Dec. 6, 2019
特別講義@久留米高専

機械学習のしくみ

杉山 麿人（国立情報学研究所，JSTさきがけ研究者）

自己紹介

- 2012.3 に京都大学で博士（情報学）取得
  - Studies on Computational Learning via Discretization
- 2014.3 まで，約2年間ドイツでポスドク
  - マックスプランク研究所，フンボルト財団ポスドクフェロー
- 2014.4 から，大阪大学 産業科学研究所で助教
- 2017.4 から，国立情報学研究所で准教授
- 2014.10–2018.3，2018.10–現在 まで，JST さきがけ研究者

例からの学習（汎化）[Schoelkopf, 2013]

- 1, 2, 4, 7, ...　　→ 次にくる数字は？

例からの学習（汎化） [Schoelkopf, 2013]

- 1, 2, 4, 7, ...　→ 次にくる数字は？
  1, 2, 4, 7, 11, 16, ...　$(a_n = a_{n-1} + n - 1)$

例からの学習（汎化） [Schoelkopf, 2013]

- 1, 2, 4, 7, ...　→ 次にくる数字は？

  1, 2, 4, 7, 11, 16, ...　$(a_n = a_{n-1} + n - 1)$

  1, 2, 4, 7, 12, 20, ...　$(a_n = a_{n-1} + a_{n-2} + 1)$

例からの学習（汎化） [Schoelkopf, 2013]

- 1, 2, 4, 7, ...　→ 次にくる数字は？

  1, 2, 4, 7, 11, 16, ...　$(a_n = a_{n-1} + n - 1)$

  1, 2, 4, 7, 12, 20, ...　$(a_n = a_{n-1} + a_{n-2} + 1)$

  1, 2, 4, 7, 13, 24, ...　$(a_n = a_{n-1} + a_{n-2} + a_{n-3})$

例からの学習（汎化） [Schoelkopf, 2013]

- 1, 2, 4, 7, ...　→ 次にくる数字は？

  1, 2, 4, 7, 11, 16, ...　$(a_n = a_{n-1} + n - 1)$
  1, 2, 4, 7, 12, 20, ...　$(a_n = a_{n-1} + a_{n-2} + 1)$
  1, 2, 4, 7, 13, 24, ...　$(a_n = a_{n-1} + a_{n-2} + a_{n-3})$
  1, 2, 4, 7, 14, 28　(28の約数)

例からの学習（汎化） [Schoelkopf, 2013]

- 1, 2, 4, 7, ...　→ 次にくる数字は？

  1, 2, 4, 7, 11, 16, ...　$(a_n = a_{n-1} + n - 1)$
  1, 2, 4, 7, 12, 20, ...　$(a_n = a_{n-1} + a_{n-2} + 1)$
  1, 2, 4, 7, 13, 24, ...　$(a_n = a_{n-1} + a_{n-2} + a_{n-3})$
  1, 2, 4, 7, 14, 28　(28の約数)
  1, 2, 4, 7, 1, 1, 5, ...　($\pi = 3.1415\ldots$ と$e = 2.718\ldots$)

例からの学習（汎化） [Schoelkopf, 2013]

- 1, 2, 4, 7, ...　→ 次にくる数字は？

  1, 2, 4, 7, 11, 16, ...　$(a_n = a_{n-1} + n - 1)$
  1, 2, 4, 7, 12, 20, ...　$(a_n = a_{n-1} + a_{n-2} + 1)$
  1, 2, 4, 7, 13, 24, ...　$(a_n = a_{n-1} + a_{n-2} + a_{n-3})$
  1, 2, 4, 7, 14, 28　(28の約数)
  1, 2, 4, 7, 1, 1, 5, ...　($\pi = 3.1415\ldots$ と$e = 2.718\ldots$)

- 1229 個！以上ある（https://oeis.org）

学習とは？

- どれが「正しい」答え（汎化）だろうか？
  → 答えることはできない（どれも正しいとも言える）
  - 「万能（普遍的な）の答え」というものは存在しない
  - 参考：醜いアヒルの子定理，no free lunch 定理
- 機械学習での目的：
  過去の経験（データ）を一般化する規則を見つける
  - 過去と同じくらい良く未来を予測する

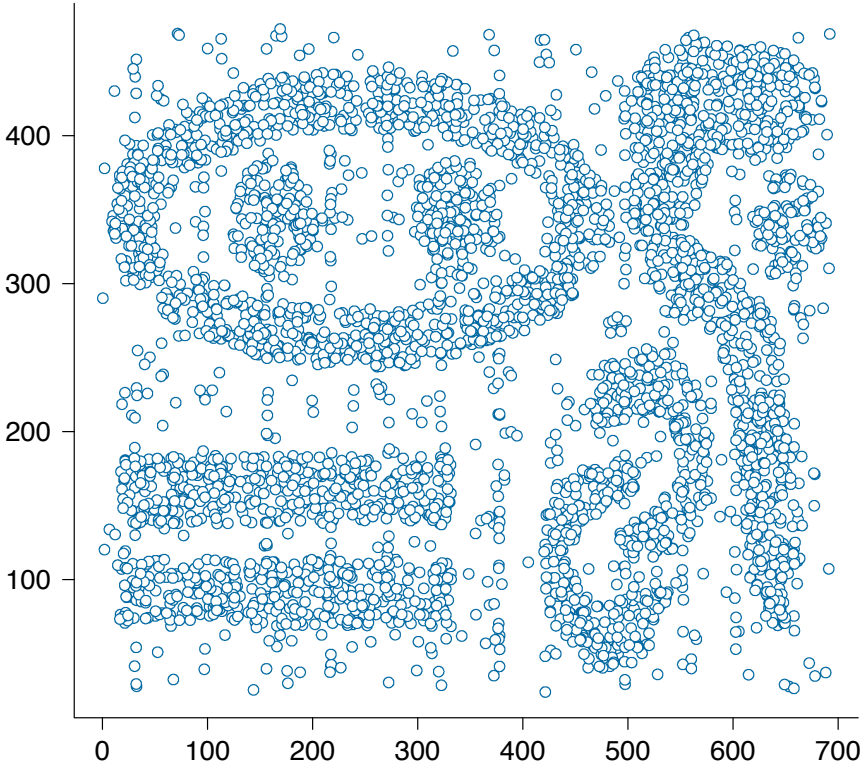
学習の定式化の構成要素

1. 学習の対象は何か？
2. 学習の対象はどうやって表現されるのか？(モデル)
3. データはどうやって与えられるのか？
4. どうやって学習するのか？(学習アルゴリズム)
5. 学習結果をどのように評価するのか？

クラスタリング

- 似たもの同士のグループを見つける
  - 教師なし学習の代表的なタスク
- 代表的な手法：
  - *k*-means, DBSCAN, 階層クラスタリング，…
- 以下では，各データ点は d 次元特徴ベクトル $\boldsymbol{x} \in \mathbb{R}^d$ で表現されると仮定
  - 入力は，データ点の集合 $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$

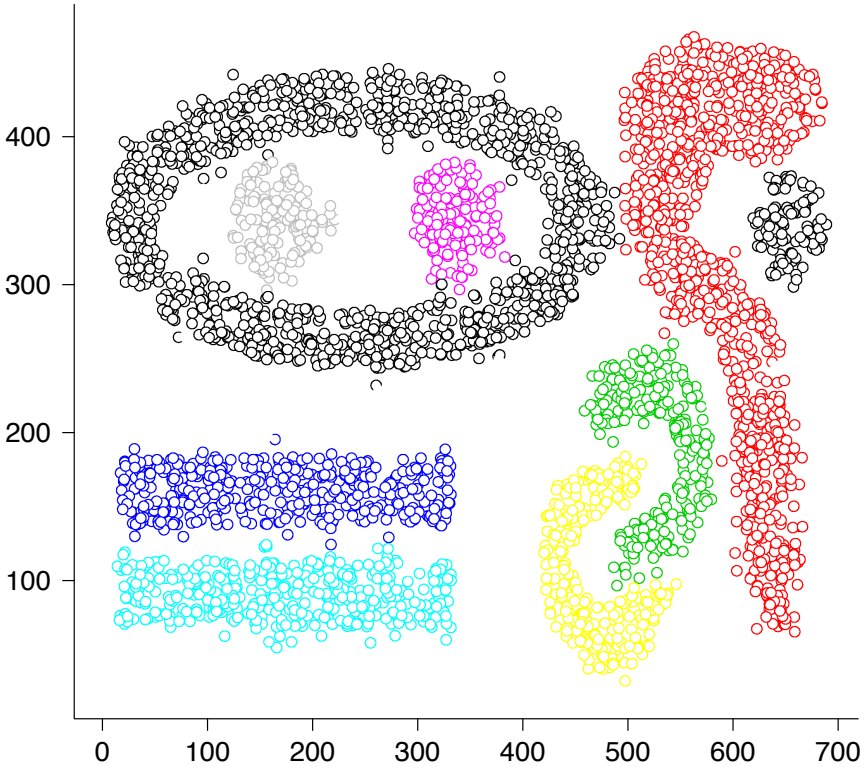
クラスタリングの例



- データ集合 $X \subset \mathbb{R}^2$:

1	355.60	270.21
2	549.28	351.71
3	520.08	215.48
4	575.15	166.68
⋮	⋮	⋮
4000	309.395	365.09

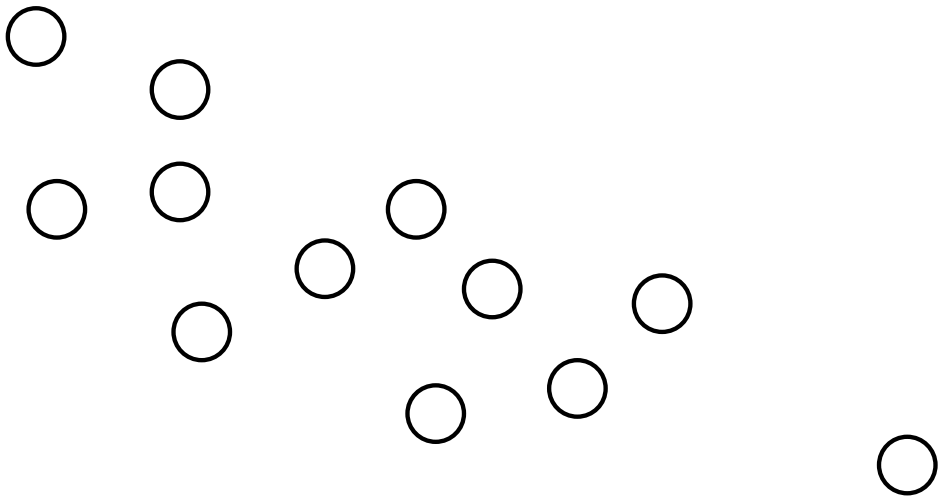
クラスタリング結果の例



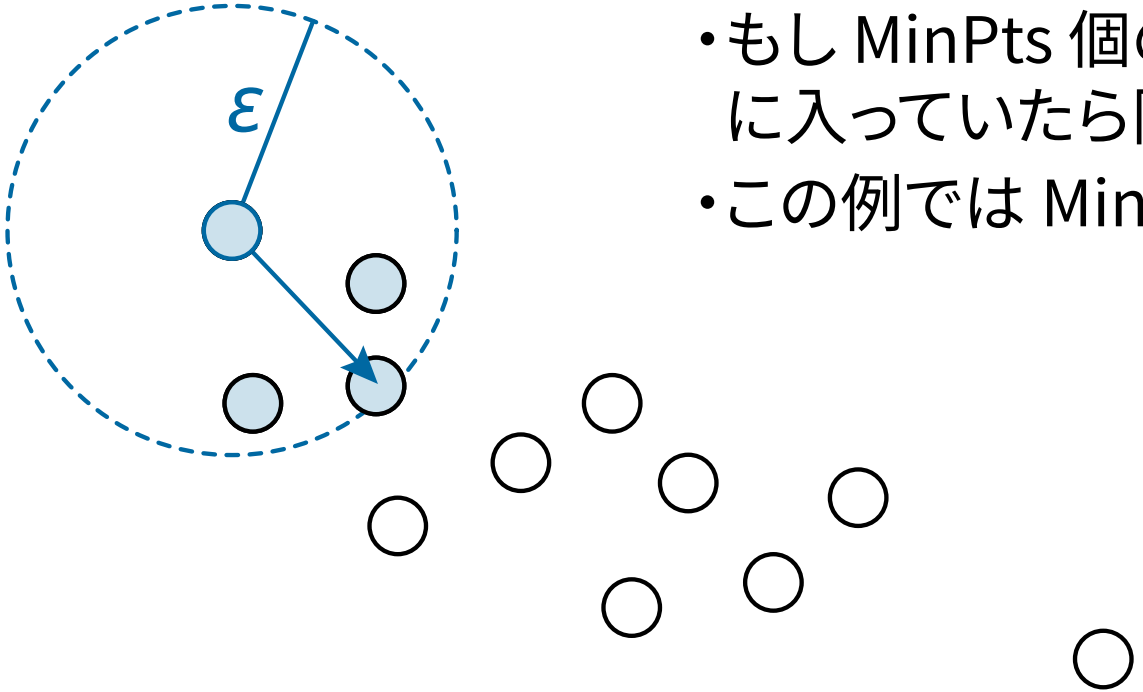
- データ集合 $X \subset \mathbb{R}^2$:

1	355.60	270.21
2	549.28	351.71
3	520.08	215.48
4	575.15	166.68
⋮	⋮	⋮
4000	309.395	365.09

DBSCAN [Ester et al., 1996]

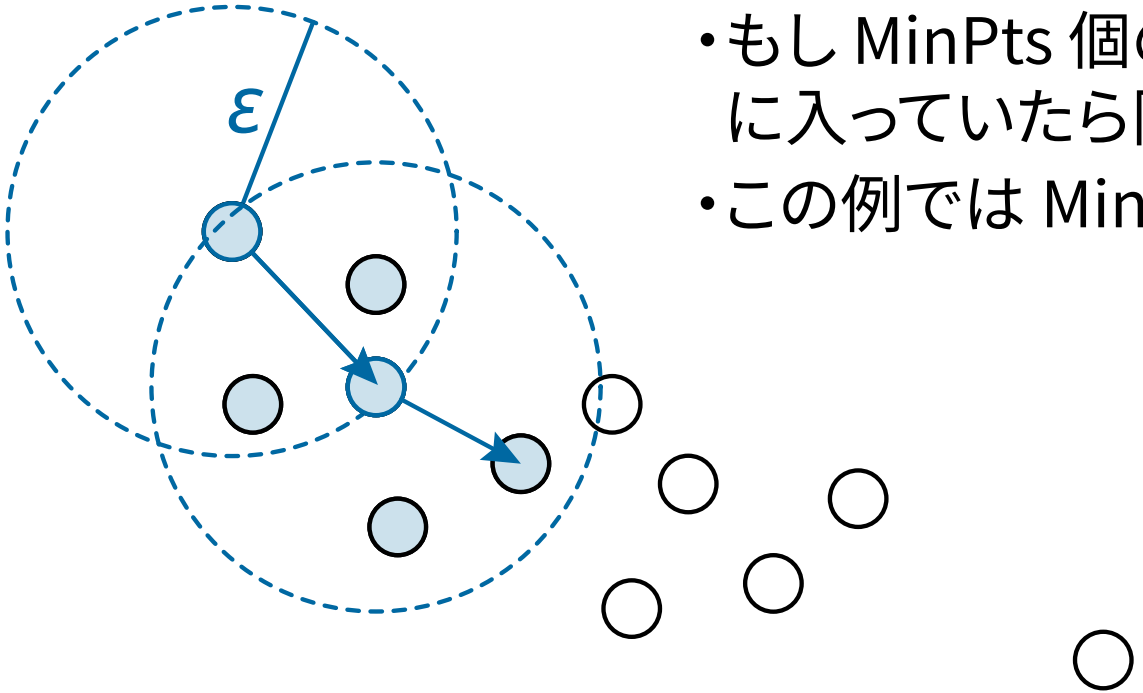


DBSCAN [Ester et al., 1996]



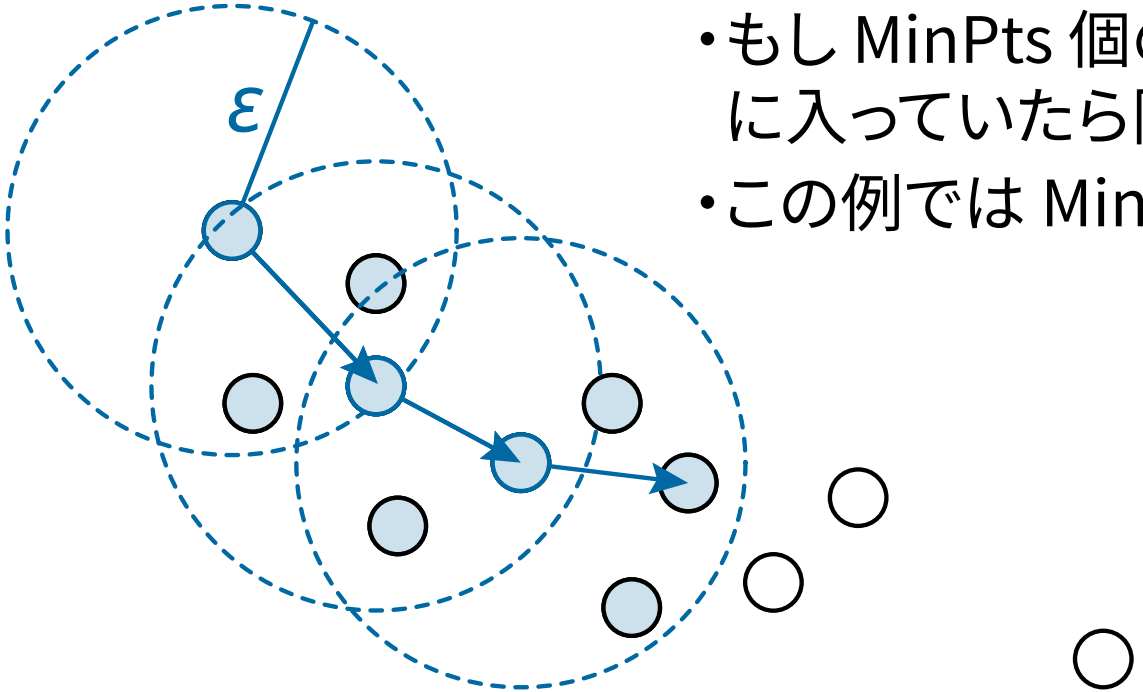
- もし MinPts 個の点が同じ円の中に入っていたら同じクラスタとみなす
- この例では MinPts = 3

DBSCAN [Ester et al., 1996]



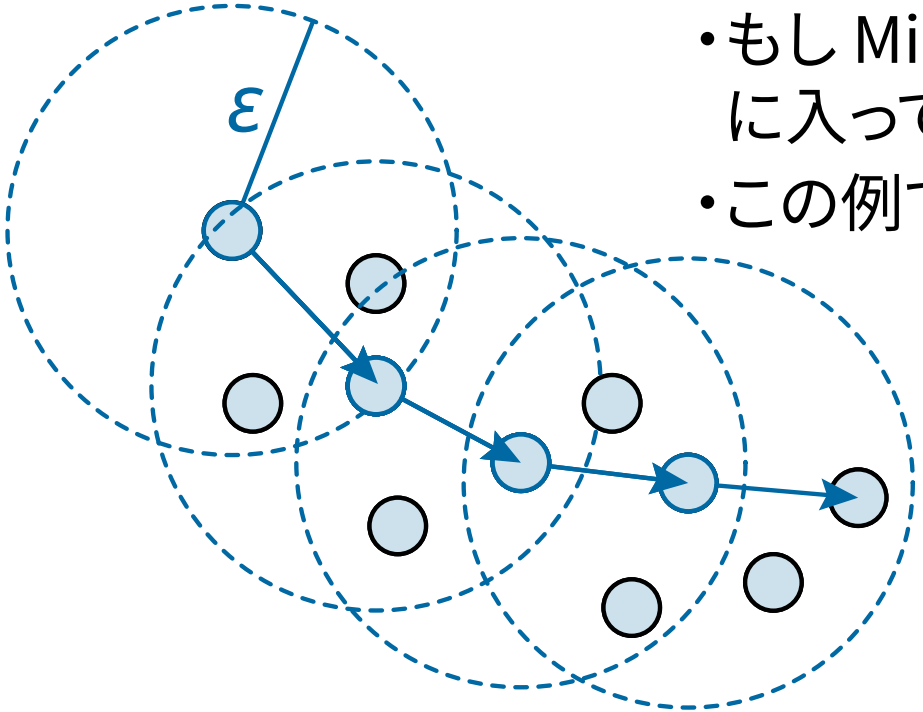
- もし MinPts 個の点が同じ円の中に入っていたら同じクラスタとみなす
- この例では MinPts = 3

DBSCAN [Ester et al., 1996]



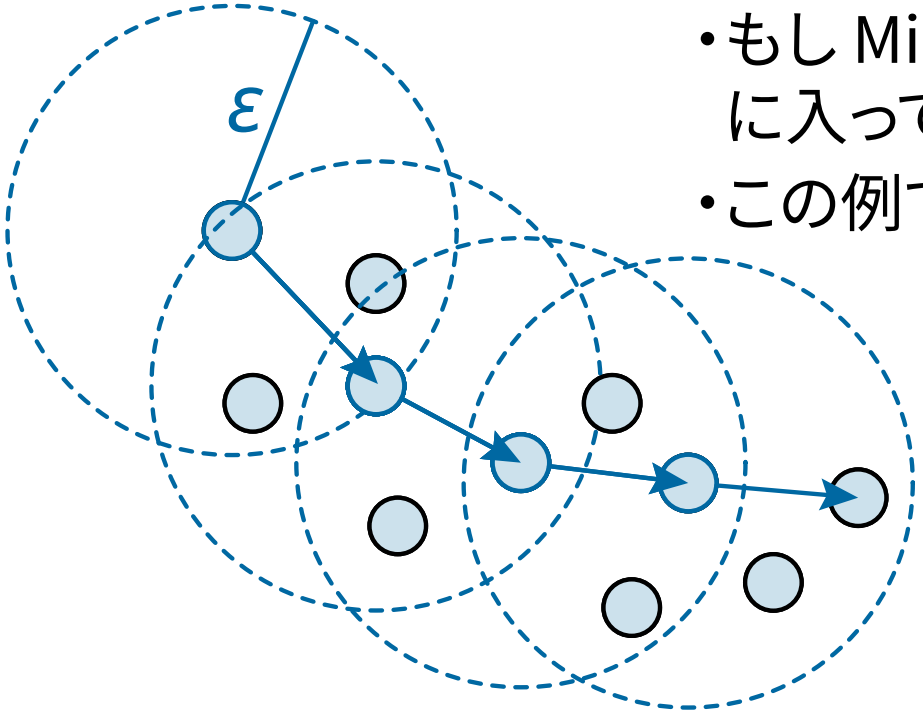
- もし MinPts 個の点が同じ円の中に入っていたら同じクラスタとみなす
- この例では MinPts = 3

DBSCAN [Ester et al., 1996]



- もし MinPts 個の点が同じ円の中に入っていたら同じクラスタとみなす
- この例では MinPts = 3

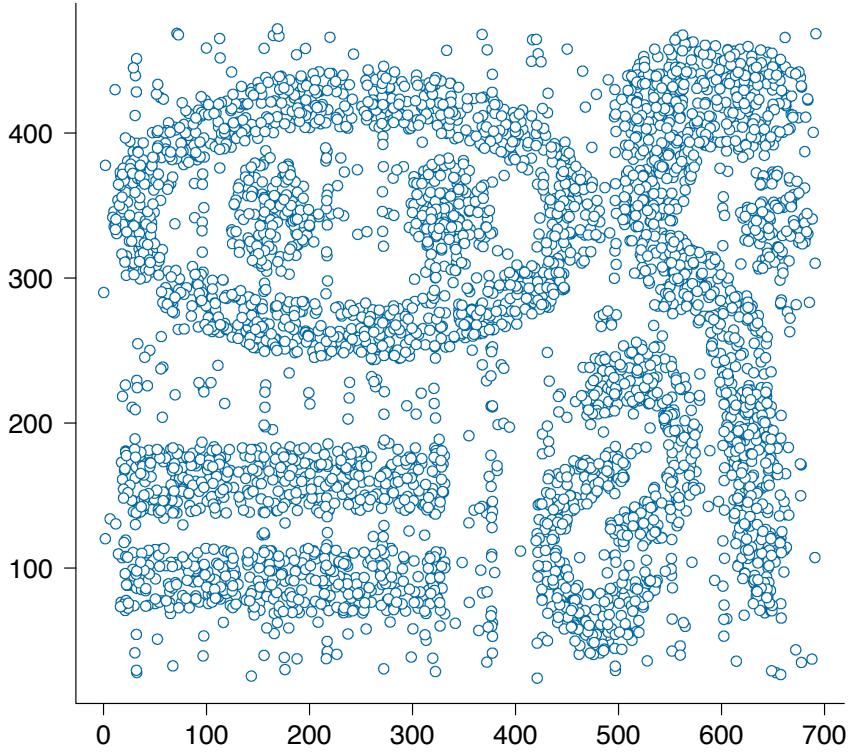
DBSCAN [Ester et al., 1996]

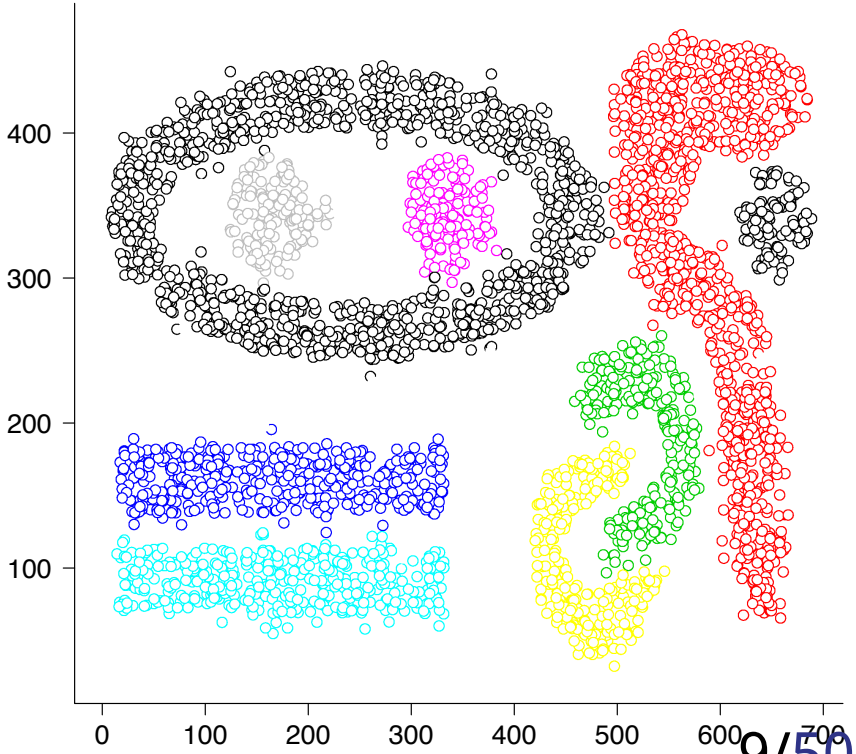


- もし MinPts 個の点が同じ円の中に入っていたら同じクラスタとみなす
- この例では MinPts = 3

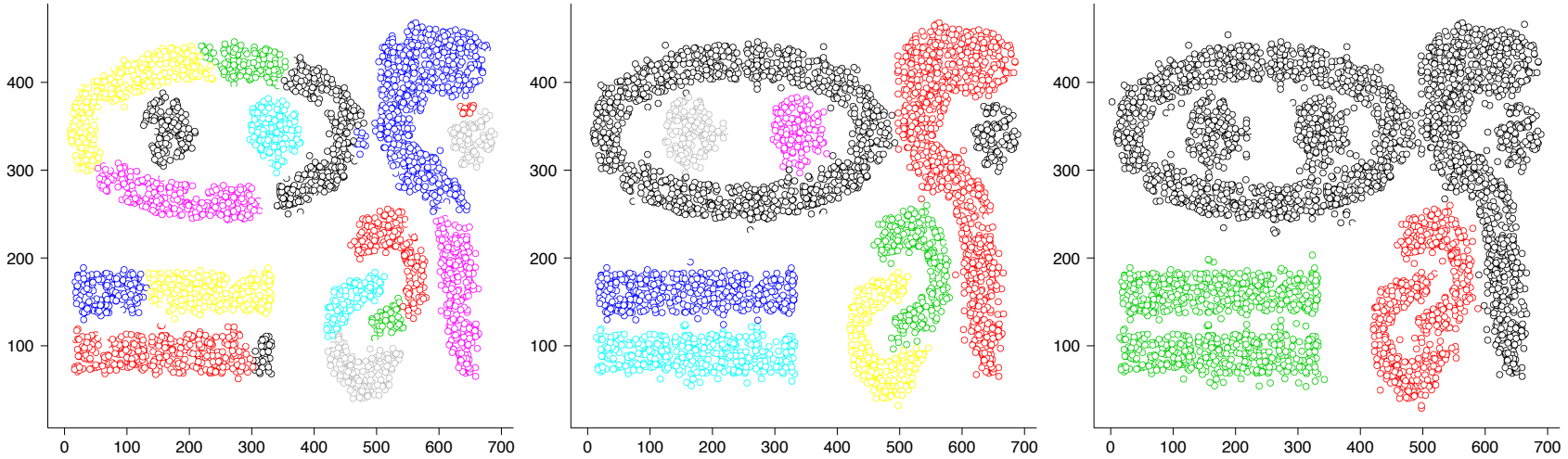
どの点からも到達できなかったらノイズ

DBSCANの結果（$\epsilon = 14$, MinPts $= 10$）



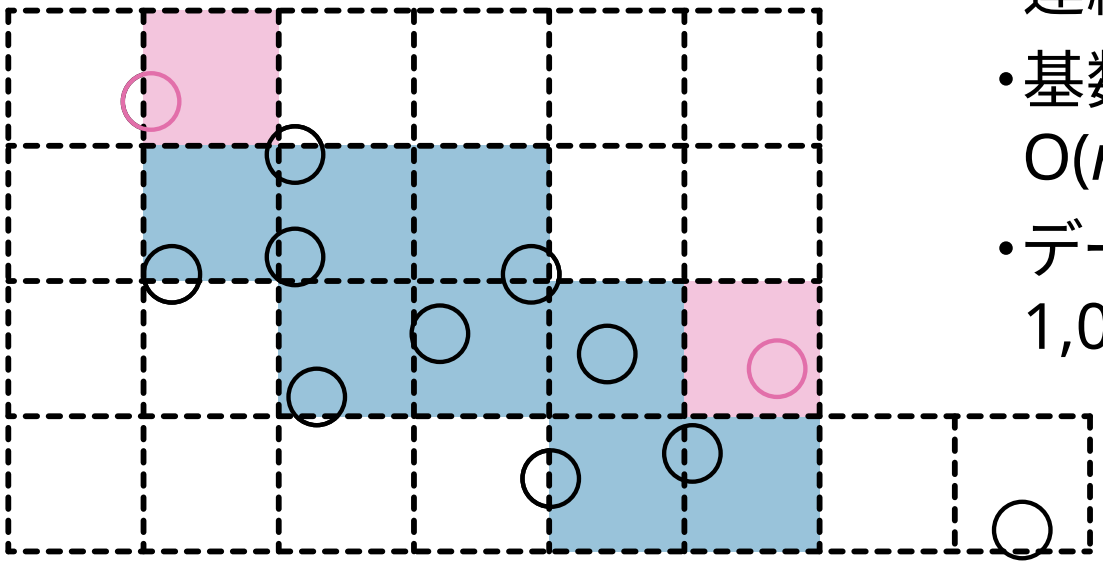


クラスタリング結果は任意



- $\epsilon = 12, 14, 16$（左から右），MinPts $= 10$
  - クラスタリング結果の解釈は要注意

BOOL [Sugiyama & Yamamoto, 2011]

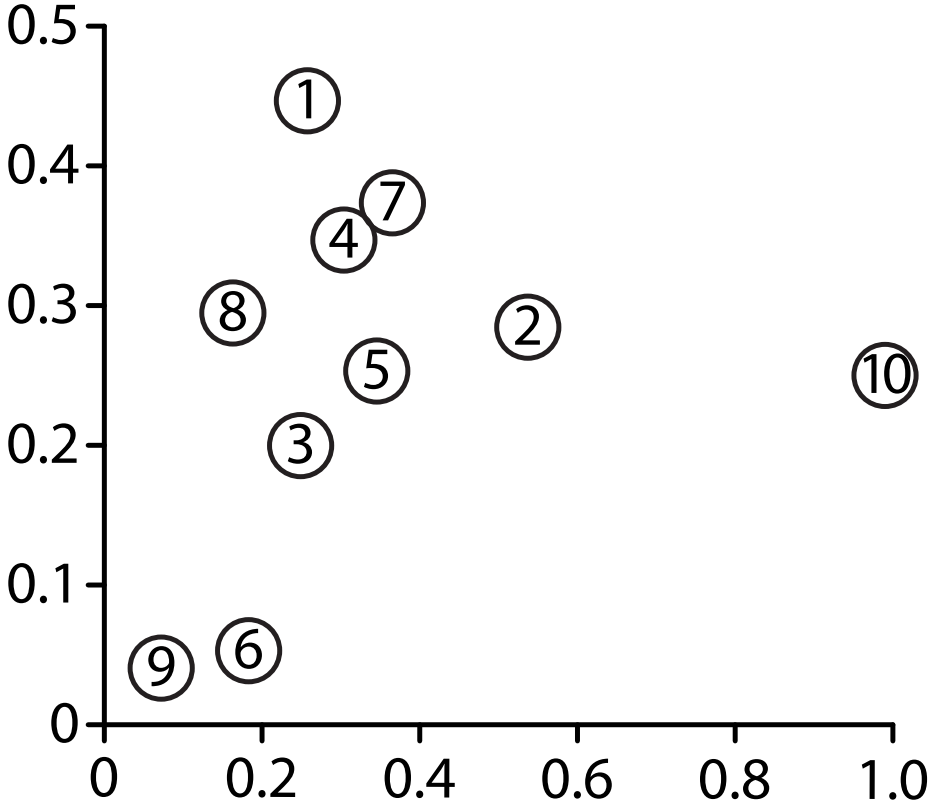


- データを離散化して，連続していれば繋げる
- 基数ソートを使えば $O(n^2) \Rightarrow O(n)$
- データ数10,000で，1,000倍程度高速化

外れ値検出（異常検知）

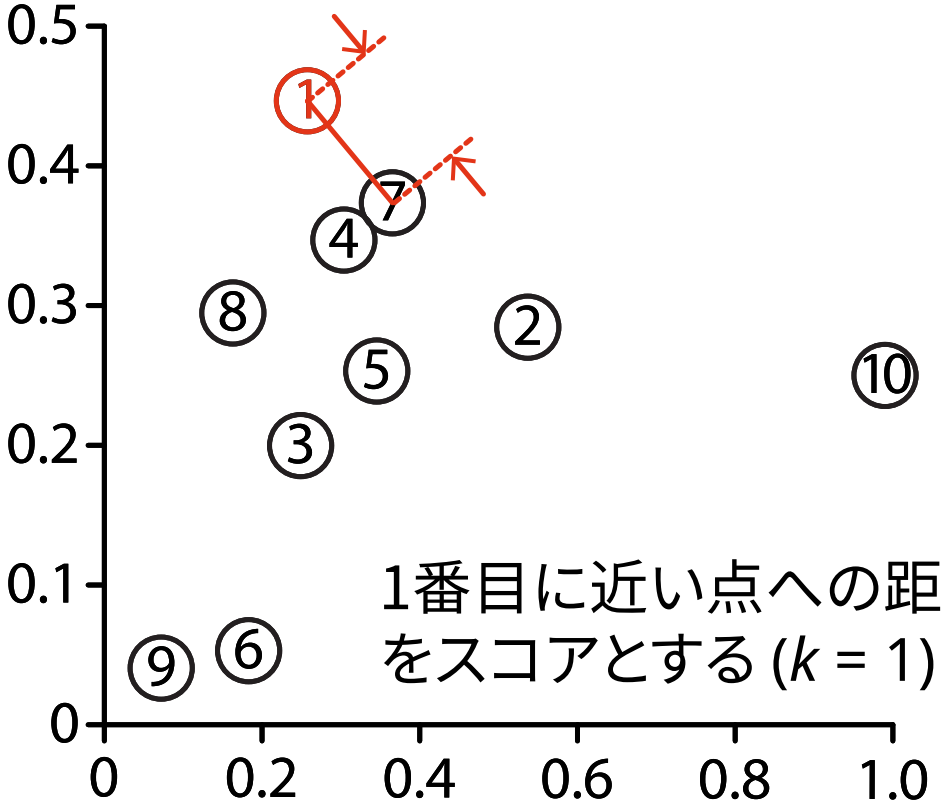
- 仲間はずれのデータ点を見つける
- 代表的な手法：
  - kth-NN（k近傍距離）, LOF, iForest, …
- クラスタリングの場合と同様，
  入力はデータ点の集合 $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\} \subset \mathbb{R}^d$

*k*th-NN（1/2） [Bay & Schwabacher, 2003]



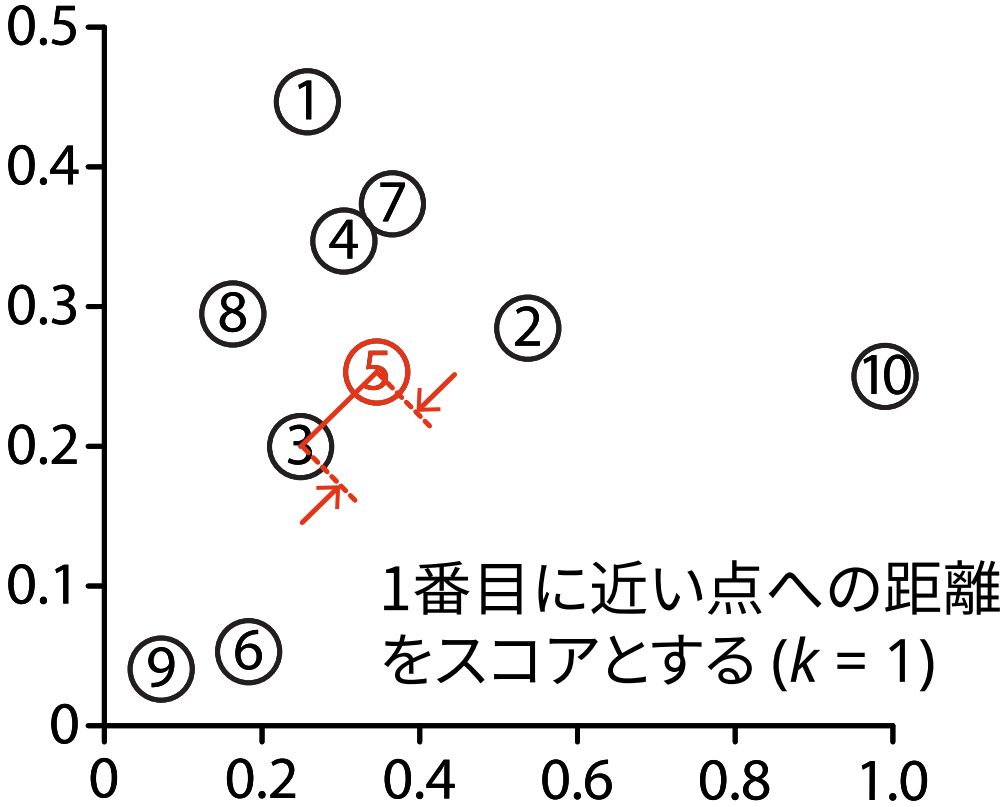
id	score

*k*th-NN（1/2） [Bay & Schwabacher, 2003]



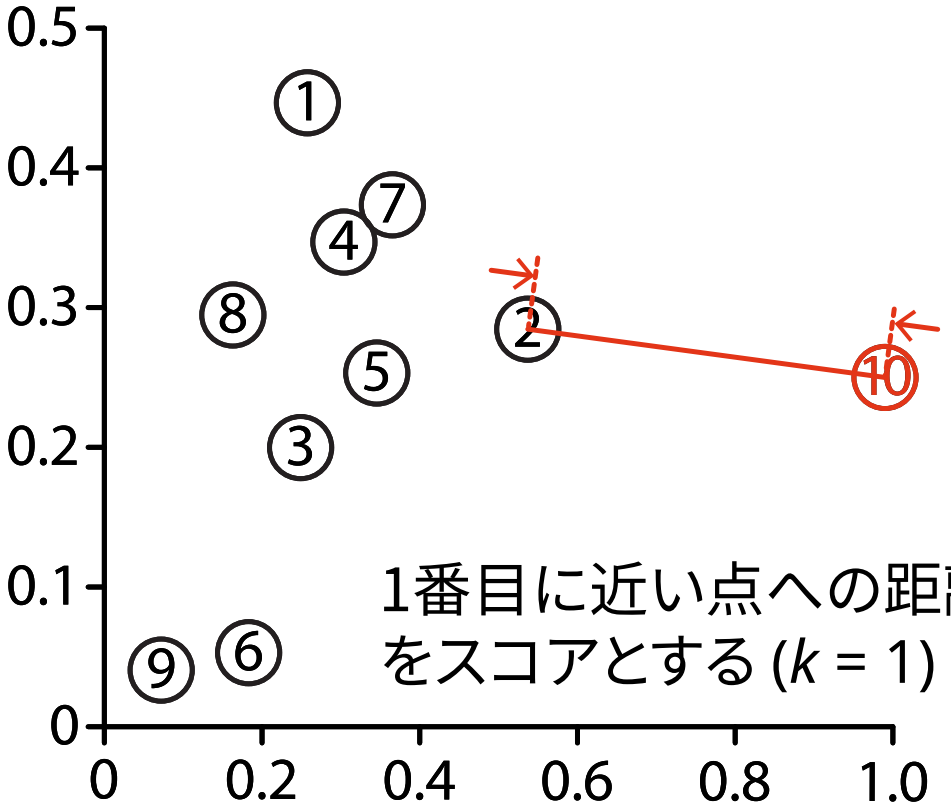
id	score
1	0.109

*k*th-NN（1/2） [Bay & Schwabacher, 2003]



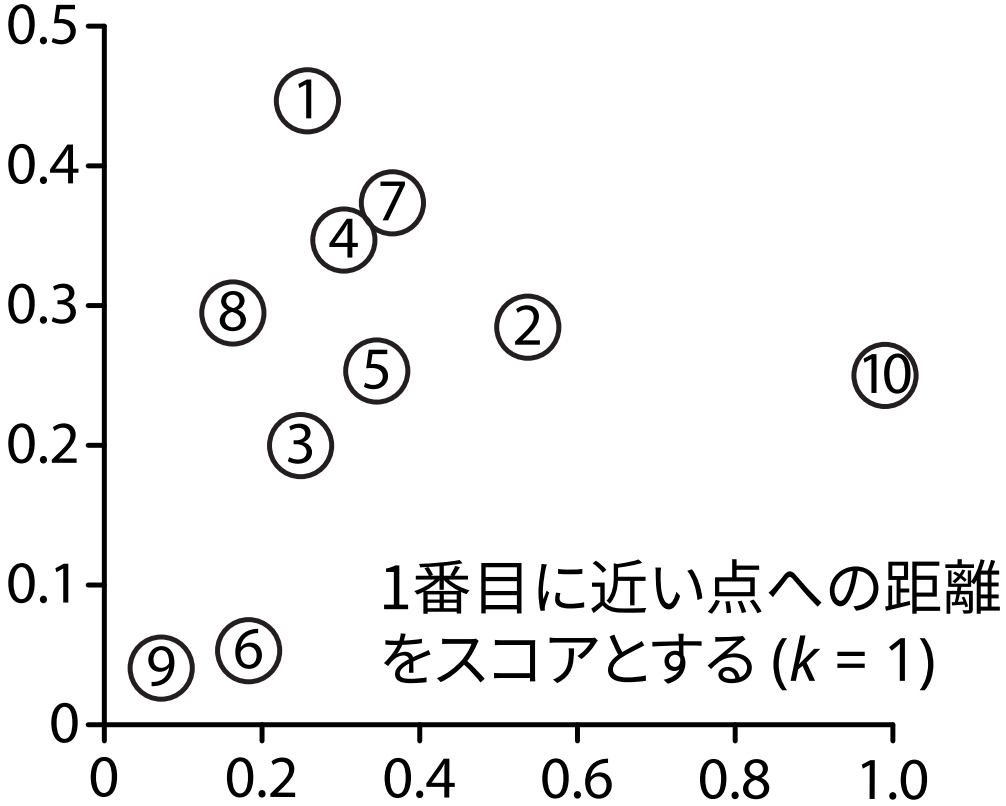
id	score
1	0.109
5	0.103

*k*th-NN（1/2） [Bay & Schwabacher, 2003]



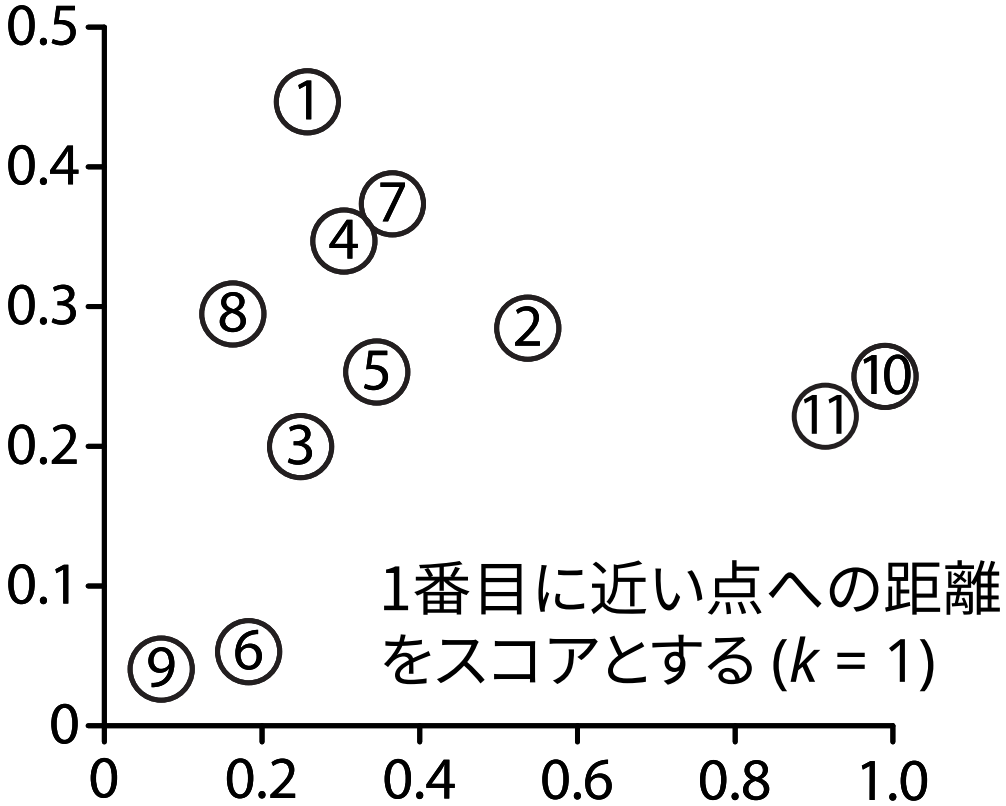
id	score
10	0.454
1	0.109
5	0.103

*k*th-NN（1/2） [Bay & Schwabacher, 2003]



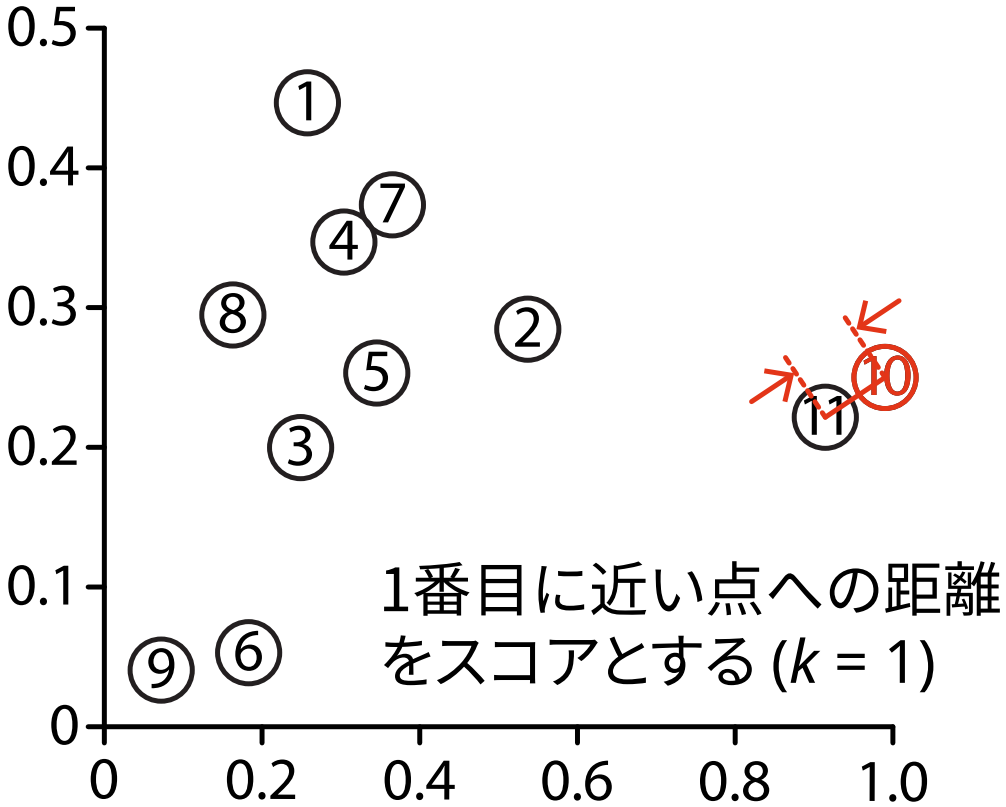
id	score
10	0.454
2	0.193
8	0.128
6	0.112
9	0.112
3	0.110
1	0.109
5	0.103
4	0.067
7	0.067

*k*th-NN（2/2） [Bay & Schwabacher, 2003]



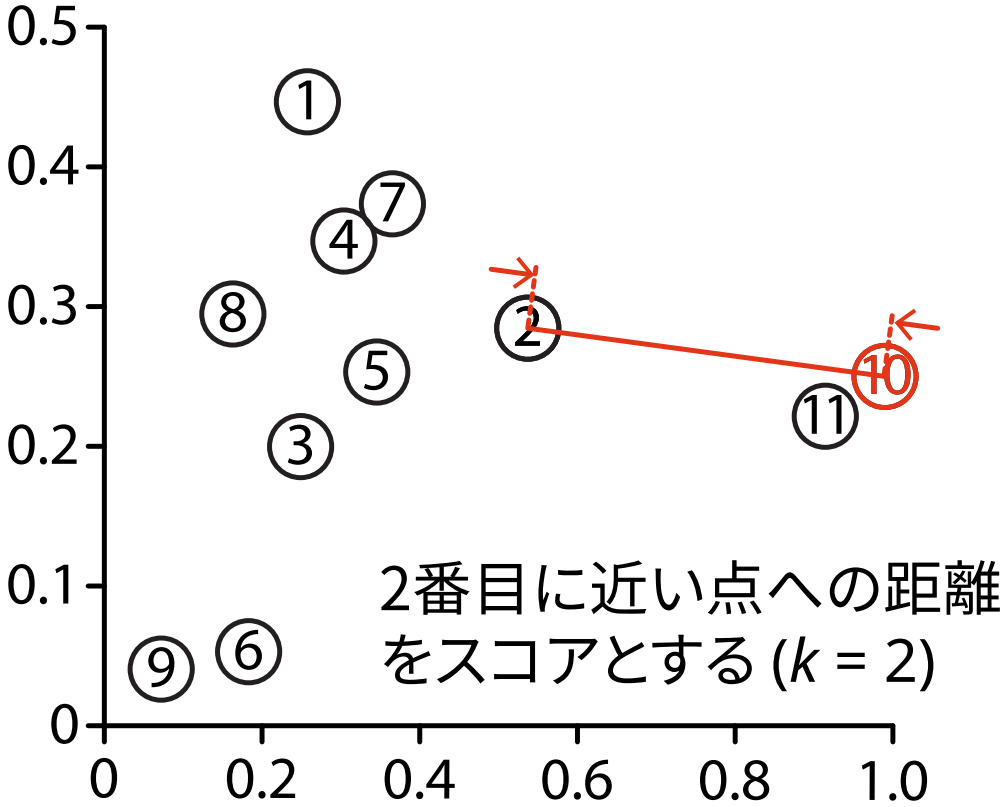
id	score
10	0.454
2	0.193
8	0.128
6	0.112
9	0.112
3	0.110
1	0.109
5	0.103
4	0.067
7	0.067

*k*th-NN（2/2） [Bay & Schwabacher, 2003]



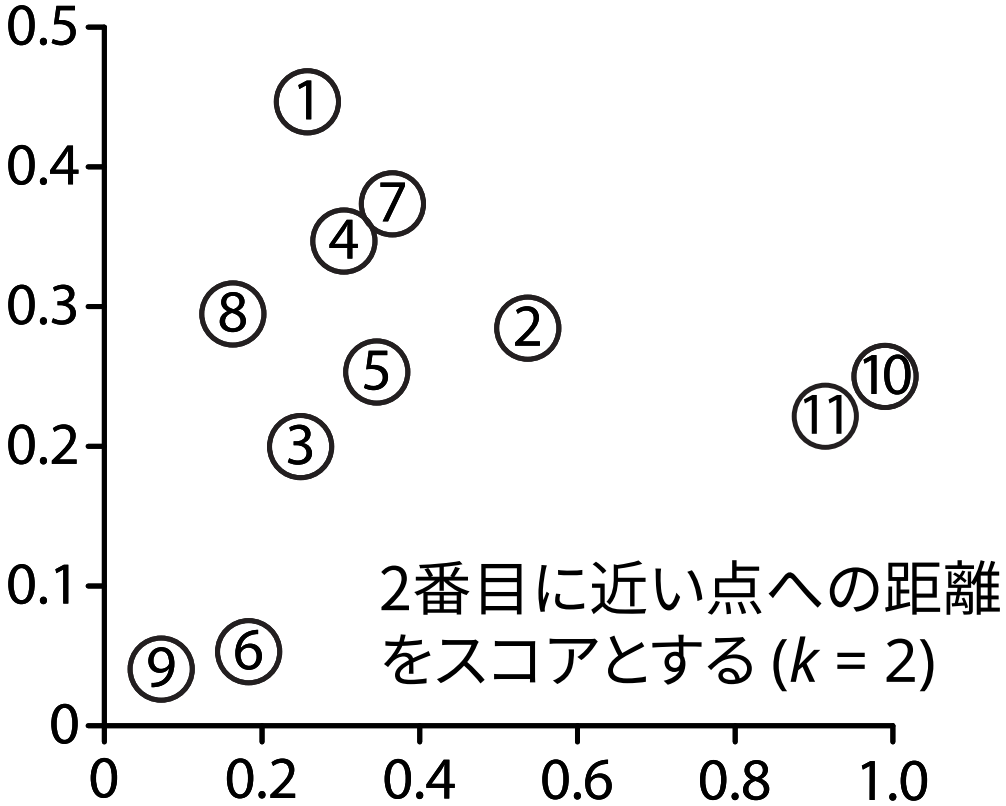
id	score
2	0.193
8	0.128
6	0.112
9	0.112
3	0.110
1	0.109
5	0.103
4	0.067
7	0.067
10	0.028
11	0.028

*k*th-NN（2/2） [Bay & Schwabacher, 2003]



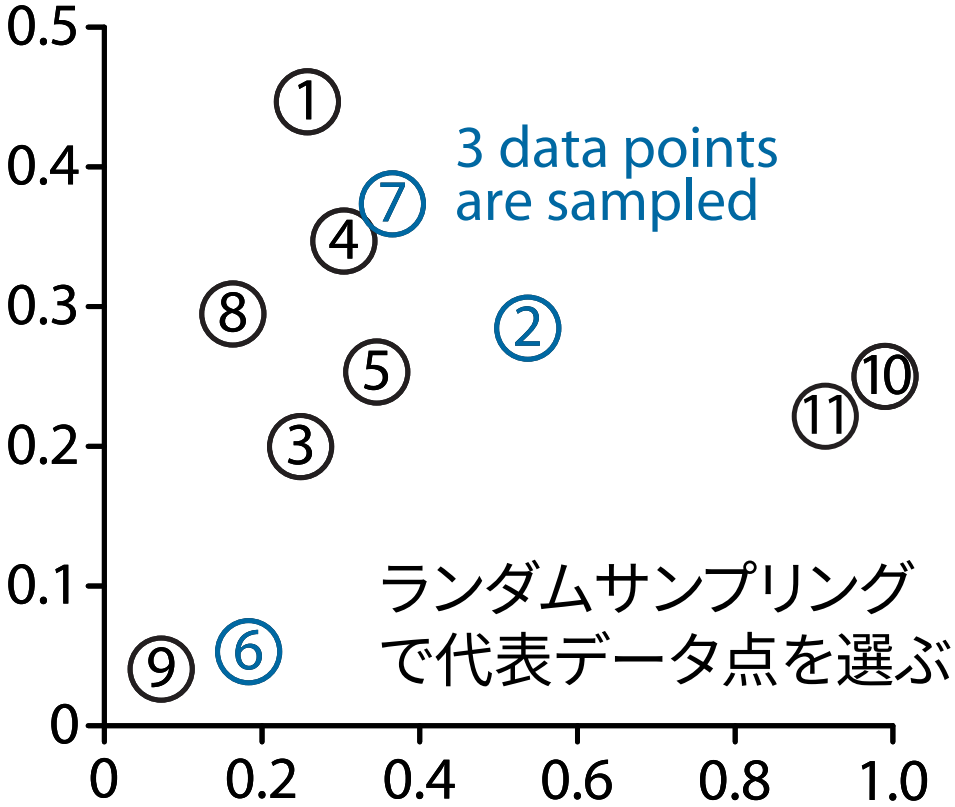
id	score
2	0.193
8	0.128
6	0.112
9	0.112
3	0.110
1	0.109
5	0.103
4	0.067
7	0.067
10	0.028
11	0.028

*k*th-NN（2/2） [Bay & Schwabacher, 2003]



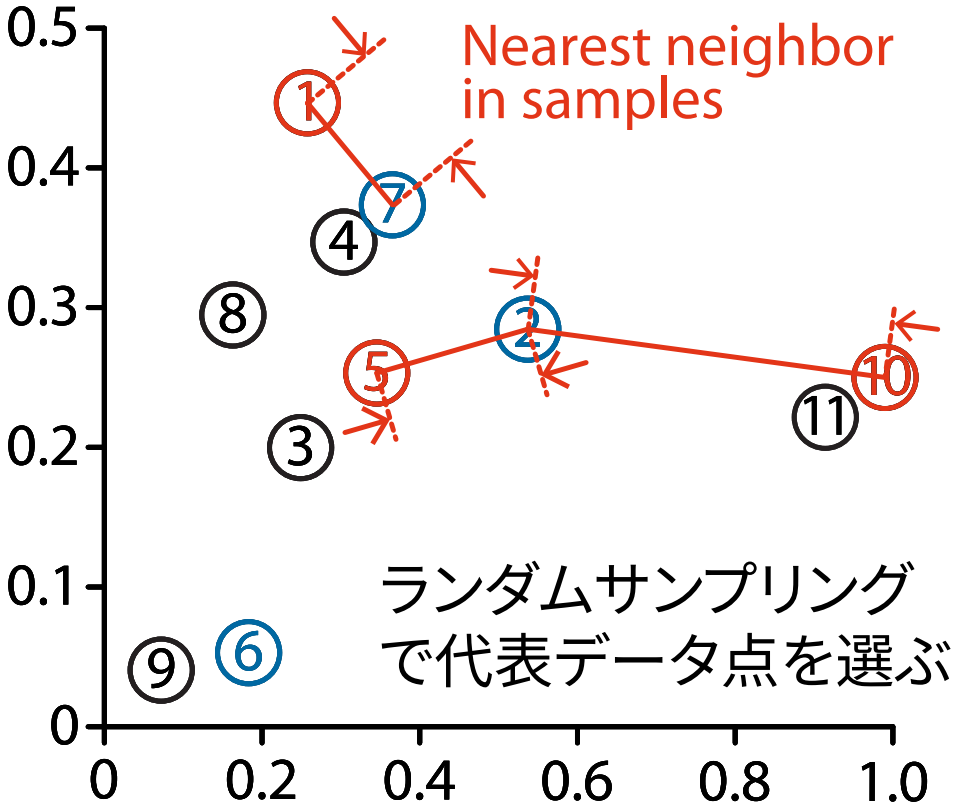
id	score
10	0.454
11	0.436
9	0.238
2	0.194
6	0.161
8	0.150
1	0.130
3	0.128
7	0.122
5	0.110
4	0.103

サンプリング法 [Sugiyama & Borgwardt, 2013]



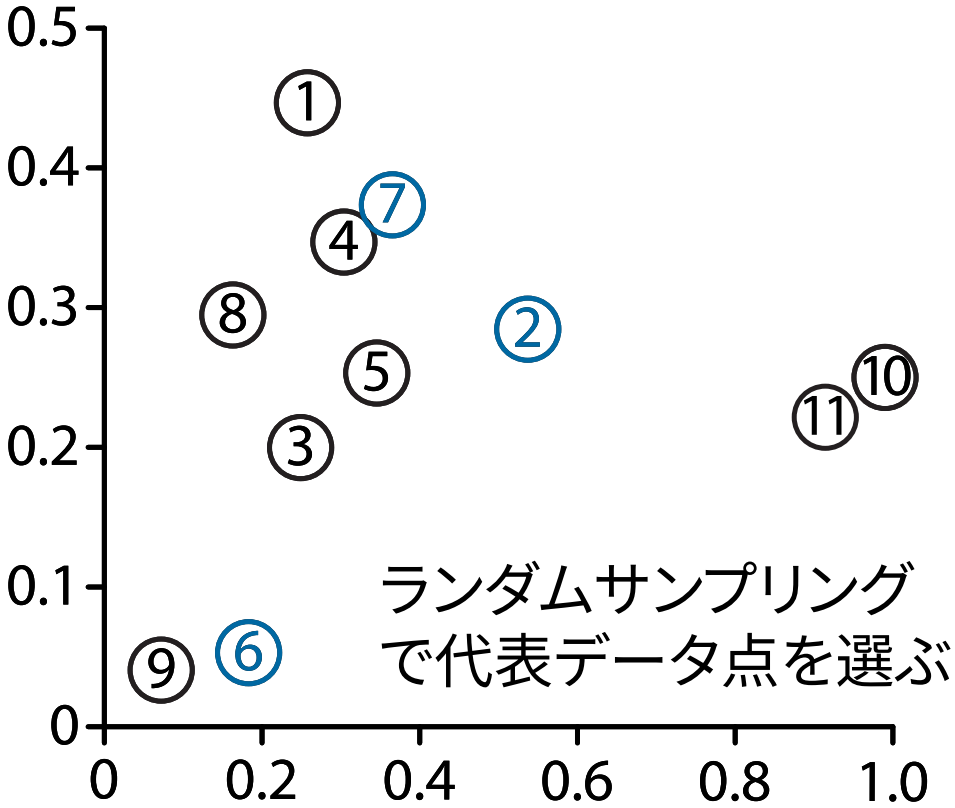
id	score

サンプリング法 [Sugiyama & Borgwardt, 2013]



id	score
10	0.454
1	0.130
5	0.122

サンプリング法 [Sugiyama & Borgwardt, 2013]



id	score
10	0.454
11	0.436
6	0.369
8	0.217
2	0.193
7	0.193
3	0.161
1	0.130
5	0.122
9	0.112
4	0.067

Algorithm 1: kth-NN

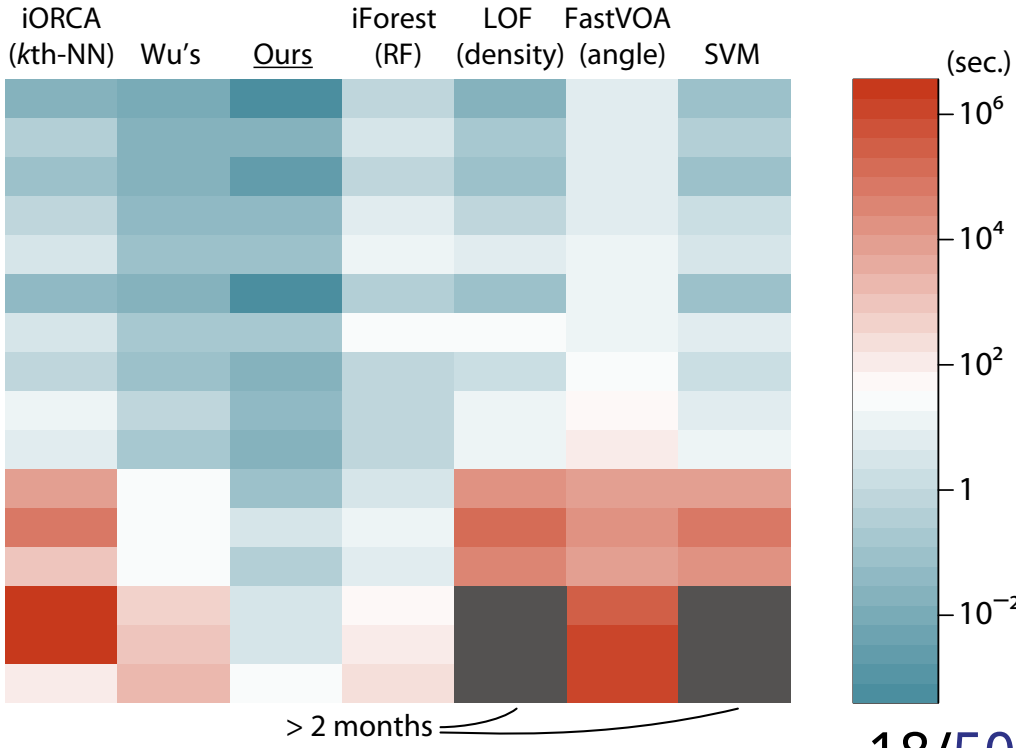
Initialize $M \in R^{n \times n}$, $\boldsymbol{q} \in R^n$
foreach $\boldsymbol{x}_i \in X$ **do**
    foreach $\boldsymbol{x}_j \in X$ **do**
        $m_{ij} \leftarrow d(\boldsymbol{x}_i, \boldsymbol{x}_j)$
foreach $i \in \{1, 2, \ldots, n\}$ **do**
    $q_i \leftarrow k$th largest value in ith row of M
Output $\boldsymbol{q}$

Algorithm 2: Sugiyama-Borgwardt サンプリング法

$S \leftarrow$ Subsample of X, initialize $M \in R^{n \times |S|}$, $\boldsymbol{q} \in R^n$
foreach $\boldsymbol{x}_i \in X$ **do**
    foreach $\boldsymbol{s}_j \in S$ **do**
        $m_{ij} \leftarrow d(\boldsymbol{x}_i, \boldsymbol{s}_j)$
foreach $i \in \{1, 2, \ldots, n\}$ **do**
    $q_i \leftarrow$ Largest value in ith row of M
Output $\boldsymbol{q}$

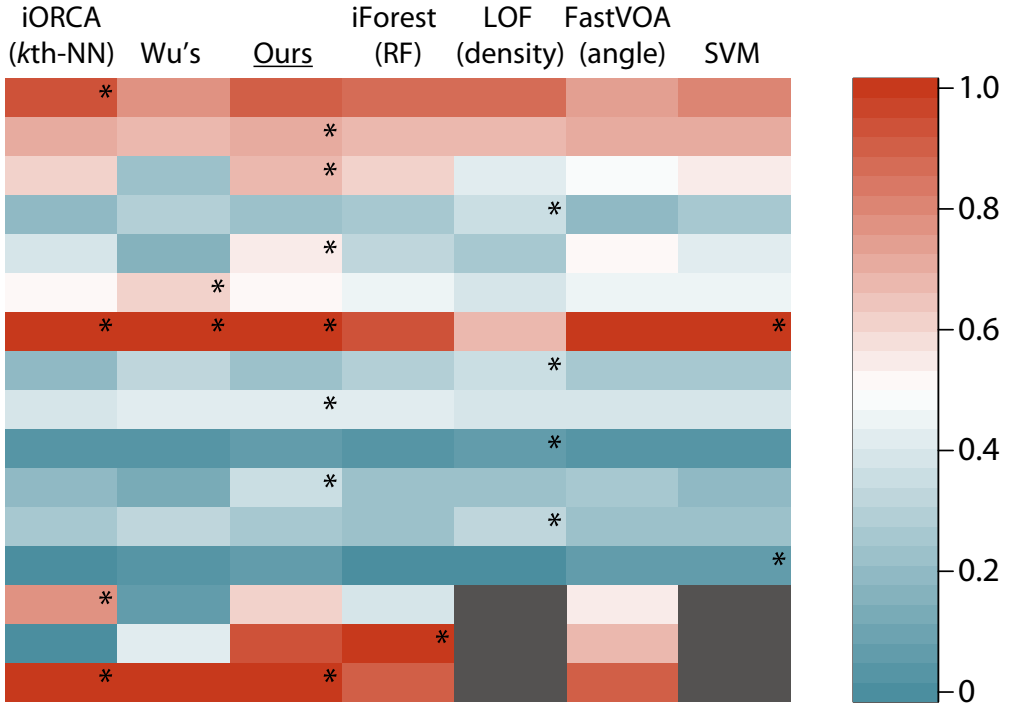
外れ値検出手法の適用結果（実行時間）

	# of objects	# of outliers	# of dims
Ionosphere	351	126	34
Arrhythmia	452	207	274
Wdbc	569	212	30
Mfeat	600	200	649
Isolet	960	240	617
Pima	768	268	8
Gaussian*	1000	30	1000
Optdigits	1688	554	64
Spambase	4601	1813	57
Statlog	6435	626	36
Skin	245057	50859	3
Pamap2	373161	125953	51
Covtype	286048	2747	10
Kdd1999	4898431	703067	6
Record	5734488	20887	7
Gaussian*	10000000	30	20



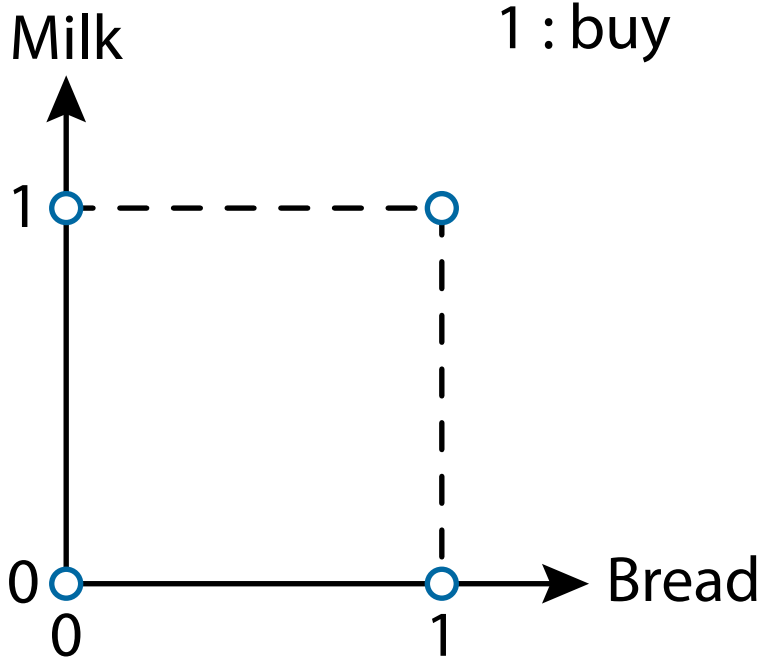
外れ値検出手法の適用結果（検出性能）

	# of objects	# of outliers	# of dims
Ionosphere	351	126	34
Arrhythmia	452	207	274
Wdbc	569	212	30
Mfeat	600	200	649
Isolet	960	240	617
Pima	768	268	8
Gaussian*	1000	30	1000
Optdigits	1688	554	64
Spambase	4601	1813	57
Statlog	6435	626	36
Skin	245057	50859	3
Pamap2	373161	125953	51
Covtype	286048	2747	10
Kdd1999	4898431	703067	6
Record	5734488	20887	7
Gaussian*	10000000	30	20



2値データ → パターンマイニング

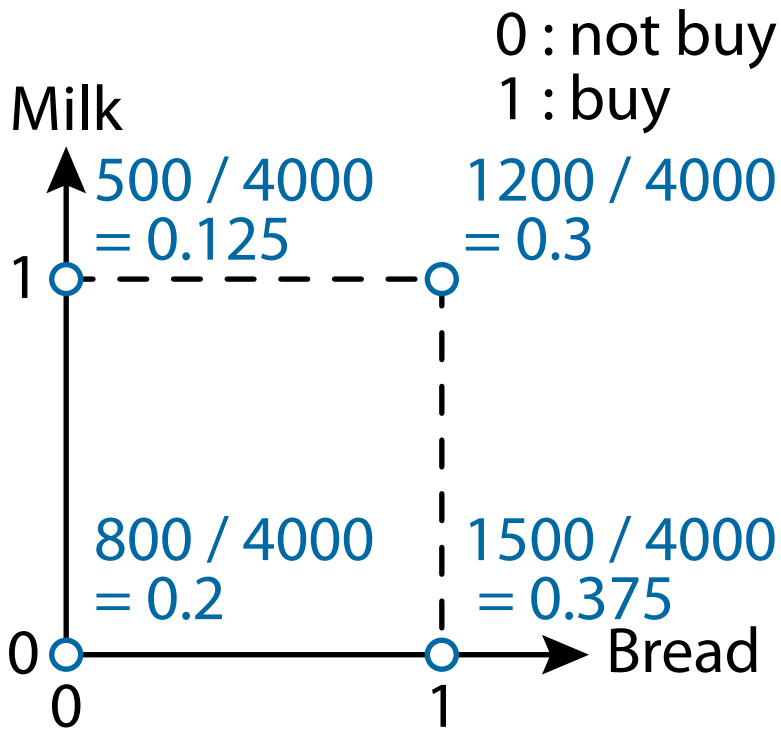
0 : not buy
1 : buy



- データ集合:

	Bread	Milk
1	0	1
2	1	1
3	1	1
4	1	0
⋮	⋮	⋮
4000	1	0

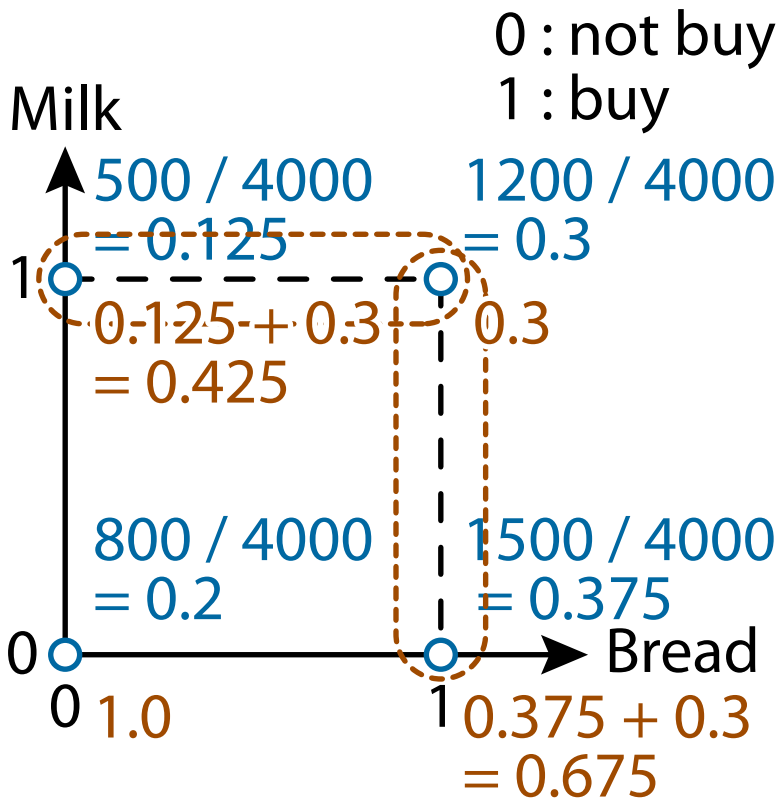
頻出パターンの発見例（確率）



- データ集合:

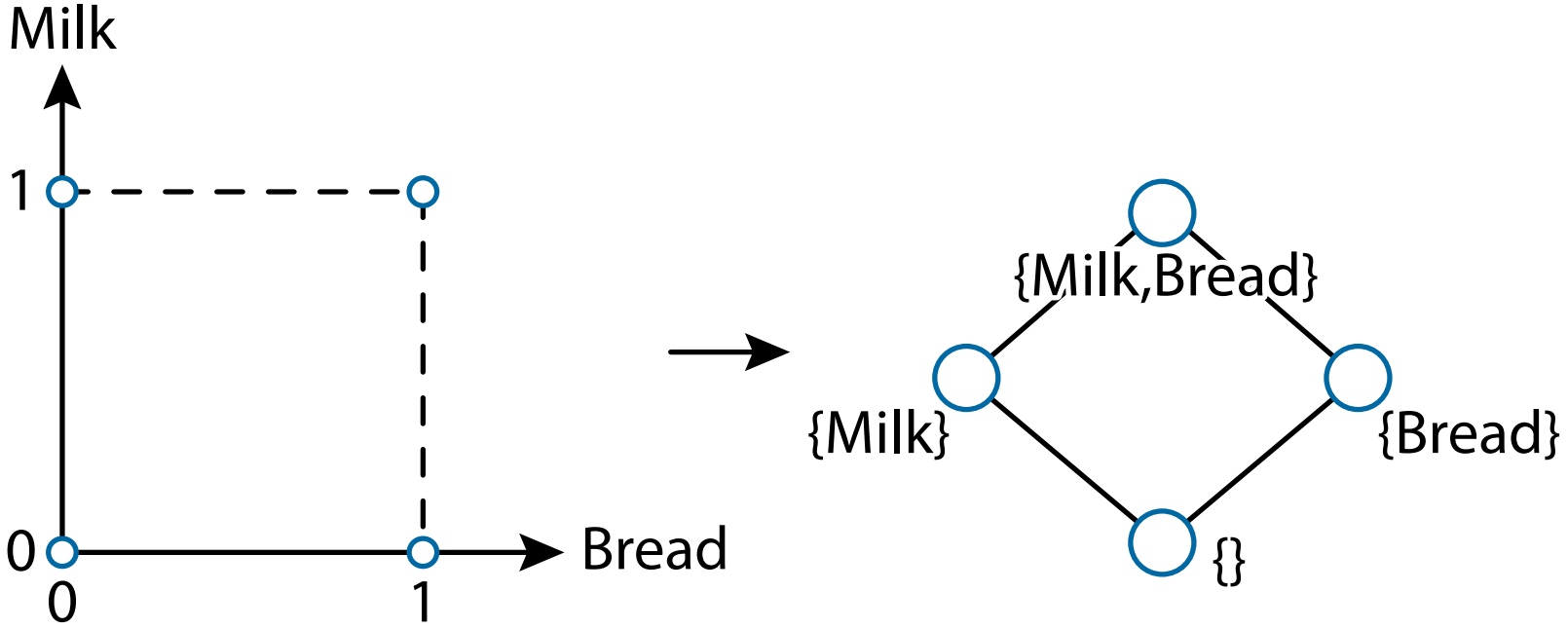
	Bread	Milk
1	0	1
2	1	1
3	1	1
4	1	0
⋮	⋮	⋮
4000	1	0

頻出パターンの発見例（頻度）

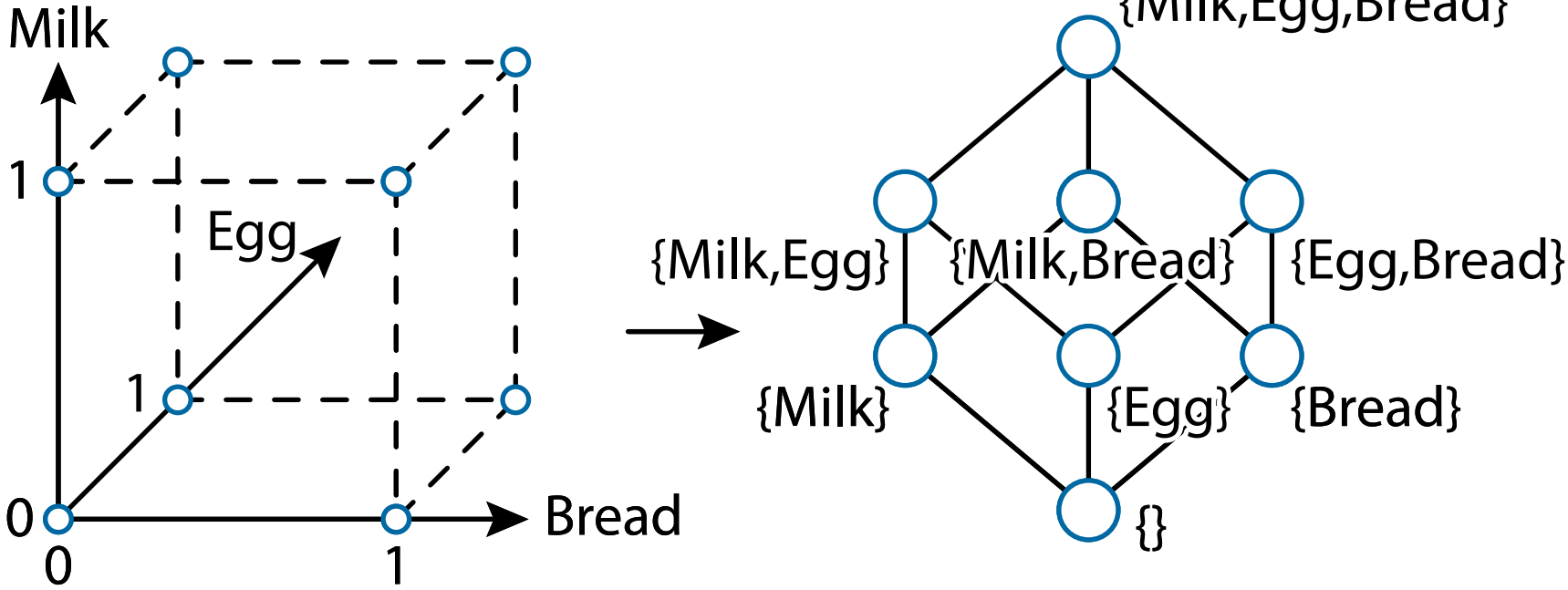


- 67.5% のお客さんが {bread} を購入
- 42.5% のお客さんが {milk} を購入
- 30.0% のお客さんが {bread, milk} を購入
- 商品の組合せをパターンと呼ぶ

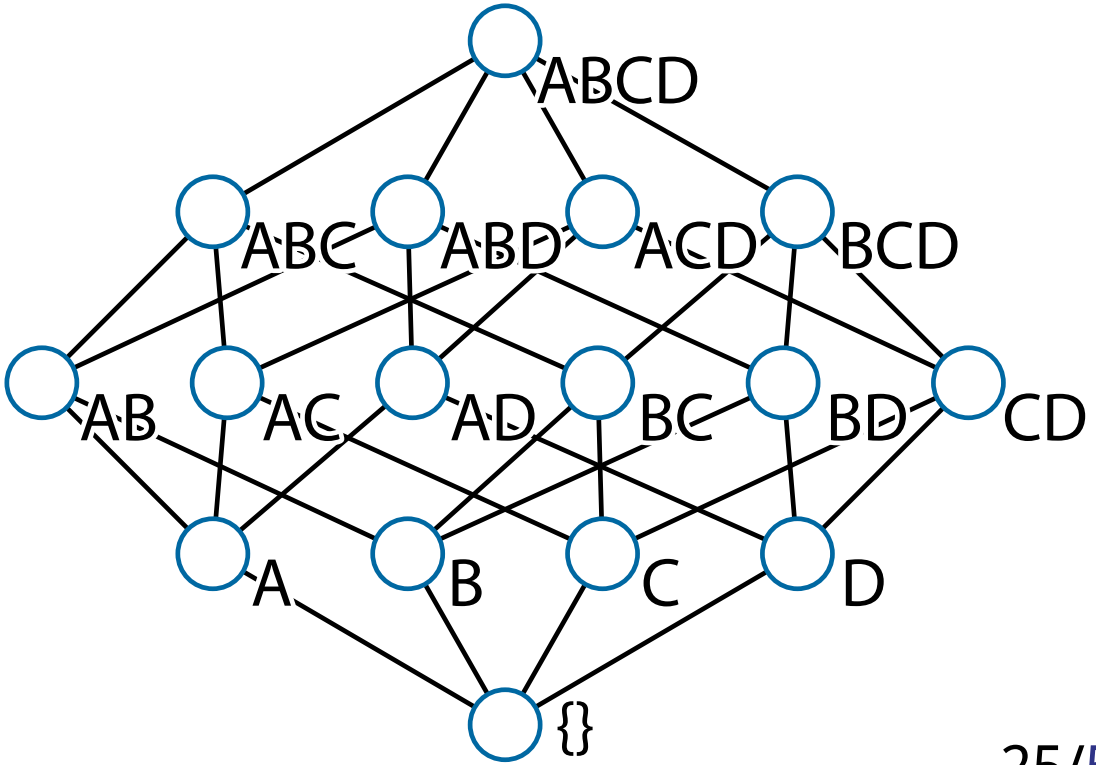
束（ラティス）での表現（2変数）



束（ラティス）での表現（3変数）



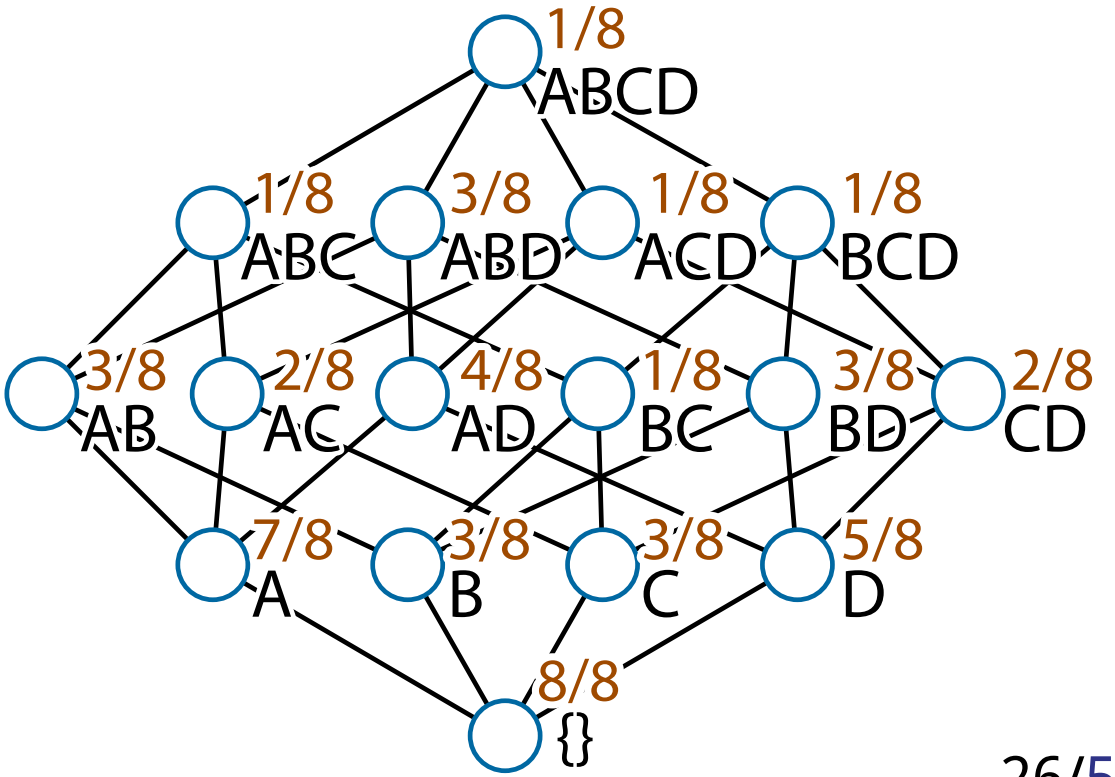
4変数の場合の束



各組合せの頻度

データ集合：

	A	B	C	D
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1



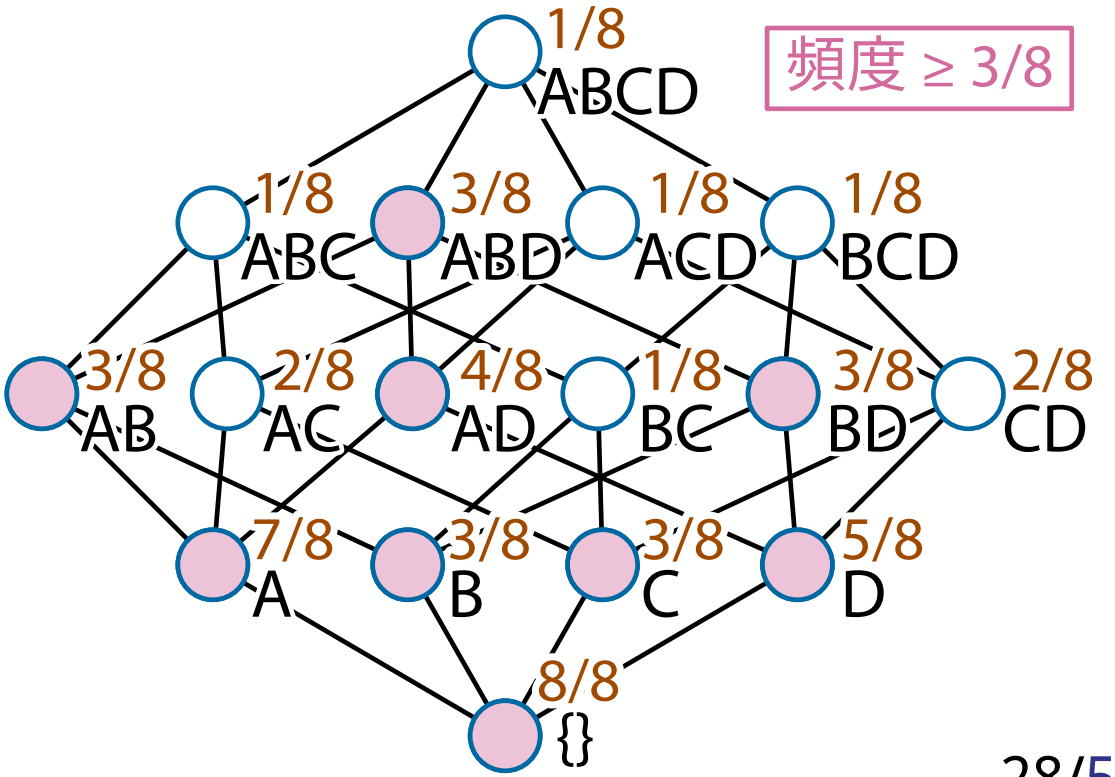
パターンマイニング

- 頻出するパターン（変数の組合せ）を見つける
- 代表的な手法：
  - Apriori, FP-growth, LCM, ...
- 入力は，2値ベクトルの集合 $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_n\}, \boldsymbol{x}_i \in \{0,1\}^d$
  - ただし，X 中の要素の重複を許す
- 出力は，パターン（アイテム集合）の集合：
  $\{s \subseteq \{1,2,\ldots,d\} \mid \eta(s) > \sigma\}$
  - $\eta(s)$ は s の頻度，σ はユーザが決める閾値

頻出パターンを探す

データ集合：

	A	B	C	D
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1



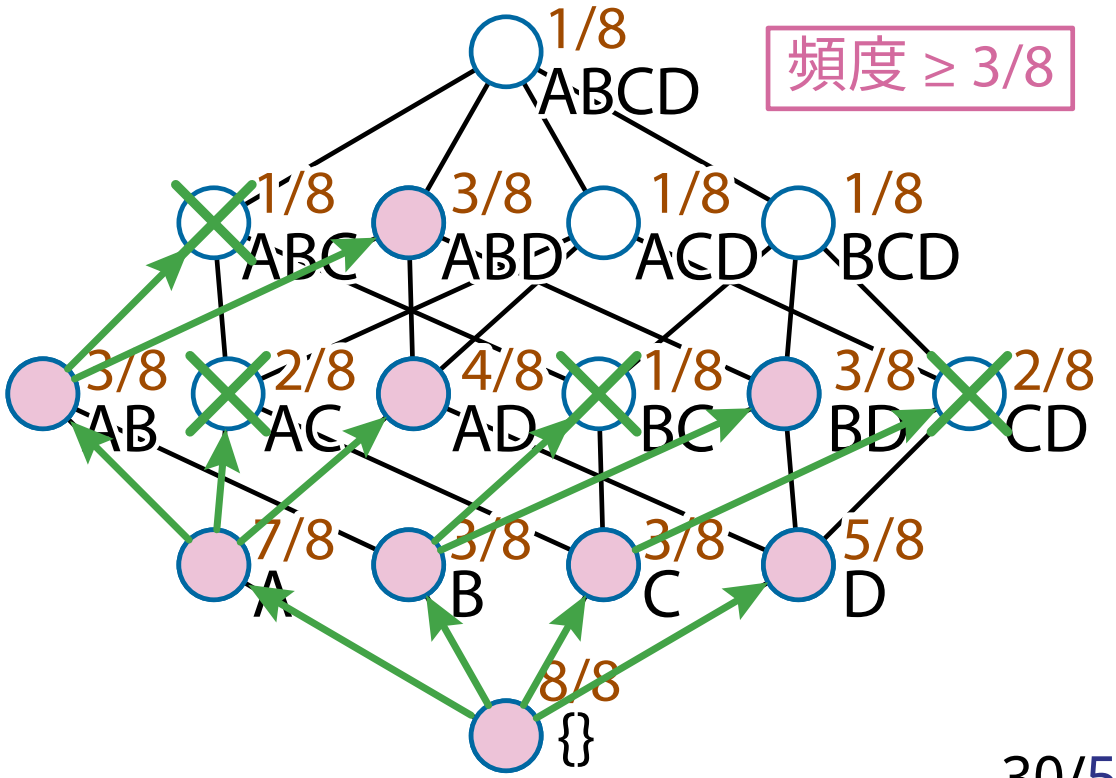
単純な全列挙 → 組合せ爆発！！ [YouTube]

変数の数	パターン数	おおよその計算時間
20	2^{20}	0.00059 sec.
30	2^{30}	0.6 sec.
40	2^{40}	10.2 min.
50	2^{50}	174 hours.
70	2^{70}	7 million days
100	2^{100}	8 thousand billion days

Apriori による列挙 [Agrawal & Srikant, 1994]

データ集合：

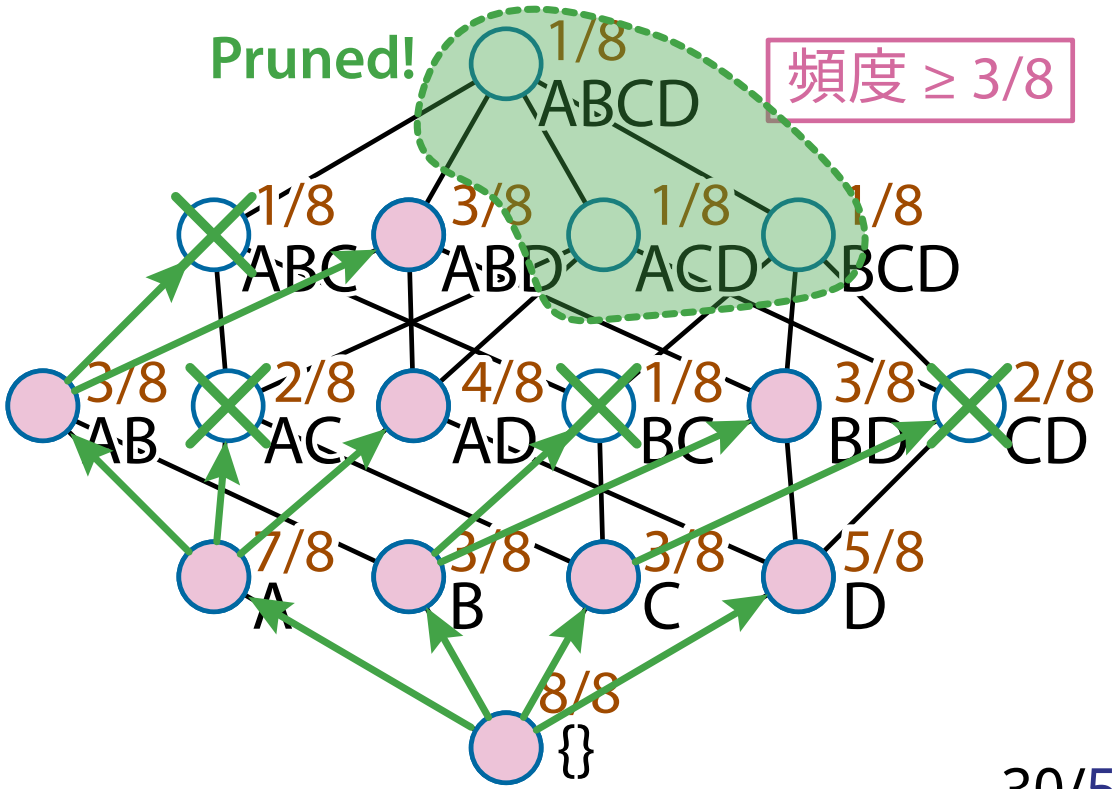
	A	B	C	D
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1



Apriori による列挙 [Agrawal & Srikant, 1994]

データ集合：

	A	B	C	D
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1



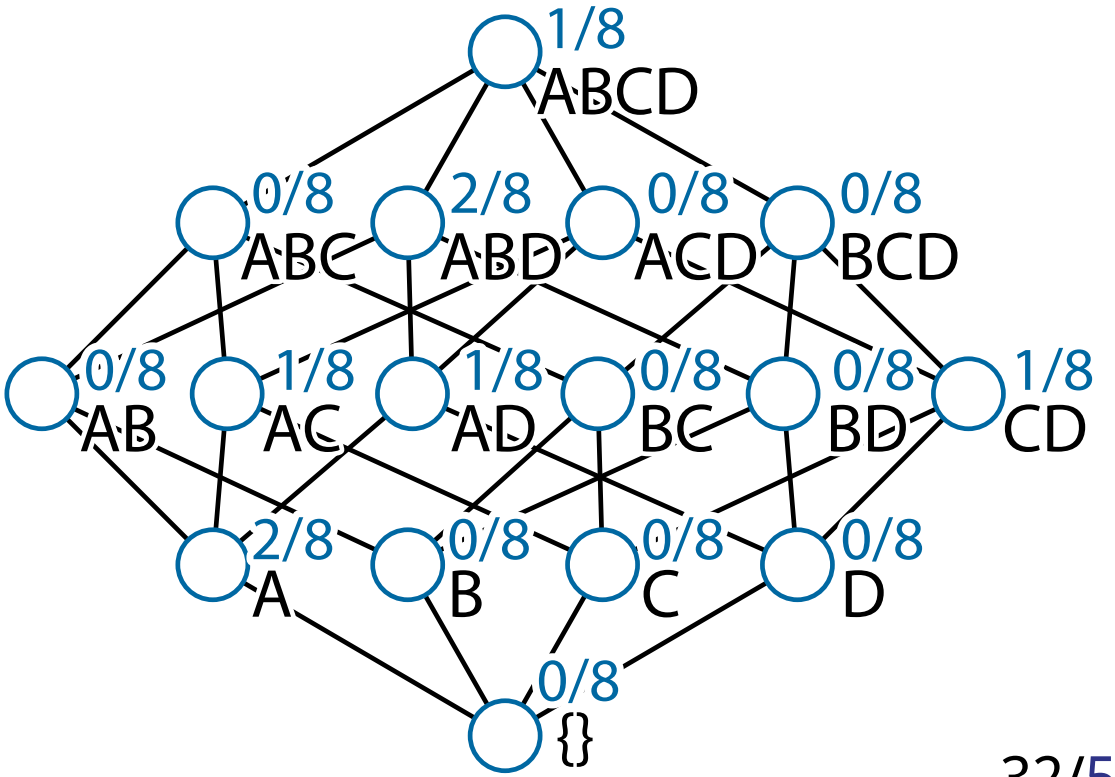
Algorithm 3: Apriori アルゴリズム

```
1 PatternMining(σ)
2     PatternEnumeration (⊥, σ)
3 PatternEnumeration(x, σ)
4     foreach s ⊃ x and |s| = |x| + 1 do
5         if η(s) ≥ σ then
6             Output s
7             PatternEnumeration (s, σ)
```

束上の（経験）確率分布

データ集合：

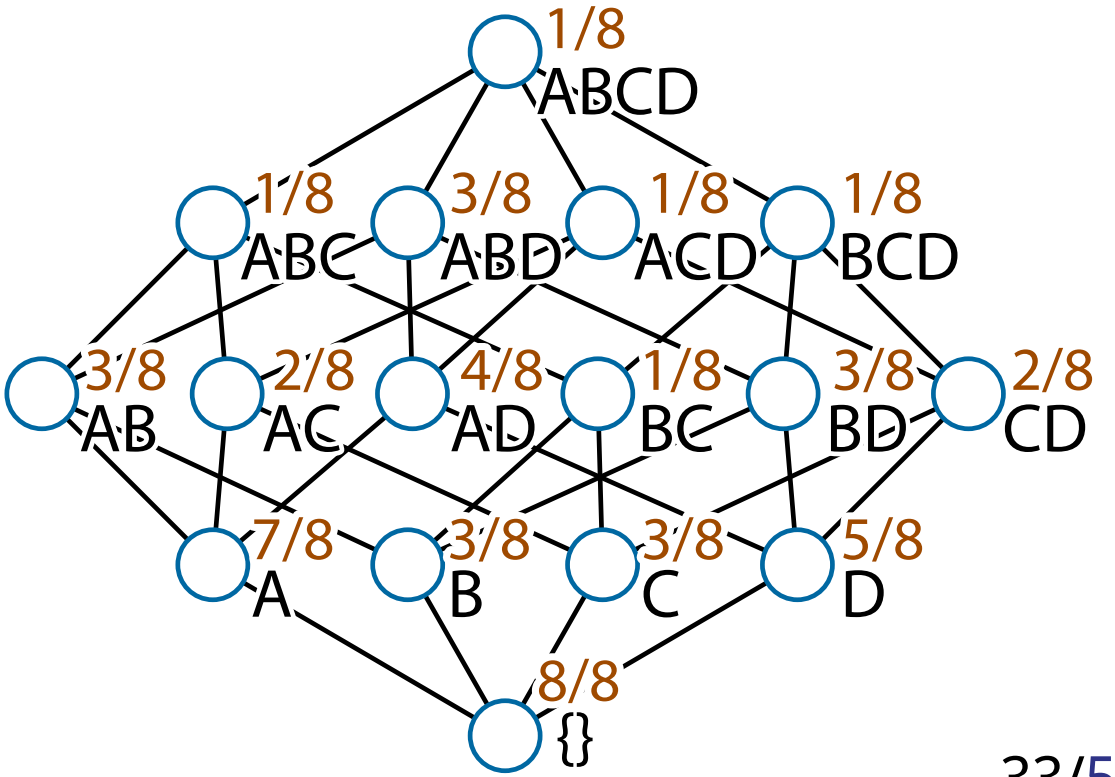
	A	B	C	D
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1



確率と頻度

データ集合：

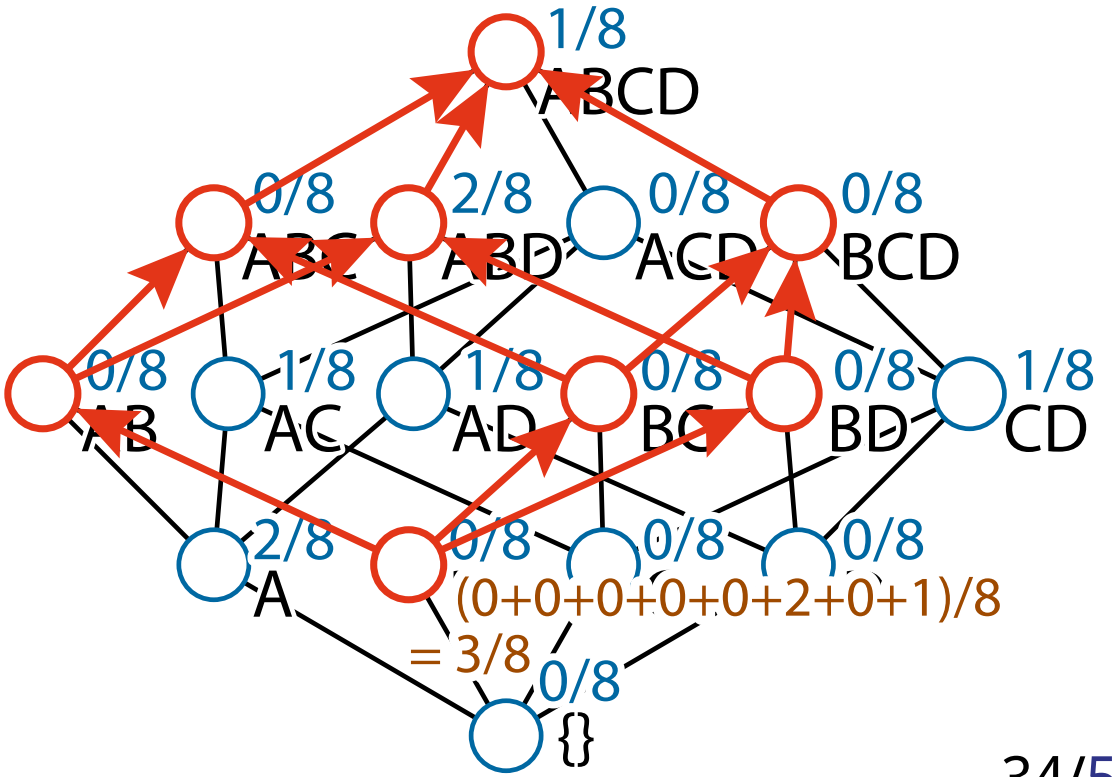
	A	B	C	D
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1



上方集合に含まれる確率の和 = 頻度

データ集合：

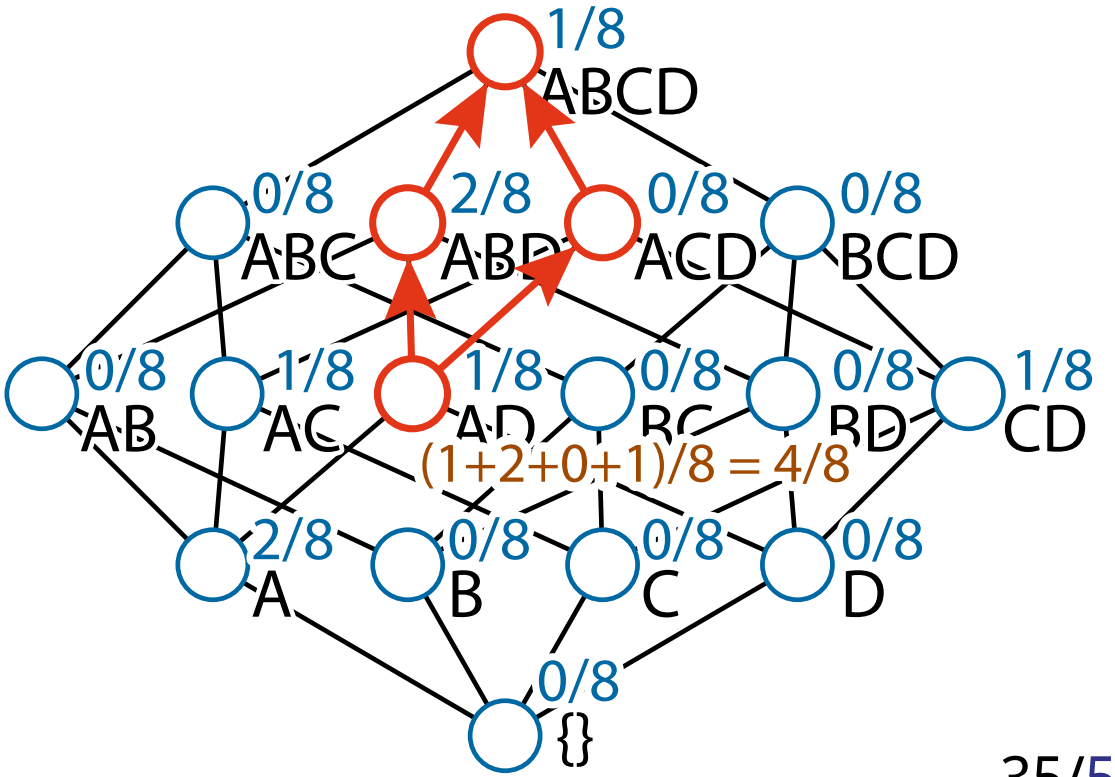
	A	B	C	D
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1



上方集合に含まれる確率の和 = 頻度

データ集合：

	A	B	C	D
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1

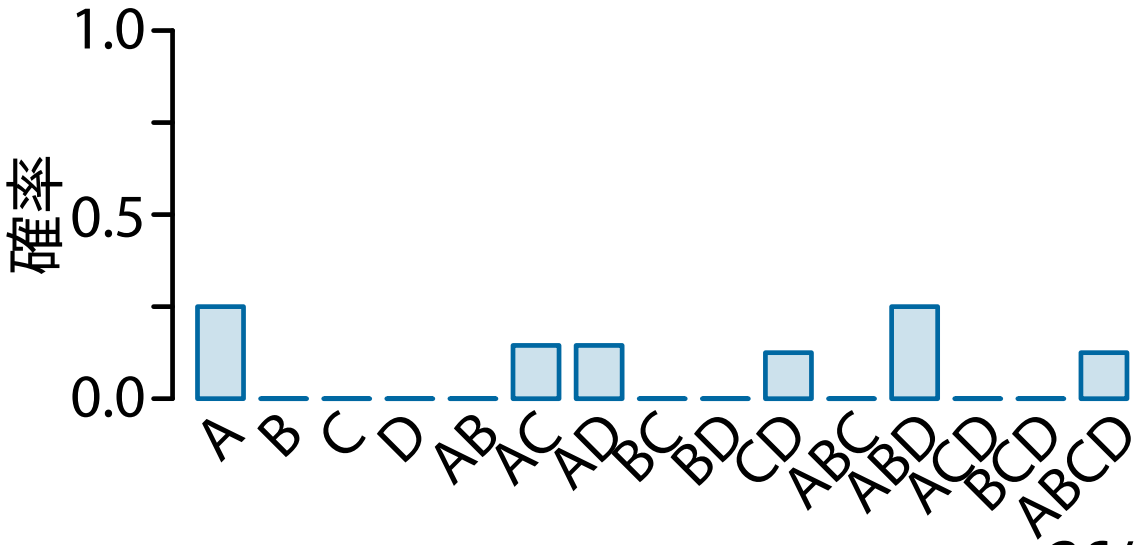


確率分布の推定

データ集合：

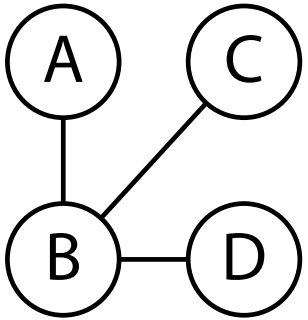
	A	B	C	D
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1

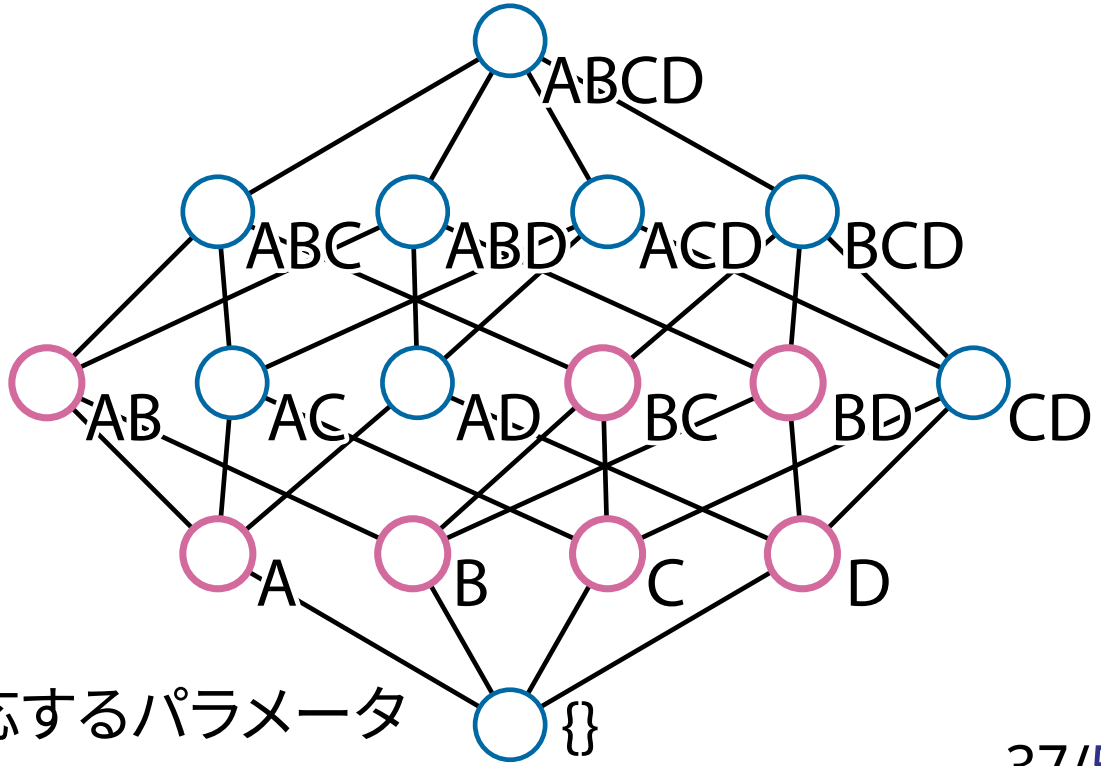
経験分布 ➔ 真の分布は？



ボルツマンマシン [Ackley, Hinton & Sejnowski, 1985]

ボルツマンマシン

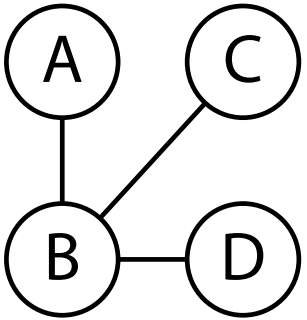


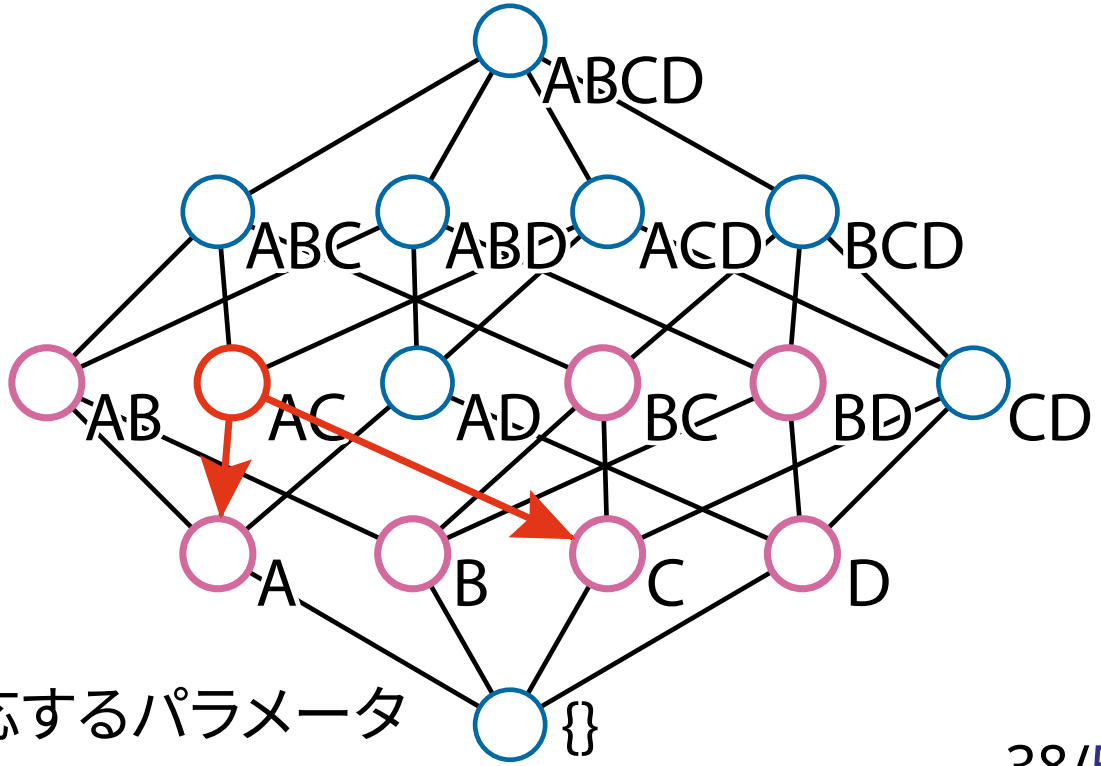


○：頂点と辺に対応するパラメータ

下方集合に含まれるパラメータの和 → 確率

ボルツマンマシン

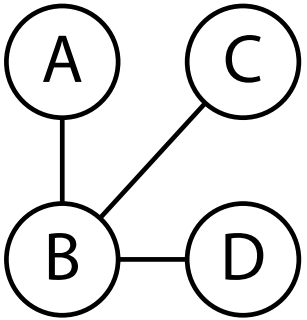


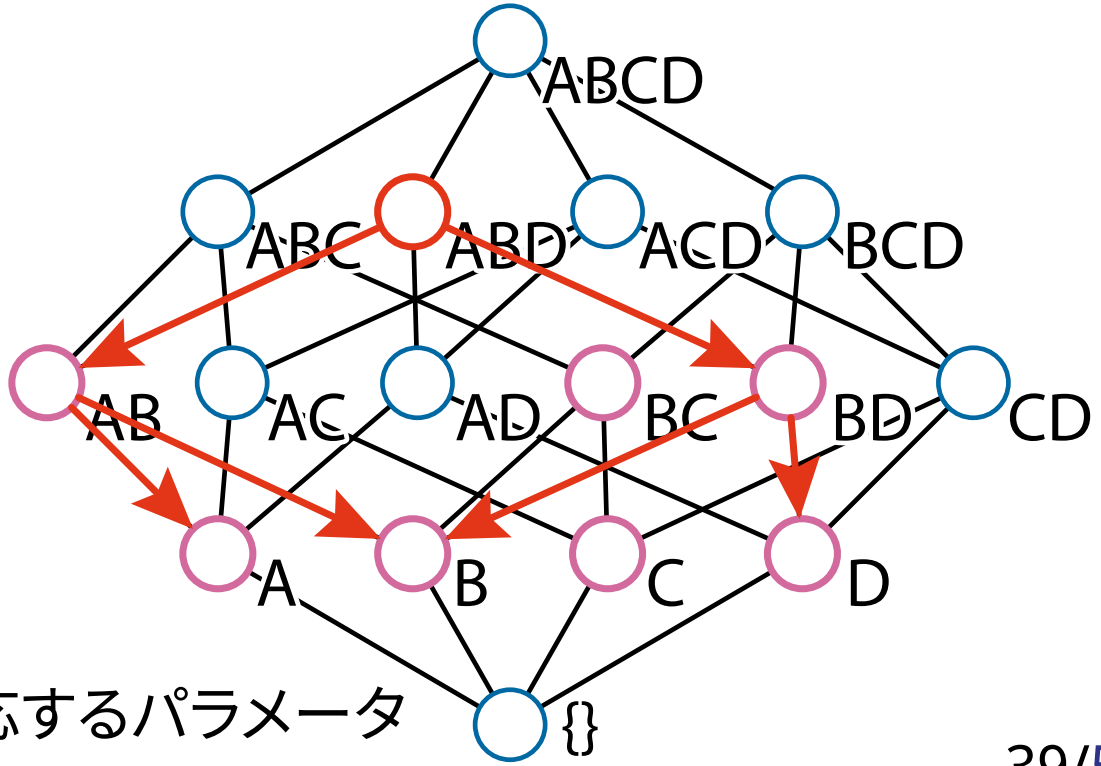


○：頂点と辺に対応するパラメータ

下方集合に含まれるパラメータの和 → 確率

ボルツマンマシン





○：頂点と辺に対応するパラメータ

ボルツマンマシンでの確率計算

- ボルツマンマシン：無向グラフ $G = (V, E)$
  - $V = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\},\ E = \{(\mathrm{A}, \mathrm{B}), (\mathrm{B}, \mathrm{C}), (\mathrm{B}, \mathrm{D})\}$
- パラメータ $\theta = (\theta_{\mathrm{A}}, \theta_{\mathrm{B}}, \theta_{\mathrm{C}}, \theta_{\mathrm{D}}, \theta_{\mathrm{AB}}, \theta_{\mathrm{BC}}, \theta_{\mathrm{BD}})$
- モデル分布の確率：

$$p(\mathrm{AC}; \theta) = \exp(\theta_{\mathrm{A}} + \theta_{\mathrm{C}}) / Z$$

$$p(\mathrm{ABD}; \theta) = \exp(\theta_{\mathrm{A}} + \theta_{\mathrm{B}} + \theta_{\mathrm{D}} + \theta_{\mathrm{AB}} + \theta_{\mathrm{BD}}) / Z$$

$$Z = \exp(-\theta_{\emptyset})$$ （正規化定数）

パラメータ θ の学習：最尤推定

- データ集合 $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ に対して，
  尤度 L を最大化するパラメータ θ を探す

  $L_X(\theta) = p(\boldsymbol{x}_1; \theta) \cdot p(\boldsymbol{x}_2; \theta) \cdot \cdots \cdot p(\boldsymbol{x}_n; \theta)$

- 基本戦略は勾配法：

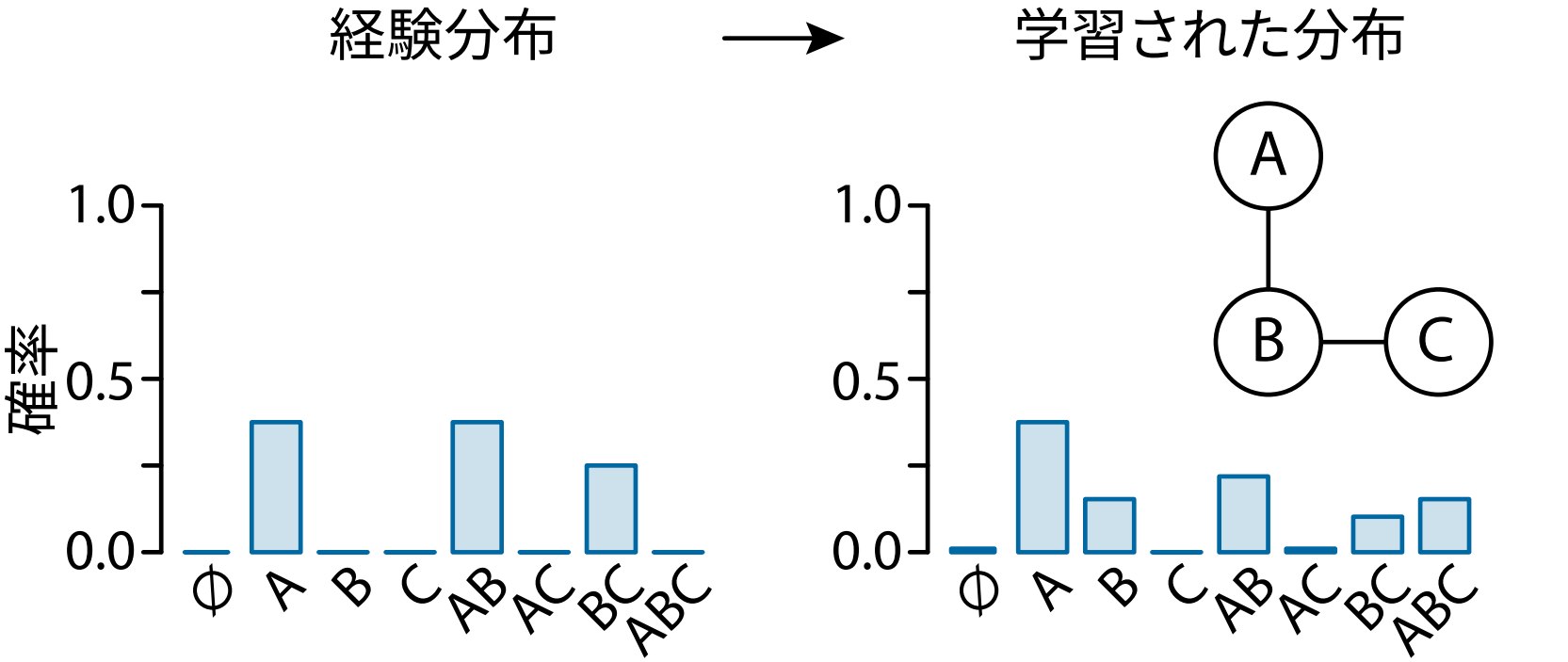
  (i) 適当な θ からスタート

  (ii) 各時点でゴールの方向（勾配）を計算

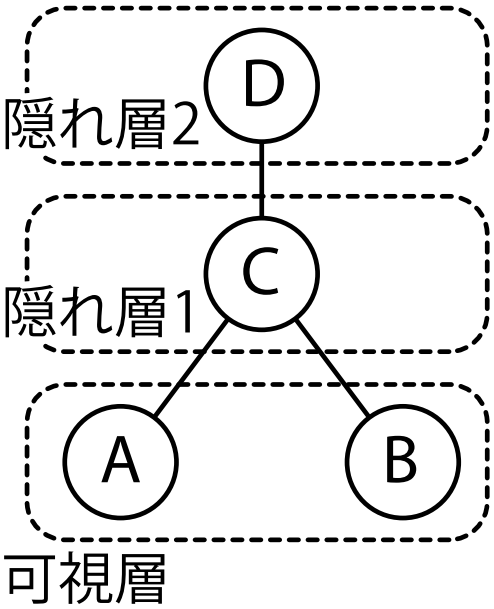
    - 勾配は，経験分布の頻度とモデル分布の頻度の差

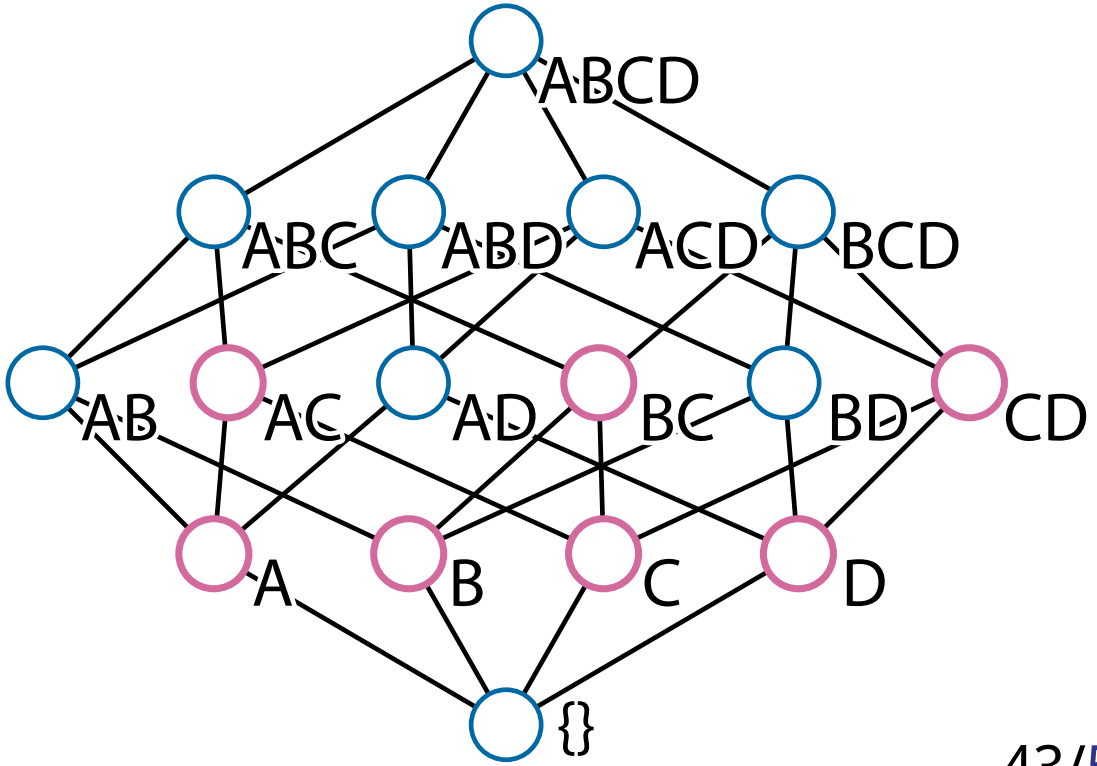
  (iii) その方向に少しだけ θ を動かす → (ii)へ

学習結果の例



深層ボルツマンマシン [Salakhutdinov & Hinton, 2009]





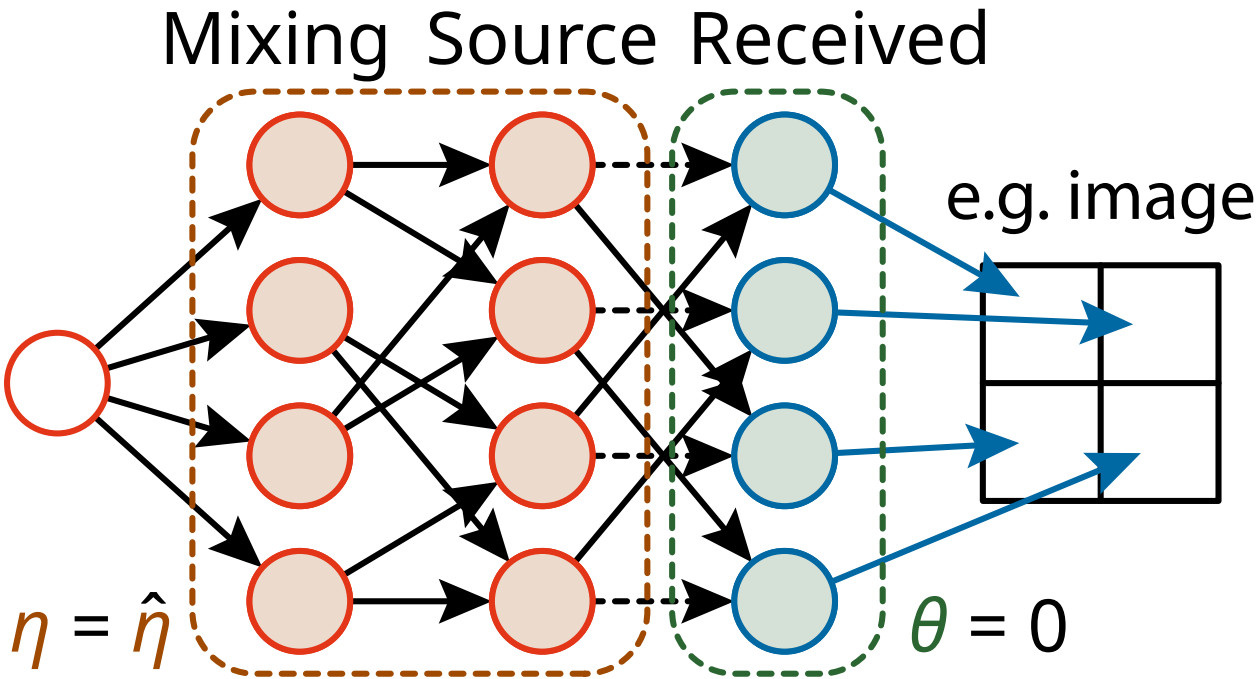
バイアス–バリアンストレードオフ

- パラメータを増やしていくと，尤度は上がっていくが，データに過度に適合するようになる
- 極端な例：
  - 全パラメータを使うと，データの経験分布そのものを出力
    → バイアス 0，バリアンス大
  - パラメータなしだと，データに関わらず一様分布を出力
    → バイアス大，バリアンス 0
- 必ずバイアスとバリアンスのトレードオフがある

情報幾何との関連 [スライド]

- 「確率分布を全部集めた集合」（多様体）を考える
- ボルツマンマシンのパラメータ θ と，確率から得られる頻度 η は，それぞれこの多様体の座標系になっている
- さらに，θ と η は必ず直交しており，双対平坦な座標系
- θ と η をうまく組み合わせると，最尤推定の枠組みを様々なタスクに適用できるようになる
  - 行列のバランス化，テンソル分解，信号分離，...

Blind Source Separation [arXiv]

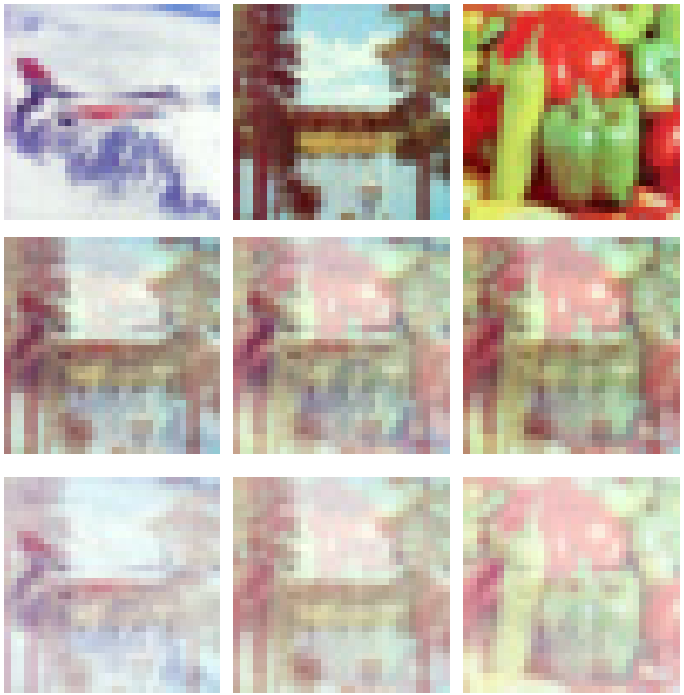


BSSの結果

Source

Received

Reconst.



Method	RMSE
IGBSS	**0.27032**
FastICA	0.43630
NMF	0.62195
DicLearn	0.37167

まとめ

- 機械学習とは：（計算機の）「学習」についての科学
  - 目的：過去の経験（データ）を一般化する規則を見つける
- 機械学習の研究では，主に以下に取り組む
  - 実世界の現象を数理モデルで表す
  - 「学習」という行為を定式化する
  - 計算機上で実行するためのアルゴリズムを設計し実装する
  - 実世界での要求に応じて学習結果を評価する
- **研究に興味があったら，ぜひ総研大へ！**

機械学習一般についての情報源

- たくさんの書籍
  - K.Murphy, *Machine Learning: A Probabilistic Perspective*, The MIT Press
  - M.J.Zaki, W.Meira Jr., *Data Mining and Analysis*, Cambridge University Press
  - 中川裕志, **東京大学工学教程 情報工学 機械学習**, 丸善出版
- Coursera などの講義動画
- Kaggle でのコンペティション

機械学習の研究についての情報源

- 機械学習研究の最先端は，国際会議で論文として発表
  - 機械学習・データマイニング
    - ICML (International Conference on Machine Learning)
    - NeurIPS (Neural Information Processing Systems)
    - KDD (Knowledge Discovery and Data Mining)
  - 人工知能全般
    - IJCAI (Inter. Joint Conference on Artificial Intelligence)
    - AAAI (Conference on Artificial Intelligence)