.9\}\}$ とする
  - $\mathscr{C}_{\geqslant 2} = \{\{0.4, 0.5, 0.6\}\}$ であり，0.1 と 0.9 はノイズ
- クラスタサイズの下限 Nを入力パラメータとする



COOL によるノイズ除去

- COOL ではノイズ除去を容易に実現できる
- 分割 $\mathscr{C}$ に対して，$\mathscr{C}_{\geqslant N} := \{C \in \mathscr{C} \mid \#C \geqslant N\}$ と定義する
  - クラスタ C を，$\#C < N$ のときノイズとみなす
- 例：$\mathscr{C} = \{\{0.1\}, \{0.4, 0.5, 0.6\}, \{0.9\}\}$ とする
  - $\mathscr{C}_{\geqslant 2} = \{\{0.4, 0.5, 0.6\}\}$ であり，0.1 と 0.9 はノイズ
- クラスタサイズの下限 Nを入力パラメータとする



グレイコード

- 閉区間 $[0, 1]$ の実数を 0，1，$\perp$ で符号化

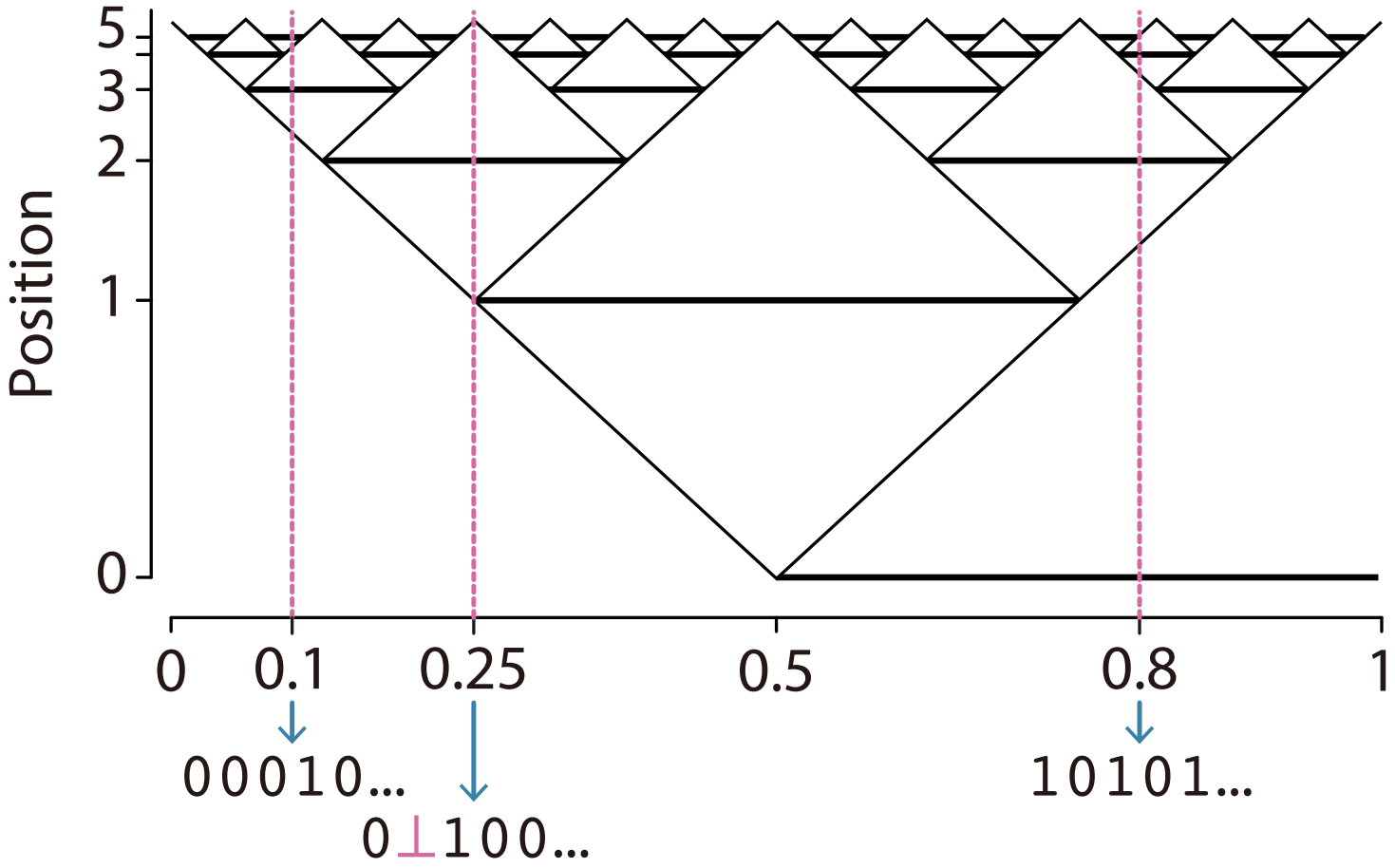
  Binary: $0.1 \to 00011\ldots, 0.25 \to 00111\ldots$

  Gray: $0.1 \to 00010\ldots, 0.25 \to 0\perp100\ldots$

- もともとは，自然数の 2 進符号化とは異なる符号化方式
  - 特に，アナログ → デジタル変換に用いられる [Knuth, 2005]

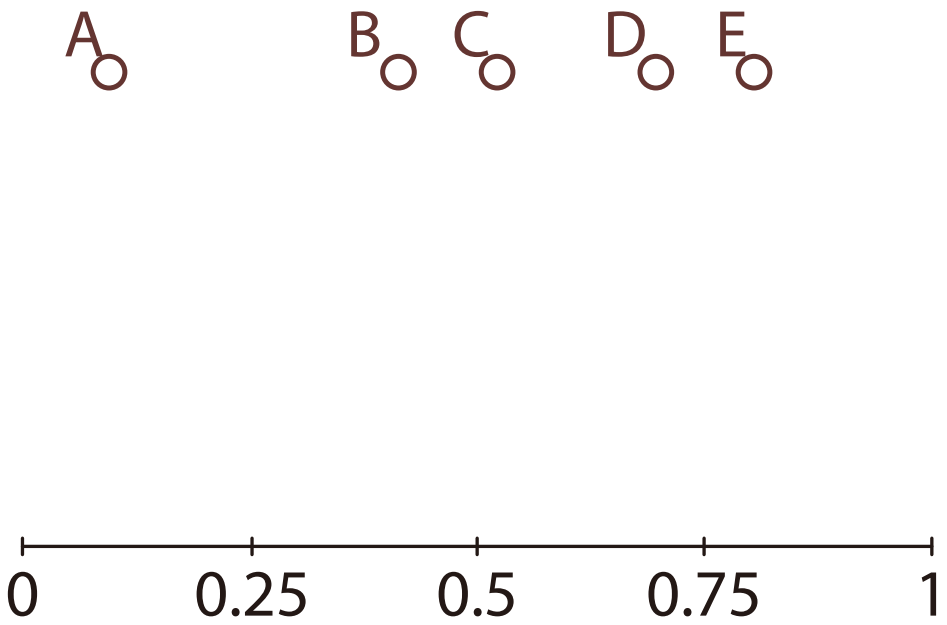
> グレイコード埋め込みとは，$x \in [0, 1]$ を以下の無限列 $p_0p_1p_2\ldots$ に写す単射 φ_{G}
> - $2^{-i}m - 2^{-(i+1)} < x < 2^{-i}m + 2^{-(i+1)}$ が奇数 m に対して成り立つなら $p_i := 1$，偶数 m に対して成り立つなら $p_i := 0$，整数 m に対して $x = 2^{-i}m - 2^{-(i+1)}$ なら $p_i := \perp$
> - ベクトル $\boldsymbol{x} = (x^1, \ldots, x^d)$ に対して $\varphi_{\mathrm{G}}(\boldsymbol{x}) = p_1^1 \ldots p_1^d p_2^1 \ldots p_2^d \ldots$

グレイコード埋め込み



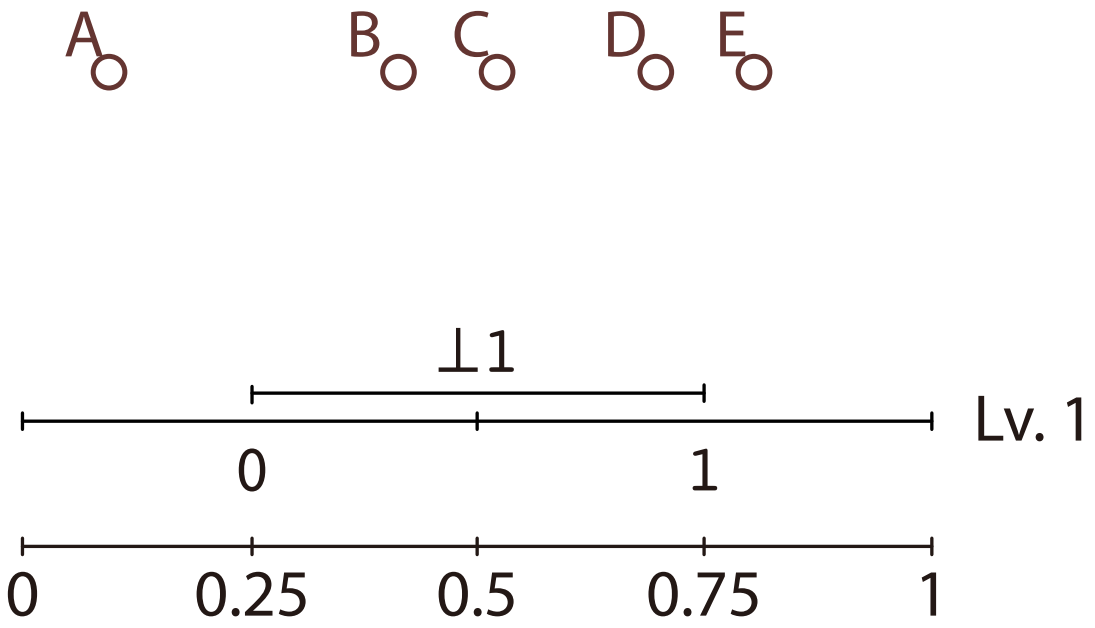
グレイコードでの COOL（G-COOL）

id	value
A	0.113
B	0.398
C	0.526
D	0.701
E	0.796



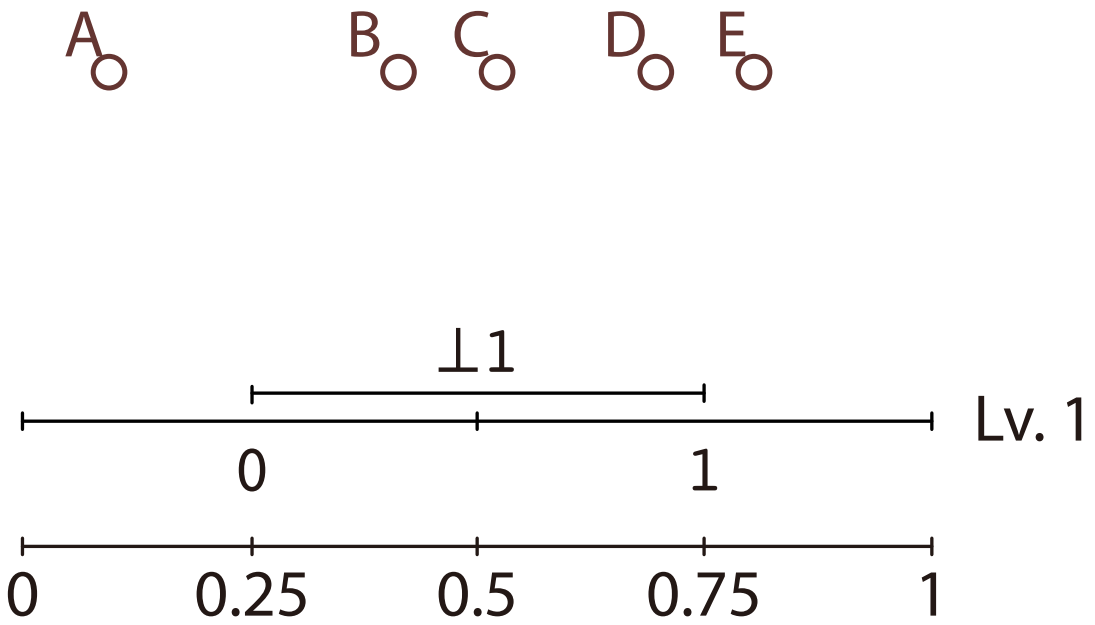
グレイコードでの COOL（G-COOL）

id	value
A	0.113
B	0.398
C	0.526
D	0.701
E	0.796



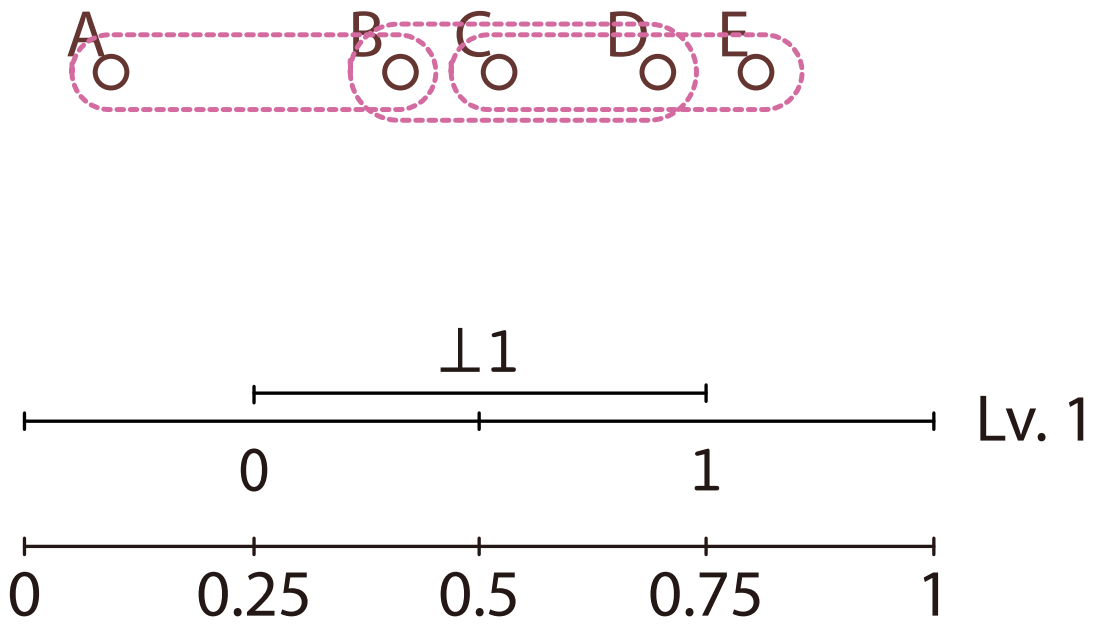
グレイコードでの COOL（G-COOL）

id	value
A	0
B	0, ⊥1
C	1, ⊥1
D	1, ⊥1
E	1



グレイコードでの COOL（G-COOL）

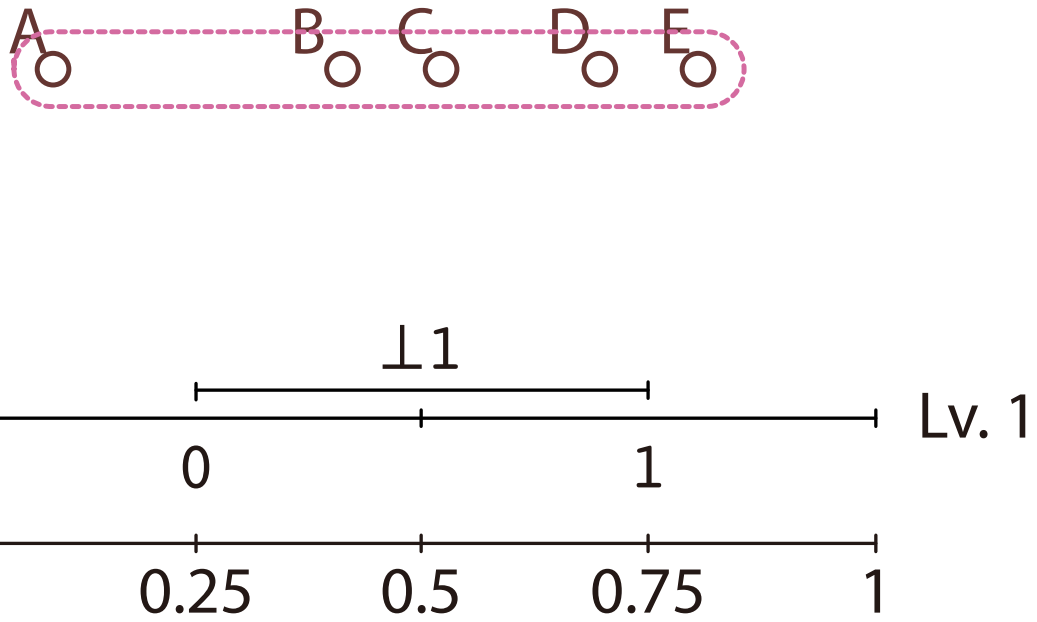
id	value
A	0
B	0, ⊥1
C	1, ⊥1
D	1, ⊥1
E	1



グレイコードでの COOL（G-COOL）

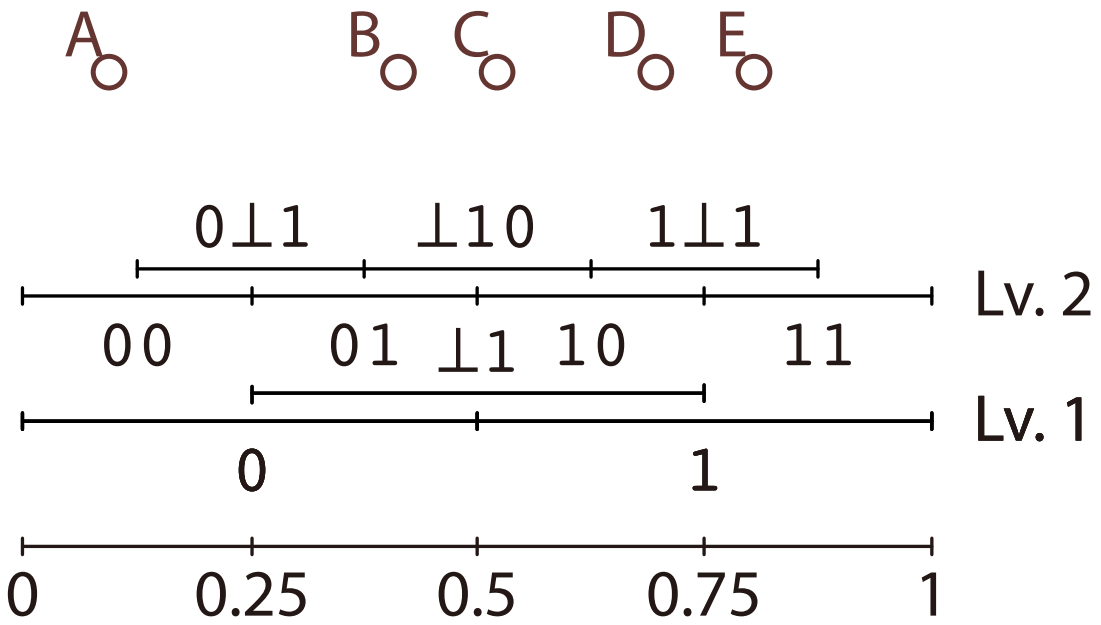
id	value
A	0
B	0, ⊥1
C	1, ⊥1
D	1, ⊥1
E	1

$$\mathrm{MCL} = 1 \cdot 2 = 2$$



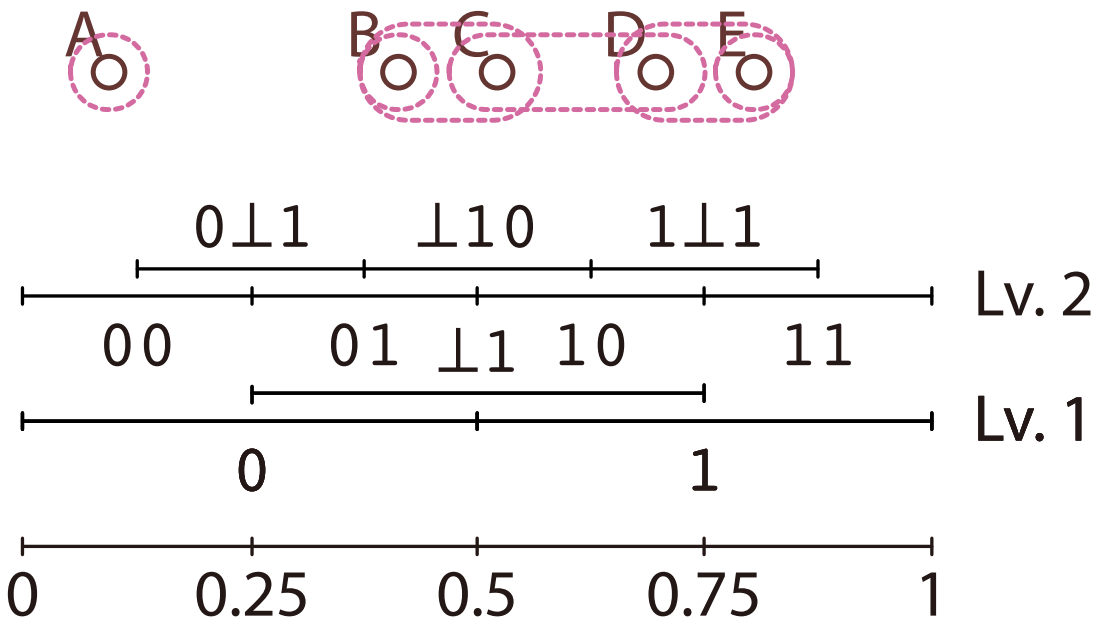
グレイコードでの COOL（G-COOL）

id	value
A	00
B	01, ⊥10
C	10, ⊥10
D	10, 1⊥1
E	11, 1⊥1



グレイコードでの COOL（G-COOL）

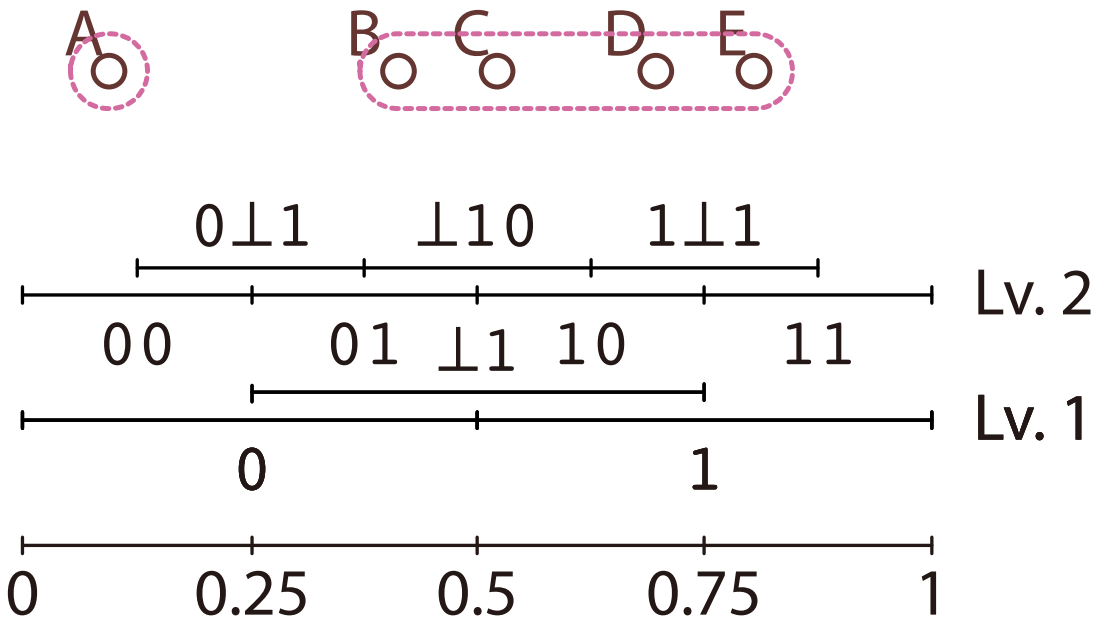
id	value
A	00
B	01, ⊥10
C	10, ⊥10
D	10, 1⊥1
E	11, 1⊥1



グレイコードでの COOL（G-COOL）

id	value
A	00
B	01, ⊥10
C	10, ⊥10
D	10, 1⊥1
E	11, 1⊥1

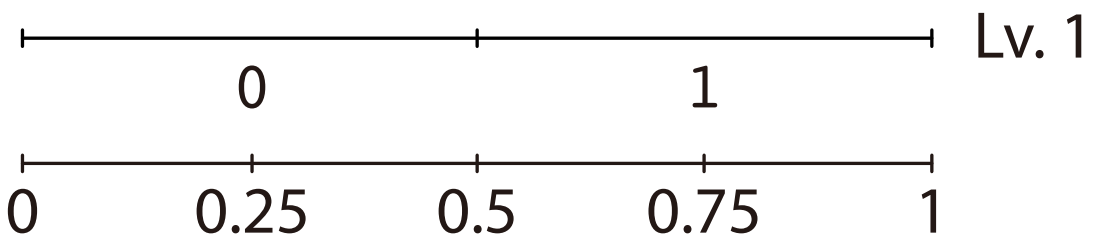
$$\mathrm{MCL} = 2 \cdot 3 = 6$$



2 進符号化での COOL

id	value
A	0
B	0
C	1
D	1
E	1

MCL = 1 + 1 = 2



G-COOL の理論的解析

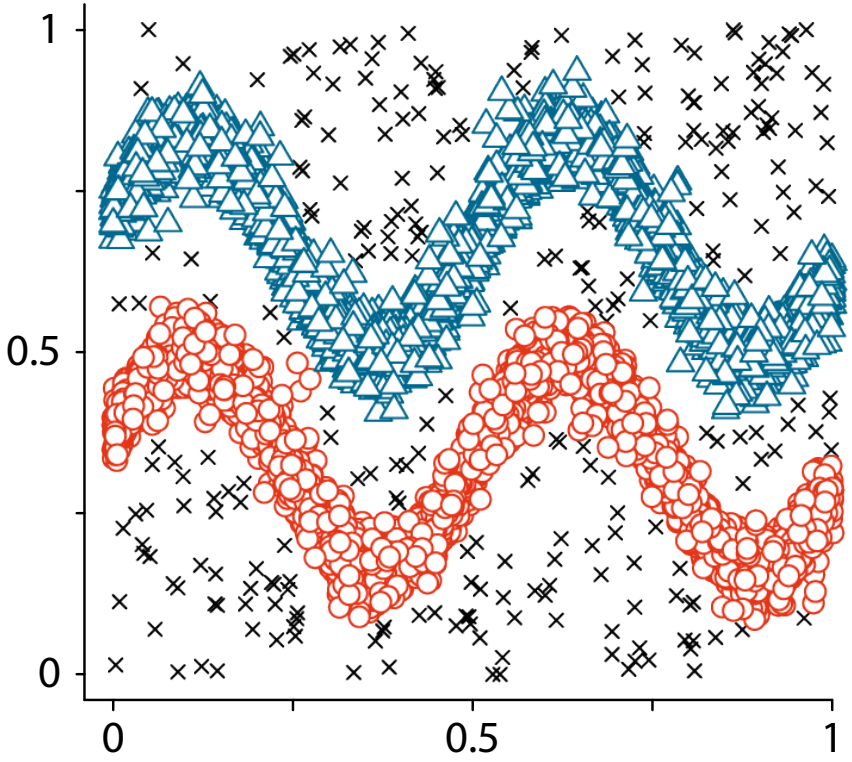
- COOL での固定された符号化方式としてグレイコードを使う
  - 内的結合と外的分離を達成する

- **定理：レベル k 分割 $\mathscr{C}^k$ に対して，$x, y \in X$ は $d_\infty(x, y) < 2^{-(k+1)}$ のとき同じクラスタに属する**
  - したがって，x, y は $d_\infty(x, y) \geqslant 2^{-(k+1)}$ ならば異なるクラスタ
  - $d_\infty(x, y) = \max_{i \in \{1,\ldots,d\}} |x_i - y_i|$ (L_∞ metric)
    - 2 つの隣接する区間が重なりあうため，それらが併合される

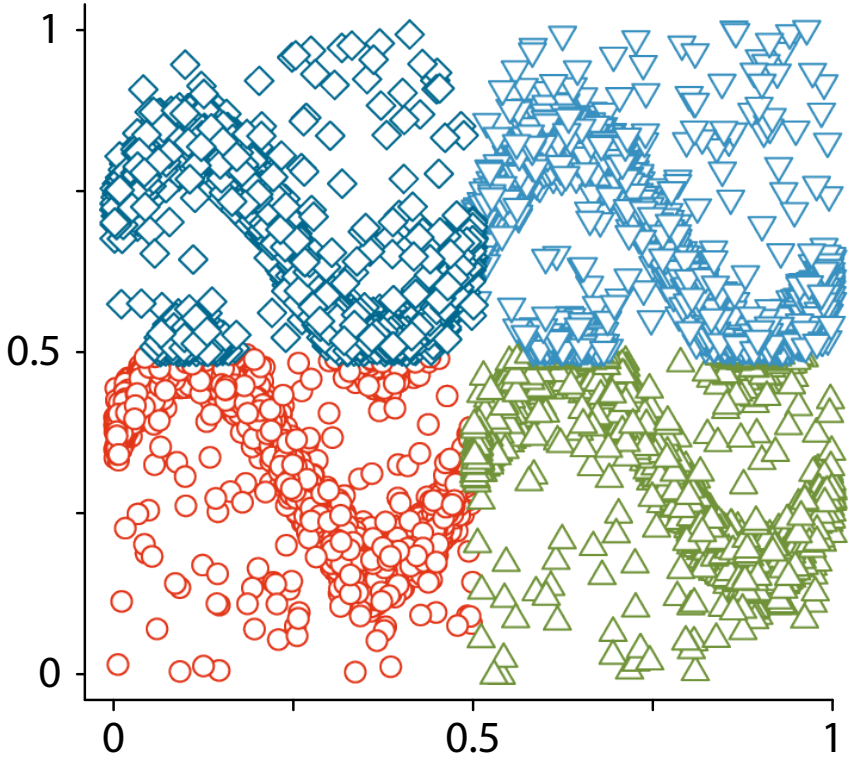
- **系：最適な分割 $\mathscr{C}_{op}$ において，任意の $x \in C$（$C \in \mathscr{C}_{op}$）に対して，その最近傍 $y \in C$**
  - y は x の最近傍 $\iff$ $y \in \operatorname{argmin}_{y \in X} d_\infty(x, y)$

G-COOL の能力

G-COOL



COOL with the binary encoding



まとめ

- コード化主導でクラスタリングとその評価を統合した
  - どうやってクラスタリングの良さを測るか，どうやって良いクラスタを見つけるか，という 2 つの問に実効的な解を与えた
    - 距離計算やデータ分布はまったく用いない
- 鍵となるアイデア：
  1. 実数値データの符号化方式を固定する
     - クラスタの圧縮に着目してMCLを導入した
     - MCL を用いてクラスタリングを最適化問題として定式化し，それを線形時間で解くアルゴリズムCOOLを構築した
  2. グレイコード
     - 理論的・実験的に G-COOL の有効性を示した

目次

- 第一部：理論
  - フラクタルとハウスドルフ距離を用いた図形の学習
    — 計算可能な 2 値分類を目指して
- 第二部：理論から実践
  - 符号化ダイバージェンス
  - MCL とグレイコードを用いたクラスタリング
  - 2 進符号化を活用した高速クラスタリング
- 第三部：形式概念と共に
  - 離散値・連続値混在データからの半教師あり学習
  - 半教師ありのマルチラベルクラス分類によるリガンド発見

2 進符号化を活用した高速クラスタリング

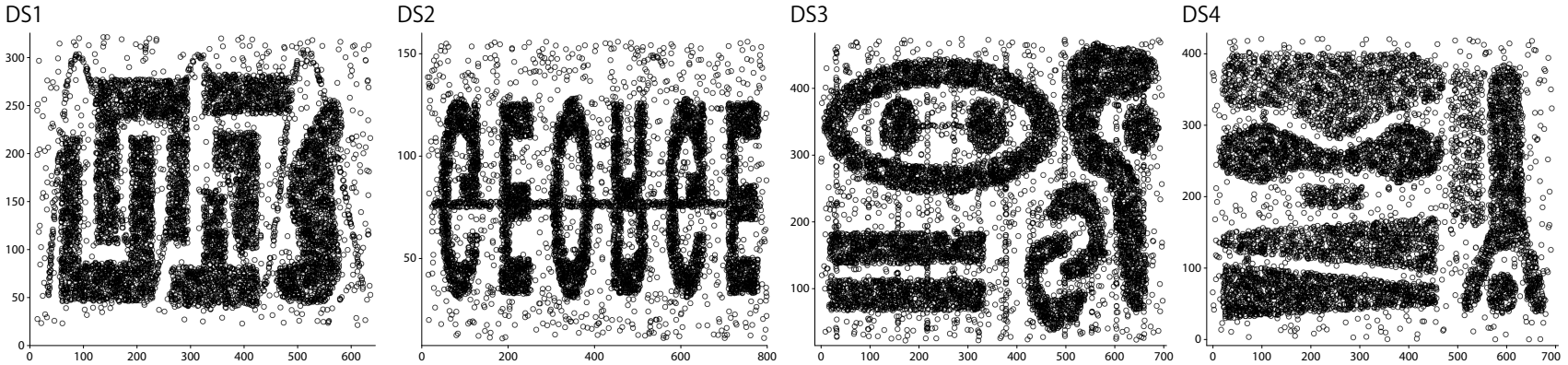
- Sugiyama, M., Yamamoto, A.:
  A Fast and Flexible Clustering Algorithm Using Binary Discretization,
  ICDM 2011
- Sugiyama, M., Yamamoto, A.:
  Fast Spatial Clustering Using Binary Discretization,
  Journal of Intelligent Information Systems (submitted)

概要・結果

1. *BOOL (Bianry cOding Oriented cLustering)*
   - 実数値データに対するクラスタリングアルゴリズム
   - 任意形状を抽出可能
   - ノイズに頑健
   - 入力パラメータに対して頑健
2. **BOOL は *K*-means より高速で，かつ任意形状を抽出可能なアルゴリズムのなかで最速**
   - 任意形状を抽出可能な最新のアルゴリズム [Chaoji *et al*. SDM 2011, Chaoji *et al*. KAIS 2009]よりも2桁から3桁速い

典型的なベンチマークでの評価

- 4 つの合成データ（DS1 - DS4）
  - CLUTO ウェブサイトから取得
    - `http://glaros.dtc.umn.edu/gkhome/views/cluto/`
- これらは，（空間）クラスタリングで典型的なベンチマーク
  - CHAMELEON [Karypis *et al.*, 1999], SPARCL [Chaoji *et al.*, 2009], ABACUS [Chaoji *et al.*, 2011]など

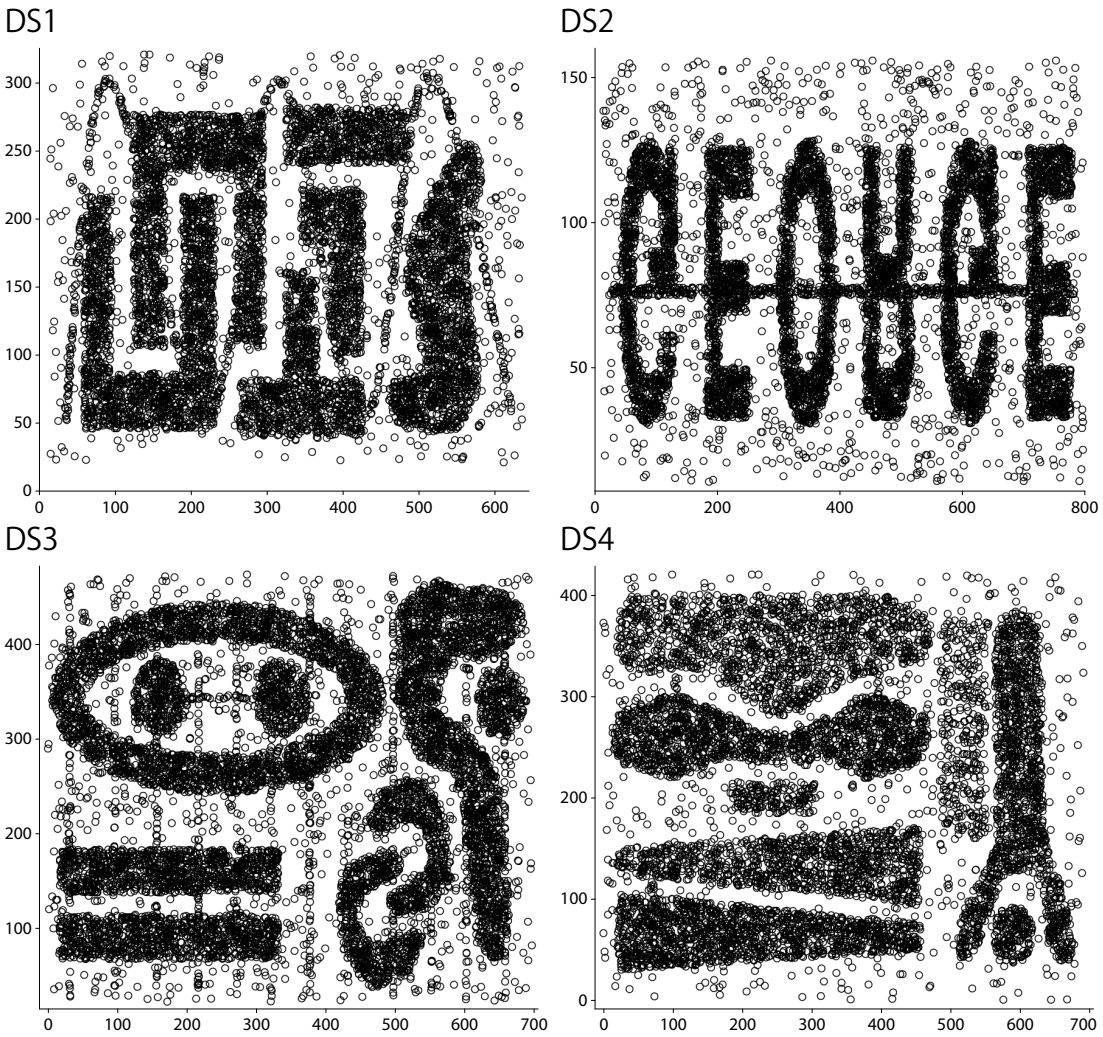


ベンチマークでの結果

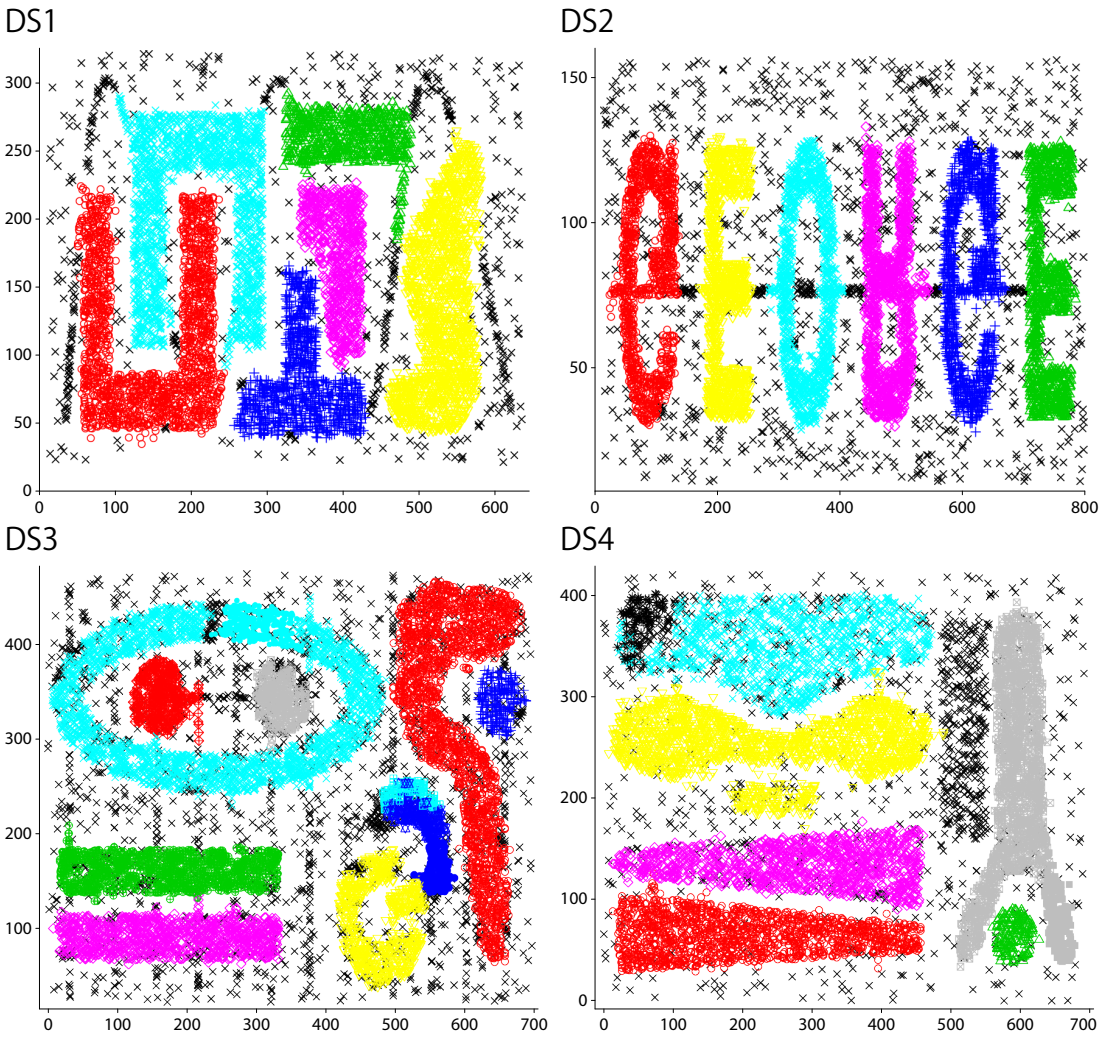
- 実行時間（秒）
  - 表において，n はデータ数を表す
  - ABACUS と SPARCL の結果は [Chaoji *et al.*, 2011] から転載
    - これらのソースが公開されていないため

		Non-convex				Convex
	n	BOOL	DBSCAN	ABACUS	SPARCL	K-means
DS1	8000	0.004	9.959	1.7	1.8	0.008
DS2	8000	0.004	10.041	1.3	1.5	0.008
DS3	10000	0.010	15.832	1.9	2.5	0.036
DS4	8000	0.005	9.947	1.7	1.8	0.018

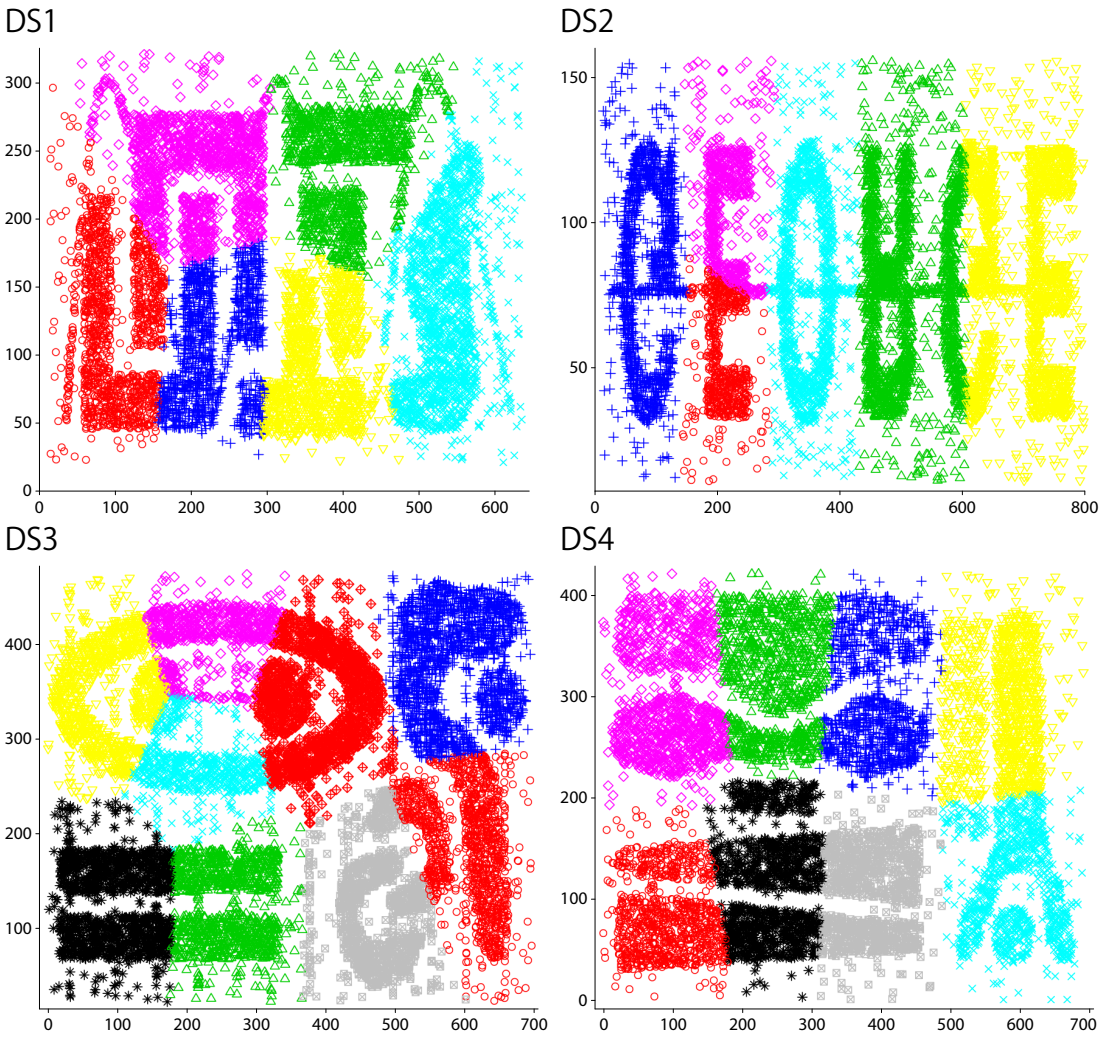
実験結果（もとのデータ）



実験結果（BOOL）



実験結果（*K*-means）



自然画像を用いた評価

- 4 つの自然画像を用いた
  - Berkeley segmentation database and benchmark から取得
    - `http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/`
- サイズは 481 × 321 で，合計 154,401 ピクセル
  - 前処理で各ピクセルから RGB (red-green-blue) 値を得る（[Chaoji *et al.*, 2011] と同じ）

Horse

Mushroom

Pyramid

Road

自然画像での結果

- 実行時間（秒）
  - 表において，n はデータ数を表す
  - ABACUS と SPARCL の結果は [Chaoji *et al.*, 2011] から転載

		Non-convex			Convex
	n	BOOL	ABACUS	SPARCL	Kmeans
Horse	154401	0.253	31.2	41.8	0.674
Mushroom	154401	0.761	29.3	–	1.449
Pyramid	154401	0.187	11.3	–	0.254
Road	154401	0.180	14.9	–	0.209

実験結果（もとのデータ）

Horse

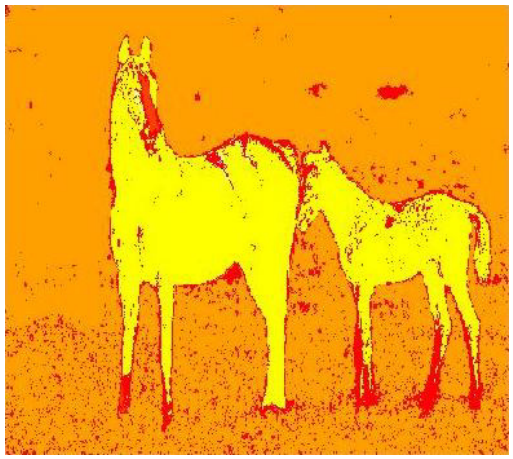
Mushroom

Pyramid

Road

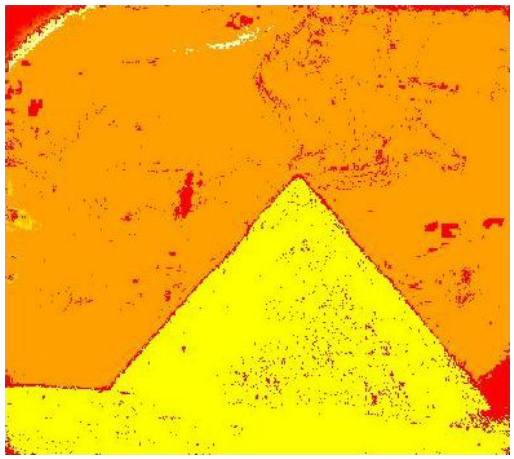
実験結果（BOOL）

Horse

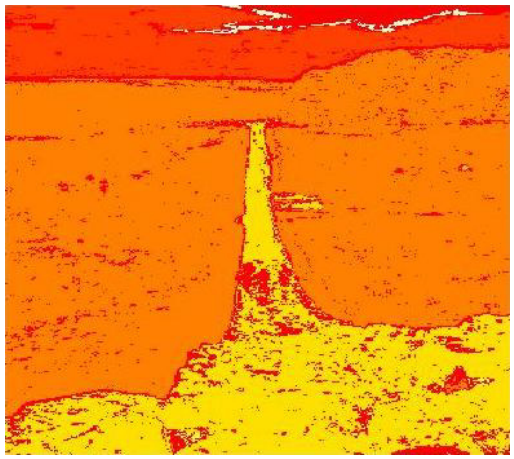


Mushroom

Pyramid

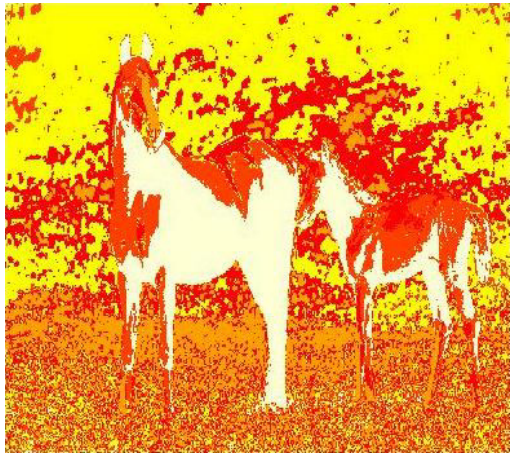


Road



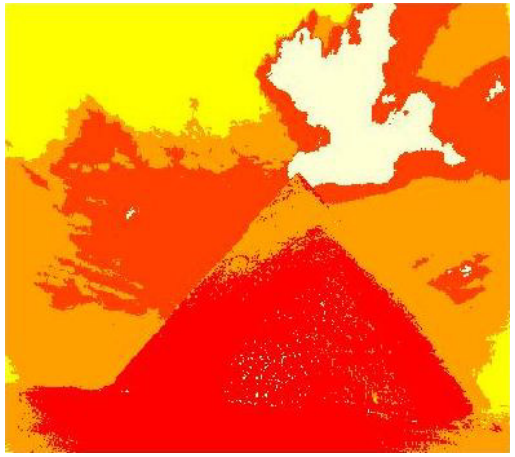
実験結果（*K*-means）

Horse

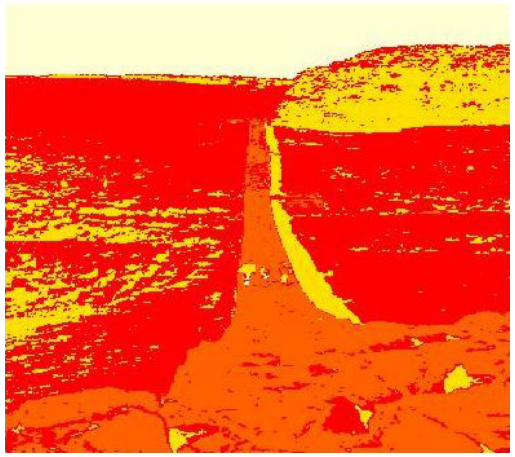


Mushroom

Pyramid



Road



UCI データを用いた結果

- 実行時間（秒）

	n	d	K	BOOL	K-means	DBSCAN
ecoli	336	7	8	0.001	0.002	0.111
sonar	208	60	2	0.004	0.005	0.149
shuttle	14500	9	7	0.025	0.065	–
wdbc	569	30	2	0.004	0.004	0.222
wine	178	13	3	0.002	0.002	0.047
wine quality	4898	11	7	0.019	0.026	7.601

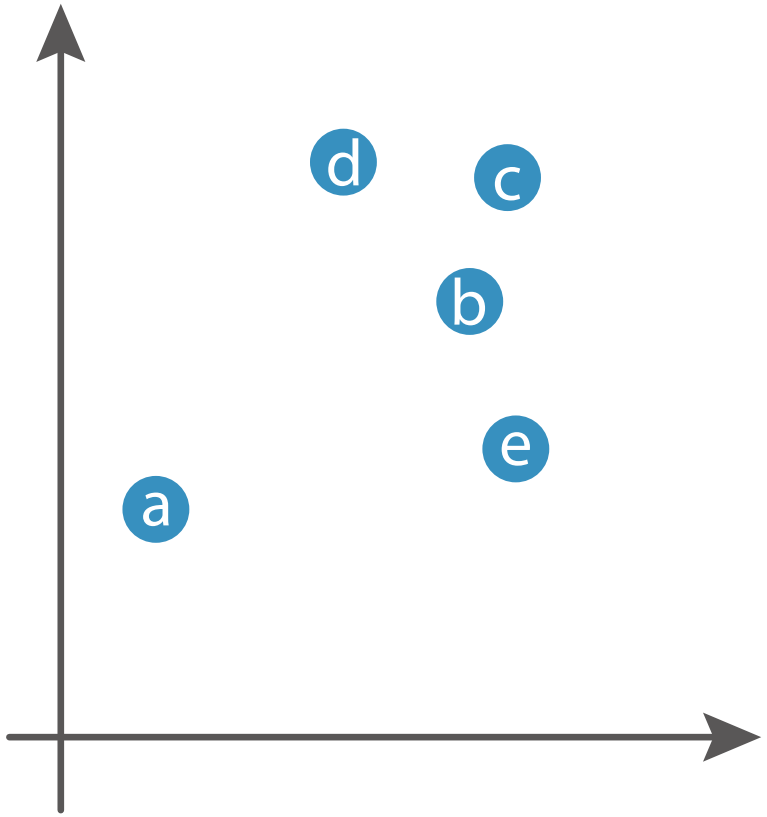
UCI データを用いた結果

- Adjusted Rand index

	n	d	K	BOOL	K-means	DBSCAN
ecoli	336	7	8	0.5745	0.4399	0.1223
sonar	208	60	2	0.0133	0.0064	0.0006
shuttle	14500	9	7	0.7651	0.1432	–
wdbc	569	30	2	0.6806	0.4914	0.5530
wine	178	13	3	0.4638	0.3347	0.2971
wine quality	4898	11	7	0.0151	0.0099	0.0134

クラスタリングのプロセス

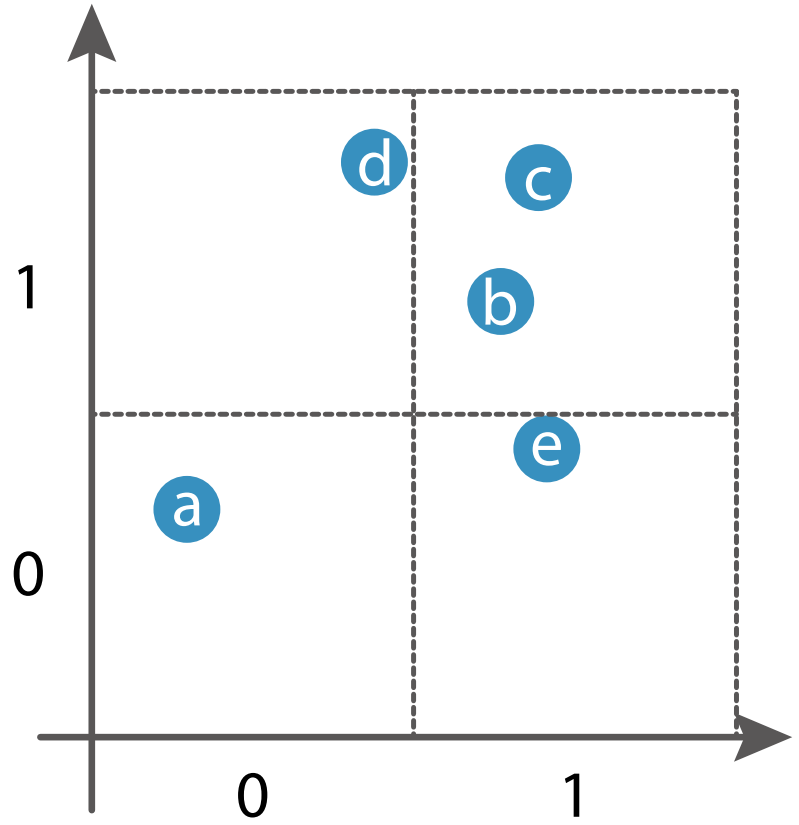
id	x	y
a	0.14	0.31
b	0.66	0.71
c	0.72	0.86
d	0.48	0.89
e	0.74	0.48



データをレベル 1 で離散化する

Discretization level: 1

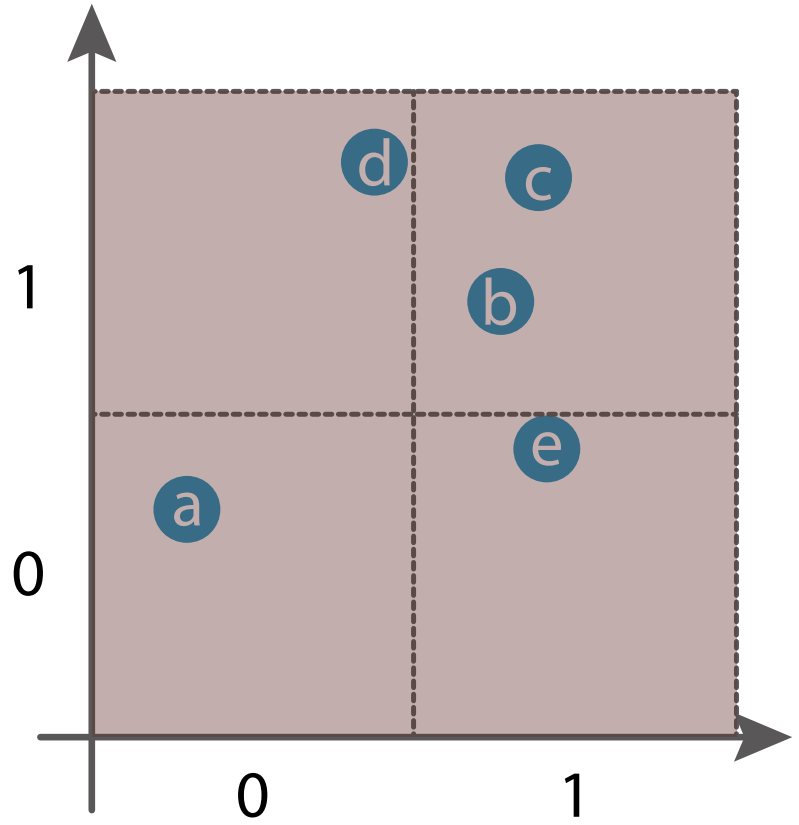
id	*x*	*y*
a	0.14	0.31
b	0.66	0.71
c	0.72	0.86
d	0.48	0.89
e	0.74	0.48



データをレベル 1 で離散化する

Discretization level: 1

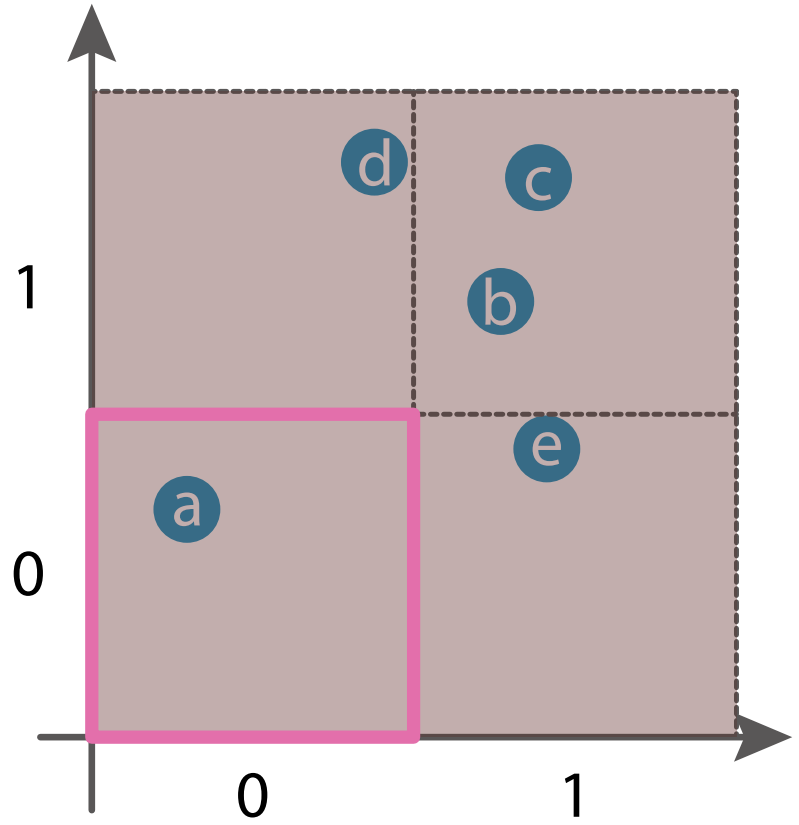
id	x	y
a	0	0
b	1	1
c	1	1
d	0	1
e	1	0



データをレベル 1 で離散化する

Discretization level: 1

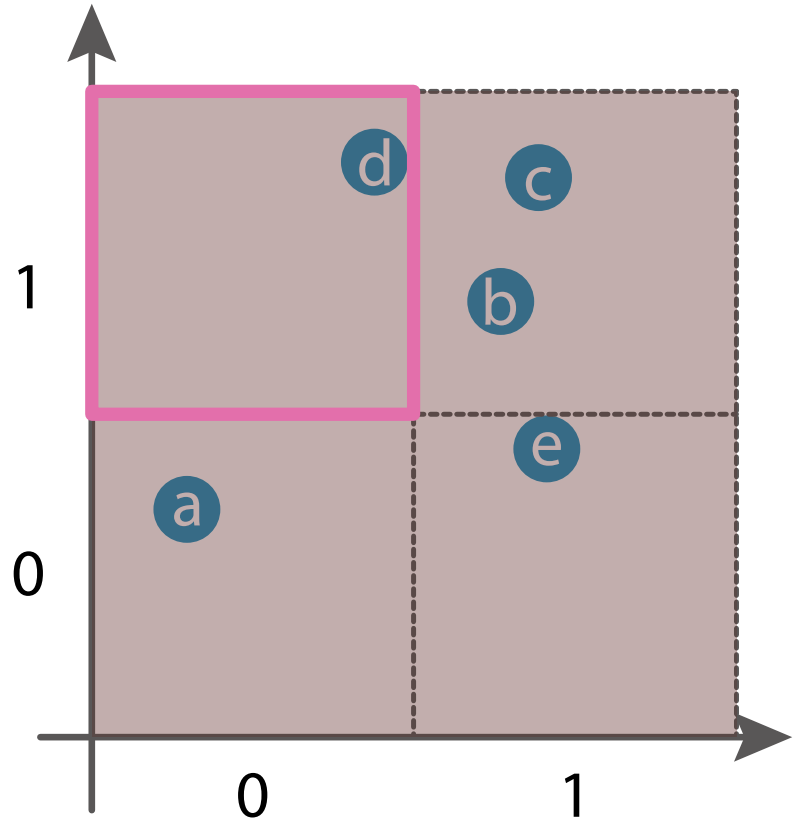
id	x	y
a	0	0
b	1	1
c	1	1
d	0	1
e	1	0



データをレベル 1 で離散化する

Discretization level: 1

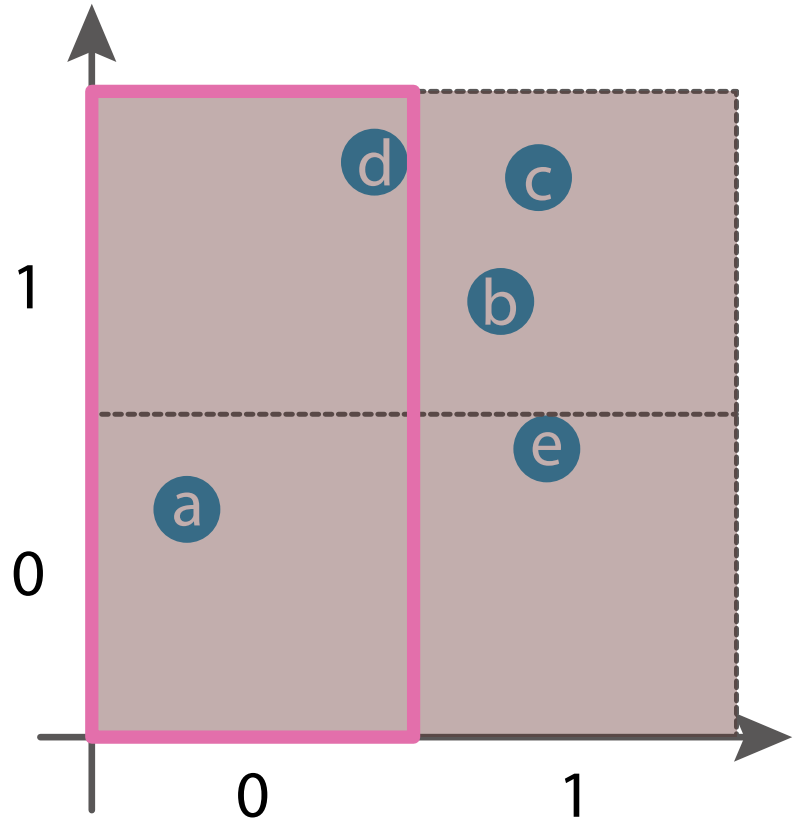
id	x	y
a	0	0
b	1	1
c	1	1
d	0	1
e	1	0



データをレベル 1 で離散化する

Discretization level: 1

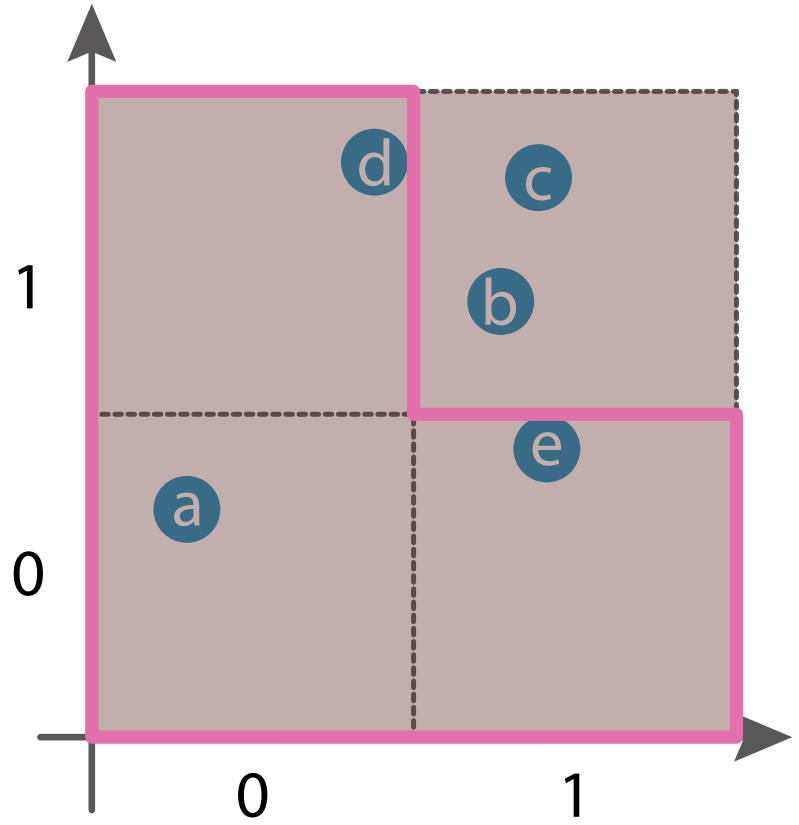
id	x	y
a	0	0
b	1	1
c	1	1
d	0	1
e	1	0



データをレベル 1 で離散化する

Discretization level: 1

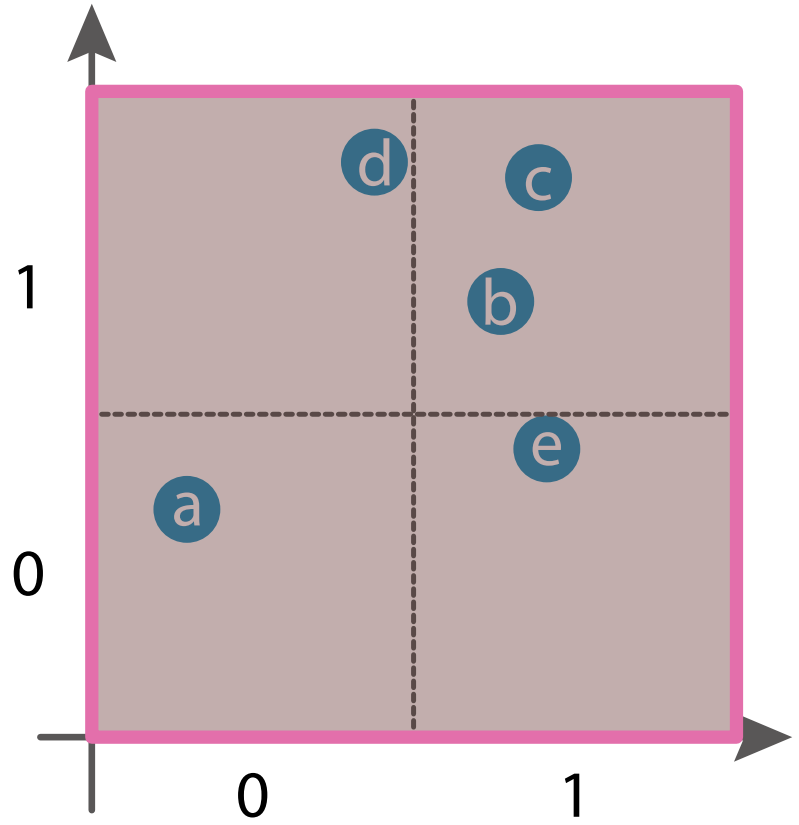
id	x	y
a	0	0
b	1	1
c	1	1
d	0	1
e	1	0



隣接クラスタを併合する

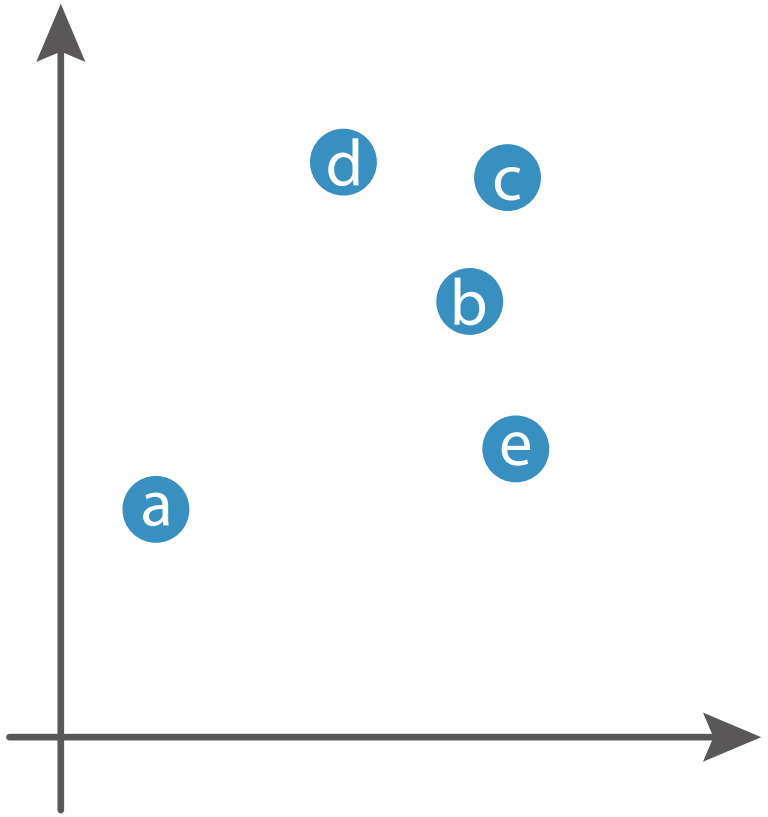
Discretization level: 1

id	*x*	*y*
a	0	0
b	1	1
c	1	1
d	0	1
e	1	0



次のレベルへ行く

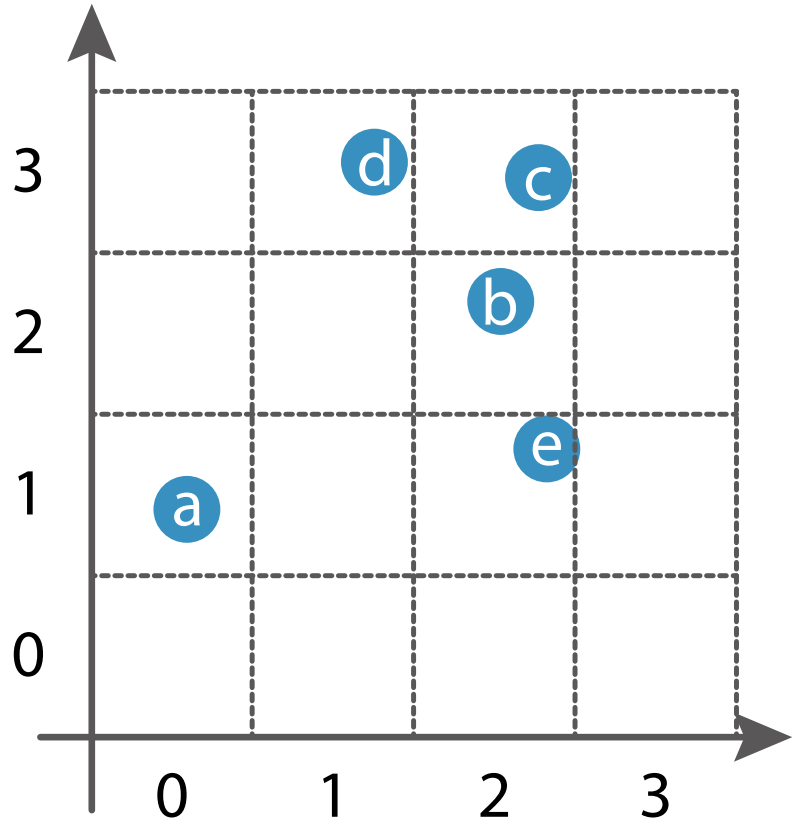
id	*x*	*y*
a	0.14	0.31
b	0.66	0.71
c	0.72	0.86
d	0.48	0.89
e	0.74	0.48



データをレベル 2 で離散化する

Discretization level: 2

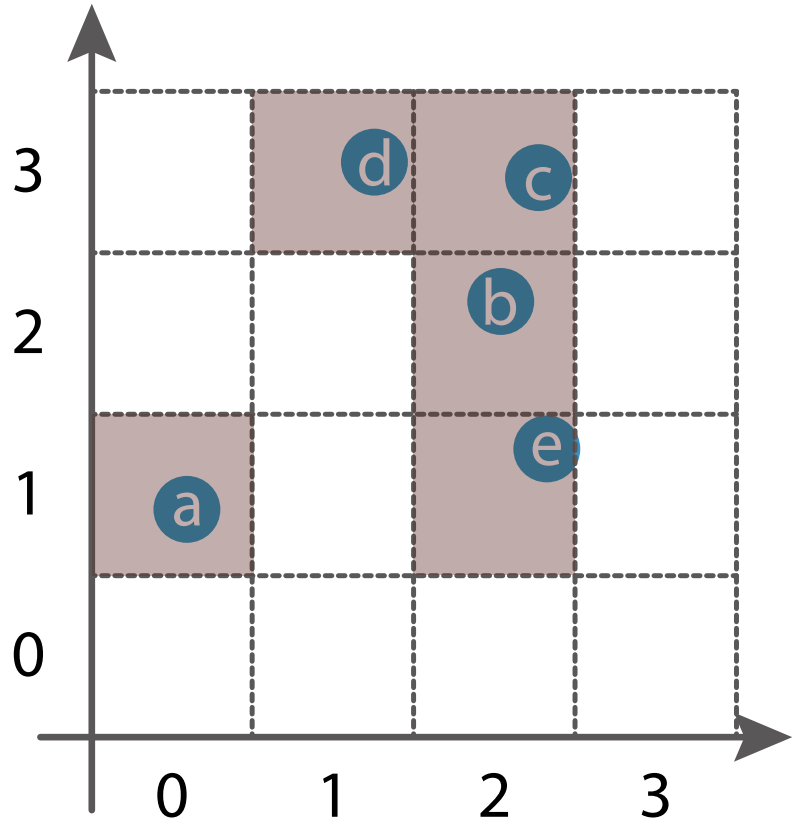
id	*x*	*y*
a	0	1
b	2	2
c	2	3
d	1	3
e	2	1



データをレベル 2 で離散化する

Discretization level: 2

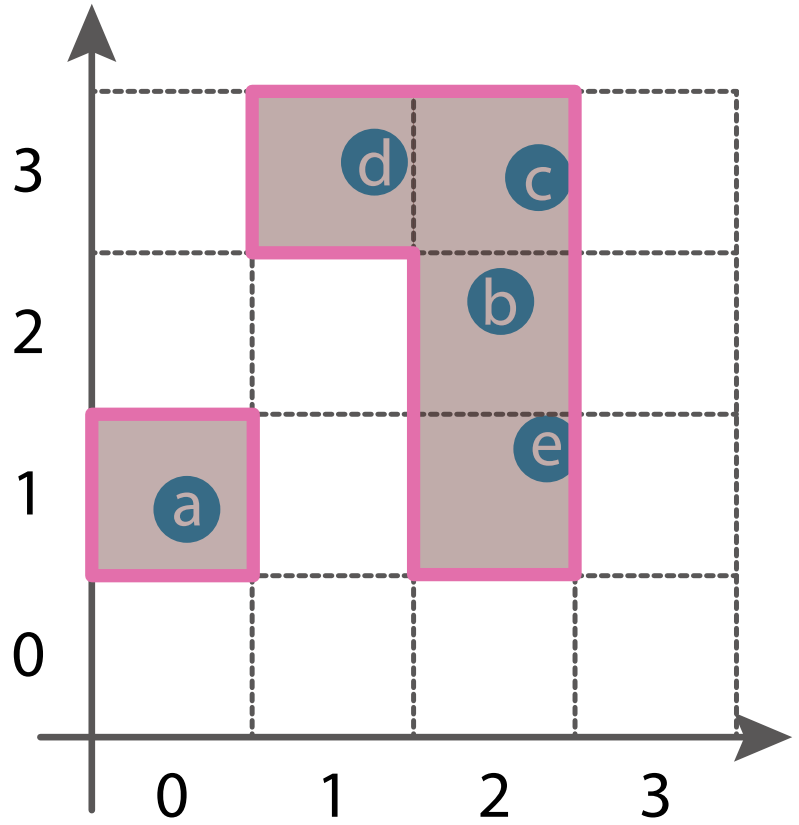
id	*x*	*y*
a	0	1
b	2	2
c	2	3
d	1	3
e	2	1



隣接クラスタを併合する

Discretization level: 2

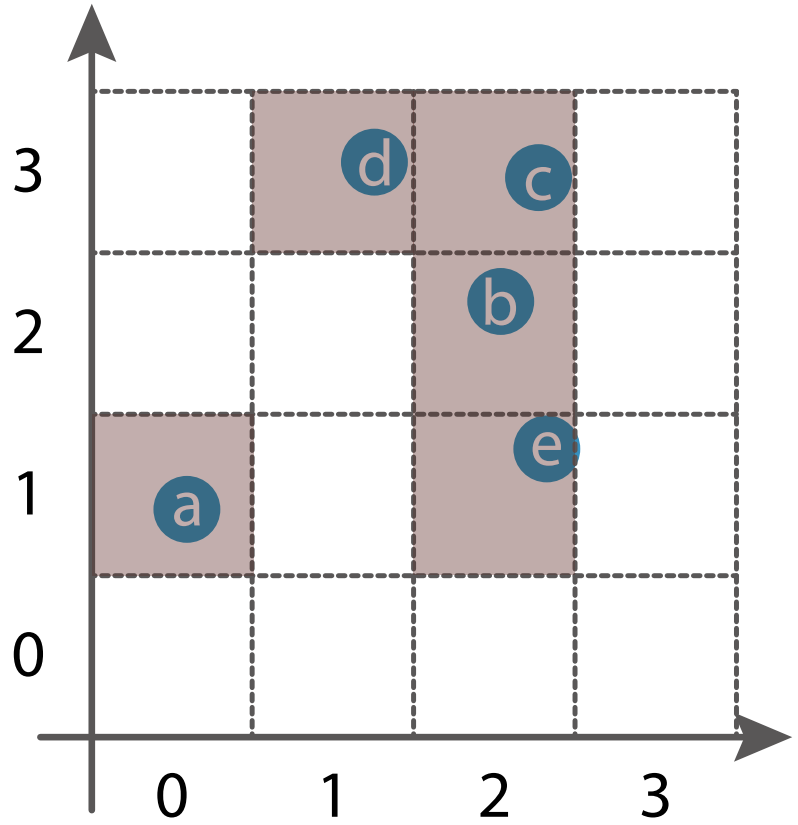
id	*x*	*y*
a	0	1
b	2	2
c	2	3
d	1	3
e	2	1



ソートによってクラスタを構築する

Discretization level: 2

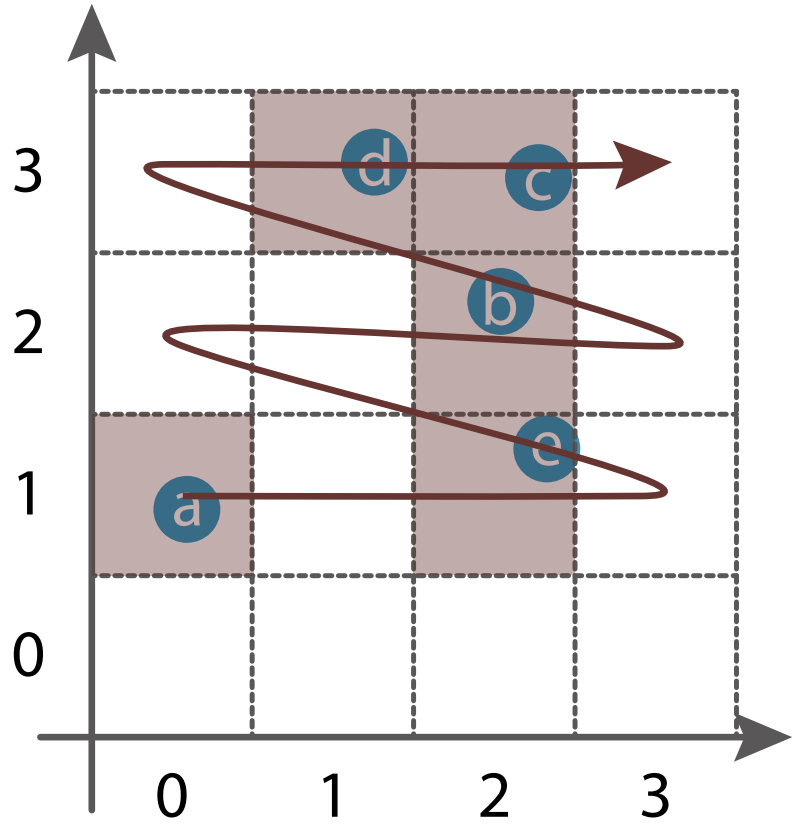
id	*x*	*y*
a	0	1
b	2	2
c	2	3
d	1	3
e	2	1



x 軸でソート → y 軸でソート

Discretization level: 2

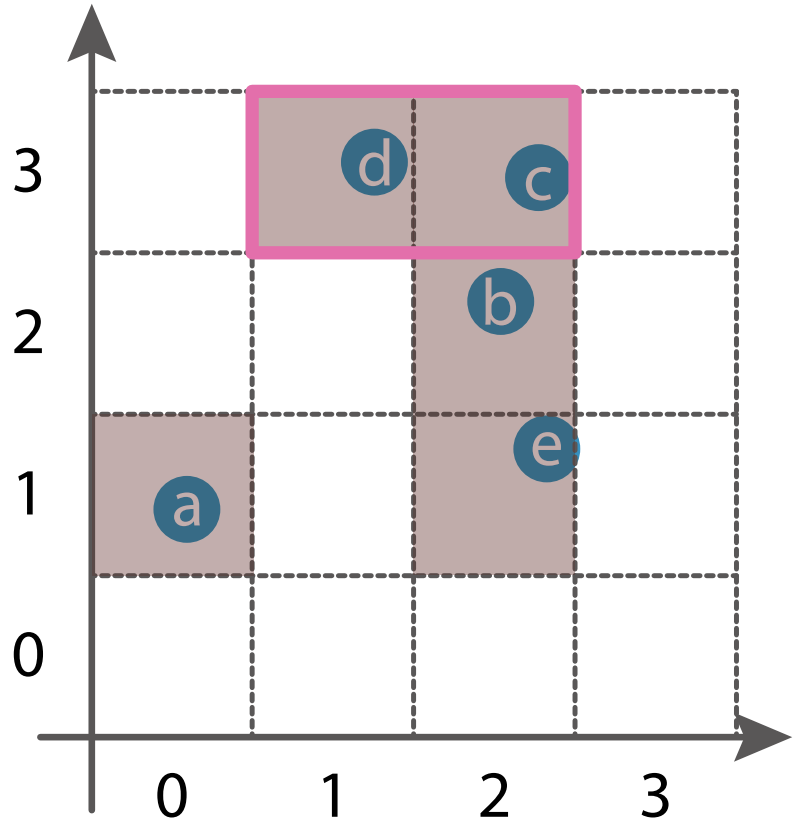
id	x	y
a	0	1
e	2	1
b	2	2
d	1	3
c	2	3



各点を次の点と比較

Discretization level: 2

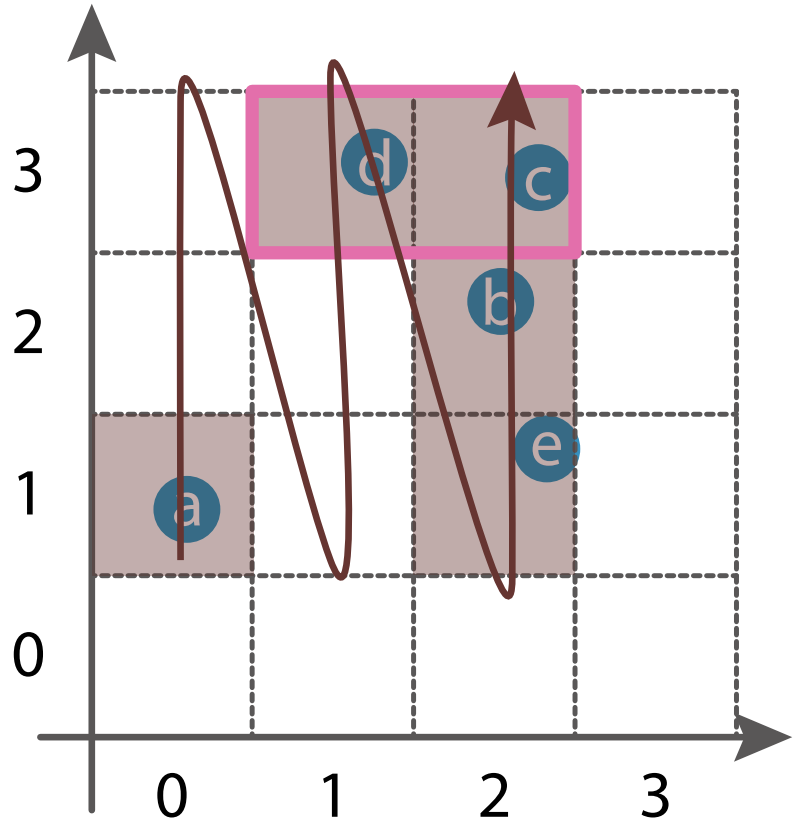
id	*x*	*y*
a	0	1
e	2	1
b	2	2
d	1	3
c	2	3



x 軸でソート

Discretization level: 2

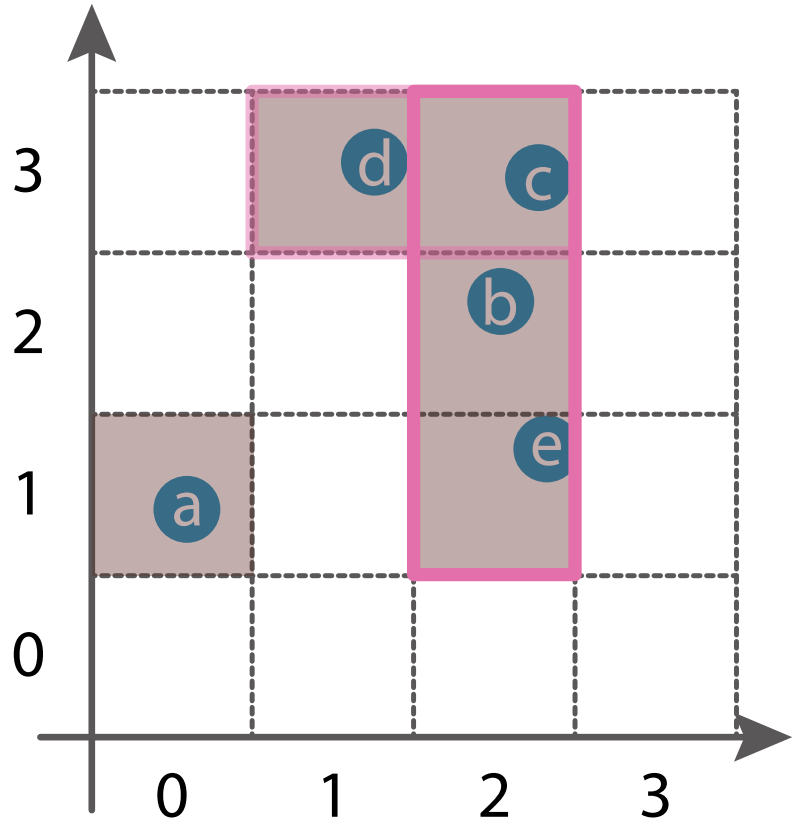
id	x	y
a	0	1
d	1	3
e	2	1
b	2	2
c	2	3



各点を次の点と比較

Discretization level: 2

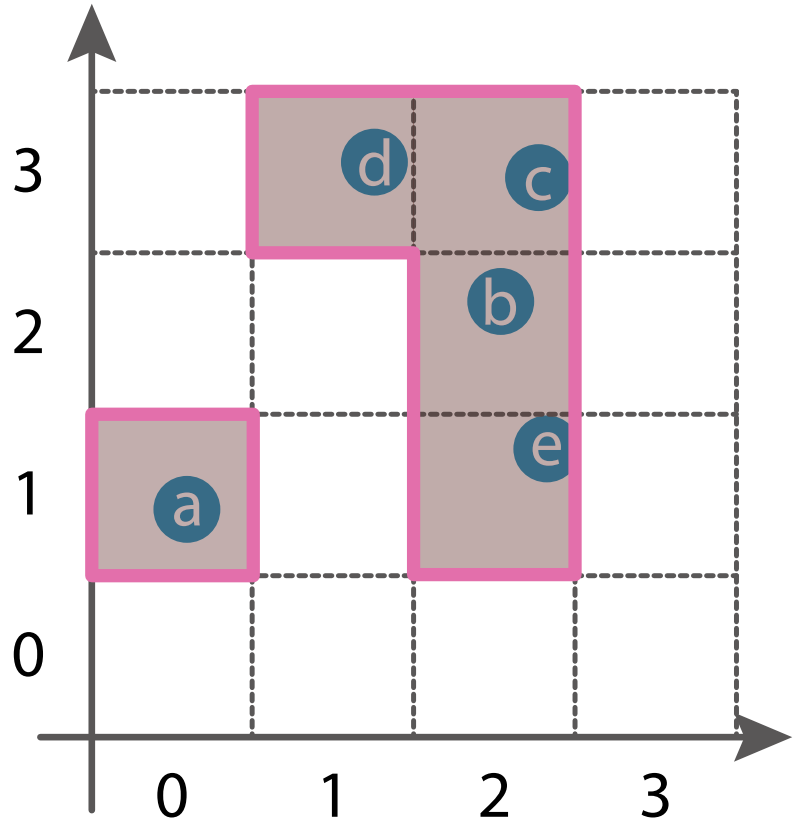
id	*x*	*y*
a	0	1
d	1	3
e	2	1
b	2	2
c	2	3



終了（... もしくは次のレベルへ）

Discretization level: 2

id	*x*	*y*
a	0	1
d	1	3
e	2	1
b	2	2
c	2	3



まとめ

- クラスタリングアルゴリズム BOOL を構築した
  - 任意形状を抽出可能なクラスタリングの
    アルゴリズムとして世界最速
  - 頑健，スケーラブル，ノイズに耐性あり
- 鍵となるアイデア：
  1. 2 進符号化による離散化
  2. データのソート

目次

- 第一部：理論
  - フラクタルとハウスドルフ距離を用いた図形の学習
    — 計算可能な 2 値分類を目指して
- 第二部：理論から実践
  - 符号化ダイバージェンス
  - MCL とグレイコードを用いたクラスタリング
  - 2 進符号化を活用した高速クラスタリング
- 第三部：形式概念と共に
  - 離散値・連続値混在データからの半教師あり学習
  - 半教師ありのマルチラベルクラス分類によるリガンド発見

離散値・連続値混在データからの半教師あり学習

- Sugiyama, M., Yamamoto, A.:
  Semi-Supervised Learning for Mixed-Type Data via Formal Concept Analysis, ICCS 2011, *LNAI* 6828
- Sugiyama, M., Yamamoto, A.:
  Semi-Supervised Learning on Closed Set Lattices, *Intelligent Data Analysis*, 17(3), 2013 (to appear)

概要・結果

- 半教師ありで順序学習をおこなうアルゴリズム
  SELF（SEmi-supervised Learning via FCA）を，
  形式概念解析（FCA）を用いて構築した

概要・結果

- 半教師ありで順序学習をおこなうアルゴリズム
  SELF（SEmi-supervised Learning via FCA）を，
  形式概念解析（FCA）を用いて構築した

- **主な貢献**
  1. 離散値・連続値混在データを扱うことができる
     - 混在データに対する初めての半教師あり学習手法
  2. 不完全なデータセットを扱うことができる
     - 欠損値と欠損ラベル
  3. クラス分類において良い精度

背景・問題点

- 多くのデータセットは離散値・連続値混在データ
  - しかし，混在データを直接扱える機械学習手法は少ない
    - 例：決定木（C4.5）など
- 多くのデータセットは不完全で，欠損値を持つ

背景・問題点

- 多くのデータセットは離散値・連続値混在データ
  - しかし，混在データを直接扱える機械学習手法は少ない
    - 例：決定木（C4.5）など
- 多くのデータセットは不完全で，欠損値を持つ
- 例：Horse-Colic Dataset （UCI）

```
2 1 530101 38.50 66 28 3 3 ? 2 5 4 4 ? ? ? 3 5 45.00 8.40 ? ? 2 2 11300 00000 00000 2
1 1 534817 39.2 88 20 ? ? 4 1 3 4 2 ? ? ? 4 2 50 85 2 2 3 2 02208 00000 00000 2
2 1 530334 38.30 40 24 1 1 3 1 3 3 1 ? ? ? 1 1 33.00 6.70 ? ? 1 2 00000 00000 00000 1
1 9 5290409 39.10 164 84 4 1 6 2 2 4 4 1 2 5.00 3 ? 48.00 7.20 3 5.30 2 1 02208 00000 00000 1
2 1 530255 37.30 104 35 ? ? 6 2 ? ? ? ? ? ? ? ? 74.00 7.40 ? ? 2 2 04300 00000 00000 2
2 1 528355 ? ? ? 2 1 3 1 2 3 2 2 1 ? 3 3 ? ? ? ? 1 2 00000 00000 00000 2
1 1 526802 37.90 48 16 1 1 1 1 3 3 3 1 1 ? 3 5 37.00 7.00 ? ? 1 1 03124 00000 00000 2
1 1 529607 ? 60 ? 3 ? ? 1 ? 4 2 2 1 ? 3 4 44.00 8.30 ? ? 2 1 02208 00000 00000 2
2 1 530051 ? 80 36 3 4 3 1 4 4 4 2 1 ? 3 5 38.00 6.20 ? ? 3 1 03205 00000 00000 2
2 9 5299629 38.30 90 ? 1 ? 1 1 5 3 1 2 1 ? 3 ? 40.00 6.20 1 2.20 1 2 00000 00000 00000 1
1 1 528548 38.10 66 12 3 3 5 1 3 3 1 2 1 3.00 2 5 44.00 6.00 2 3.60 1 1 02124 00000 00000 1
2 1 527927 39.10 72 52 2 ? 2 1 2 1 2 1 1 ? 4 4 50.00 7.80 ? ? 1 1 02111 00000 00000 2
1 1 528031 37.20 42 12 2 1 1 1 3 3 3 3 1 ? 4 5 ? 7.00 ? ? 1 2 04124 00000 00000 2
2 9 5291329 38.00 92 28 1 1 2 1 1 3 2 3 ? 7.20 1 1 37.00 6.10 1 ? 2 2 00000 00000 00000 1
```

背景・問題点

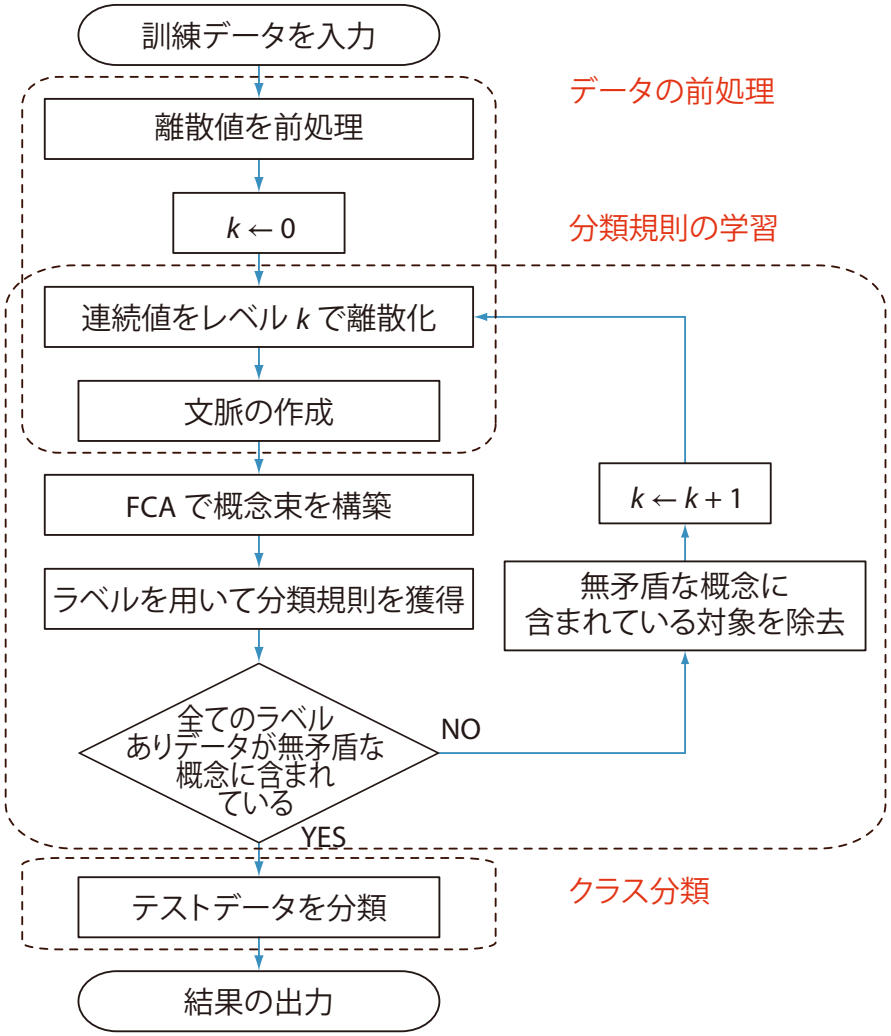
- 多くのデータセットは離散値・連続値混在データ
  - しかし，混在データを直接扱える機械学習手法は少ない
    - 例：決定木（C4.5）など
- 多くのデータセットは不完全で，欠損値を持つ
- データのラベルづけはコストが高い（お金・時間など）
  - 半教師あり学習：訓練でラベル無しデータを効果的に使う
  - しかし，混在データを直接扱う手法が存在しない

解決策

- 多くのデータセットは離散値・連続値混在データ
  - しかし，混在データを直接扱える機械学習手法は少ない
    - 例：決定木（C4.5）など
- 多くのデータセットは不完全で，欠損値を持つ
- データのラベルづけはコストが高い（お金・時間など）
  - 半教師あり学習：訓練でラベル無しデータを効果的に使う
  - しかし，混在データを直接扱う手法が存在しない

- 形式概念解析（FCA）を用いてこれらの問題を解決する
  - 頻出パターン発見で用いられている[Pasquier *et al.*, 1999]
    - FCA で得られる頻出飽和パターンは「可逆」圧縮に相当
  - 高速なアルゴリズム LCM [Uno *et al.*, 2005] が手に入る

SELF のフローチャート



SELF による学習

- 以下のデータセットから学習する（⊥ は欠損値）

H		1	2	3	ラベル
X	x_1	**T**	C	0.28	1
	x_2	**F**	A	0.54	1
	x_3	**T**	B	⊥	⊥
	x_4	**F**	A	0.79	2
	x_5	**T**	C	0.81	⊥

データ前処理（1/3）

- 以下のデータセットから学習する（⊥ は欠損値）

H		1	2	3	ラベル
X	x_1	**T**	C	0.28	1
	x_2	**F**	A	0.54	1
	x_3	**T**	B	⊥	⊥
	x_4	**F**	A	0.79	2
	x_5	**T**	C	0.81	⊥

- データ前処理で（形式）文脈を作る

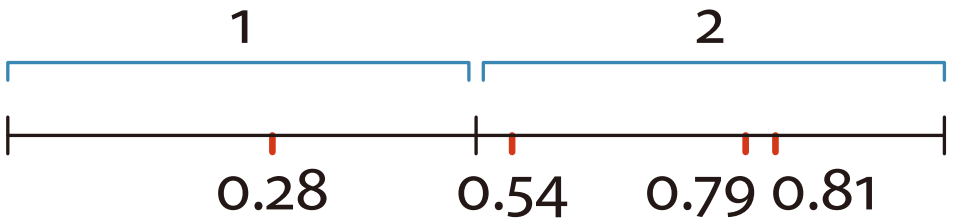
	1.**T**	1.**F**	2.A	2.B	2.C	3.1	3.2
x_1	×				×		
x_2		×	×				
x_3	×			×			
x_4		×	×				
x_5	×				×		

データ前処理（2/3）

- 以下のデータセットから学習する（⊥ は欠損値）

H		1	2	3	ラベル
X	x_1	**T**	C	0.28	1
	x_2	**F**	A	0.54	1
	x_3	**T**	B	⊥	⊥
	x_4	**F**	A	0.79	2
	x_5	**T**	C	0.81	⊥

- 連続量を2 進符号化で離散化する



データ前処理（3/3）

- 以下のデータセットから学習する（⊥ は欠損値）

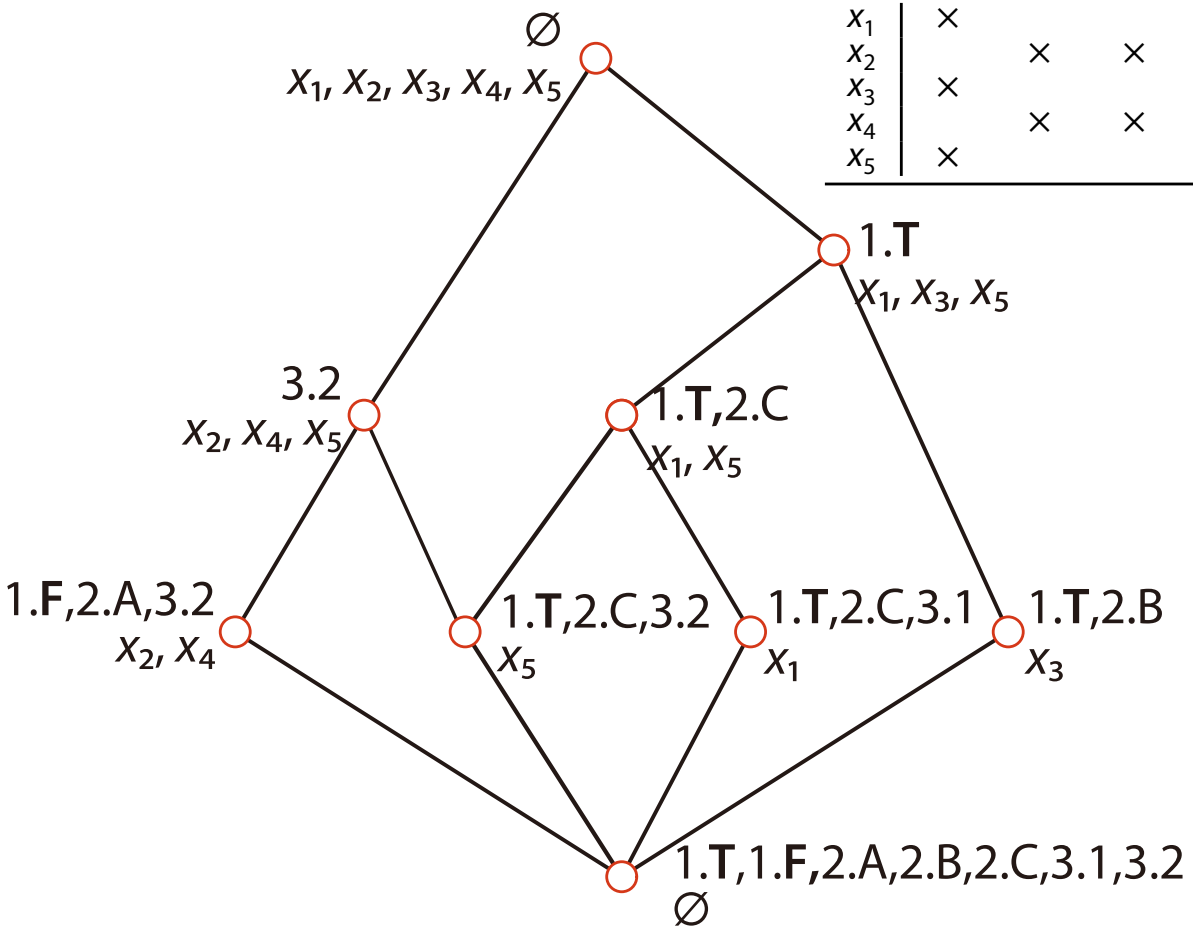
H		1	2	3	ラベル
X	x_1	**T**	C	0.28	1
	x_2	**F**	A	0.54	1
	x_3	**T**	B	⊥	⊥
	x_4	**F**	A	0.79	2
	x_5	**T**	C	0.81	⊥

- データ前処理で（形式）文脈を作る

	1.**T**	1.**F**	2.A	2.B	2.C	3.1	3.2
x_1	×				×	×	
x_2		×	×				×
x_3	×			×			
x_4		×	×				×
x_5	×				×		×

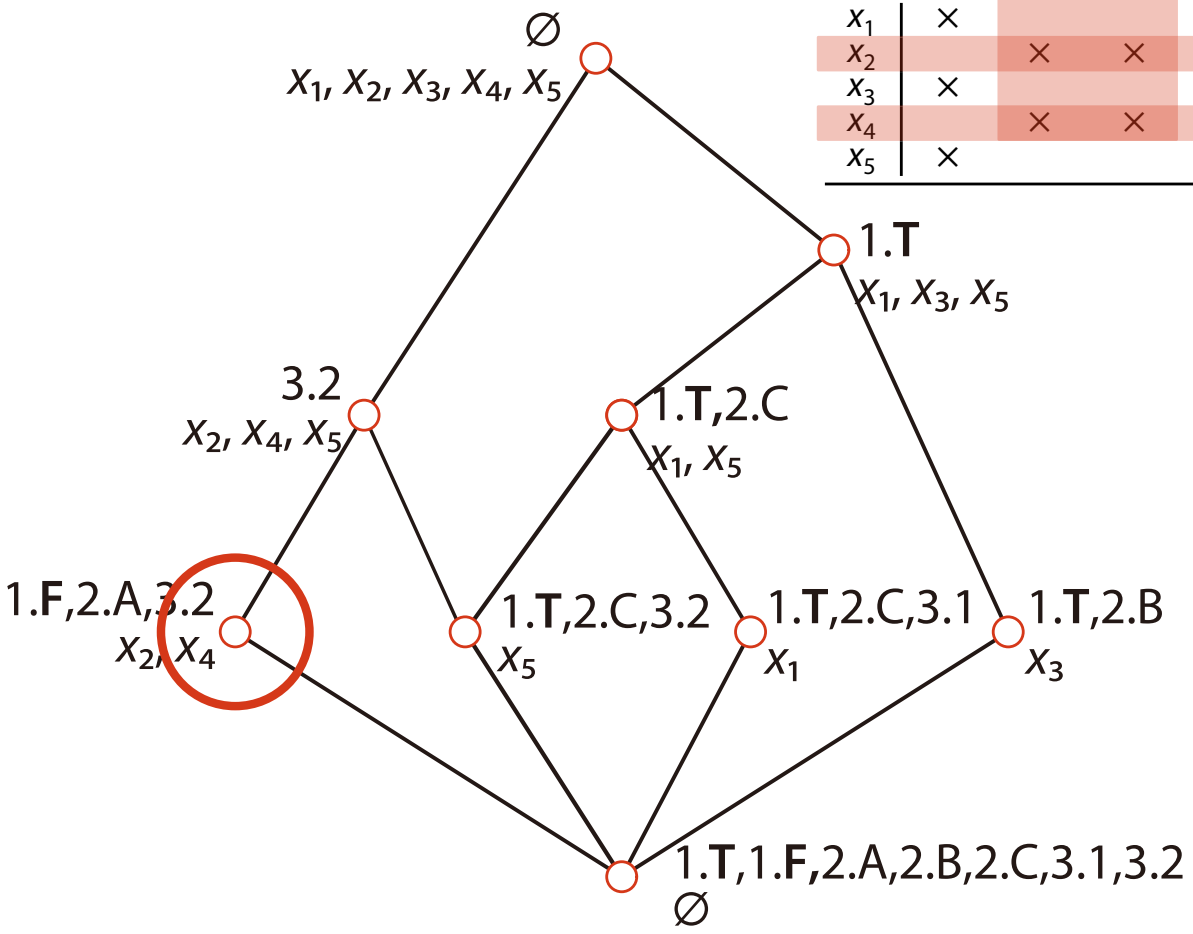
FCA で概念束を作る

	1.T	1.F	2.A	2.B	2.C	3.1	3.2
x_1	×				×	×	
x_2		×	×				×
x_3	×			×			
x_4		×	×				×
x_5	×				×		×



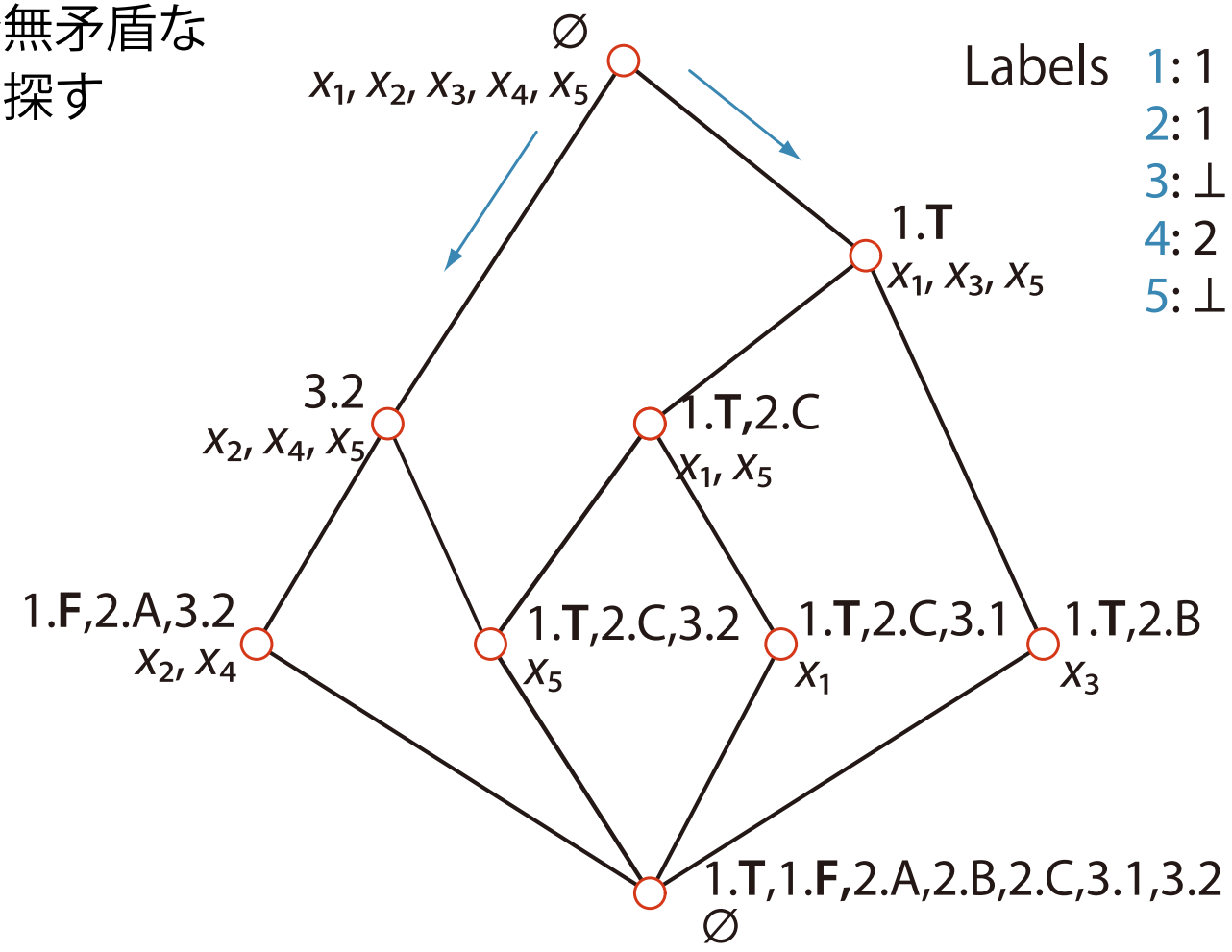
FCA で概念束を作る

	1.T	1.F	2.A	2.B	2.C	3.1	3.2
x_1	×				×	×	
x_2		×	×				×
x_3	×			×			
x_4		×	×				×
x_5	×				×		×



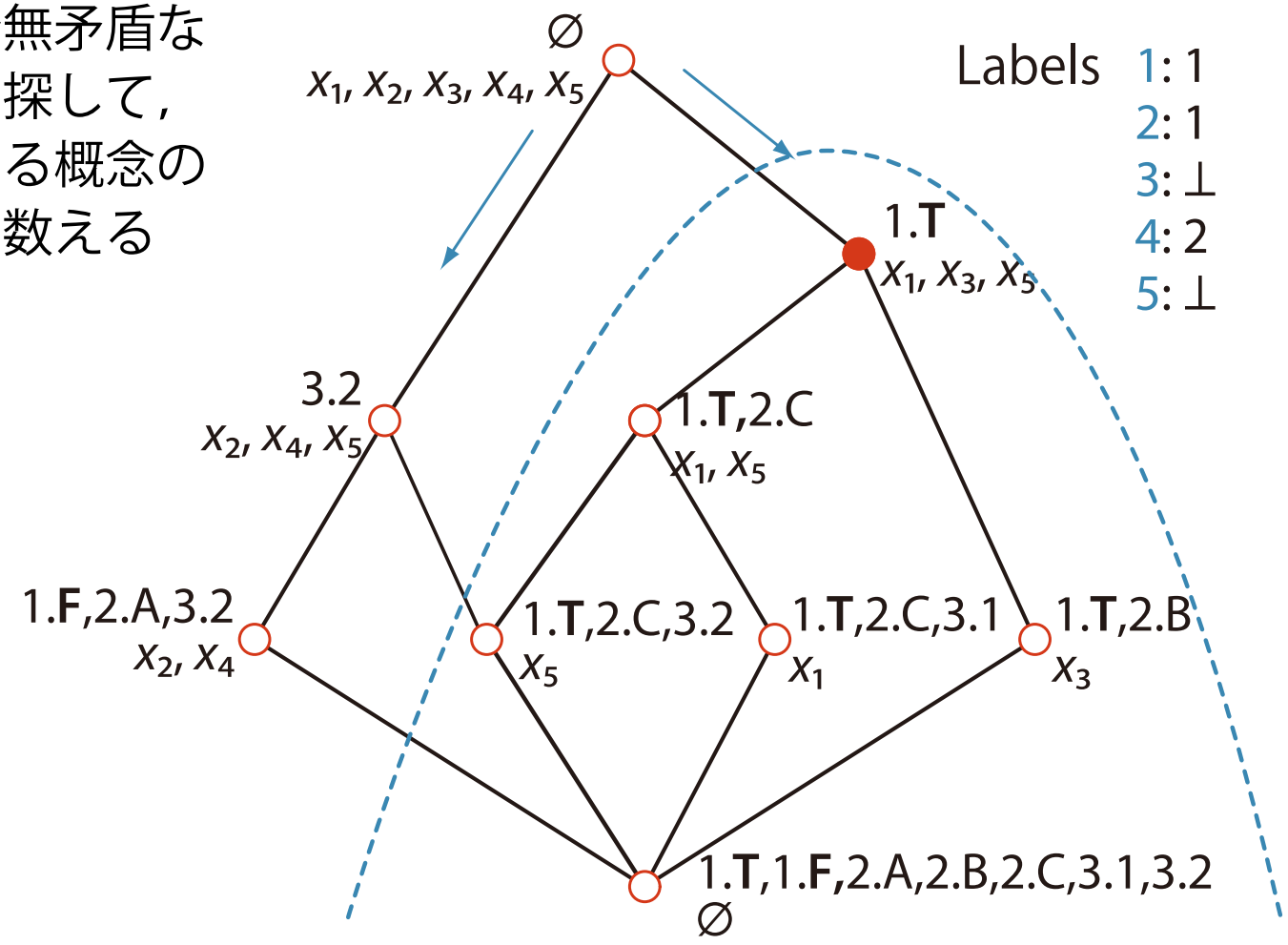
分類規則を学習する

極大で無矛盾な
概念を探す



分類規則を学習する

極大で無矛盾な
概念を探して，
下にある概念の
個数を数える



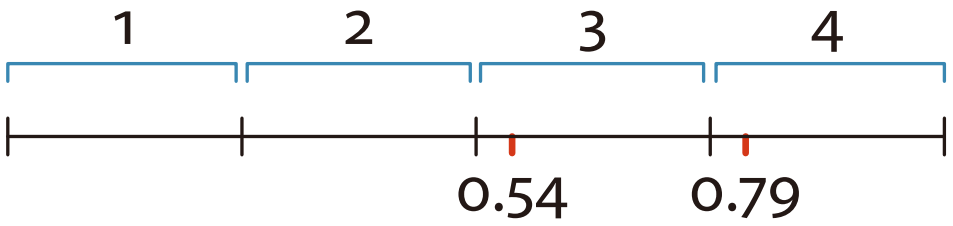
データ前処理

- 残りのデータから学習する

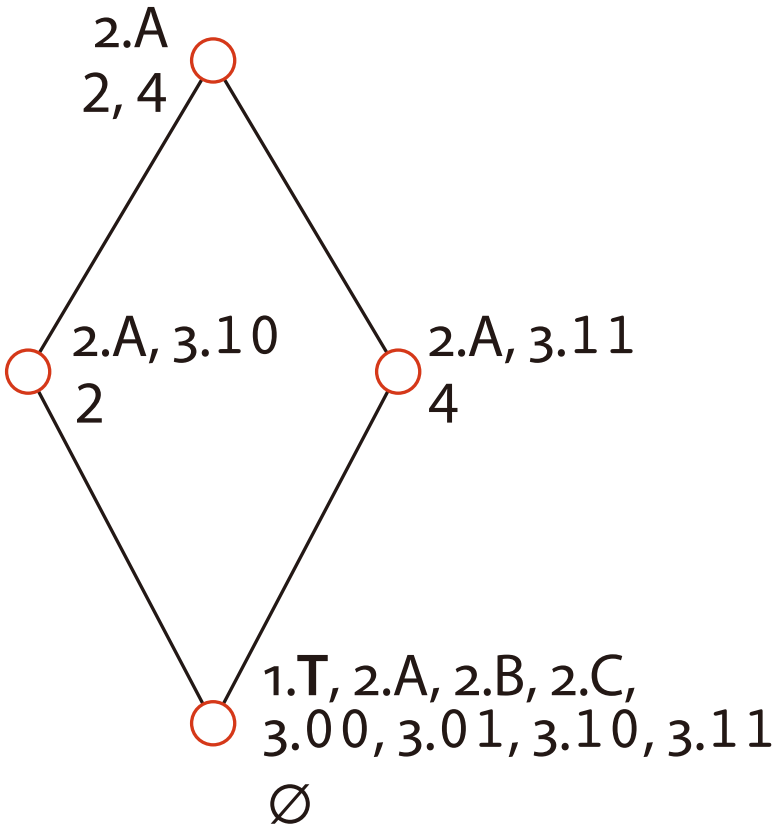
H		1	2	3	ラベル
X	x_2	**F**	A	0.54	1
	x_4	**F**	A	0.79	2

- 離散化を精密化して，以下の文脈を得る

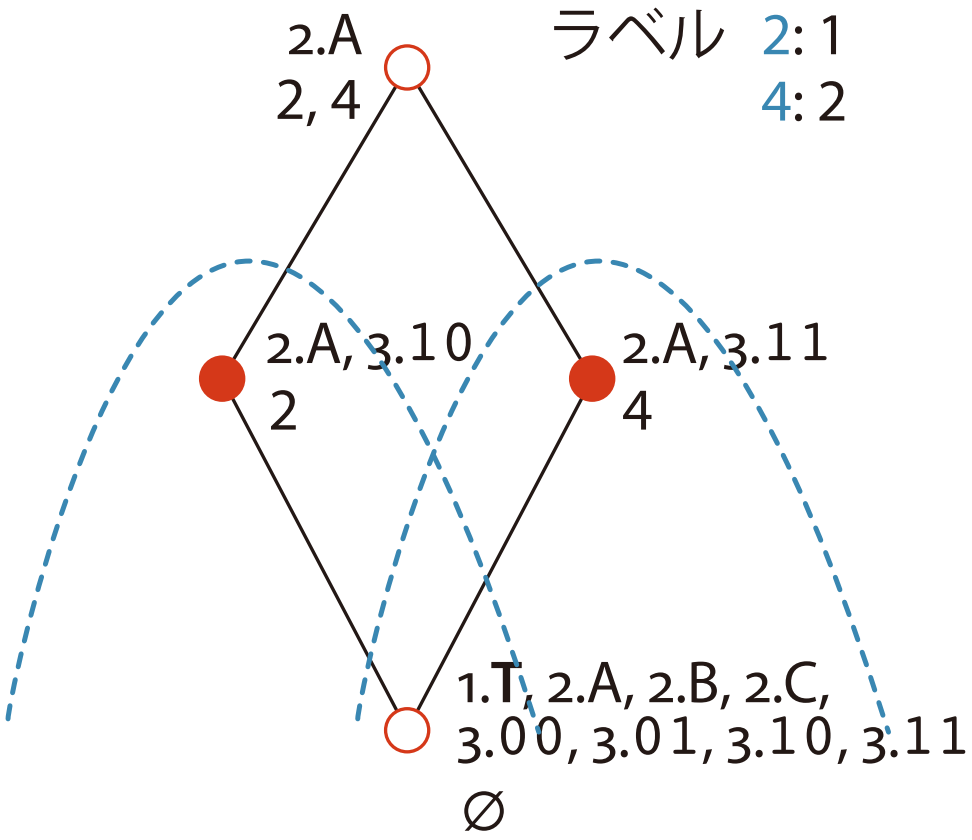
	1.T	1.F	2.A	2.B	2.C	3.1	3.2	3.3	3.4
x_2		×	×					×	
x_4		×	×						×



FCA で形式概念を作る



分類規則を学習する



ラベル無しデータを分類する

- 各離散化レベルで以下の分類規則とその重みを得た

  $\mathscr{R}_1 = \{(\{1.\mathbf{T}\}, 1)\}$, $\omega(\{1.\mathbf{T}\}, 1) = 6$

  （属性 1.T を持っているデータをクラス 1 に分類する）

  $\mathscr{R}_2 = \{(\{1.\mathbf{F}, 2.\mathrm{A}, 3.3\}, 1), (\{1.\mathbf{F}, 2.\mathrm{A}, 3.4\}, 2)\}$

  $\omega(\{1.\mathbf{F}, 2.\mathrm{A}, 3.3\}, 1) = 2$, $\omega(\{1.\mathbf{F}, 2.\mathrm{A}, 3.4\}, 2) = 2$

- 各重みを足し合わせることで，ラベルの順序学習を実現する
- ラベル $\lambda \in \mathscr{L}$ に対して，その *l*-preference を
  以下のように定義する

$$\mathrm{lPref}(\lambda) := \sum_{k=1}^{K} \sum_{R \in \mathscr{Q}} \omega(R), \text{ where}$$

$$\mathscr{Q} = \{(B, \lambda) \in \mathscr{R}_k \mid (y, b) \in I \text{ for all } b \in B\},$$

ラベル無しデータを分類する

- 各離散化レベルで以下の分類規則とその重みを得た

  $\mathscr{R}_1 = \{(\{1.\mathbf{T}\}, 1)\}$, $\omega(\{1.\mathbf{T}\}, 1) = 6$

  （属性 1.T を持っているデータをクラス 1 に分類する）

  $\mathscr{R}_2 = \{(\{1.\mathbf{F}, 2.\mathrm{A}, 3.3\}, 1), (\{1.\mathbf{F}, 2.\mathrm{A}, 3.4\}, 2)\}$

  $\omega(\{1.\mathbf{F}, 2.\mathrm{A}, 3.3\}, 1) = 2$, $\omega(\{1.\mathbf{F}, 2.\mathrm{A}, 3.4\}, 2) = 2$

- 例：
  - データ(**T**, B, 0.45)に対して，

    $\mathrm{IPref}(1) = 6$, $\mathrm{IPref}(2) = 0$
  - データ(**F**, A, 0.82)に対して，

    $\mathrm{IPref}(1) = 0$, $\mathrm{IPref}(2) = 2$

実験手法

1. 多クラス分類で SELF と他の手法を比較
2. ラベルのランキングで SELF を評価
   - どちらの場合も，訓練にラベル無しデータを使った場合と使わなかった場合を両方実験する

- SELF はR 2.11.1で実装
- 全ての概念の列挙にLCM [Uno *et al.*, 2005]（NII 宇野さん）を使用
  - 最速のアルゴリズムとして知られている
- 多クラス分類では R で実装されてる決定木，SVM（RBF カーネル），kNN（$k = 1, 5$）と比較
  - 混在データを扱えるのは決定木のみなので，他の手法は離散値をそのまま連続値とみなして適用

データセット（UCI から取得）

Name	# Data	# Classes	# Features	
			Discrete	Continuous
abalone	4177	28	1	7
allbp	2800	3	2	3
anneal	798	5	28	10
arrhythmia	452	13	5	5
australian	690	2	7	4
crx	690	2	9	6
echocardiogram	131	2	1	7
heart	270	2	7	6
hepatitis	155	2	13	6
horse colic	368	2	8	2

データセット（UCI から取得）

Name	# Data	# Classes	# Features	
			Discrete	Continuous
abalone	4177	28	1	7
allbp	2800	3	2	3
anneal	798	5	28	10
arrhythmia	452	13	5	5

- クラス数が 3 以上のものをランキングの評価に使用

ラベルのランキングの評価手法

- 文献[Cheng *et al.*, 2010]に倣って correctnessとcompletenessを採用
  - Correctness は *gamma rank correlation* [Goodman and Kruskal, 1979] と同じ
    - 正しくランク付けされたペアの数とそうでないペアの数の差を正規化したもの
- 以下のように C と D を定義

$$C := \#\{(\lambda, \lambda') \in \mathscr{L} \times \mathscr{L} \mid \lambda \prec \lambda' \text{ かつ } \lambda \prec_* \lambda'\},$$

$$D := \#\{(\lambda, \lambda') \in \mathscr{L} \times \mathscr{L} \mid \lambda \prec \lambda' \text{ かつ } \lambda' \prec_* \lambda\}.$$

  - $\prec$ は予測された順序，$\prec_*$ は正解の順序

ラベルのランキングの評価手法

- Correctness:

$$\mathrm{CR}(\prec, \prec_*) := \frac{C - D}{C + D}$$

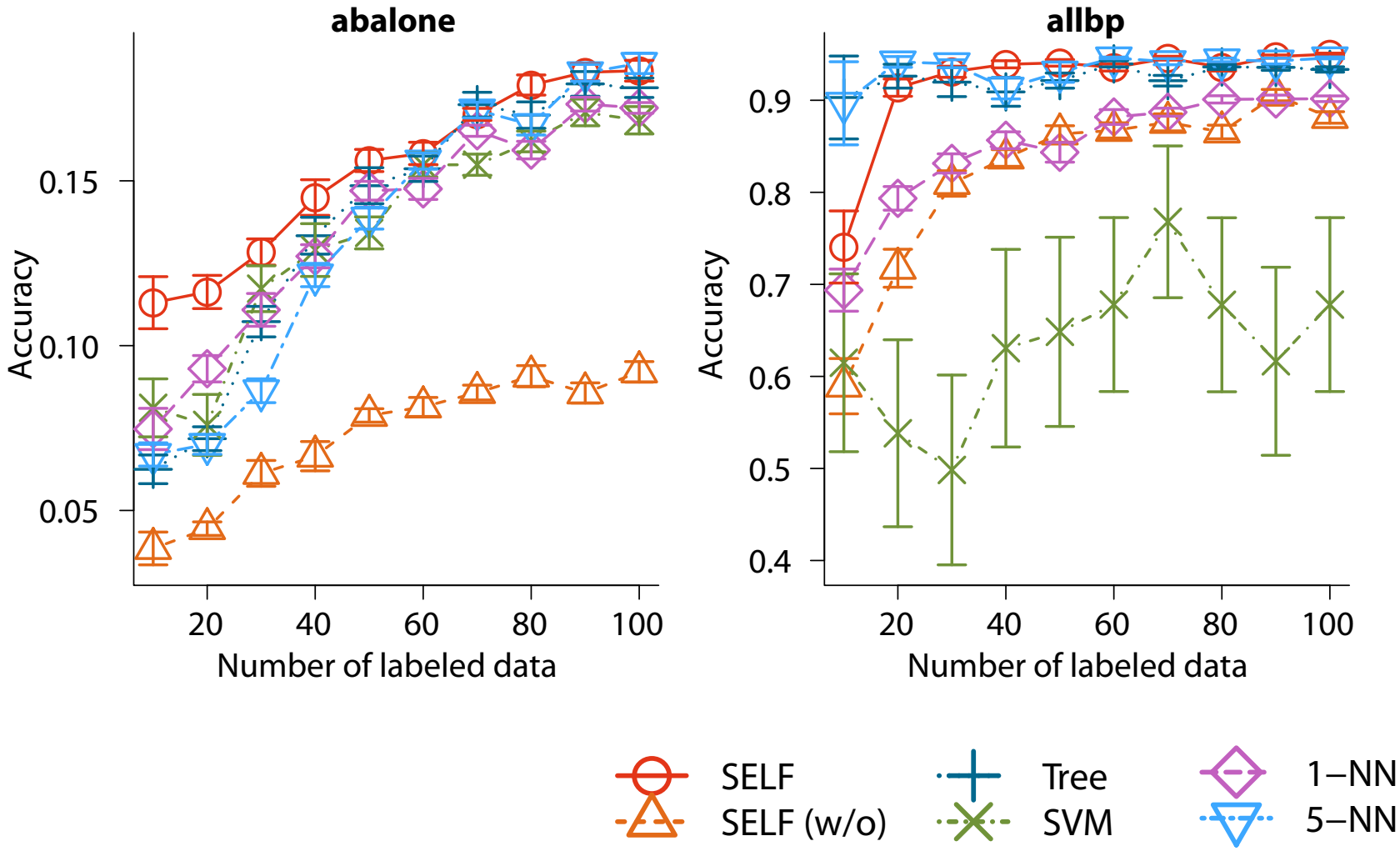
  - -1 から 1 までの値を取り，大きほど良い

- Completeness:

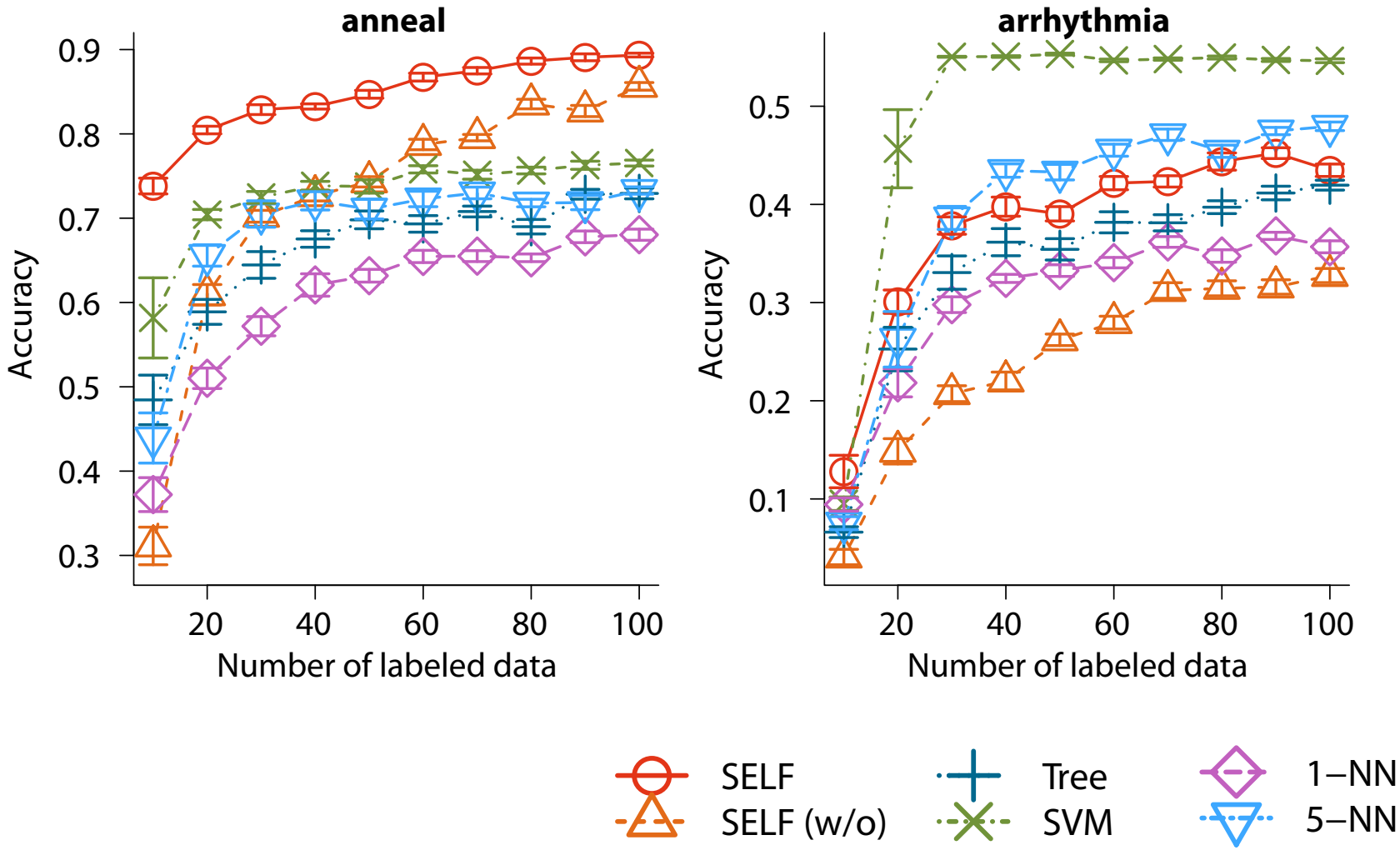
$$\mathrm{CP}(\prec) := \frac{C + D}{\#\{(\lambda, \lambda') \in \mathscr{L} \times \mathscr{L} \mid \lambda \prec_* \lambda' \text{ または } \lambda' \prec_* \lambda\}}$$

  - 0 から 1 までの値を取り，大きいほどよい

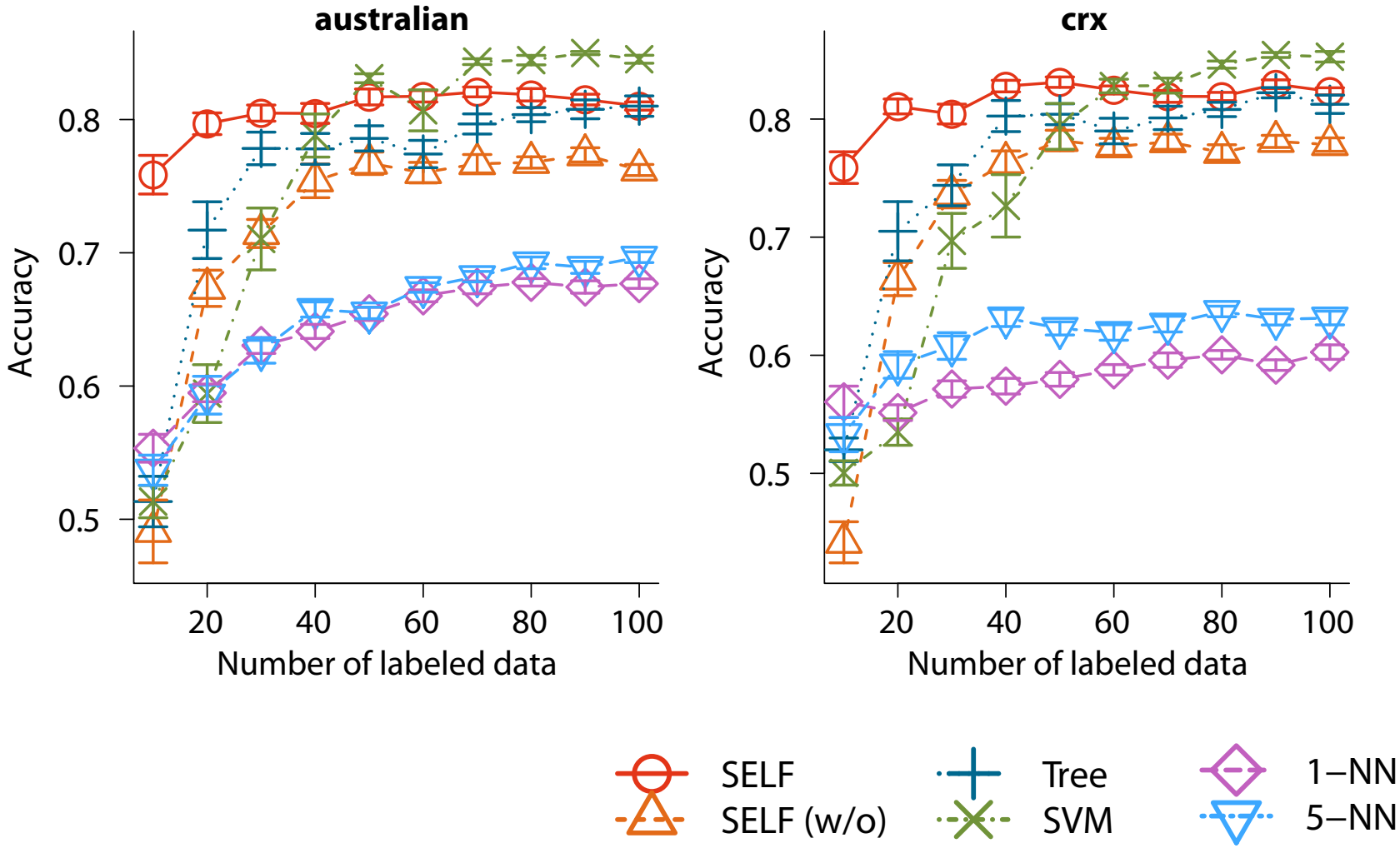
クラス分類の実験結果


abalone
Accuracy
0.05
0.10
0.15
20
40
60
80
100
Number of labeled data
allbp
Accuracy
0.4
0.5
0.6
0.7
0.8
0.9
20
40
60
80
100
Number of labeled data
SELF
SELF (w/o)
Tree
SVM
1–NN
5–NN

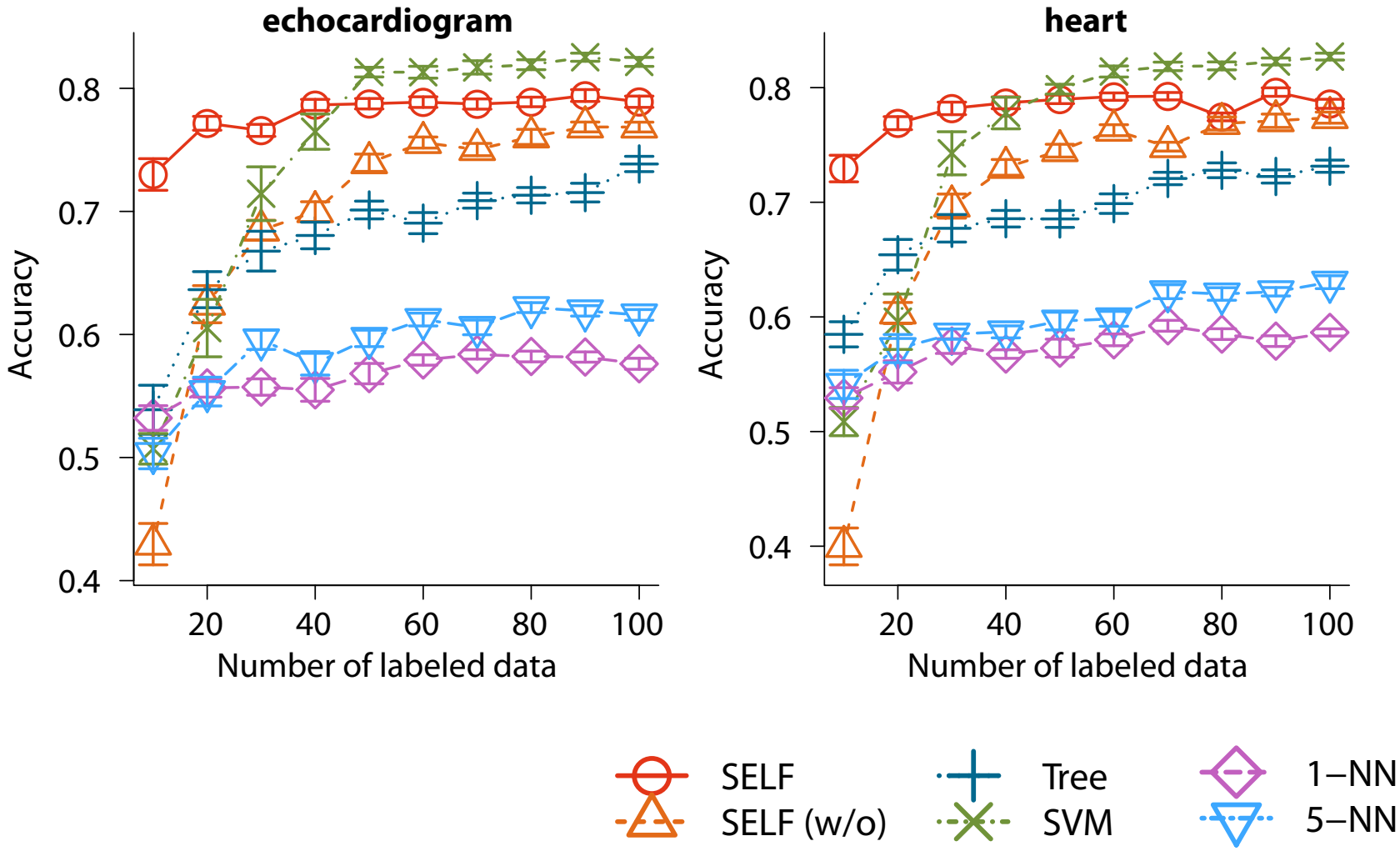
クラス分類の実験結果


anneal
Accuracy
0.9
0.8
0.7
0.6
0.5
0.4
0.3
20
40
60
80
100
Number of labeled data
arrhythmia
Accuracy
0.5
0.4
0.3
0.2
0.1
20
40
60
80
100
Number of labeled data
SELF
SELF (w/o)
Tree
SVM
1–NN
5–NN

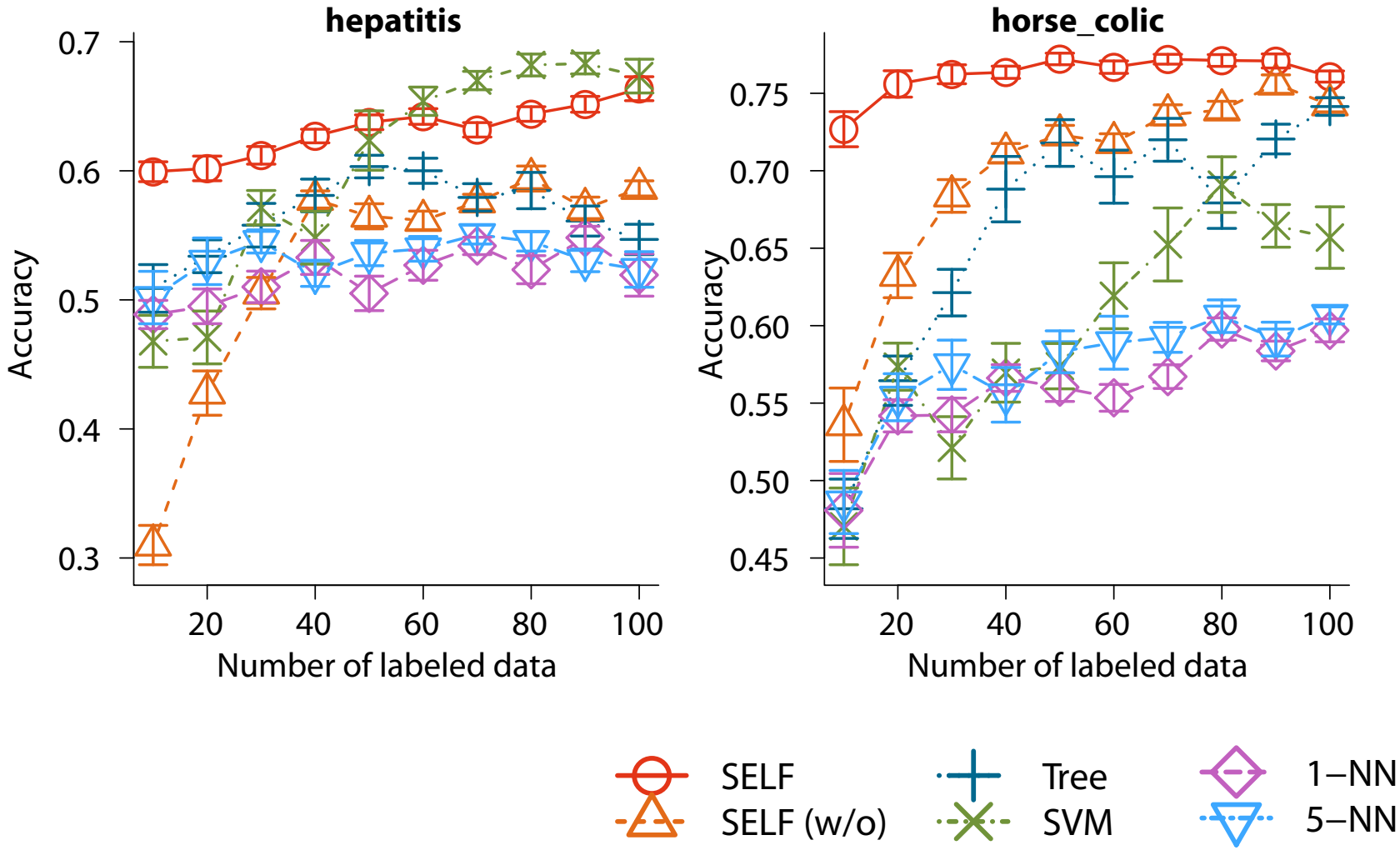
クラス分類の実験結果


australian
crx
Accuracy
Number of labeled data
0.5
0.6
0.7
0.8
20
40
60
80
100
SELF
SELF (w/o)
Tree
SVM
1–NN
5–NN

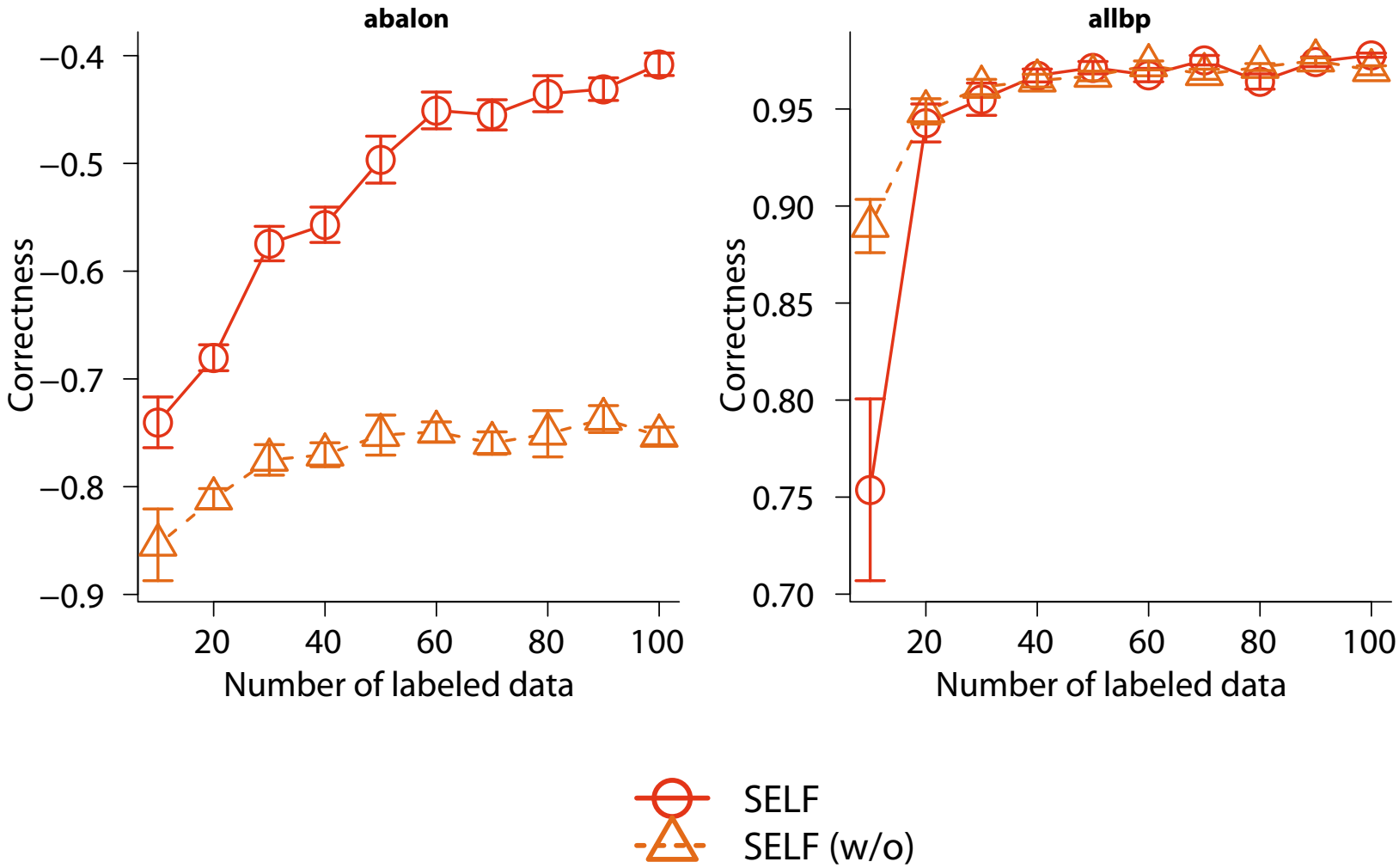
クラス分類の実験結果


echocardiogram
heart
Accuracy
Number of labeled data
0.4
0.5
0.6
0.7
0.8
20
40
60
80
100
SELF
SELF (w/o)
Tree
SVM
1–NN
5–NN

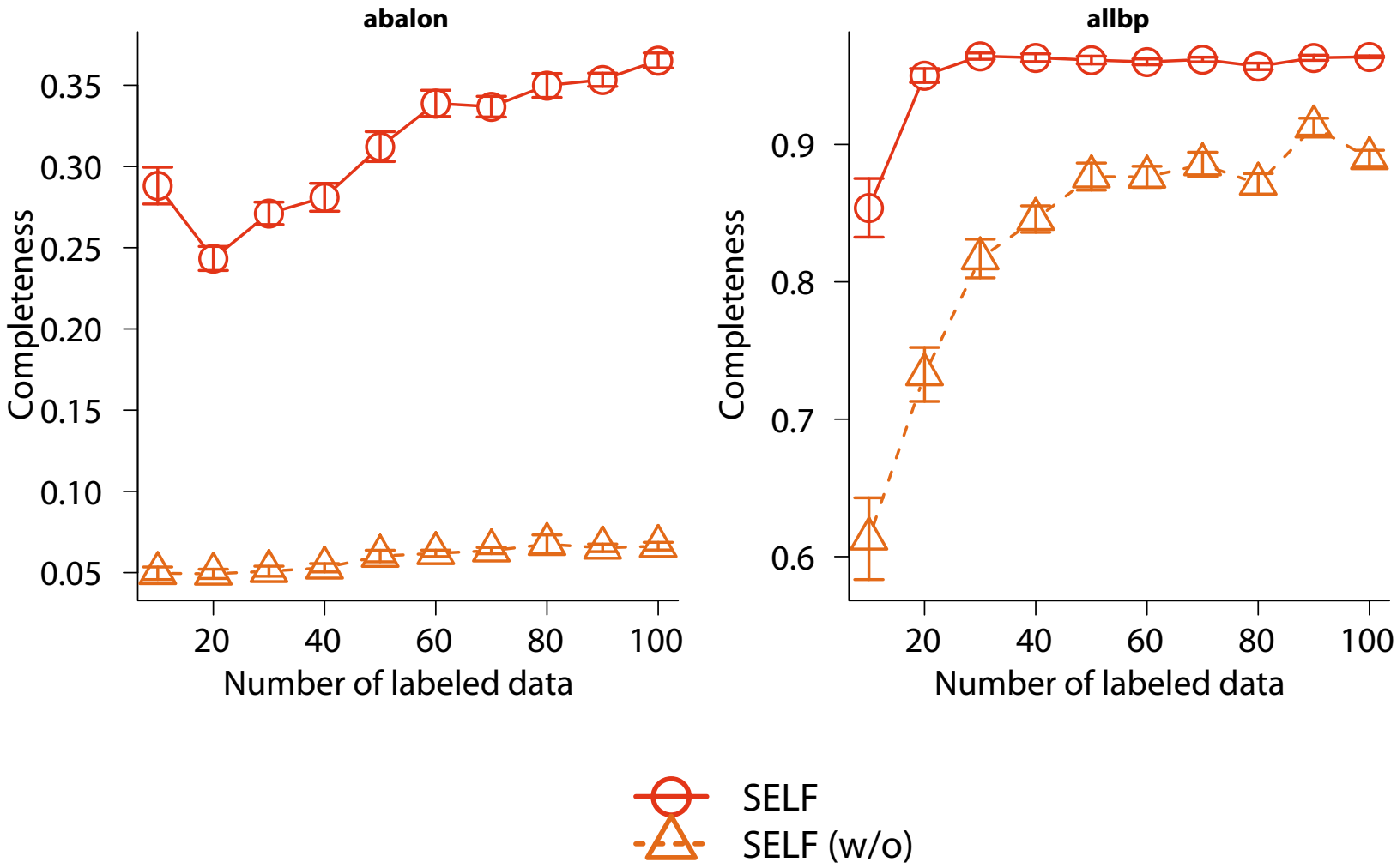
クラス分類の実験結果


hepatitis
horse_colic
Accuracy
Number of labeled data
0.7
0.6
0.5
0.4
0.3
0.75
0.70
0.65
0.60
0.55
0.50
0.45
20
40
60
80
100
SELF
SELF (w/o)
Tree
SVM
1–NN
5–NN

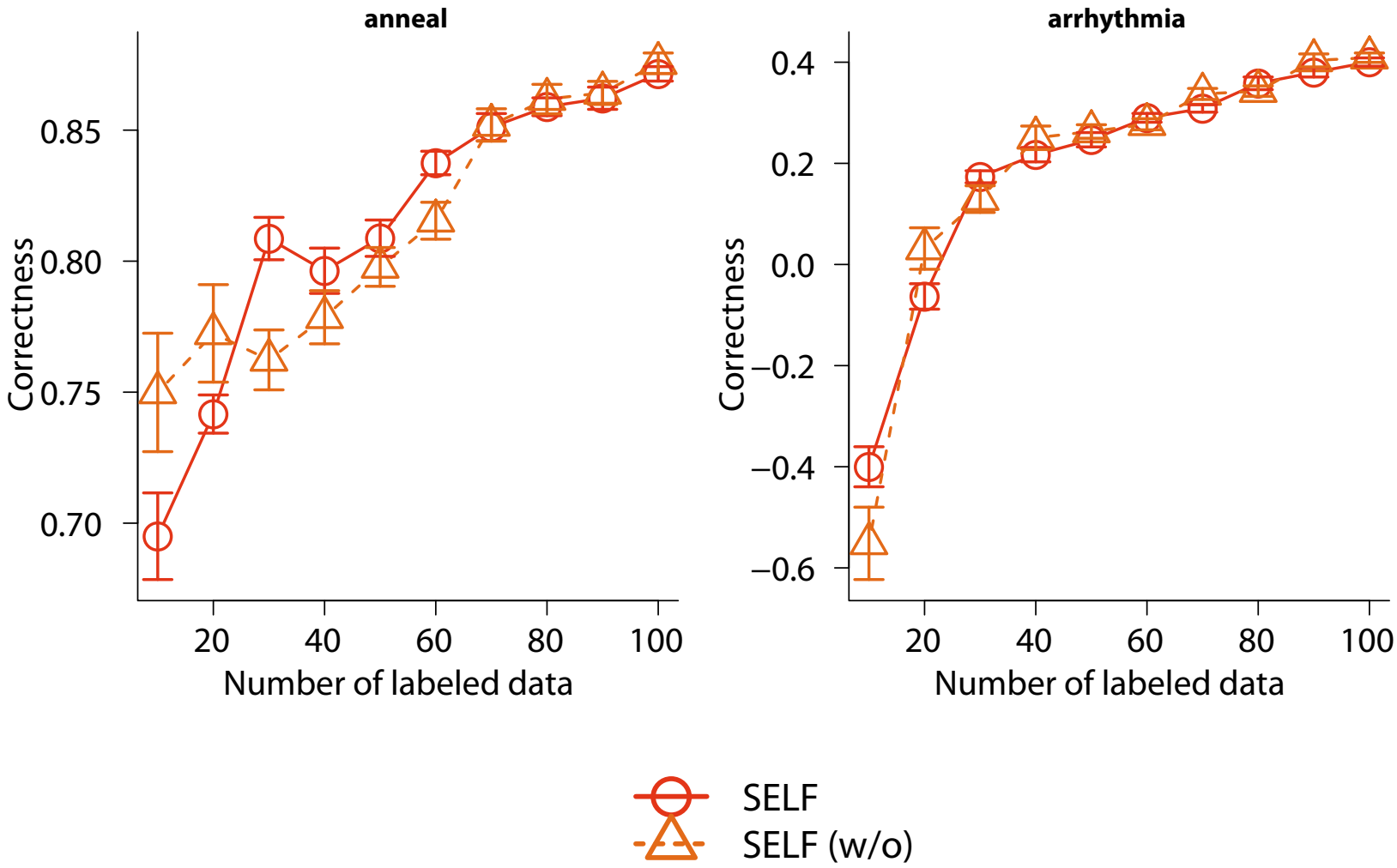
ランキングの実験結果


abalon
allbp
Correctness
Number of labeled data
−0.4
−0.5
−0.6
−0.7
−0.8
−0.9
0.95
0.90
0.85
0.80
0.75
0.70
20
40
60
80
100
SELF
SELF (w/o)

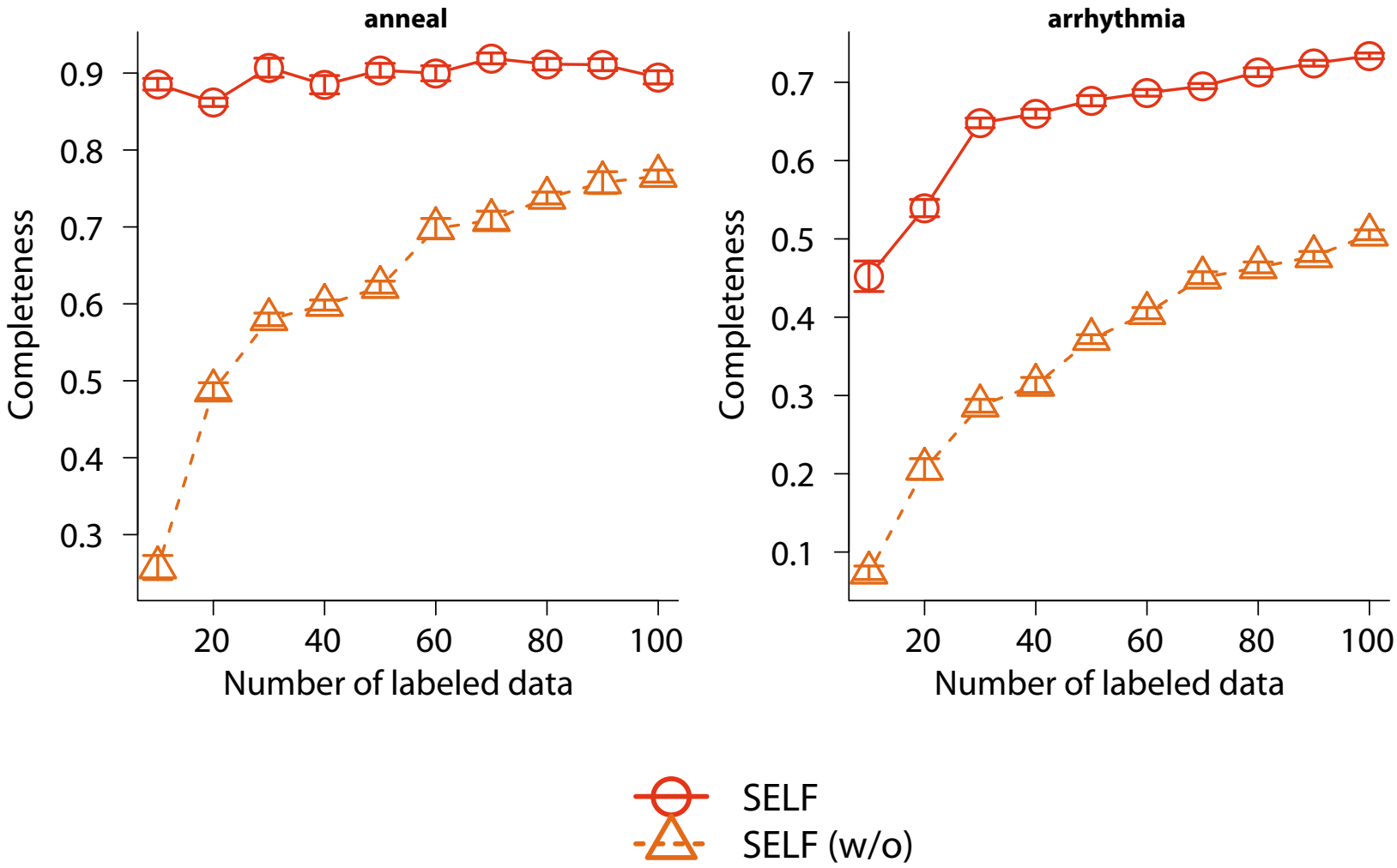
ランキングの実験結果


abalon
Completeness
0.35
0.30
0.25
0.20
0.15
0.10
0.05
20
40
60
80
100
Number of labeled data
allbp
Completeness
0.9
0.8
0.7
0.6
20
40
60
80
100
Number of labeled data
SELF
SELF (w/o)

ランキングの実験結果


anneal
Correctness
0.85
0.80
0.75
0.70
20
40
60
80
100
Number of labeled data
arrhythmia
Correctness
0.4
0.2
0.0
−0.2
−0.4
−0.6
20
40
60
80
100
Number of labeled data
SELF
SELF (w/o)

ランキングの実験結果


anneal
Completeness
0.9
0.8
0.7
0.6
0.5
0.4
0.3
20
40
60
80
100
Number of labeled data
arrhythmia
Completeness
0.7
0.6
0.5
0.4
0.3
0.2
0.1
20
40
60
80
100
Number of labeled data
SELF
SELF (w/o)

まとめ

- 離散値・連続値混在データに対する
  半教師あり学習手法SELFを提案した
  - 形式概念解析によってもとのデータに束構造を与える
  - その空間で，分類規則とその重みを学習する
- 半教師あり学習への貢献：
  - 代数的枠組を用いた新しいアプローチを示した
  - データの分布は必要ない
- 形式概念解析（FCA）への貢献：
  - FCA が半教師あり学習へ応用できることを示した

目次

- 第一部：理論
  - フラクタルとハウスドルフ距離を用いた図形の学習
    — 計算可能な 2 値分類を目指して
- 第二部：理論から実践
  - 符号化ダイバージェンス
  - MCL とグレイコードを用いたクラスタリング
  - 2 進符号化を活用した高速クラスタリング
- 第三部：形式概念と共に
  - 離散値・連続値混在データからの半教師あり学習
  - 半教師ありのマルチラベルクラス分類によるリガンド発見

半教師ありのマルチラベルクラス分類によるリガンド発見

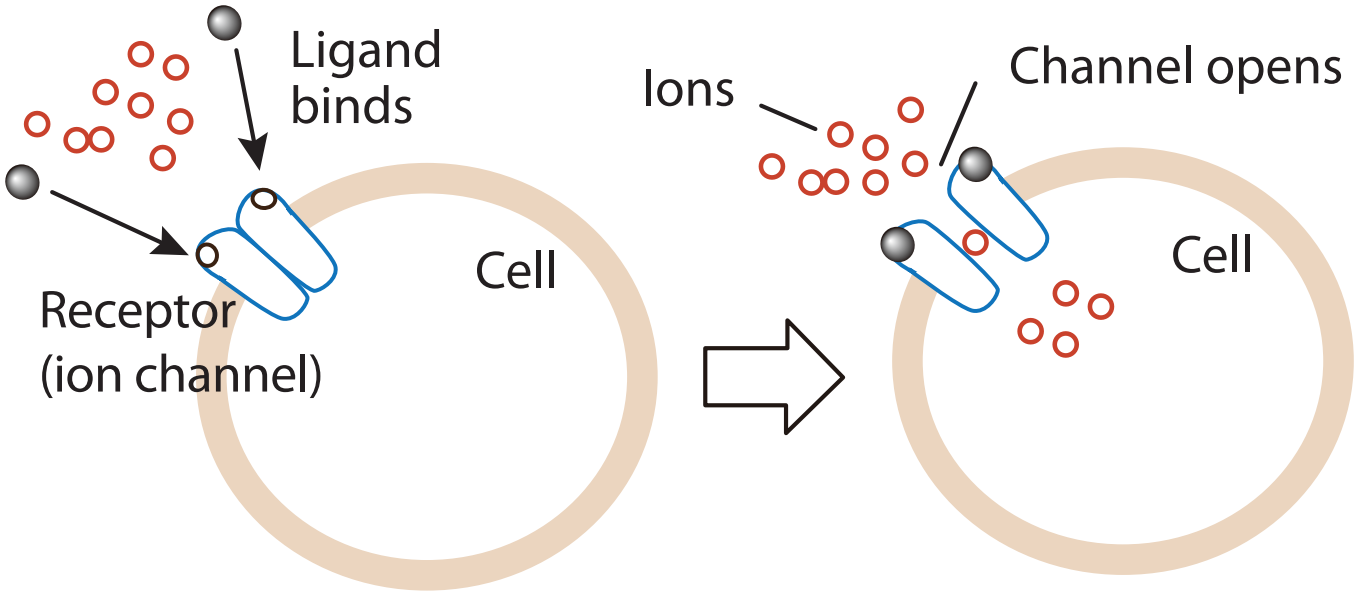
- Sugiyama, M., Imajo, K., Otaki, K., Yamamoto, A.:
  Discovering Ligands for TRP Ion Channels
  Using Formal Concept Analysis, ILP 2011
- Sugiyama, M., Imajo, K., Otaki, K., Yamamoto, A.:
  Semi-Supervised Ligand Finding Using Formal Concept Analysis,
  IPSJ TOM (accepted)

概要・結果

- **背景**：生体内では，受容器がシグナル処理に重要な役割を担っている
  - 生化学的実験において，リガンドが重要なツール
- **問題点**：リガンドの発見が難しい
  - *In silico* アプローチ が求められている
- **解決策**：リガンド発見問題をマルチラベルクラス分類として定式化し，機械学習的にアプローチする
- 新規手法 LIFT (Ligand FInding via Formal ConcepT Analysis) を提案する
  - SELF と同様のアプローチ：
    - 半教師あり学習でデータを扱う
    - 形式概念解析（FCA）を用いる

背景

- 生体内では，レセプターが種々の反応の「ゲート」の役割
  - リガンドは，特定の受容器に特異的に結合する化合物
  - 受容器を活性化（agonist / activator）させたり，不活化（antagonist / inhibitor）させたり



背景・問題

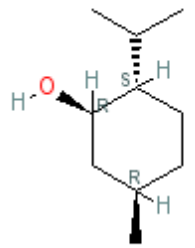
- 生体内では，レセプターが種々の反応の「ゲート」の役割
  - リガンドは，特定の受容器に特異的に結合する化合物
  - 受容器を活性化（agonist / activator）させたり，不活化（antagonist / inhibitor）させたり
- 取り扱いやすいリガンドを新たに発見するのは困難
  - リガンド候補の発見は生物学者の勘と経験に依存
  - 候補の *in vivo* や *in vitro* でのテスト実験はコストが高い
- *In silico* アプローチが求められている

戦略

- 機械学習的アプローチでデータベースから
  リガンド候補を発見する
- IUPHAR データベース を用いる
  - 全部で 1,782 個のリガンドが登録されている
  - 各リガンドは 7 つの特徴量で記述されている
    - 3 つの連続量と 4 つの離散量
  - 各リガンドに対して，それがどの受容器に結合するのかがわかる
    - クラスのラベルに対応する

IUPHAR-DB Ligand: 2430	
Ligand name	**menthol**

2D Structure



Calculated Physical-Chemical Properties

Hydrogen bond acceptors	1
Hydrogen bond donors	1
Rotatable bonds	1
Topological polar surface area	20.23
Molecular weight	156.15
XLogP	3.21
No. Lipinski's rules broken	0

Molecular properties generated using the CDK

Summary | **Biological activity** | References | Structure | Similar ligands

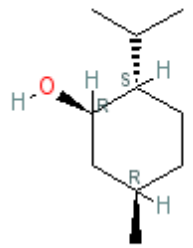
Selectivity at human ion channels

Key to terms and symbols — Click column headers to sort

Receptor	Type	Action	Affinity	Units	Concentration range (M)	Reference
TRPM8	Activator	None	4.6	pEC_{50}	-	3

IUPHAR-DB Ligand: 2430	
Ligand name	**menthol**

2D Structure



Calculated Physical-Chemical Properties	
Hydrogen bond acceptors	1
Hydrogen bond donors	1
Rotatable bonds	1
Topological polar surface area	20.23
Molecular weight	156.15
XLogP	3.21
No. Lipinski's rules broken	0

Molecular properties generated using the CDK

Feature vector

Summary | **Biological activity** | References | Structure | Similar ligands

Selectivity at human ion channels

Key to terms and symbols

Class label

Click column headers to sort

Receptor	Type	Action	Affinity	Units	Concentration range (M)	Reference
TRPM8	Activator	None	4.6	pEC_{50}	-	3

アプローチ

- リガンド発見をマルチラベルクラス分類としてモデル化する
  - リガンド：データ点
  - 受容器：クラスラベル
    - 各対象は単一のラベルでなく可能なラベルの集合を持つ
    - 順序学習（preference learning）で扱われている
- データベース内には，関係ないリガンドがたくさんある
  - ラベル無しデータとして扱い半教師あり学習をおこなう
- 形式概念解析（FCA）を使う
  - 連続量を 2 進符号化で離散化して FCA を適用する

関連研究

- クラス分類の観点からリガンド発見を扱った研究はない
  - 1 つだけ関連した研究［Ballester and Mitchell, 2010］, リガンドの親和力を予測する
  - King ら［King *et al.*, 1996］による SAR のモデル化
    - 帰納論理プログラミングによる関係の記述と理解が目的
- 受容器やリガンドに関する多くの *in silico* な研究［Moitessier *et al.*, 2008］は，背景知識を使った予測モデルの構築
  - 複合体のポテンシャルエネルギーや 2 次元配置など
- 多くのスコアリング手法が提案されている（AMBER［Cornell *et al.*, 1995］, AutoDock［Huey *et al.*, 2007］など）
- しかし，それらを利用するにはドメインの知識が必要，結果はその背景知識に依存する

クラス分類の例

- テストデータ y を分類する（分類規則の学習ではない）

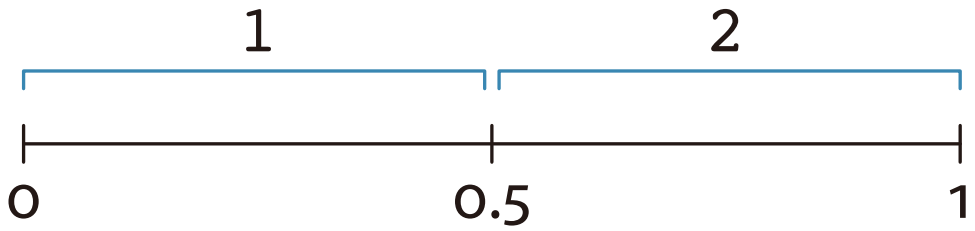
	HBD	TPS	MW	Labels	
x_1	0	0.98	0.88	A	
x_2	1	0.41	0.48	B	C
x_3	2	0.12	0.71	A	C
y	0	0.77	0.79		

レベル 1 での離散化

- テストデータ y を分類する

	HBD	TPS	MW	Labels	
x_1	0	0.98	0.88	A	
x_2	1	0.41	0.48	B	C
x_3	2	0.12	0.71	A	C
y	0	0.77	0.79		

  - 離散化レベル 1



データ前処理

- テストデータ y を分類する

	HBD	TPS	MW	Labels	
x_1	0	0.98	0.88	A	
x_2	1	0.41	0.48	B	C
x_3	2	0.12	0.71	A	C
y	0	0.77	0.79		

  - 以下のように変換される

	H.0	H.1	H.2	T.1	T.2	M.1	M.2
x_1	×				×		×
x_2		×		×		×	
x_3			×	×			×
y	×				×		×

プリファレンスの計算

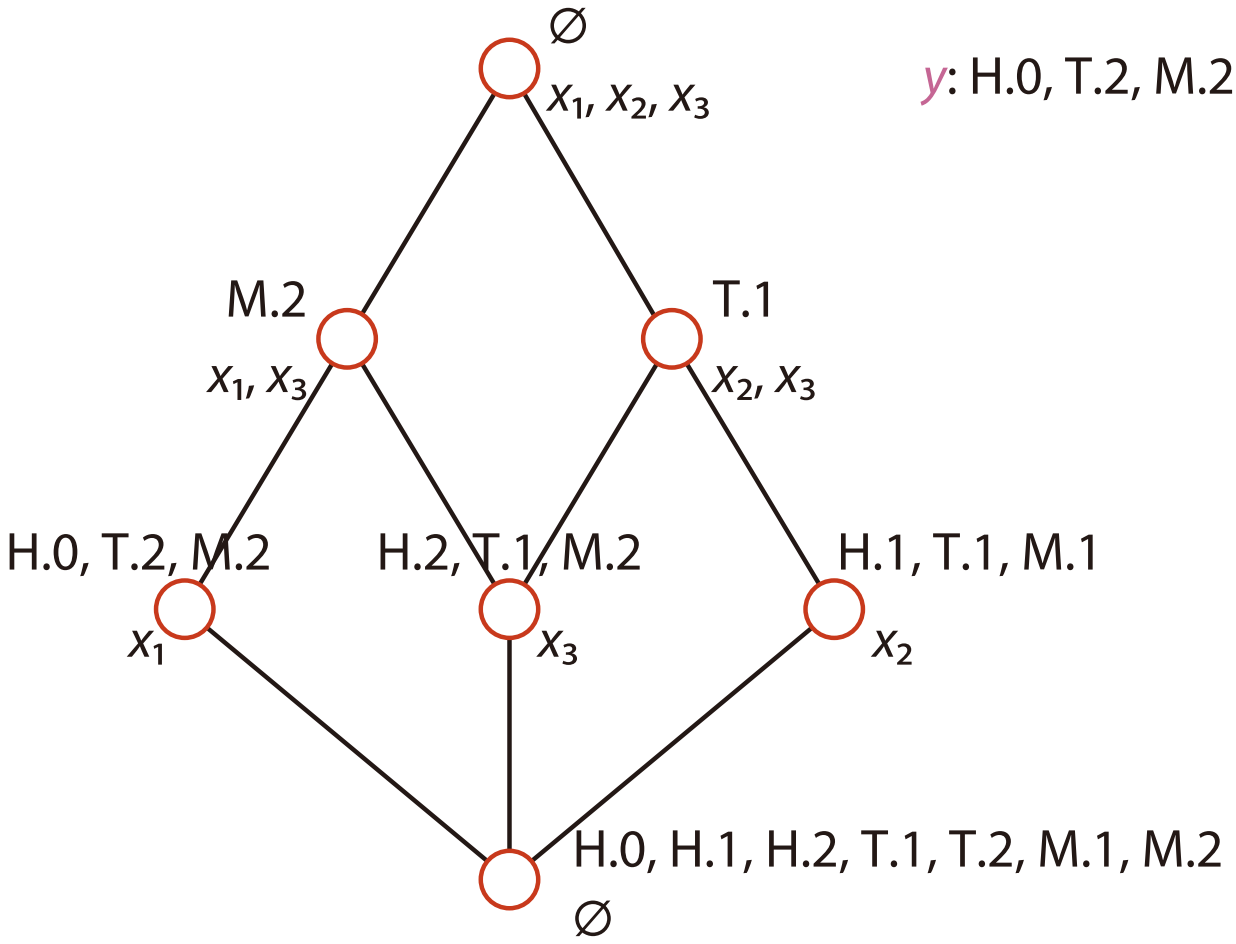
- テストデータ y を分類する

	H.0	H.1	H.2	T.1	T.2	M.1	M.2
x_1	×				×		×
x_2		×		×		×	
x_3			×	×			×
y	×				×		×

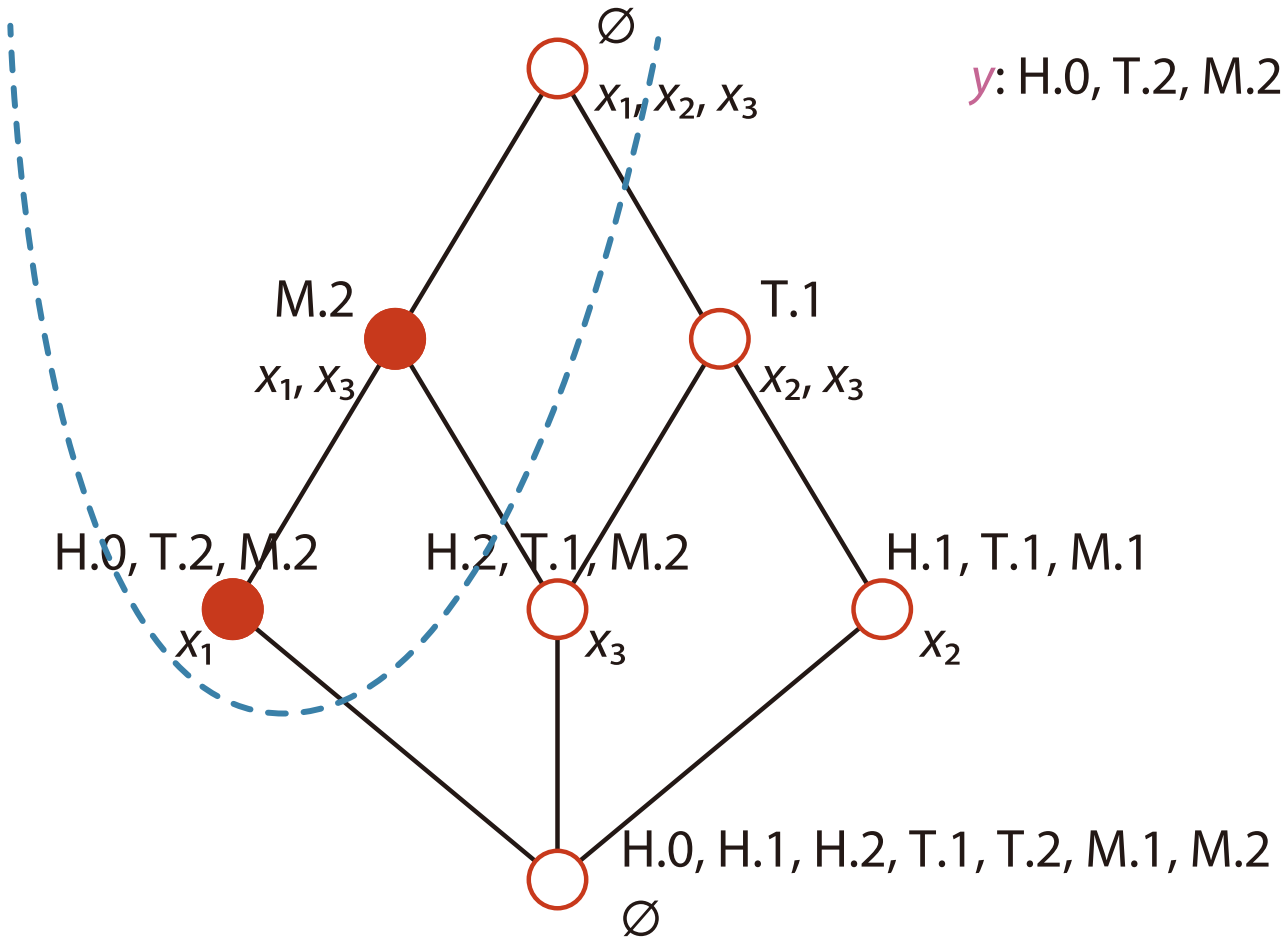
1. 訓練データ x_1, x_2, x_3 から FCA で概念束を作る
2. 無矛盾な概念を見つける
3. 各ラベルのプリファレンスを計算する

- 分類規則を獲得するのではなく，テストデータのラベルのプリファレンスを直接計算するところが SELF と異なる

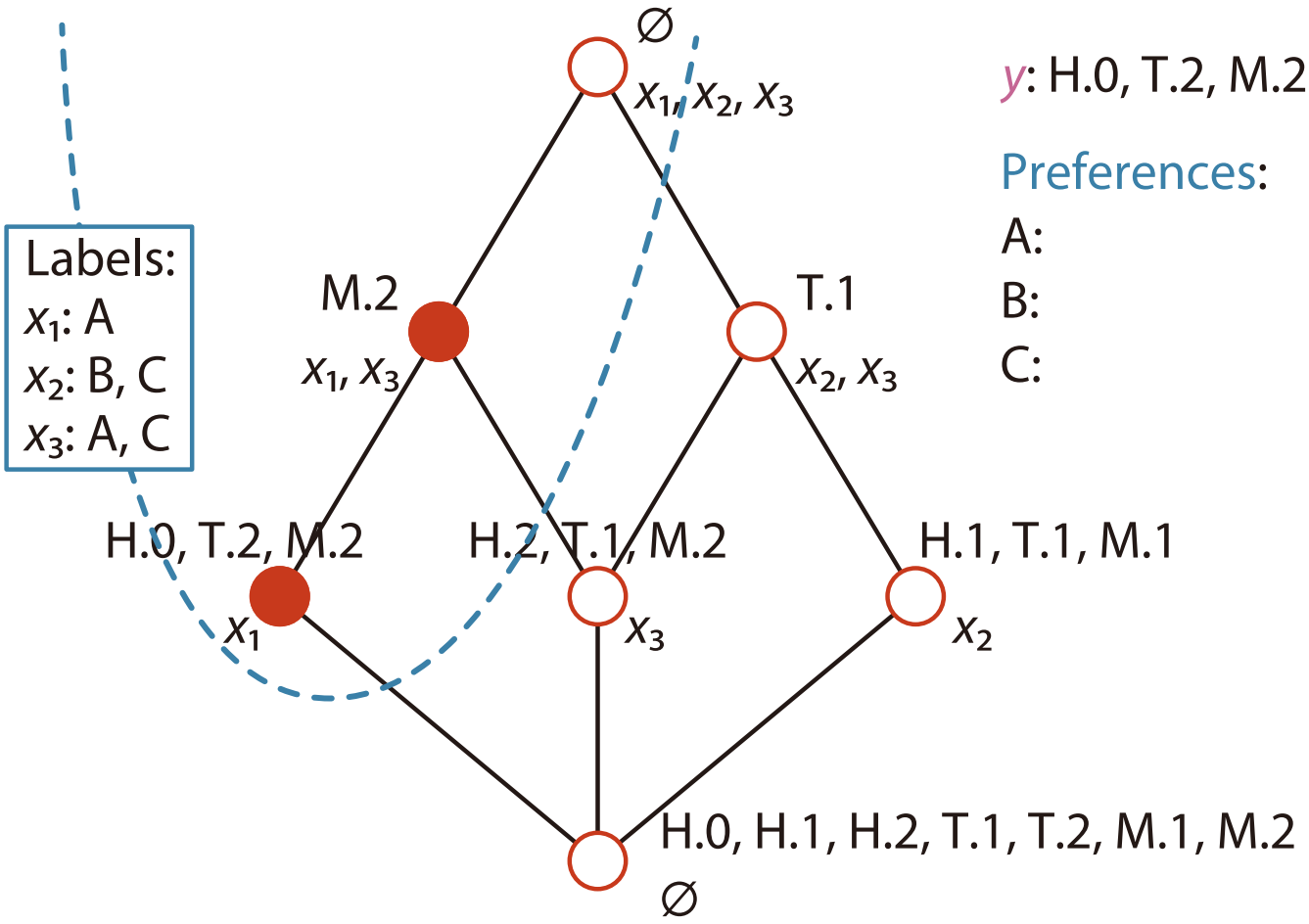
概念束上での学習


∅
x_1, x_2, x_3
y: H.0, T.2, M.2
M.2
x_1, x_3
T.1
x_2, x_3
H.0, T.2, M.2
x_1
H.2, T.1, M.2
x_3
H.1, T.1, M.1
x_2
H.0, H.1, H.2, T.1, T.2, M.1, M.2
∅

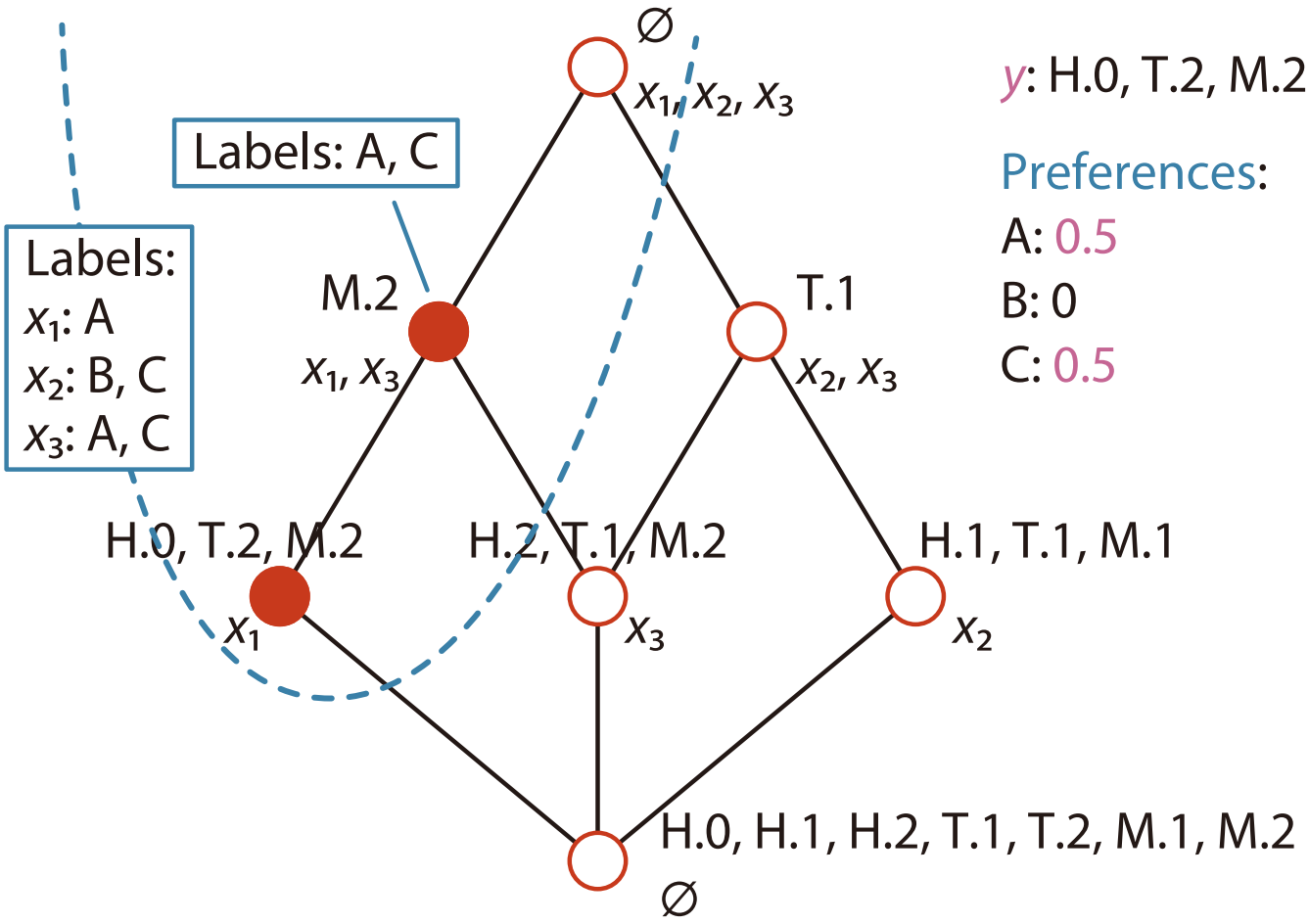
概念束上での学習


∅
x_1, x_2, x_3
y: H.0, T.2, M.2
M.2
x_1, x_3
T.1
x_2, x_3
H.0, T.2, M.2
x_1
H.2, T.1, M.2
x_3
H.1, T.1, M.1
x_2
H.0, H.1, H.2, T.1, T.2, M.1, M.2
∅

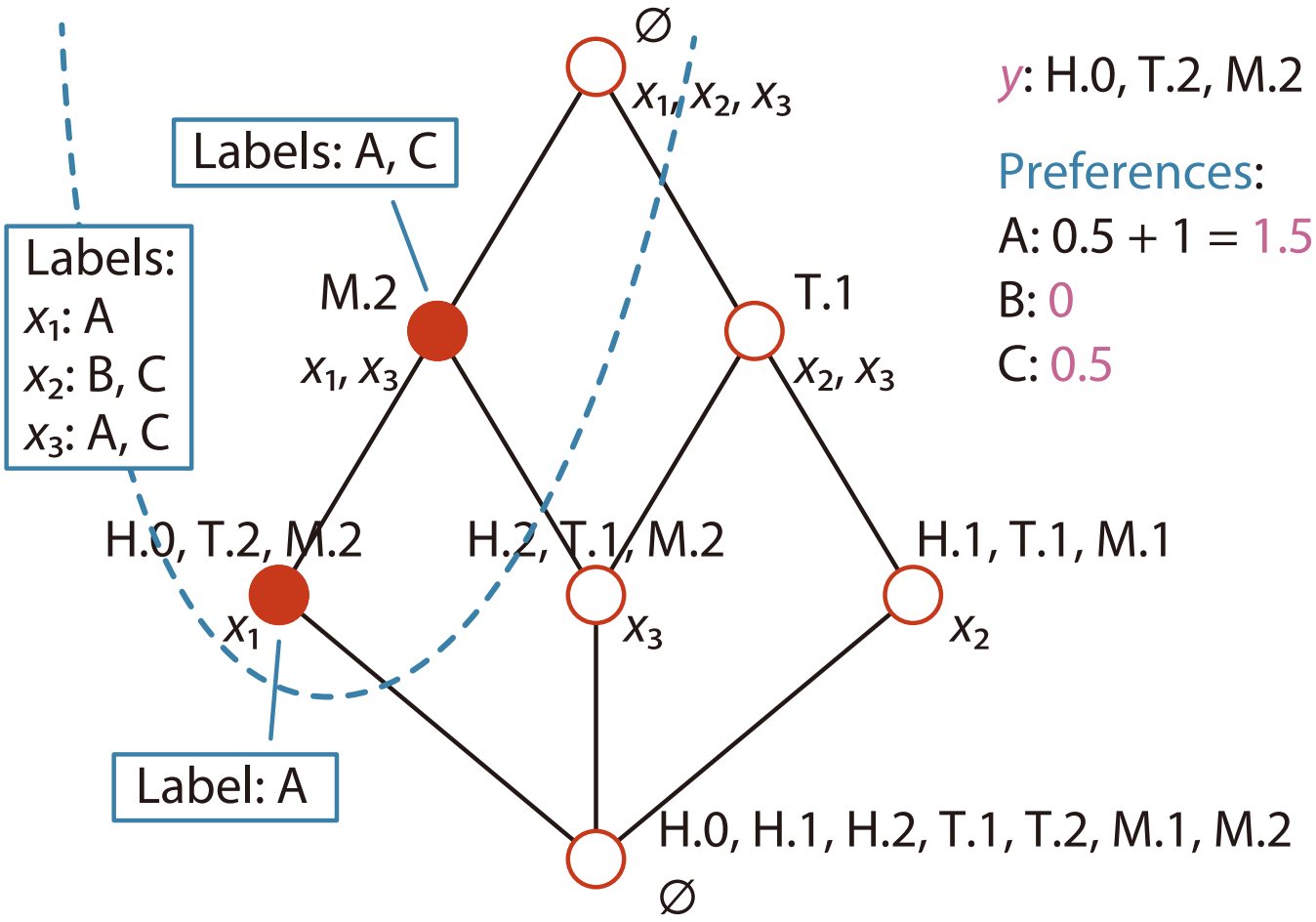
概念束上での学習


∅
x_1, x_2, x_3
Labels:
x_1: A
x_2: B, C
x_3: A, C
M.2
x_1, x_3
T.1
x_2, x_3
H.0, T.2, M.2
x_1
H.2, T.1, M.2
x_3
H.1, T.1, M.1
x_2
H.0, H.1, H.2, T.1, T.2, M.1, M.2
∅
y: H.0, T.2, M.2
Preferences:
A:
B:
C:

概念束上での学習


∅
x_1, x_2, x_3
Labels: A, C
Labels:
x_1: A
x_2: B, C
x_3: A, C
M.2
x_1, x_3
T.1
x_2, x_3
H.0, T.2, M.2
x_1
H.2, T.1, M.2
x_3
H.1, T.1, M.1
x_2
H.0, H.1, H.2, T.1, T.2, M.1, M.2
∅
y: H.0, T.2, M.2
Preferences:
A: 0.5
B: 0
C: 0.5

概念束上での学習


∅
x_1, x_2, x_3
Labels: A, C
Labels:
x_1: A
x_2: B, C
x_3: A, C
M.2
x_1, x_3
T.1
x_2, x_3
H.0, T.2, M.2
x_1
H.2, T.1, M.2
x_3
H.1, T.1, M.1
x_2
Label: A
H.0, H.1, H.2, T.1, T.2, M.1, M.2
∅
y: H.0, T.2, M.2
Preferences:
A: 0.5 + 1 = 1.5
B: 0
C: 0.5

プリファレンス

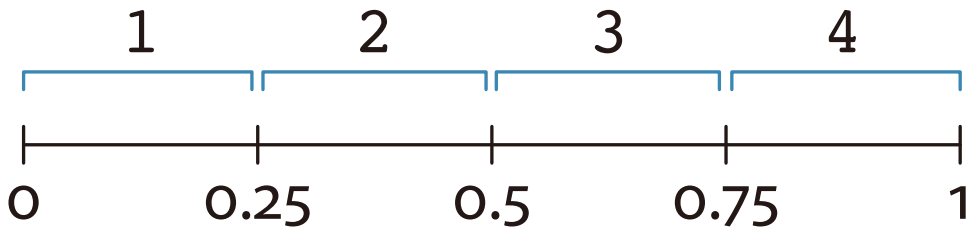
- 離散化レベル 1 では，プリファレンスは
  $\psi_y^1(\mathrm{A}) = 1.5,\ \psi_y^1(\mathrm{B}) = 0,\ \text{and}\ \psi_y^1(\mathrm{C}) = 0.5$

レベル 2 での離散化

- テストデータ y を分類する

	HBD	TPS	MW	Labels	
x_1	0	0.98	0.88	A	
x_2	1	0.41	0.48	B	C
x_3	2	0.12	0.71	A	C
y	0	0.77	0.79		

  - 離散化レベル 2



データ前処理

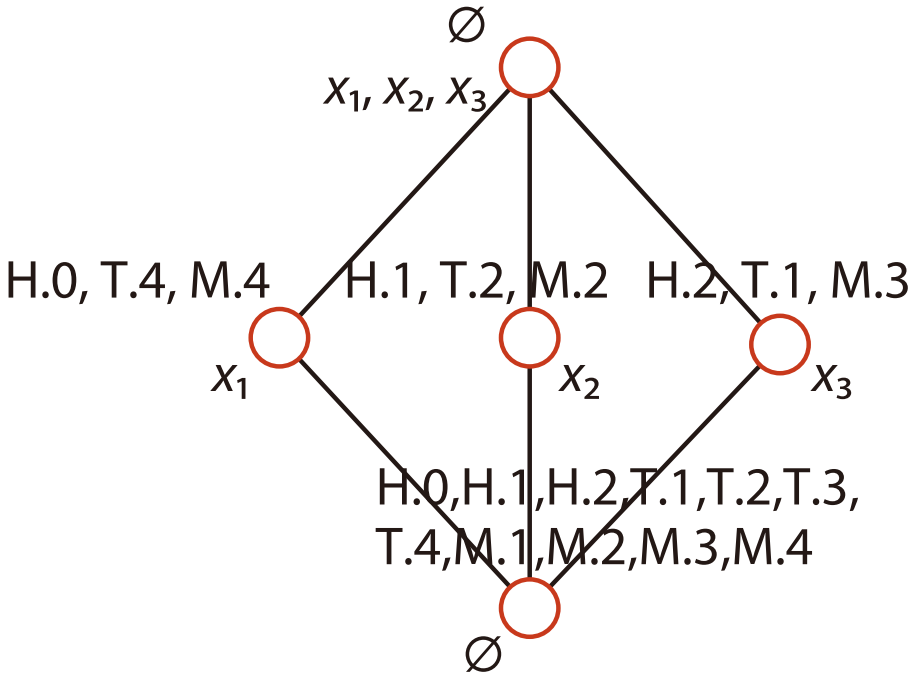
- テストデータ y を分類する

	HBD	TPS	MW	Labels	
x_1	0	0.98	0.88	A	
x_2	1	0.41	0.48	B	C
x_3	2	0.12	0.71	A	C
y	0	0.77	0.79		

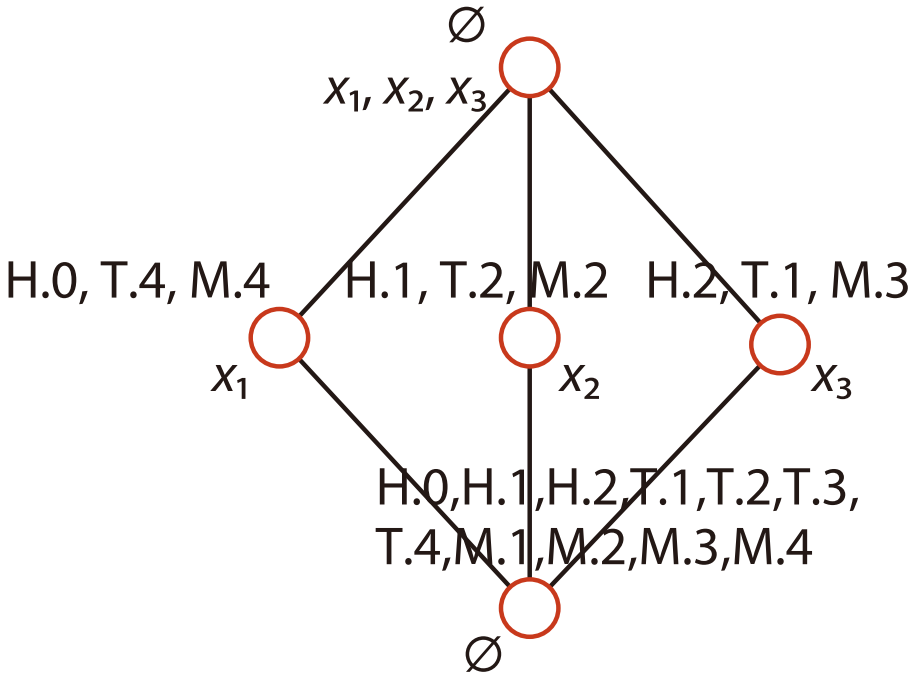
  - 以下のように変換される

	H.0	H.1	H.2	T.1	T.2	T.3	T.4	M.1	M.2	M.3	M.4
x_1	×						×				×
x_2		×			×				×		
x_3			×	×						×	
y	×						×				×

FCA による概念束

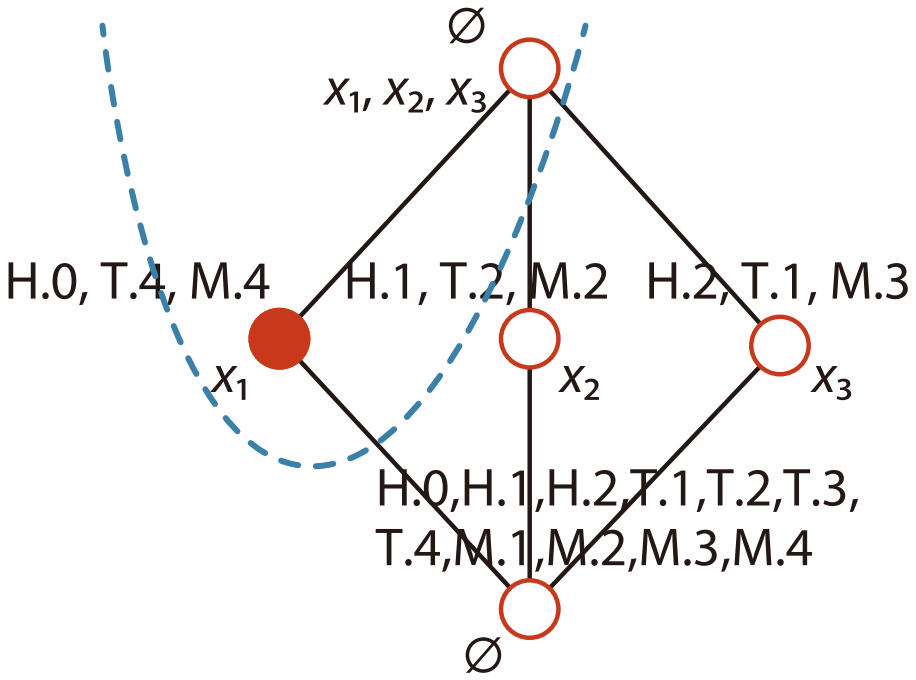

∅
x_1, x_2, x_3
H.0, T.4, M.4
H.1, T.2, M.2
H.2, T.1, M.3
x_1
x_2
x_3
H.0,H.1,H.2,T.1,T.2,T.3,
T.4,M.1,M.2,M.3,M.4
∅

概念束上での学習



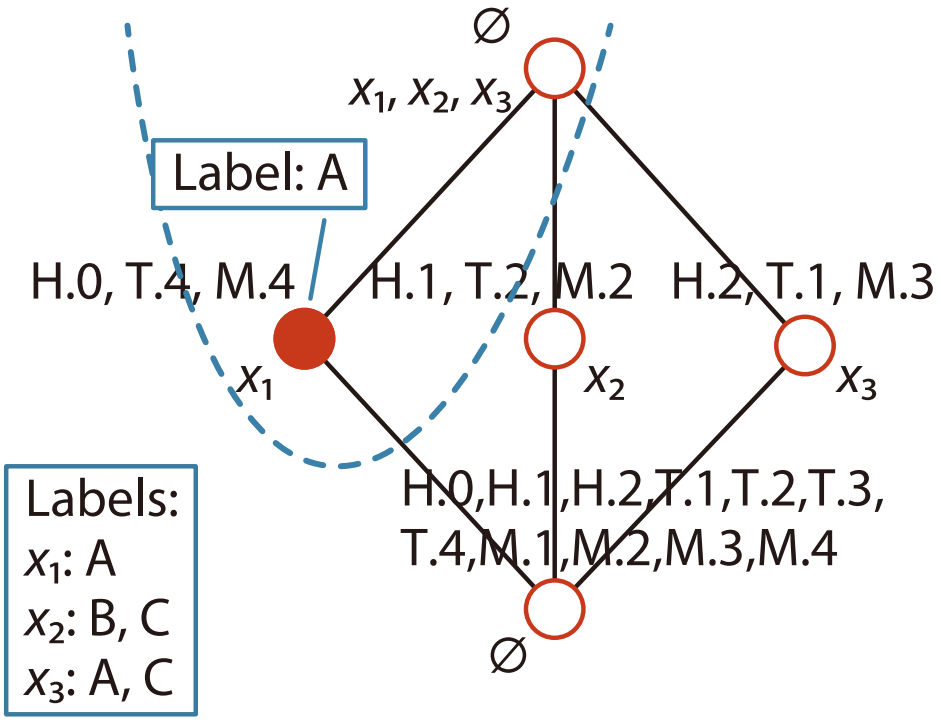
y: H.0, T.4, M.4

概念束上での学習



y: H.0, T.4, M.4

概念束上での学習



y: H.0, T.4, M.4

Preferences:

A: 1

B: 0

C: 0

プリファレンス

- 離散化レベル 1 では，プリファレンスは
  $\psi_y^1(\mathrm{A}) = 1.5,\ \psi_y^1(\mathrm{B}) = 0,\ \text{and}\ \psi_y^1(\mathrm{C}) = 0.5$
- 離散化レベル 2 では，プリファレンスは
  $\psi_y^2(\mathrm{A}) = 1,\ \psi_y^2(\mathrm{B}) = 0,\ \text{and}\ \psi_y^2(\mathrm{C}) = 0$

プリファレンス

- 離散化レベル 1 では，プリファレンスは
  $\psi_y^1(\mathrm{A}) = 1.5,\ \psi_y^1(\mathrm{B}) = 0,\ \text{and}\ \psi_y^1(\mathrm{C}) = 0.5$
- 離散化レベル 2 では，プリファレンスは
  $\psi_y^2(\mathrm{A}) = 1,\ \psi_y^2(\mathrm{B}) = 0,\ \text{and}\ \psi_y^2(\mathrm{C}) = 0$
- 各ラベルに対するプリファレンスは，
  $\psi_y(\mathrm{A}) = 1.5 + 1 = 2.5,$
  $\psi_y(\mathrm{B}) = 0 + 0 = 0,$
  $\psi_y(\mathrm{C}) = 0.5 + 0 = 0.5$
  - y はラベル A と C に分類される
  - y のラベルランキングは A > C > B
    - 離散化レベルの最大値 $k_{\max} = 2$ はユーザが与える

プリファレンスの定義

- 文脈 $(\{y\}, M, I)$ と概念 (A, B) に対して，y は (A, B) にm-無矛盾 $\Longleftrightarrow$ $B \subseteq \{m \in M \mid (y, m) \in I\}$ かつ $B \neq \emptyset$
- データセット $\tau = (H, X)$ と $\upsilon = (H, y)$（$|\upsilon| = 1$）とする 各離散化レベル k と各ラベル $\lambda \in \mathscr{L}$ に対して，λ の離散化レベル k での y に関する m-プリファレンスを以下のように定義

$$\psi_y^k(\lambda|\tau) := \sum_{A \in \mathbf{A}} \#\Lambda(A)^{-1}, \text{ where}$$

$$\mathbf{A} := \{A \mid y \text{ は } (A, B) \in \mathscr{B}^k(\tau)\,(\lambda \in \Lambda(A)) \text{ に } m\text{-無矛盾}\}$$

  - $\#\Lambda(A)^{-1} = 0 \Longleftrightarrow \#\Lambda(A) = 0$ と仮定
- 各ラベル $\lambda \in \mathscr{L}$ と y に対して，λ の m-プリファレンスは：

$$\psi_y(\lambda|\tau) := \sum_{k=1}^{k_{\max}} \psi_y^k(\lambda|\tau)$$

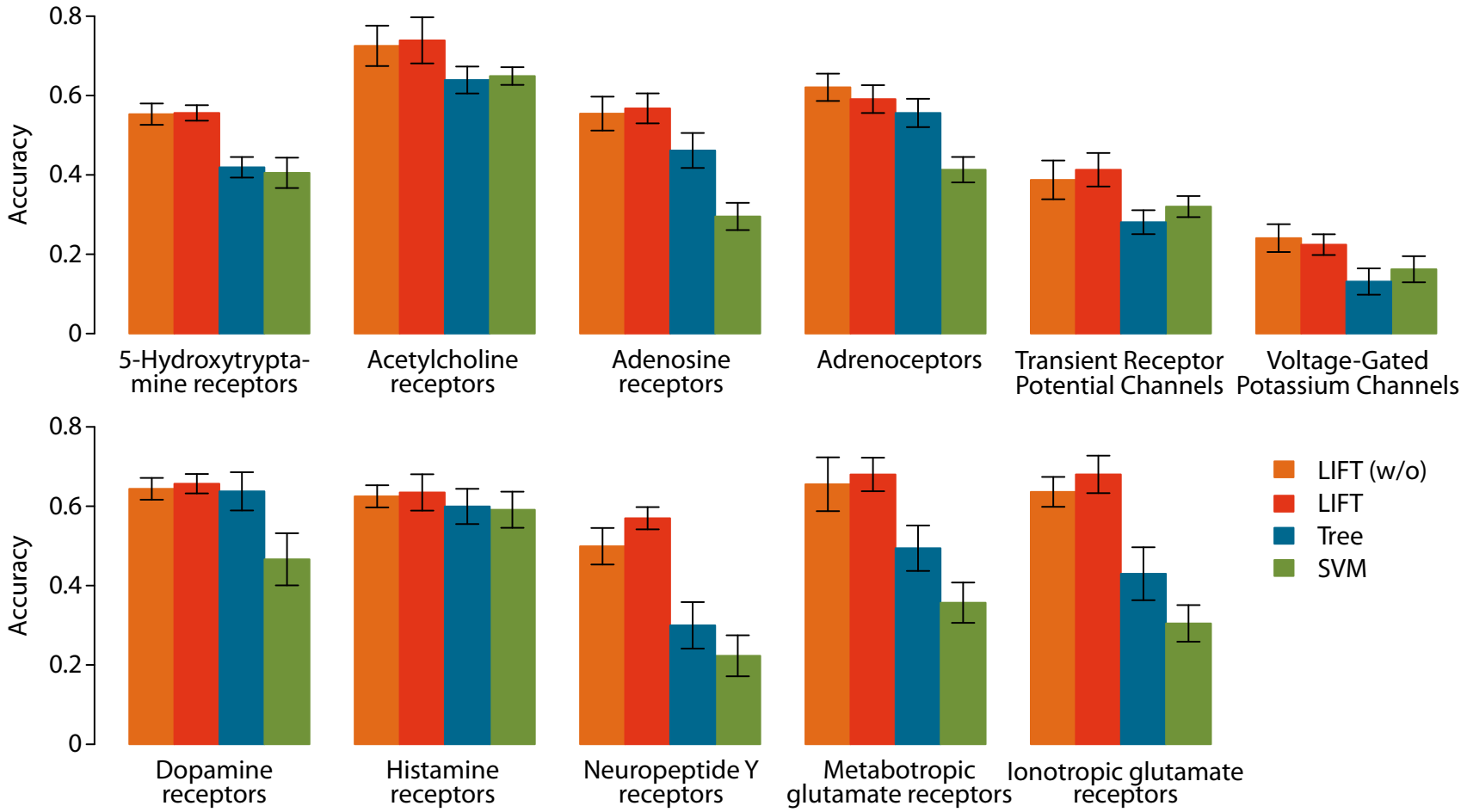
実験手法

- LIFT はR version 2.12.2 で実装
  - LIFT は概念束の構築にLCM [Uno *et al.*, 2005] を使う
- IUPHAR データベースから全 1,782 リガンドを収集
- ラベル無しデータの有効性を検証するため，以下の 2 つを実行
  1. ラベル付きデータのみを訓練で使用
  2. すべてのリガンドデータをラベル無しデータとして訓練で使用
- 最大離散化レベル k_{max} は 5
- 10-fold 交差検定
- 比較手法：SVM (RBF カーネル)と R で実装された決定木

受容器の族

Family name	# Ligands (Data size)	# Receptors (Class size)
5-Hydroxytryptamine receptors	286	53
Acetylcholine receptors	100	68
Adenosine receptors	162	40
Adrenoceptors	111	35
Dopamine receptors	69	40
Histamine receptors	120	37
Neuropeptide Y receptors	76	34
Metabotropic glutamate receptors	73	9
Transient receptor potential channels	78	58
Voltage-gated potassium channels	61	71
Ionotropic glutamate receptors	81	14

実験結果


Accuracy
0.8
0.6
0.4
0.2
0
5-Hydroxytrypta-mine receptors
Acetylcholine receptors
Adenosine receptors
Adrenoceptors
Transient Receptor Potential Channels
Voltage-Gated Potassium Channels
Dopamine receptors
Histamine receptors
Neuropeptide Y receptors
Metabotropic glutamate receptors
Ionotropic glutamate receptors
LIFT (w/o)
LIFT
Tree
SVM

まとめ

- リガンド発見問題をマルチラベルクラス分類として定式化
- 半教師あり学習手法 LIFT を提案
  - FCA を用いる
- 他の機械学習手法に比べて，LIFT がリガンド発見に有効であることを示した
  - 生物学，生化学への貢献が期待される

全体のまとめ

連続的な対象の離散化とそこからの学習という
2 つのプロセスを融合し，理論から実践までに渡って
学習の計算論的側面を解析した

- 現代の機械学習における主要な課題群：
  - 計算論学習理論
  - 教師あり学習（クラス分類）
  - 教師なし学習（クラスタリング）
  - 半教師あり学習
  - 順序学習

  に対して，一貫して新規の手法を提供した

研究成果（1/2）

- 著書：
  - 小林茂夫, 杉山麿人.: 生命科学研究に成功するための統計法ノート, 講談社, 2009.
- ジャーナル論文：
  - Sugiyama, M., Yamamoto, A.: Semi-Supervised Learning on Closed Set Lattices, *Intelligent Data Analysis*, accepted
  - Sugiyama, M., Imajo, K., Otaki, K., Yamamoto, A.: Semi-Supervised Ligand Finding Using Formal Concept Analysis, *IPSJ TOM*, accepted

研究成果（2/2）

- 国際会議（査読付）：
  - Sugiyama, M., Hirowatari, E., Tsuiki, H., Yamamoto, A.: Learning Figures with the Hausdorff Metric by Fractals, ALT 2010
  - Sugiyama, M., Yamamoto, A.: The Coding Divergence for Measuring the Complexity of Separating Two Sets, ACML 2010
  - Sugiyama, M., Yamamoto, A.: Semi-Supervised Learning for Mixed-Type Data via Formal Concept Analysis, ICCS 2011
  - Sugiyama, M., Yamamoto, A.: The Minimum Code Length for Clustering Using the Gray Code, ECML PKDD 2011
  - Sugiyama, M., Yamamoto, A.: A Fast and Flexible Clustering Algorithm Using Binary Discretization, ICDM 2011
  - Sugiyama, M., Yoshioka, T., Yamamoto, A.: High-throughput Data Stream Classification on Trees, ALSIP 2011
- その他，査読なし国内・国際会議 7 回

付録

戦略

- 離散化プロセスそのものを学習に取り込むことで，離散化と学習を一体化したモデルを構築したい
- 離散化プロセスを適切に扱うために計算可能性解析での標準的な計算モデルである実効的計算を用いる
  - 計算機への入力の精度が上がるにつれて，出力の近似が次第に良くなる
- Gold 型学習モデルに基づき実効的学習を提案
  - 学習機械への入力（データ）の精度が上がるにつれて，出力（仮説）の近似が次第に良くなる
    - 出力の精度は，汎化誤差に対応する

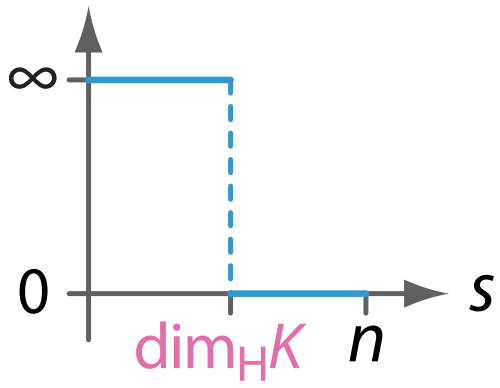
標準的な 2 値分類での用語との対応

		学習対象の図形 K	
		w は正例 $(\rho(w) \cap K \neq \emptyset)$	w は負例 $(\rho(w) \cap K = \emptyset)$
仮説 H	$h(w) = 1$ $(\rho(w) \cap \kappa(H) \neq \emptyset)$	True positive	False positive (Type I error)
	$h(w) = 0$ $(\rho(w) \cap \kappa(H) = \emptyset)$	False negative (Type II error)	True negative

- 有限列 w：テストデータ
- 関数 h：仮説 H で構成される分類器

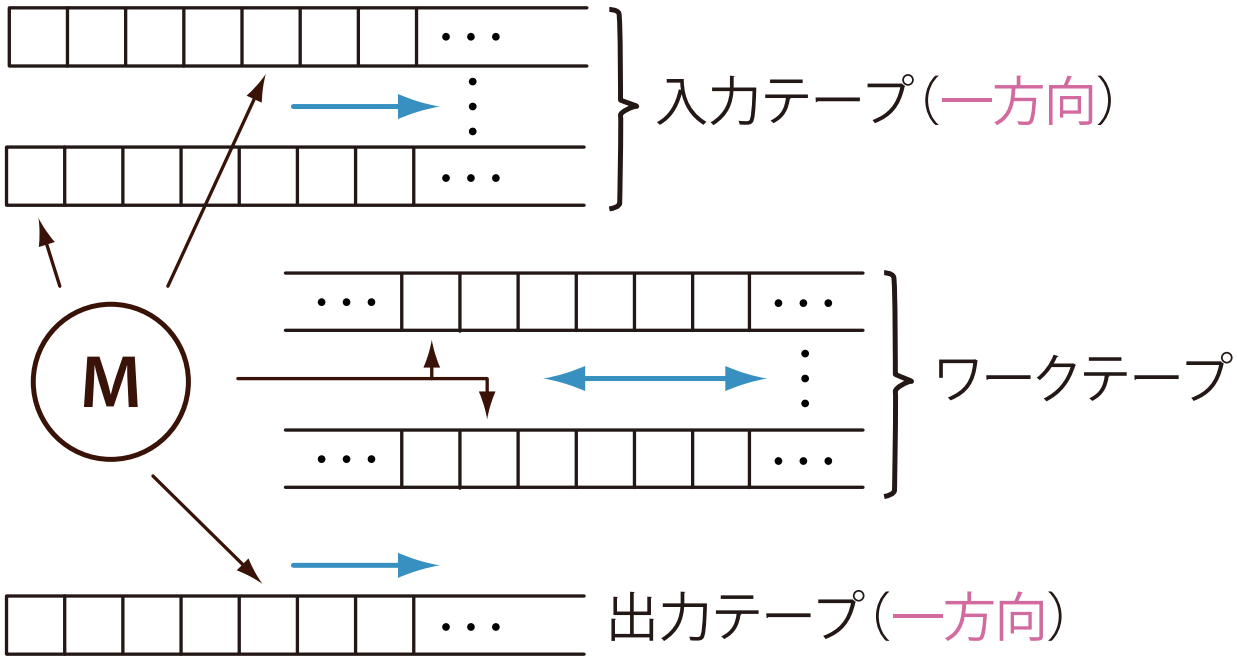
ハウスドルフ次元（$\dim_H$）の定義

- ハウスドルフ次元はフラクタル幾何における中心的概念
  - どのくらい空間を占めているかを表す
  - ハウスドルフ測度で定義される
- ハウスドルフ測度は，長さや面積の一般化
  - 集合の「被覆」で定義される
- K の s 次元ハウスドルフ測度は，$\lim_{\varepsilon \to 0} \mathcal{H}^s_\varepsilon(K)$
  - 可算集合 $\mathcal{U}$ は K の ε 被覆 $\iff$
    $\forall U \in \mathcal{U},\ |U| \leqslant \varepsilon$ かつ $X \subset \bigcup_{U \in \mathcal{U}} U$
  - $\mathcal{H}^s_\varepsilon(K)$
    $= \inf\{ \sum_{U \in \mathcal{U}} |U|^s \mid \mathcal{U}$ は K の ε 被覆$\}$



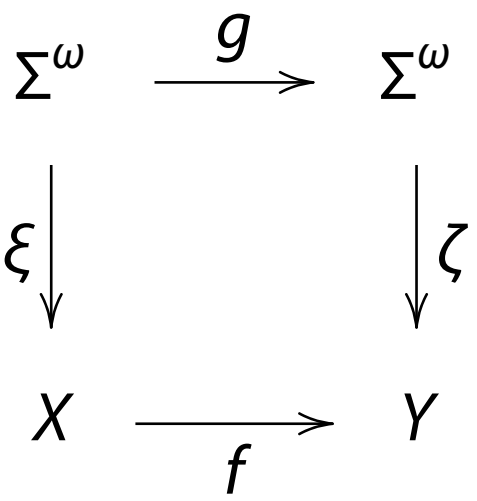
TTE における無限列の計算

- TTE（Type-2 Theory Effectivity）では，連続的対象（特に実数）の計算をその表現（通常は無限列）の変換として捉える
- 無限列に対する計算をタイプ 2 マシンによって実現する



TTE による計算可能性の定義

- ξ と ζ をそれぞれ X と Y の表現とする
  - 表現とは無限列 Σ^ω から X（または Y）への全射
- 要素 $x \in X$ がξ-計算可能 $\Longleftrightarrow$
  ある計算可能な p が存在して，$\xi(p) = x$
- 関数 $f :\subseteq X \to Y$ が(ξ, ζ)-計算可能 $\Longleftrightarrow$
  ある計算可能な関数 g が存在して，任意の $p \in \mathrm{dom}(\xi)$ に対して，$f \circ \xi(p) = \zeta \circ g(p)$
  - g を f の(ξ, ζ)-実現と呼ぶ



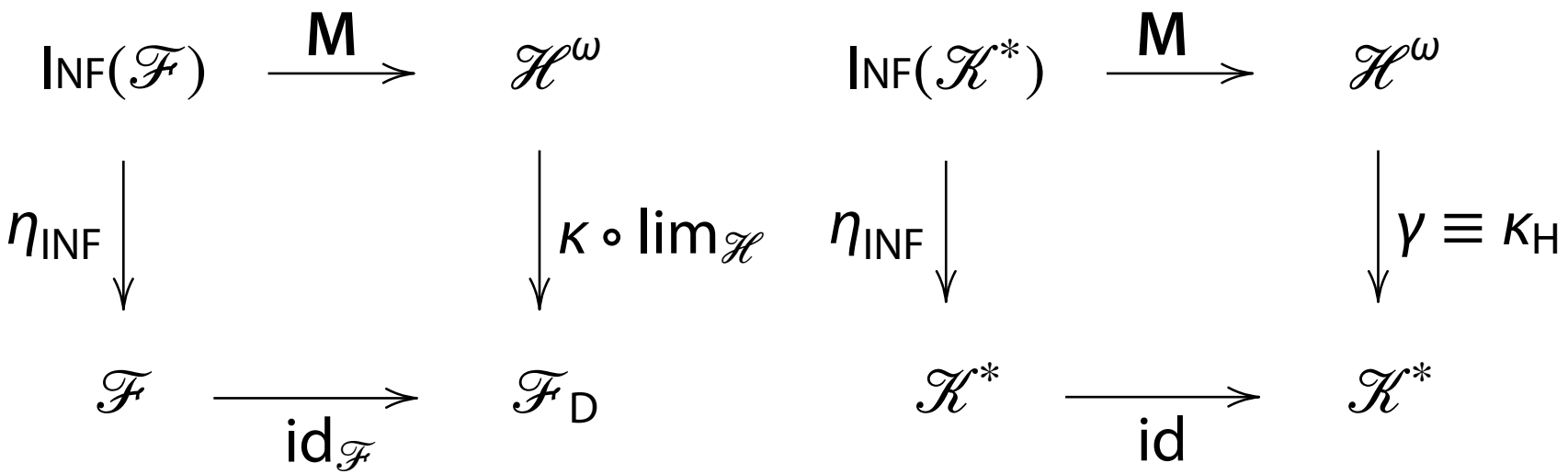
極限での学習と計算の関係

- $\mathscr{F} \subseteq \mathscr{K}^*$ は**FigEx-Inf** 学習可能 $\Longleftrightarrow$
  恒等写像 $\mathrm{id}_{\mathscr{F}} : \mathscr{F} \to \mathscr{F}_{\mathrm{D}}$ が$(\eta_{\mathrm{INF}}, \kappa \circ \lim_{\mathscr{H}})$-計算可能
  - 図形 K とその完全提示 σ に対して $\eta_{\mathrm{INF}}(\sigma) := K$
  - 写像 $\lim_{\mathscr{H}} : \mathscr{H}^{\omega} \to \mathscr{H}$, $\lim_{\mathscr{H}}(\tau) := H$（$\tau$ は H に収束）
  - $\mathrm{Inf}(\mathscr{F})$ は $K \in \mathscr{F}$ の完全提示全体からなる集合
- **FigEx-Txt** 学習でも同様の結果

$$\begin{array}{ccc} \mathrm{Inf}(\mathscr{F}) & \xrightarrow{\mathbf{M}} & \mathscr{H}^{\omega} \\ \Big\downarrow{\scriptstyle \eta_{\mathrm{INF}}} & & \Big\downarrow{\scriptstyle \kappa \circ \lim_{\mathscr{H}}} \\ \mathscr{F} & \xrightarrow[\mathrm{id}_{\mathscr{F}}]{} & \mathscr{F}_{\mathrm{D}} \end{array}$$

実効的学習と計算の関係

- $\mathscr{F} \subseteq \mathscr{K}^*$ は**FigEfEx-Inf** 学習可能 $\Longleftrightarrow$
  ある計算可能関数 g が存在して，
  - g は恒等写像 $\mathrm{id}_{\mathscr{F}}$ の$(\eta_{\mathrm{INF}}, \kappa \circ \lim_{\mathscr{H}})$-実現，かつ
  - g は恒等写像 $\mathrm{id} : \mathscr{K}^* \to \mathscr{K}^*$ の$(\eta_{\mathrm{INF}}, \gamma)$-実現



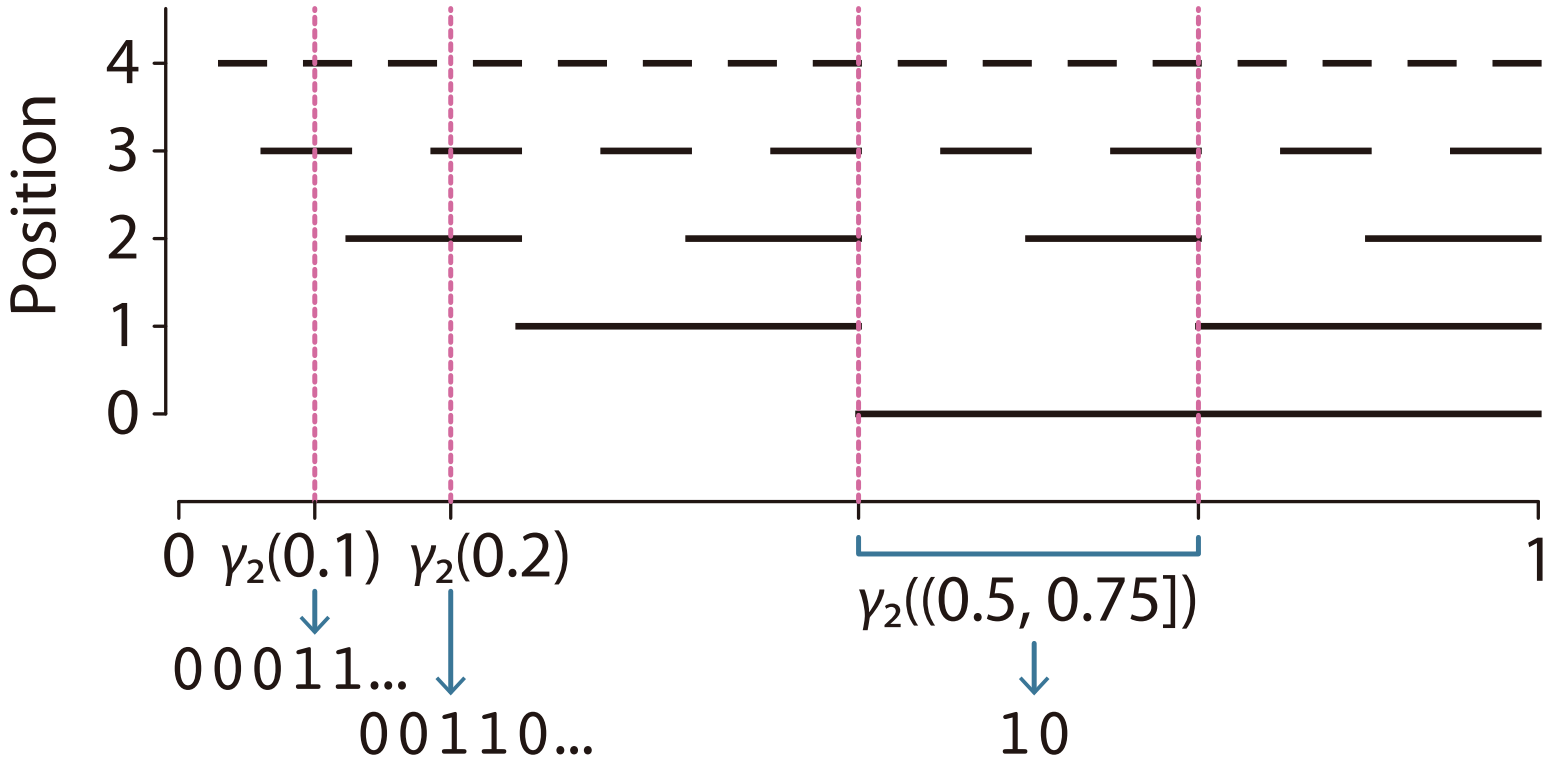
実験科学への貢献

- 実験科学ではコントロール実験が仮説検証の標準的方法
  - 例：2 つの群を比較して新薬の効果を検証する.
    ひとつはプラシーボ（コントロール群),
    もうひとつは新薬（処理群）を与える
- 典型的な手法は統計的仮説検定（*t* 検定など）だが,
  現場で使いづらいことが多い [Johnson, 1999]
  - 検証不可能な仮定や任意に設定可能な *p* の閾値など
- 2 つのクラスを比較すればよい
  → 機械学習で扱うことのできる問題
- 符号化ダイバージェンスが適用可能

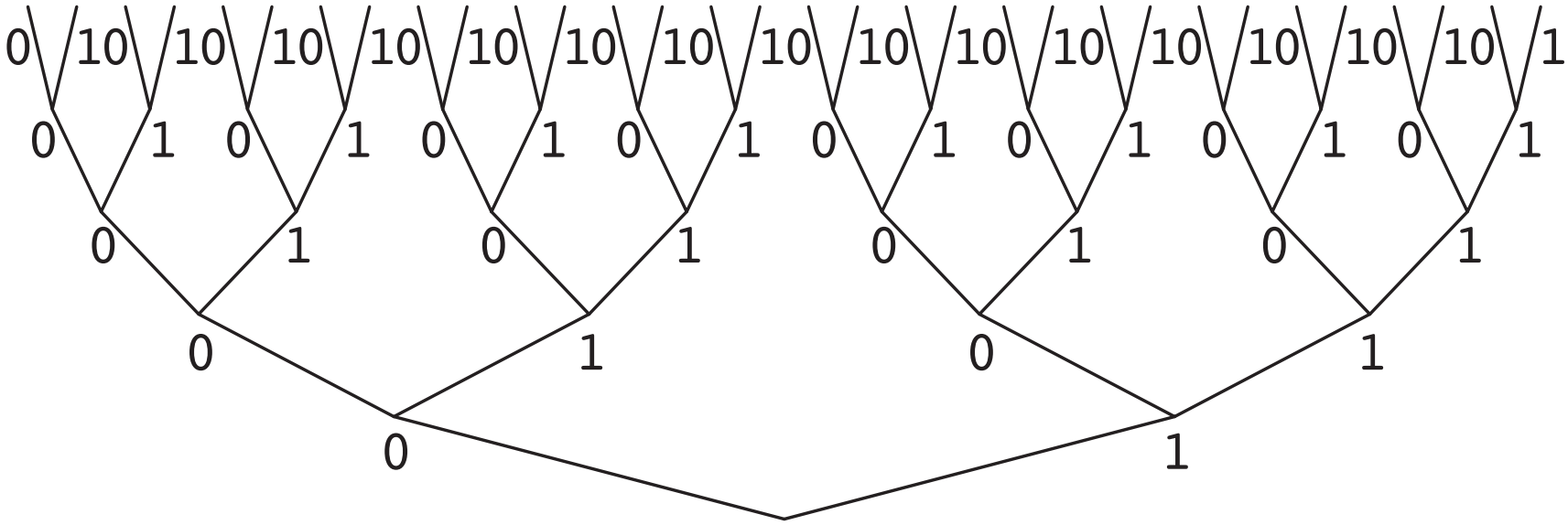
カントール空間への埋め込み

- カントール位相 $\tau_{\Sigma^\omega} := \{\uparrow W \mid W \subseteq \Sigma^*\}$,
  カントール空間とは位相空間 $(\Sigma^\omega, \tau_{\Sigma^\omega})$
  - カントール空間とは，アルファベット Σ 上の無限列の集合 Σ^ω に導かれる標準的な位相空間
  - $\uparrow w = \{p \in \Sigma^\omega \mid w \sqsubseteq p\}$
  - $\uparrow W = \{p \in \Sigma^\omega \mid \exists w \in W.\, w \sqsubseteq p\}$
    - $\{w\Sigma^\omega \mid w \in \Sigma^*\}$ は基底となる
  - 集合 $P \subseteq \Sigma^\omega$ が開集合のとき，P は有限で観察可能
- d 次元ユークリッド空間 $\mathbb{R}^d$ からカントール空間への埋め込み $\gamma :\subseteq \mathbb{R}^d \to \Sigma^\omega$ は，実数値データの符号化に対応
  - 離散化されたデータは開集合の基底
  - 第一部で扱った ρ とは写像の向きが反対

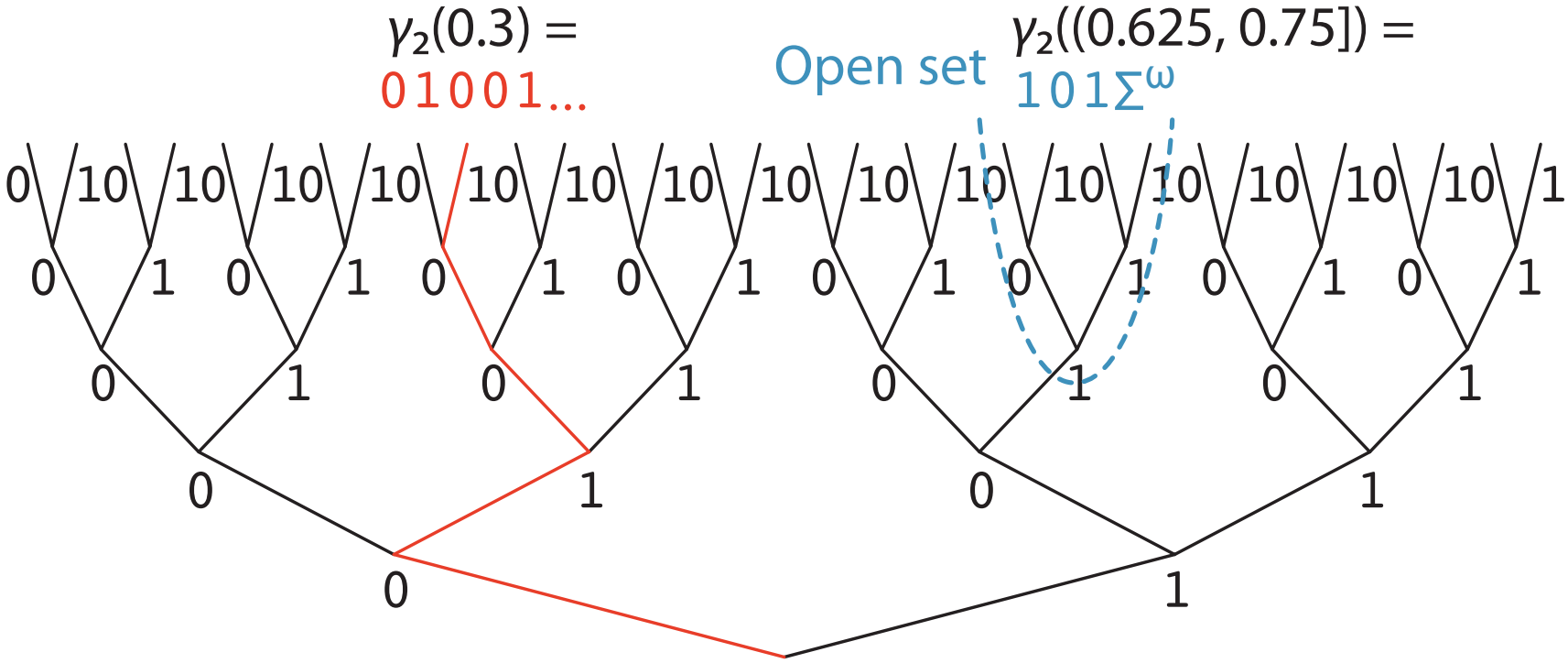
例：2 進埋め込み


Position
4
3
2
1
0
0
$\gamma_2(0.1)$
$\gamma_2(0.2)$
1
$\gamma_2((0.5, 0.75])$
00011...
00110...
10

2 進埋め込みの木表現


0 10 10 10 10 10 10 10 10 10 10 10 10 10 10 1
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 1
0 1 0 1
0 1
0
1

2 進埋め込みの木表現



符号化ダイバージェンスの学習

function MAIN(X, Y, $k_{\max}$)
  $(H_1, H_2) \leftarrow$ LEARNING(X, Y, $\emptyset$, $\emptyset$, 0, $k_{\max}$)
  return $\frac{1}{\|X\|} \sum_{v \in H_1} |v| + \frac{1}{\|Y\|} \sum_{w \in H_2} |w|$

function LEARNING(X, Y, H_1, H_2, k, $k_{\max}$)
  $V \leftarrow$ OBSERVE(X, k), $\quad W \leftarrow$ OBSERVE(Y, k)
  $H_1 \leftarrow H_1 \cup \{ v \in V \mid v \notin W \}$, $\quad H_2 \leftarrow H_2 \cup \{ w \in W \mid w \notin V \}$
  $X \leftarrow \{ x \in X \mid x \notin \rho(H_1 \Sigma^\omega) \}$, $\quad Y \leftarrow \{ y \in Y \mid y \notin \rho(H_2 \Sigma^\omega) \}$
  if $X = \emptyset$ and $Y = \emptyset$ then return (H_1, H_2)
  else if $k = k_{\max}$ then return $(H_1 \cup V, H_2 \cup W)$
  else return LEARNING(X, Y, H_1, H_2, $k+1$, $k_{\max}$)

function OBSERVE(X, k)
  return $\{ \gamma(x)[n] \mid x \in X \}$ $(n = (k+1)d - 1)$

COOL アルゴリズム

Input: A data set X, two lower bounds K and N
Output: The optimal partition $\mathscr{C}_{\text{op}}$ and noises

function COOL(X, K, N)

1: Find partitions $\mathscr{C}^1_{\geqslant N}, \ldots, \mathscr{C}^m_{\geqslant N}$ such that $\|\mathscr{C}^{m-1}_{\geqslant N}\| < K \leqslant \|\mathscr{C}^m_{\geqslant N}\|$
2: $(\mathscr{C}_{\text{op}}, \text{MCL}(\mathscr{C}_{\text{op}})) \leftarrow$ FINDCLUSTERS(X, K, $\{\mathscr{C}^1_{\geqslant N}, \ldots, \mathscr{C}^m_{\geqslant N}\}$)
3: return $(\mathscr{C}_{\text{op}}, X \setminus \bigcup \mathscr{C}_{\text{op}})$

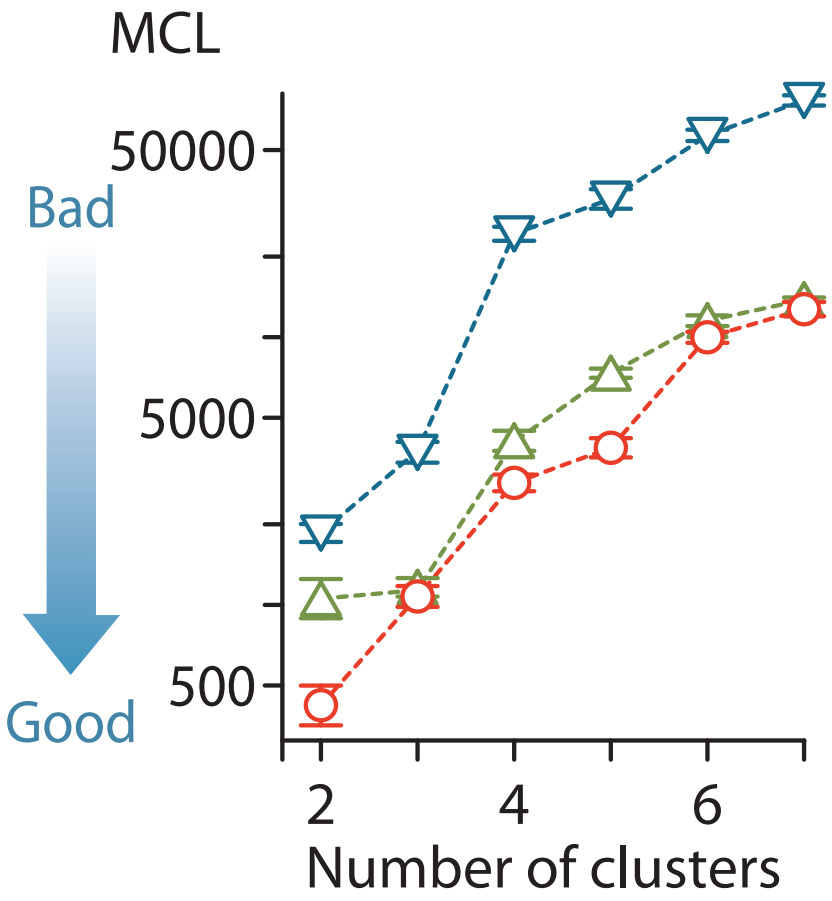
function FINDCLUSTERS(X, K, $\{\mathscr{C}^1, \ldots, \mathscr{C}^m\}$)

1: Find k such that $\|\mathscr{C}^{k-1}\| < K$ and $\|\mathscr{C}^k\| \geqslant K$
2: $\mathscr{C}_{\text{op}} \leftarrow \mathscr{C}^k$
3: if $K = 2$ then return $(\mathscr{C}_{\text{op}}, \text{MCL}(\mathscr{C}_{\text{op}}))$
4: for each C in $\mathscr{C}^1 \cup \ldots \cup \mathscr{C}^{k-1}$
5: $(\mathscr{C}, L) \leftarrow$ FINDCLUSTERS($X \setminus C$, $K - 1$, $\{\mathscr{C}^1, \ldots, \mathscr{C}^k\}$)
6: if $\text{MCL}(\mathscr{C} \cup C) < \text{MCL}(\mathscr{C}_{\text{op}})$ then $\mathscr{C}_{\text{op}} \leftarrow C \cup \mathscr{C}$
7: return $(\mathscr{C}_{\text{op}}, \text{MCL}(\mathscr{C}_{\text{op}}))$

実験手法

- 合成データと実データを用いて，G-COOL を DBSCANと*K*-meansと比較して評価する
  - 合成データは R パッケージ *clusterGeneration* を持ちいて生成
    - 各クラスタは $n = 1,500$ で $d = 3$
  - 実データはEarth-as-Artから得た地理的画像
    - 200 × 200 ピクセルにして，2 値画像に変換する
  - すべてのデータは min-max 正規化で前処理する
- G-COOL は R (version 2.12.1) で実装
- 以下の基準を用いる
  - MCL, connectivity, Silhouette width（正解ラベルはなし）
  - adjusted Rand index（正解ラベルを用いる）

実験結果（合成データ）

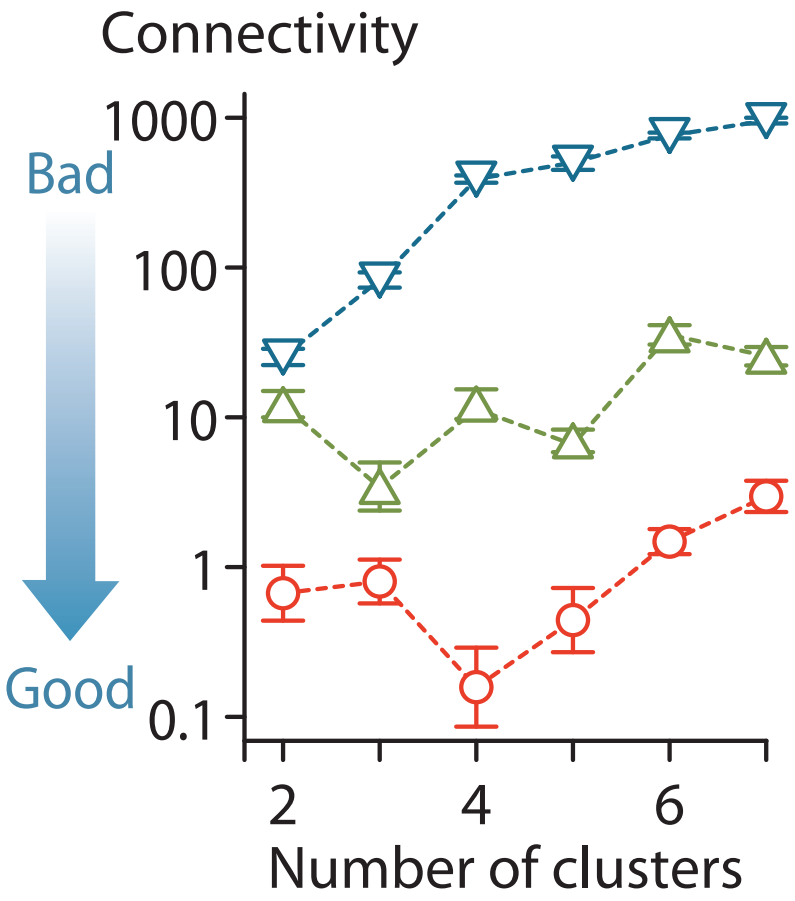


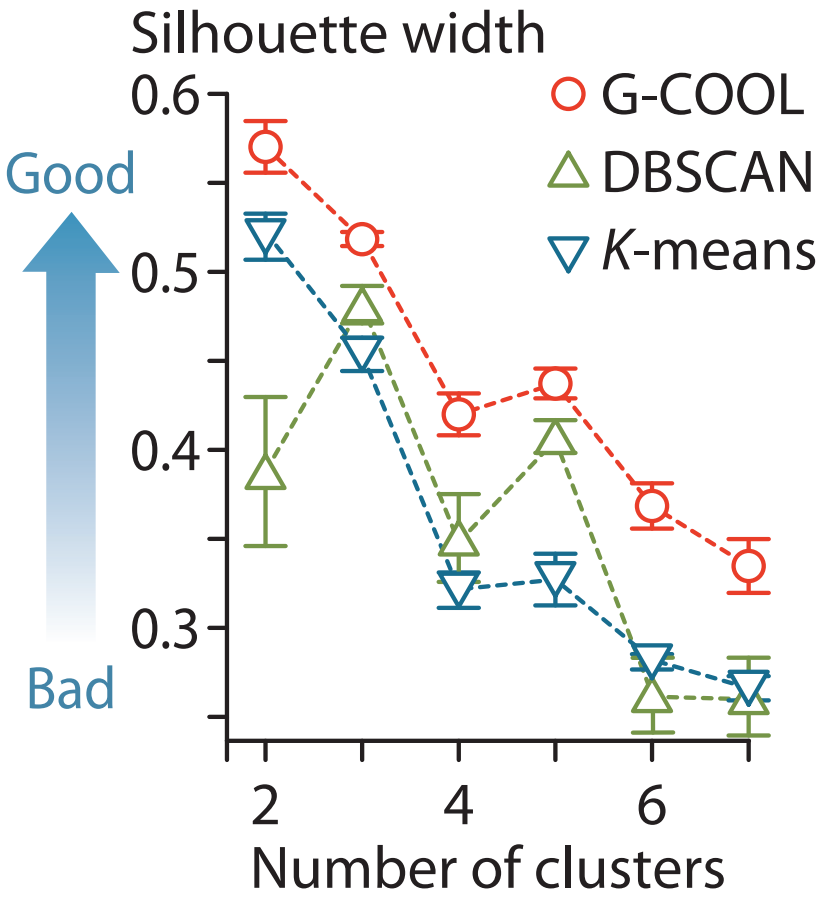
○ G-COOL
△ DBSCAN
▽ *K*-means

Data show mean ± s.e.m.

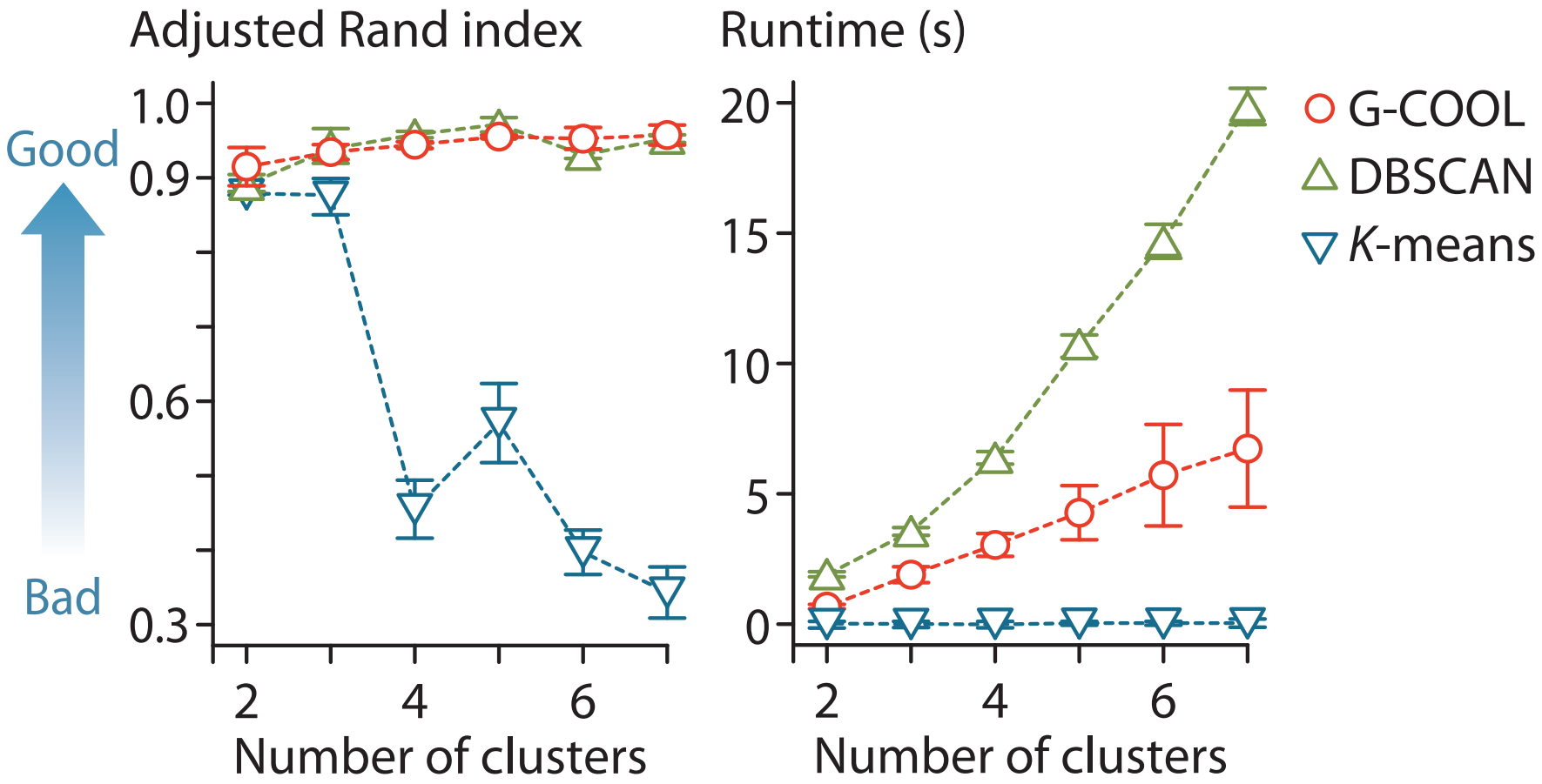
Each experiment was performed 20 times

実験結果（合成データ）

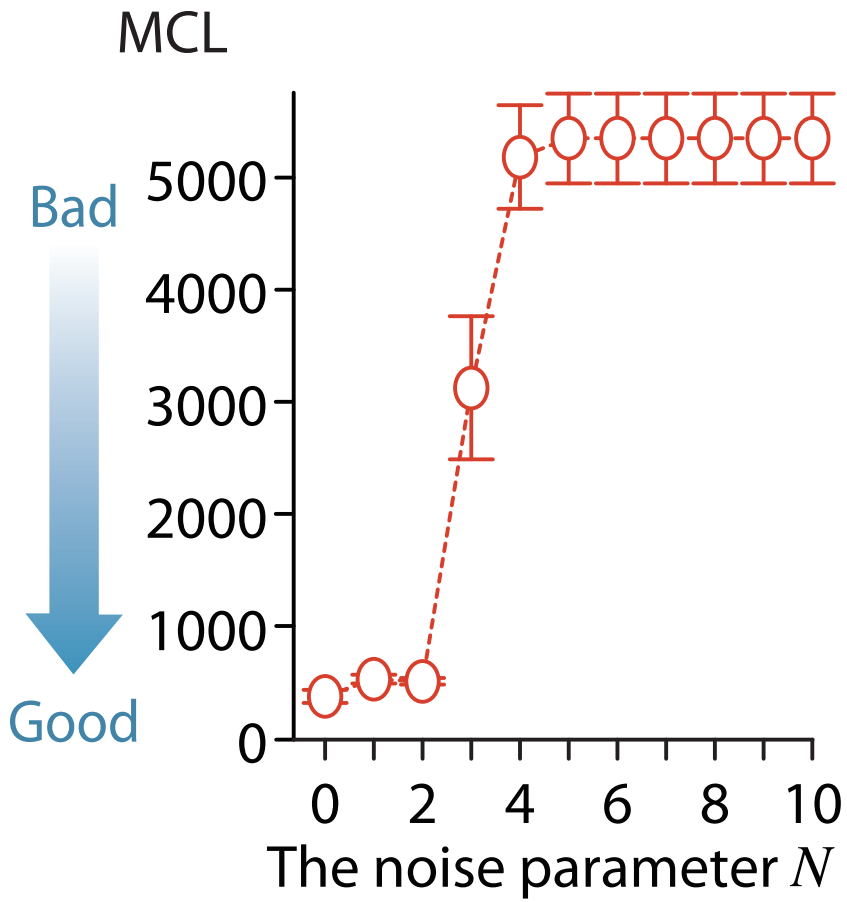

Connectivity
Bad
Good
1000
100
10
1
0.1
2
4
6
Number of clusters


Silhouette width
Good
Bad
0.6
0.5
0.4
0.3
G-COOL
DBSCAN
K-means
2
4
6
Number of clusters

実験結果（合成データ）


Adjusted Rand index
Runtime (s)
Good
Bad
1.0
0.9
0.6
0.3
20
15
10
5
0
2
4
6
Number of clusters
G-COOL
DBSCAN
K-means

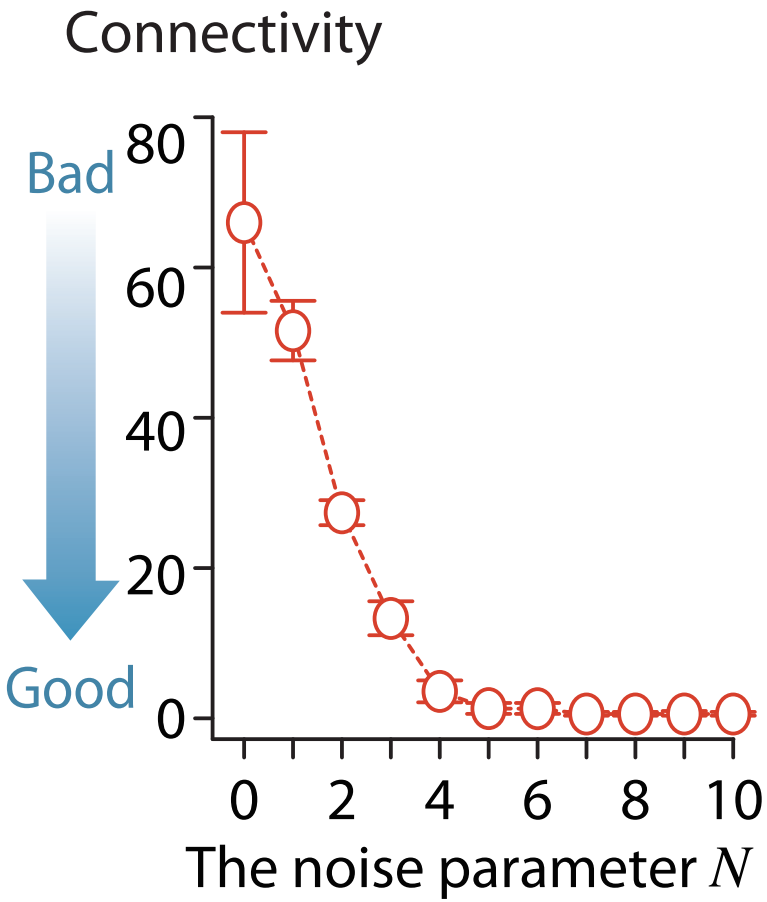
実験結果（合成データ）

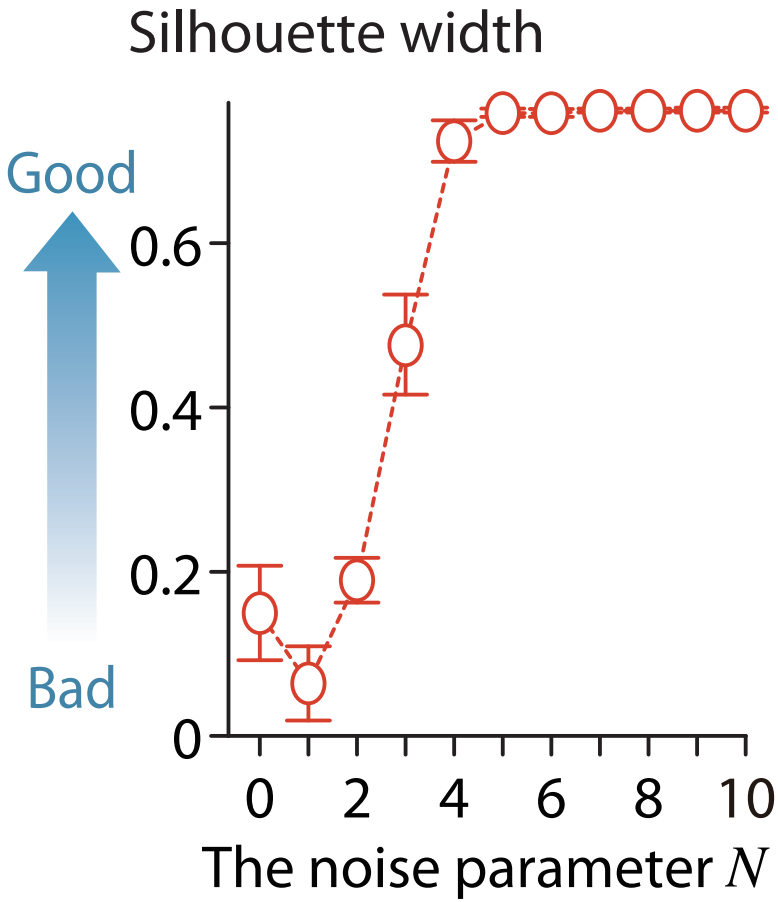


Data show mean ± s.e.m.

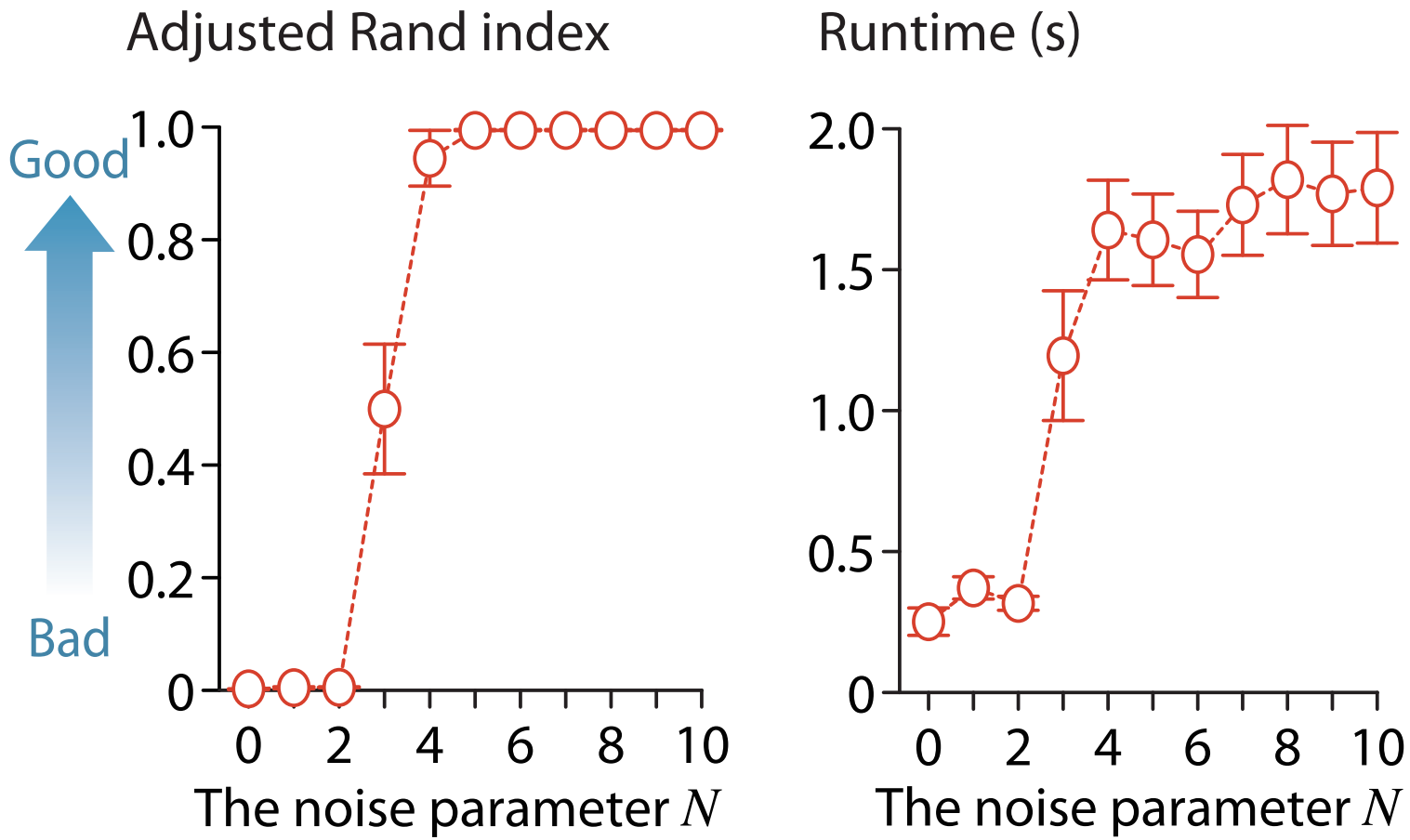
Each experiment was performed 20 times

実験結果（合成データ）

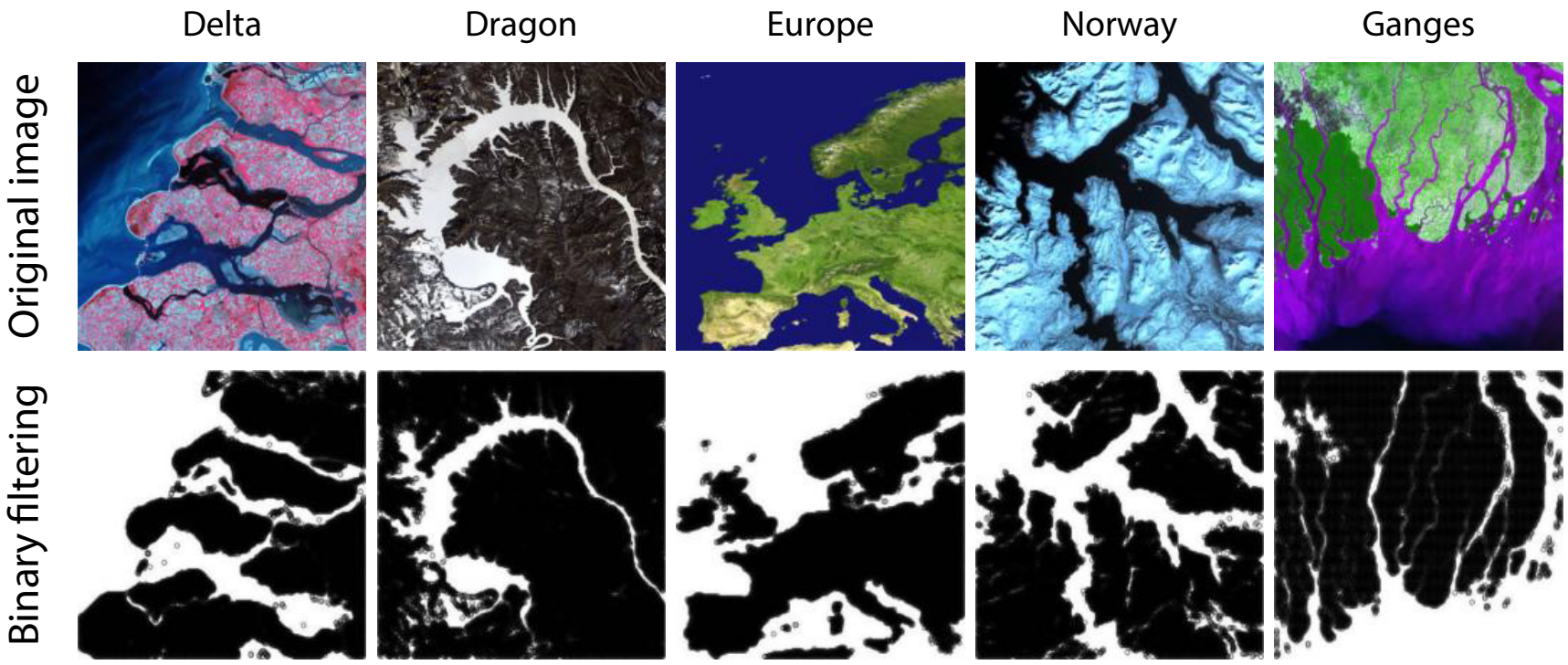

Connectivity
Bad
Good
80
60
40
20
0
0
2
4
6
8
10
The noise parameter N


Silhouette width
Good
Bad
0.6
0.4
0.2
0
0
2
4
6
8
10
The noise parameter N

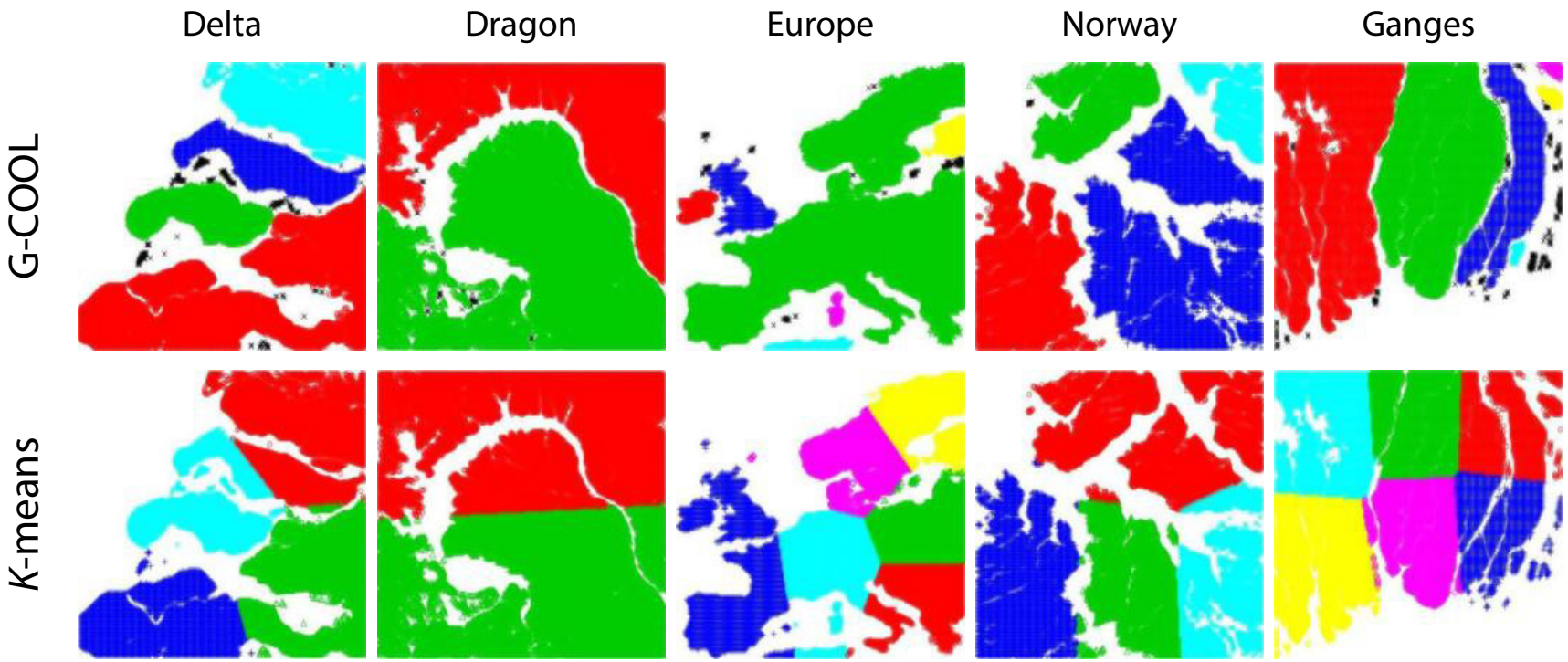
実験結果（合成データ）


Adjusted Rand index
Good
Bad
1.0
0.8
0.6
0.4
0.2
0
0
2
4
6
8
10
The noise parameter N
Runtime (s)
2.0
1.5
1.0
0.5
0
0
2
4
6
8
10
The noise parameter N

実験結果（実データ）


Delta
Dragon
Europe
Norway
Ganges
Original image
Binary filtering

実験結果（実データ）



実験結果（実データ）

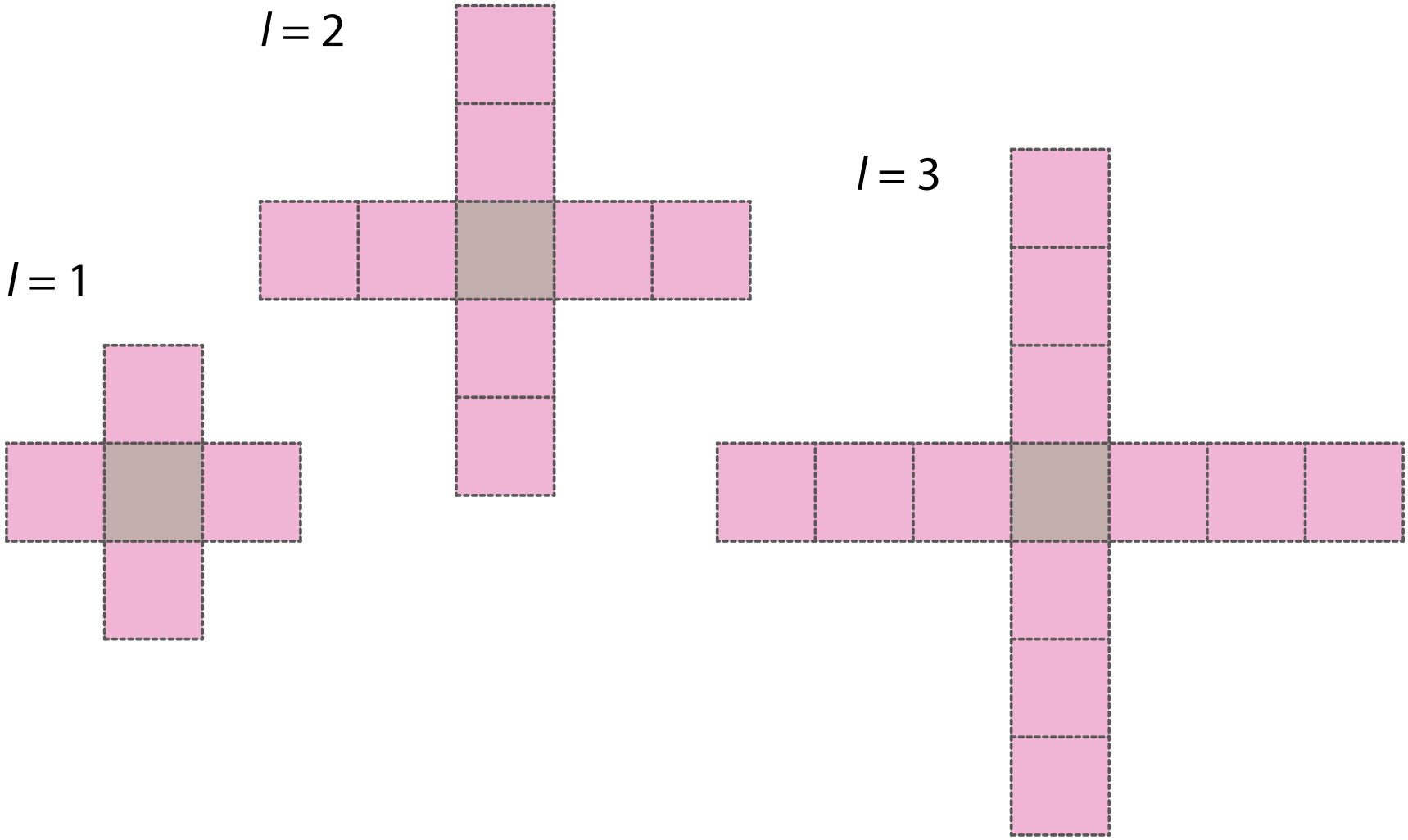
Name	n	K	Running time (s)		MCL	
			GC	KM	GC	KM
Delta	20748	4	1.158	0.012	4010	4922
Dragon	29826	2	0.595	0.026	3906	7166
Europe	17380	6	2.404	0.041	2320	12210
Norway	22771	5	0.746	0.026	1820	6114
Ganges	18019	6	0.595	0.026	2320	12526

GC: G-COOL, KM: *K*-means

様々なクラスタリング手法

- 多くのアルゴリズムが提案されている
  - 詳しくは [Berkhin, 2006; Halkidi *et al.*, 2001; Jain *et al.*, 1999]
- Partitional algorithms
  - ABACUS [Chaoji *et al.*, 2011], SPARCL [Chaoji *et al.*, 2009]
- Mass-based algorithms [Ting and Wells, 2010]
- Density-based algorithms
  - DBSCAN [Ester *et al.*, 1996]
- Hierarchical clustering algorithms
  - CURE [Guha *et al.*, 1998], CHAMELEON [Karypis *et al.*, 1999]
- Grid-based algorithms
  - STING [Wang *et al.*, 1997]

距離パラメータ *l*



BOOL アルゴリズム (1/3)

Input: Database X,
lower bound on number of clusters K,
noise parameter N, and
distance parameter l
Output: Partition $\mathscr{C}$

function BOOL(X, K, N)
8: $k \leftarrow 1$ // k is level of discretization
9: **repeat**
10: $\mathscr{C} \leftarrow$ MAKEHIERARCHY(X, k, N)
11: $k \leftarrow k + 1$
12: **until** $\#\mathscr{C} \geqslant K$
13: output $\mathscr{C}$

BOOL アルゴリズム (2/3)

function MAKEHIERARCHY(X, k, N)

1: $\mathscr{C} \leftarrow \{\{x\} \mid x \in X\}$
2: $h \leftarrow d$ // d is number of attributes of X
3: $X_D \leftarrow \Delta^k(X)$ // discretize X at level k
4: $X \leftarrow S_{\pi_1(X_D)} \circ S_{\pi_2(X_D)} \circ \ldots \circ S_{\pi_d(X_D)}(X)$
5: **repeat**
6: $X \leftarrow S_{\pi_h(X_D)}(X)$
7: $\mathscr{C} \leftarrow$ AGGL(X, $\mathscr{C}$, h)
8: $h \leftarrow h - 1$
9: **until** $h = 0$
10: $\mathscr{C} \leftarrow \{C \in \mathscr{C} \mid \#C \geqslant N\}$
11: output $\mathscr{C}$

BOOL アルゴリズム (3/3)

function AGGL(X, $\mathscr{C}$, h)

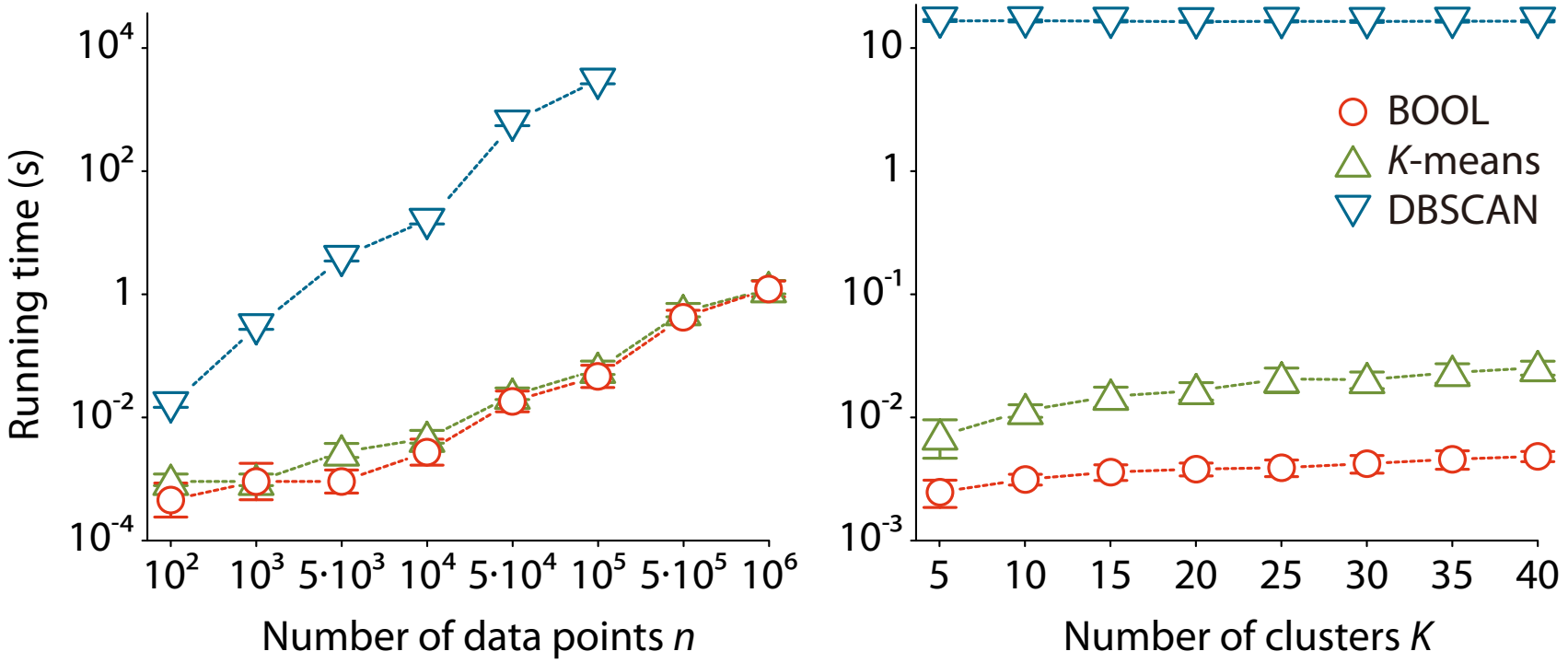
1: **for each** object x of X
2: $y \leftarrow$ successive object of x
3: **if** $d_0(\Delta^k(x), \Delta^k(y)) \leqslant 1$ and $d_\infty(\Delta^k(x), \Delta^k(y)) \leqslant l$ **then**
4: delete $C \ni x$ and $D \ni y$ from $\mathscr{C}$, and add $C \cup D$
5: **end if**
6: **end for**
7: output $\mathscr{C}$

BOOL によるノイズ除去

- ノイズ除去はクラスタリングにおいて重要
- 分割 $\mathscr{C}$ に対して，$\mathscr{C}_{\geqslant N} := \{C \in \mathscr{C} \mid \#C \geqslant N\}$ と定義する
  - クラスタ C を，$\#C < N$ のときノイズとみなす
- 例：$\mathscr{C} = \{\{0.1\}, \{0.4, 0.5, 0.6\}, \{0.9\}\}$ とする
  - $\mathscr{C}_{\geqslant 2} = \{\{0.4, 0.5, 0.6\}\}$ であり，0.1 と 0.9 はノイズ
- クラスタサイズの下限 N をノイズパラメータとして入力

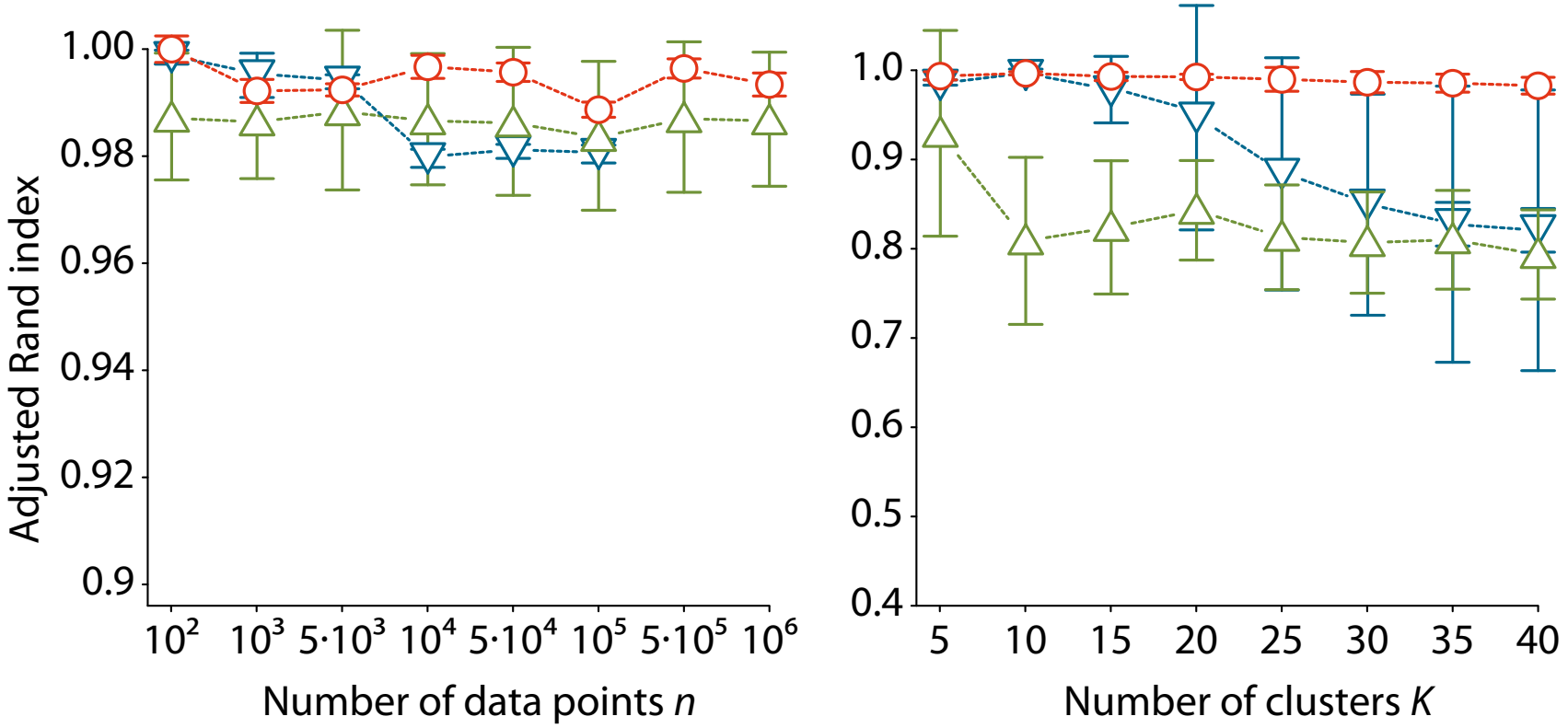
データ数とクラスタ数に関する速度

- ランダムに生成された合成データ
  - $K = 5$（左），$n = 10000$（右）



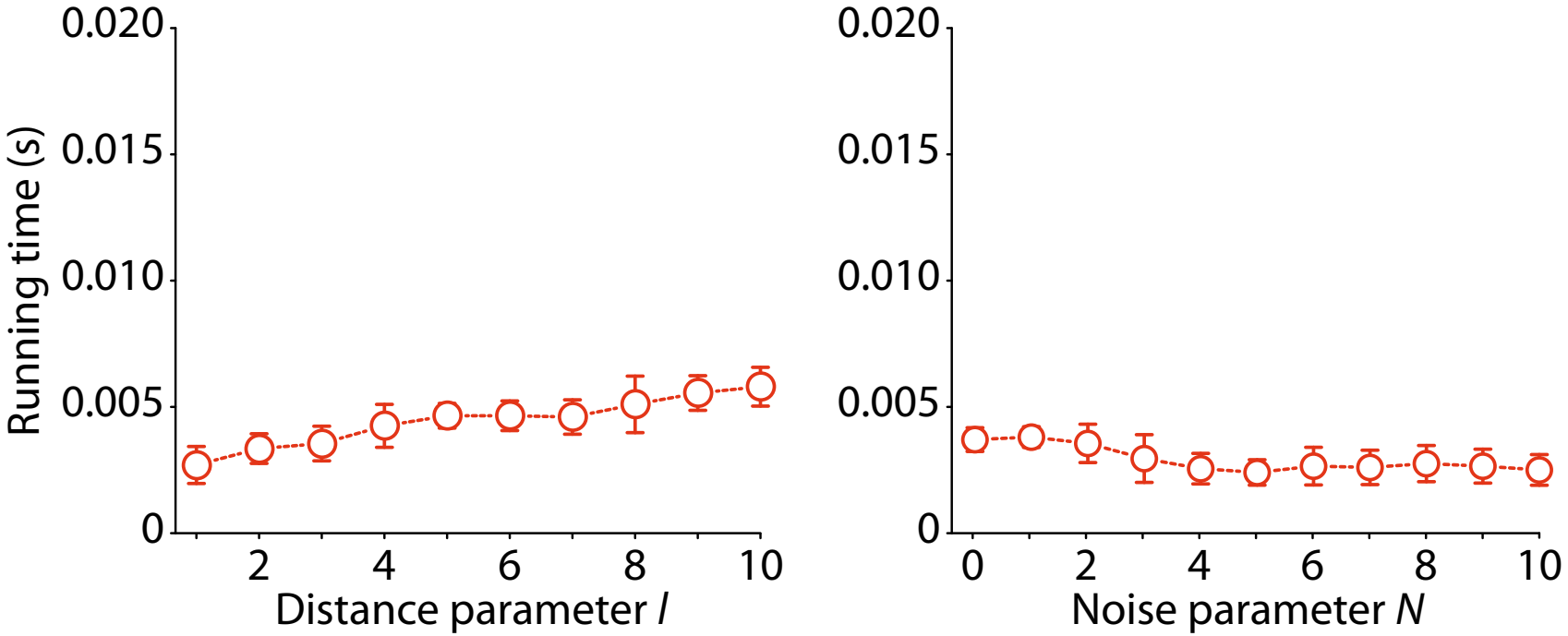
データ数とクラスタ数に関する質

- ランダムに生成された合成データ
  - $K = 5$（左），$n = 10000$（右）



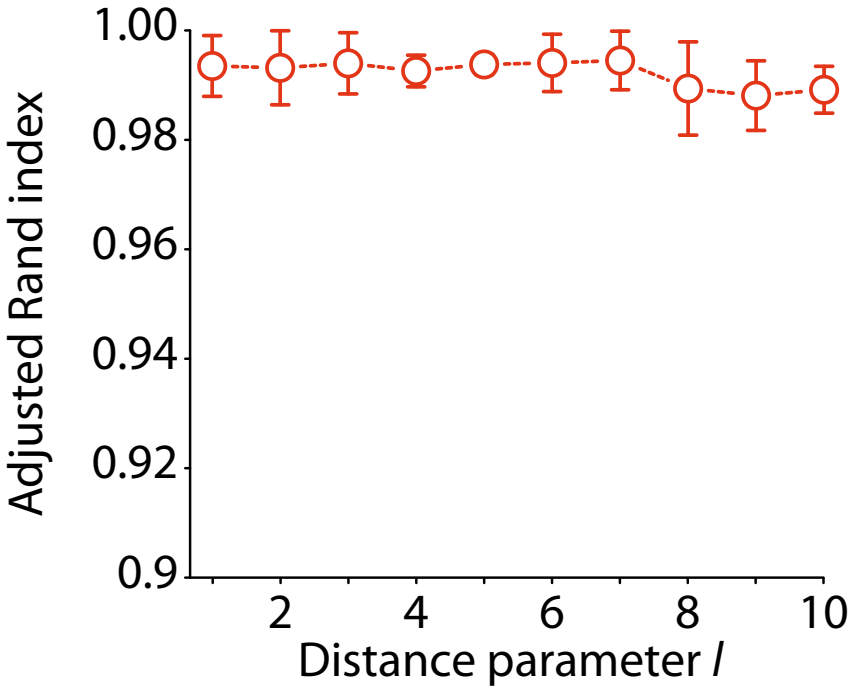
入力パラメータに関する速度

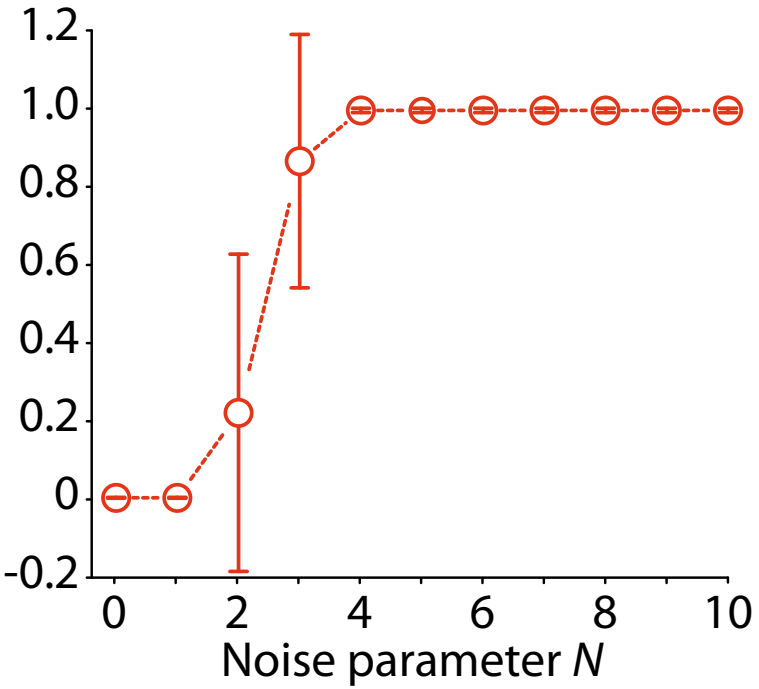
- ランダムに生成された合成データ
  - $K = 5$, $n = 10000$



入力パラメータに関するクラスタの質

- ランダムに生成された合成データ
  - $K = 5$，$n = 10000$





混在データ

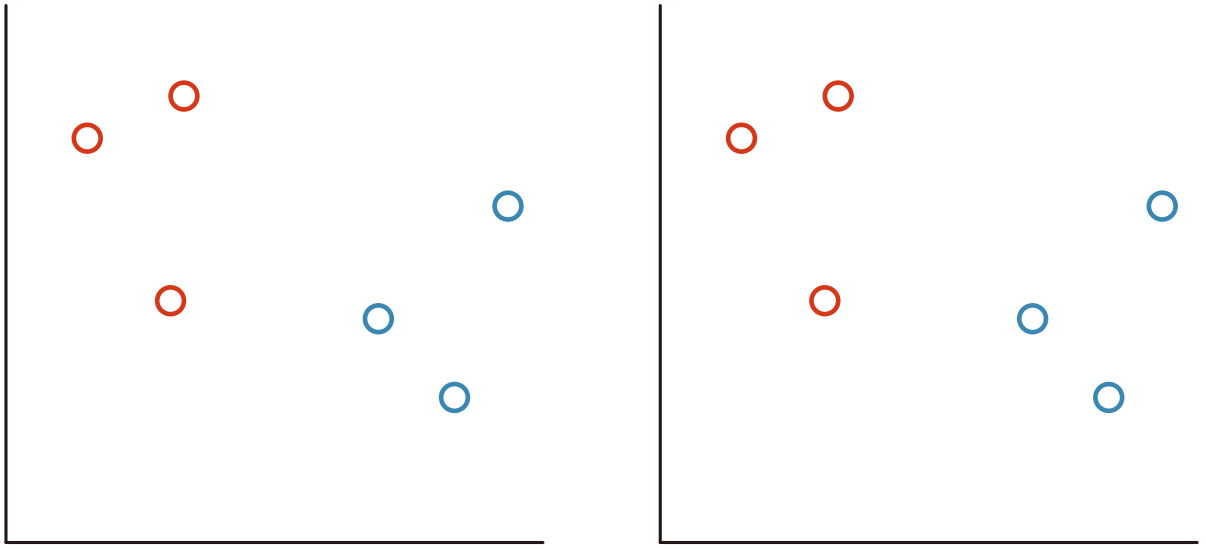
- 多くの離散値・連続値混在データが存在する
  - 例：不正侵入検出のためのトラフィックデータ
  - 離散変数：2 値（**T**, **F**），名義 (A, B, . . . , *v*) など
    - 例：性別，購買履歴，アンケートの回答，...
  - 連続変数：実数値（$\mathbb{R}$）
    - 測定や観測で得られるデータが多い
- 混在データを直接扱える機械学習手法は少ない
  - 例：決定木（C4.5）など
- 現代的な手法が求められている

不完全データ

- NULL（⊥）値を持つ不完全なデータをどのように扱うのか，は重要な問題
  - 様々な種類のNULLがある
- 以下の 2 種類のNULLを扱う
  - 欠損値：データのある属性の値が欠損
  - 欠損ラベル：データのクラスラベルが欠損
- さらに，離散化された実数値データの誤差も考慮する
  - 例：実数 $\pi = 3.1415\ldots$ が 3.1 に離散化されると，それに続くビットはわからない
  - この誤差は，多くの統計的手法では無視されている

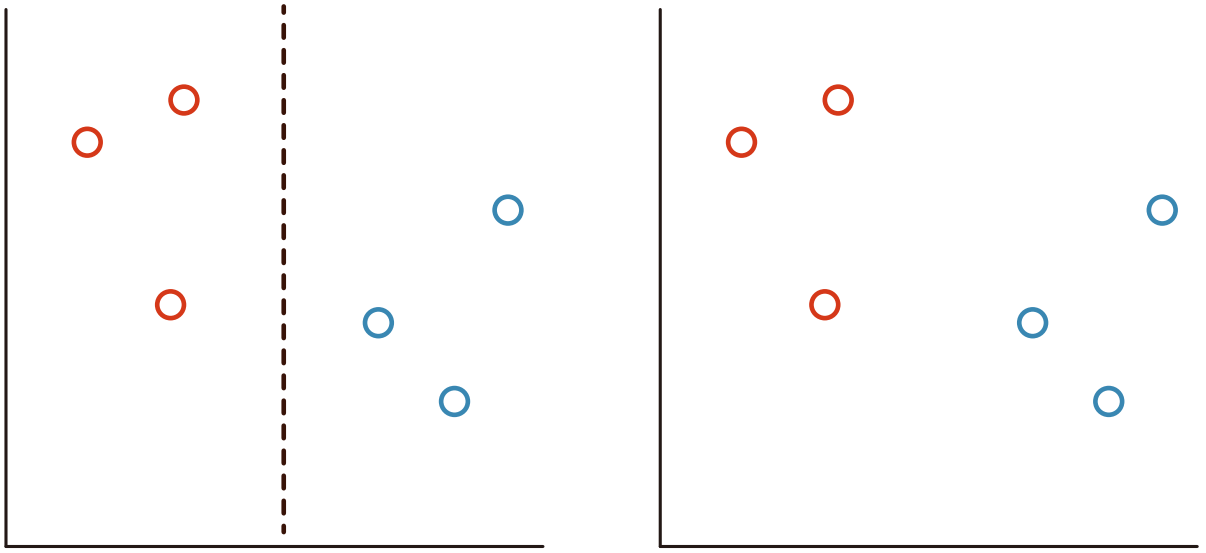
半教師あり学習

- クラス分類の特殊系
- 目標：（大量の）ラベル無しデータを効果的に使うことで，より良い分類器を構築する [Zhu and Goldberg, 2009]



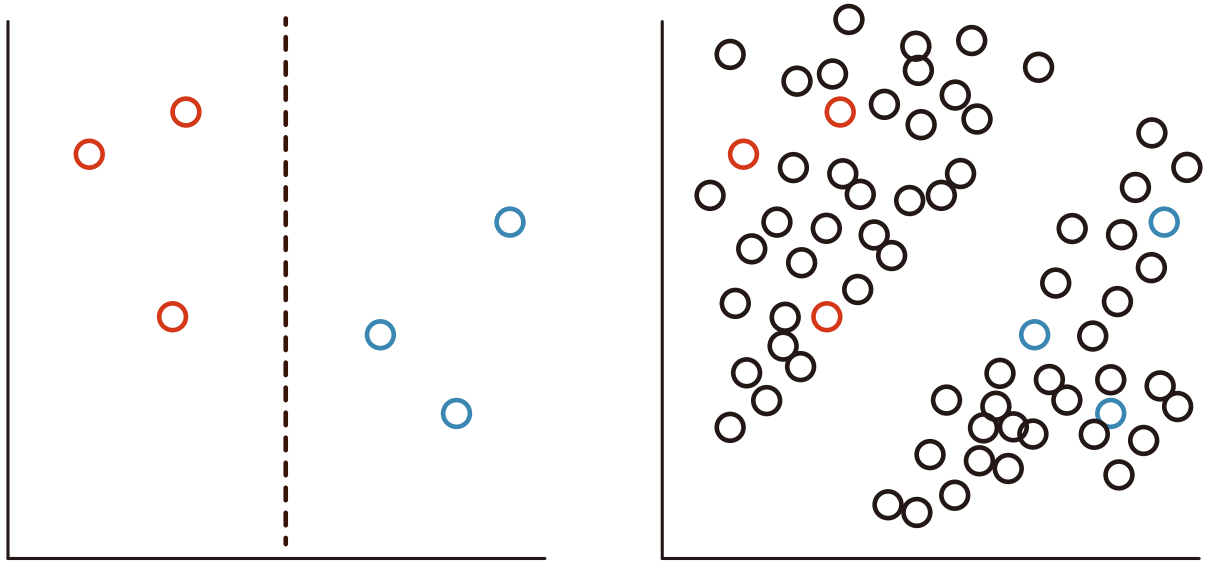
半教師あり学習

- クラス分類の特殊系
- 目標：（大量の）ラベル無しデータを効果的に使うことで，より良い分類器を構築する [Zhu and Goldberg, 2009]



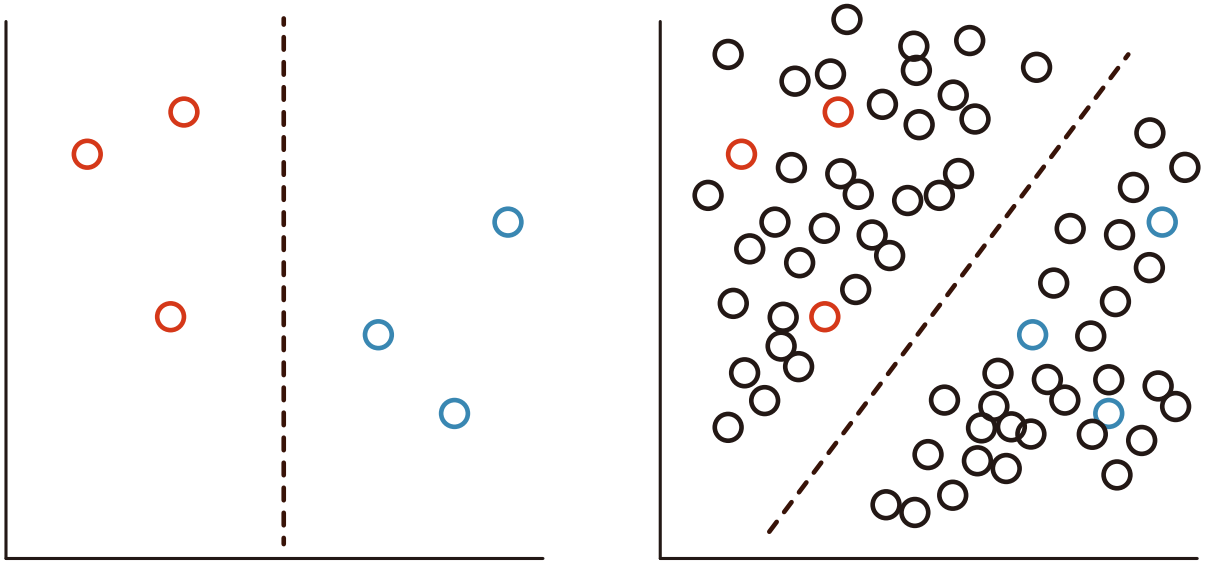
半教師あり学習

- クラス分類の特殊系
- 目標：（大量の）ラベル無しデータを効果的に使うことで，より良い分類器を構築する [Zhu and Goldberg, 2009]



半教師あり学習

- クラス分類の特殊系
- 目標：（大量の）ラベル無しデータを効果的に使うことで，より良い分類器を構築する [Zhu and Goldberg, 2009]

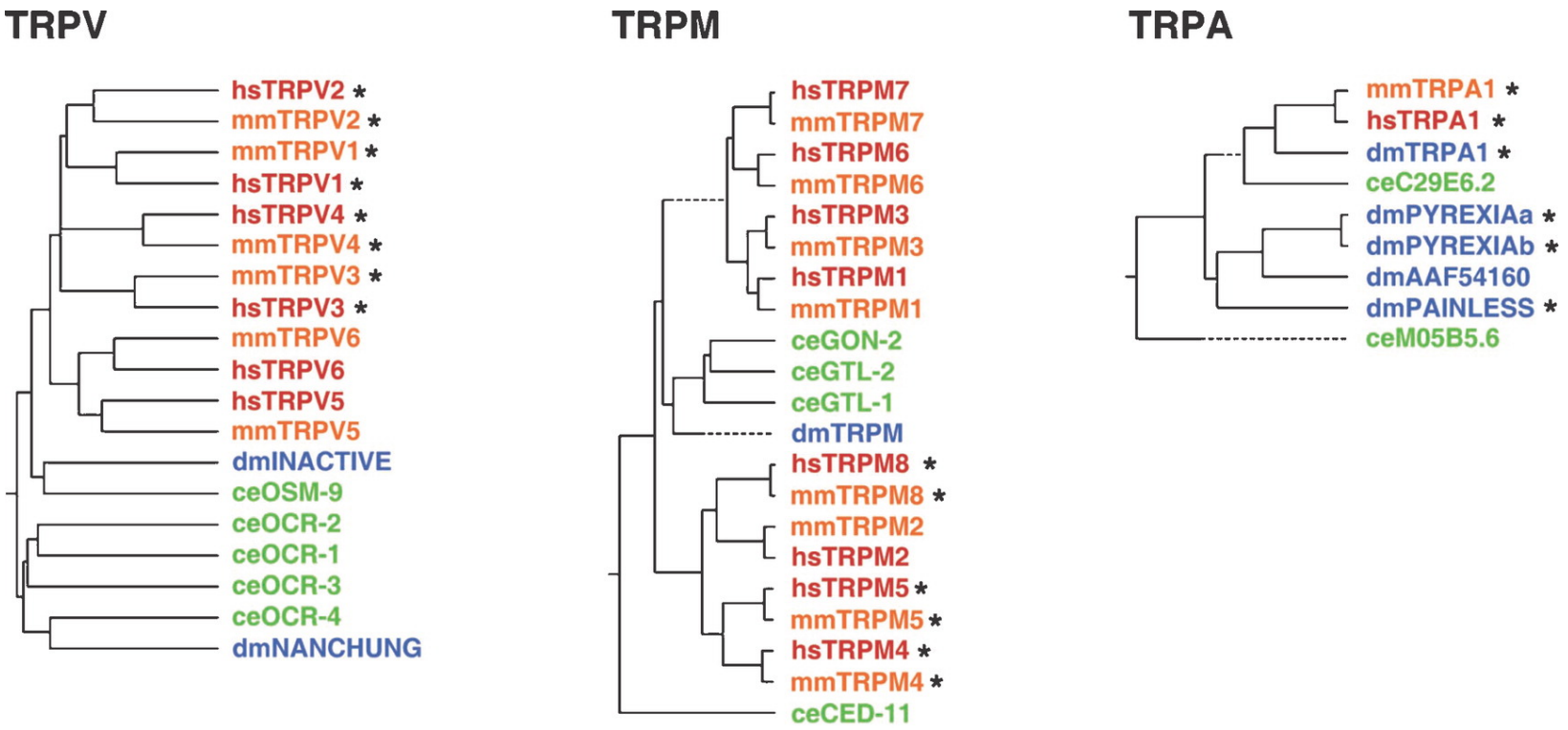


半教師あり学習における問題点

- クラス分類の特殊系
- 目標：（大量の）ラベル無しデータを効果的に使うことで，
  より良い分類器を構築する [Zhu and Goldberg, 2009]
- しかし，離散値・連続値混在データを直接扱う
  半教師あり学習手法は存在しない

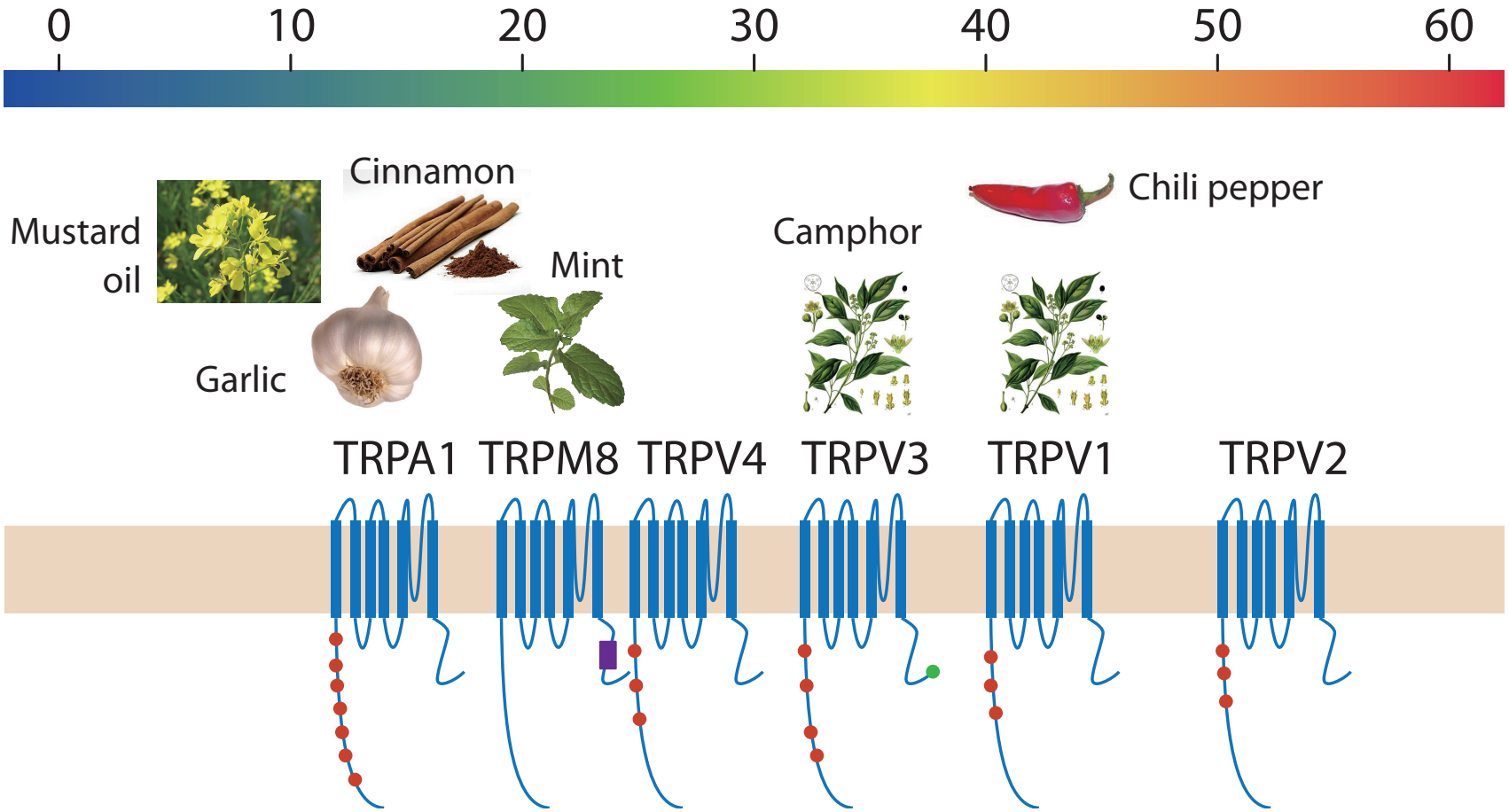
例：TRP イオンチャネル

- TRP イオンチャネルはリガンドファミリーの１つ

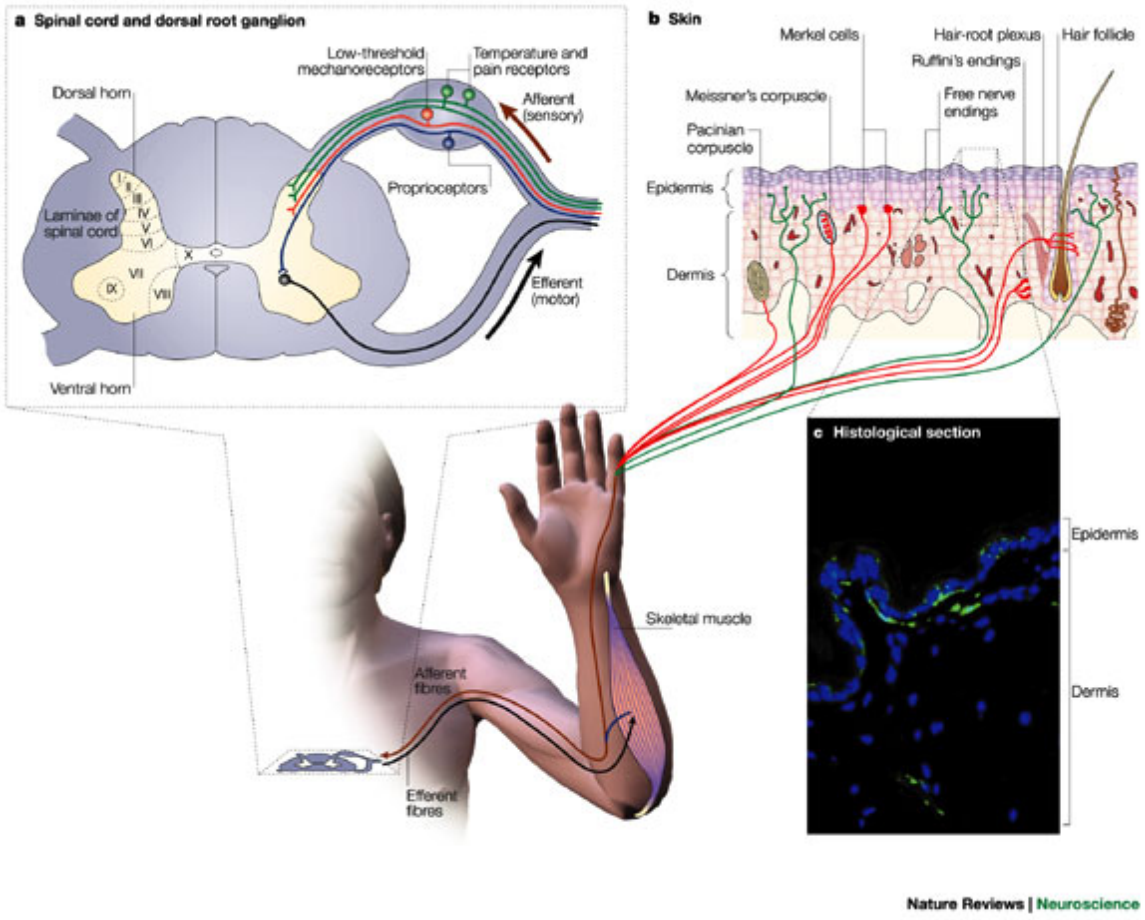


Dhaka A, et al. 2006.
Annu. Rev. Neurosci. 29:135–61

ThermoTRPs


0
10
20
30
40
50
60
Mustard oil
Cinnamon
Garlic
Mint
Camphor
Chili pepper
TRPA1
TRPM8
TRPV4
TRPV3
TRPV1
TRPV2

ThermoTRPs のシグナル経路



Patapoutian, A *et al.*, Nat. Rev. Neurosci. 4, 529–539

データ前処理アルゴリズム

Input: Table $\tau = (H, X)$ and discretization level k
Output: Context (G, M^k, I^k)

function Context(τ, k)
8: $G \leftarrow \text{set}(X)$
9: for each feature $h \in H$
10: if Drom(h) $= \mathbb{N}$ then $(M_h, I_h) \leftarrow$ ContextD(X, h)
11: else if Drom(h) $= \mathbb{R}$ then $(M_h, I_h) \leftarrow$ ContextC(X, h, k)
12: end if
13: combine $(M_{\text{HBA}}, I_{\text{HBA}}), (M_{\text{HBD}}, I_{\text{HBD}}), \ldots, (M_{\text{NLR}}, I_{\text{NLR}})$ into (M^k, I^k)
14: return (G, M^k, I^k)

データ前処理アルゴリズム

function ContextD(X, h)
1: $M \leftarrow \{h.m \mid m \in x(h) \text{ such that } x \in \text{set}(X)\}$
2: $I \leftarrow \{(x, h.m) \mid x \in \text{set}(X) \text{ and } x(h) = m\}$
3: return (M, I)

function ContextC(X, h, k)
1: $M \leftarrow \{1, 2, \ldots, 2^k\}, I \leftarrow \emptyset$
2: Normalize the set $\{x(h) \mid x \in \text{set}(X)\}$
3: for each $x \in \text{set}(X)$
4: if $x(h) = 0$ then $I \leftarrow I \cup \{(x, h.1)\}$
5: else if $x(h) \neq 0$ then
6: $I \leftarrow I \cup \{(x, h.a)\}$, where $(a-1) \cdot 2^{-k} < x(h) \leqslant a \cdot 2^{-k}$
7: end if
8: end for
9: return (M, I)

LIFT アルゴリズム (1/2)

Input: Tables $\tau = (H, X)$ and $\upsilon = (H, y)$, and maximum level $k_{\max}$
Output: Preference ψ_y for each label $\lambda \in \mathscr{L}$

function LIFT(τ, υ, $k_{\max}$)
1: $k \leftarrow 1$ // k is discretization level
2: for each label $\lambda \in \mathscr{L}$
3: $\psi_y(\lambda|\tau) \leftarrow 0$ // initialization
4: end for
5: return Learning(τ, υ, k, $k_{\max}$)

LIFT アルゴリズム (2/2)

function Learning($\tau, \upsilon, k, k_{\max}$)

1: $(G(\tau), M^k(\tau), I^k(\tau)) \leftarrow \text{Context}(\tau, k)$ // make a context from τ
2: $(G(\upsilon), M^k(\upsilon), I^k(\upsilon)) \leftarrow \text{Context}(\upsilon, k)$ // make a context from υ
3: make a concept lattice $k(\tau)$ from $(G(\tau), M^k(\tau), I^k(\tau))$ by FCA
4: for each label $\lambda \in \mathscr{L}$
5: compute the preference $\psi_y^k(\lambda|X)$ at discretization level k
6: $\psi_y(\lambda|X) \leftarrow \psi_y(\lambda|X) + \psi_y^k(\lambda|X)$
7: end for
8: if $k = k_{\max}$ then
9: return $(\psi_y(\lambda|\tau))_{\lambda \in \mathscr{L}}$
10: else
11: return Learning$(\tau, \upsilon, k+1, k_{\max})$
12: end if

参考文献

[Ballester and Mitchell, 2010] P. J. Ballester and J. B. O. Mitchell. A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. *Bioinformatics*, 26(9):1169–1175, 2010.

[Berkhin, 2006] P. Berkhin. A survey of clustering data mining techniques. *Grouping Multidimensional Data*, pages 25–71, 2006.

[Chaoji *et al.*, 2009] V. Chaoji, M. A. Hasan, S. Salem, and M. J. Zaki. SPARCL: An effective and efficient algorithm for mining arbitrary shape-based clusters. *Knowledge and Information Systems*, 21(2):201–229, 2009.

[Chaoji *et al.*, 2011] V. Chaoji, G. Li, H. Yildirim, and M. J. Zaki. ABACUS: Mining arbitrary shaped clusters from large datasets based on backbone identification. In *Proceedings of 2011 SIAM International Conference on Data Mining*, pages 295–306, 2011.

[Cheng *et al.*, 2010] W. Cheng, M. Rademaker, B. De Baets, and E. Hüllermeier. Predicting partial orders: Ranking with abstention. In J. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, editors,

Machine Learning and Knowledge Discovery in Databases, volume 6321 of *Lecture Notes in Computer Science*, pages 215–230. Springer, 2010.

[Cornell *et al.*, 1995] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *Journal of the American Chemical Society*, 117(19):5179–5197, 1995.

[Ester *et al.*, 1996] M. Ester, H. P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, volume 96, pages 226–231, 1996.

[Falconer, 2003] Kenneth Falconer. *Fractal Geometry: Mathematical Foundations and Applications*. Wiley, November 2003.

[Gohlke *et al.*, 2000] H. Gohlke, M. Hendlich, and G. Klebe. Knowledge-based scoring function to predict protein-ligand interactions1. *Journal of molecular biology*, 295(2):337–356, 2000.

[Goodman and Kruskal, 1979] L. Goodman and W. Kruskal. *Measures of As-*

sociation for Cross Classifications. Springer, 1979.

[Guha *et al.*, 1998] S. Guha, R. Rastogi, and K. Shim. CURE: An efficient clustering algorithm for large databases. *Information Systems*, 26(1):35–58, 1998.

[Halkidi *et al.*, 2001] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2):107–145, 2001.

[Huang *et al.*, 2006] N. Huang, C. Kalyanaraman, K. Bernacki, and M. P. Jacobson. Molecular mechanics methods for predicting protein–ligand binding. *Physical Chemistry Chemical Physics*, 8(44):5166–5177, 2006.

[Huey *et al.*, 2007] R. Huey, G. M. Morris, A. J. Olson, and D. S. Goodsell. A semiempirical free energy force field with charge-based desolvation. *Journal of computational chemistry*, 28(6):1145–1152, 2007.

[Jain *et al.*, 1999] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[Johnson, 1999] D.H. Johnson. The insignificance of statistical significance testing. *The journal of wildlife management*, 63(3):763–772, 1999.

[Karypis *et al.*, 1999] G. Karypis, H. Eui-Hong, and V. Kumar. CHAMELEON: Hierarchical clustering using dynamic modeling. *Computer*, 32(8):68–75, 1999.

[King *et al.*, 1996] R. D. King, S. H. Muggleton, A. Srinivasan, and M. J. E. Sternberg. Structure-activity relationships derived by machine learning: The use of atoms and their bond connectivities to predict mutagenicity by inductive logic programming. *Proceedings of the National Academy of Sciences*, 93(1):438–442, 1996.

[Knuth, 2005] D. E. Knuth. *The Art of Computer Programming, Volume 4, Fascicle 2: Generating All Tuples and Permutations*. Addison-Wesley Professional, 2005.

[Kontkanen *et al.*, 2005] P. Kontkanen, P. Myllymäki, W. Buntine, J. Rissanen, and H. Tirri. An MDL framework for data clustering. In P. Grünwald, I. J. Myung, and M. Pitt, editors, *Advances in Minimum Description Length: Theory and Applications*. MIT Press, 2005.

[Moitessier *et al.*, 2008] N. Moitessier, P. Englebienne, D. Lee, J. Lawandi, and C. R. Corbeil. Towards the development of universal, fast and highly accurate docking/scoring methods: a long way to go. *British journal of pharmacology*, 153(S1):S7–S26, 2008.

[Mukouchi and Arikawa, 1995] Y. Mukouchi and S. Arikawa. Towards a mathematical theory of machine discovery from facts. *Theoretical Computer Science*, 137(1):53–84, 1995.

[Pasquier *et al.*, 1999] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Efficient mining of association rules using closed itemset lattices. *Information Systems*, 24(1):25–46, 1999.

[Schröder, 2002] M. Schröder. *Admissible representations for continuous computations*. PhD thesis, dem Fachbereich Informatik, der FernUniversit at – Gesamthochschule in Hagen, 2002.

[Ting and Wells, 2010] K. M. Ting and J. R. Wells. Multi-dimensional mass estimation and mass-based clustering. In *Proceedings of 10th IEEE International Conference on Data Mining*, pages 511–520, 2010.

[Tsuiki, 2002] H. Tsuiki. Real number computation through Gray code embedding. *Theoretical Computer Science*, 284(2):467–485, 2002.

[Uno *et al.*, 2005] T. Uno, M. Kiyomi, and H. Arimura. LCM ver. 3: Collaboration of array, bitmap and prefix tree for frequent itemset mining. In *Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations*, pages 77–86. ACM,

2005.

[Wang *et al.*, 1997] W. Wang, J. Yang, and R. Muntz. STING: A statistical information grid approach to spatial data mining. In *Proceedings of the 23rd International Conference on Very Large Data Bases*, pages 186–195, 1997.

[Weihrauch, 2000] K. Weihrauch. *Computable Analysis: An Introduction*. Springer, November 2000.

[Zhu and Goldberg, 2009] X. Zhu and A. B. Goldberg. *Introduction to semi-supervised learning*. Morgan and Claypool Publishers, 2009.