

BDDによる最適領域抽出手法

(株) 富士通研究所
樋口 博之

自己紹介

■ 所属：富士通研究所

- ERATO出向中の岩下さんと同じ部署

■ 研究分野

■ もともとLSI設計CAD

- ・ BDDなどを利用した論理回路の自動合成・解析
- ・ SRAMの歩留解析(BDDなどから離れていた)

■ BDDなどのLSI CAD技術を他分野に応用する研究開発

- ・ BDDのおさらい必要.

■ 企業側でBDDなど離散構造技術の適用を進めたい.

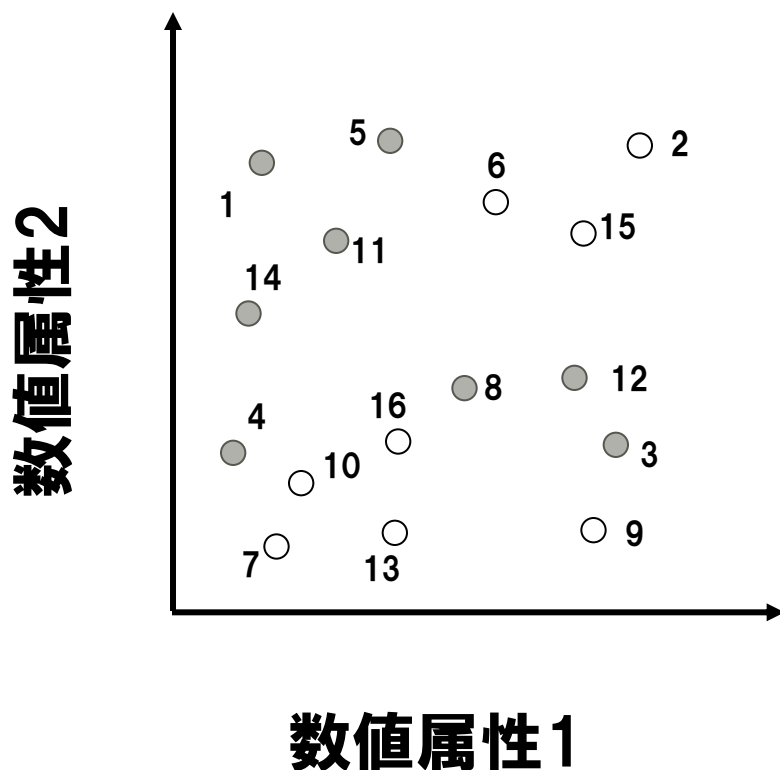
- よい適用先・利用方法のアドバイスあればぜひお教えてください.

最適領域抽出問題

最適領域抽出問題

= データマイニングの数値属性相関ルール抽出問題

例) 属性が2次元(数値属性1, 2)の場合



データ = (属性1, 属性2, yes/no)

○ = yes (ルール成立)

● = no (ルール不成立)

ある領域に対するゲイン

= 「yes点の数」 - 「no点の数」



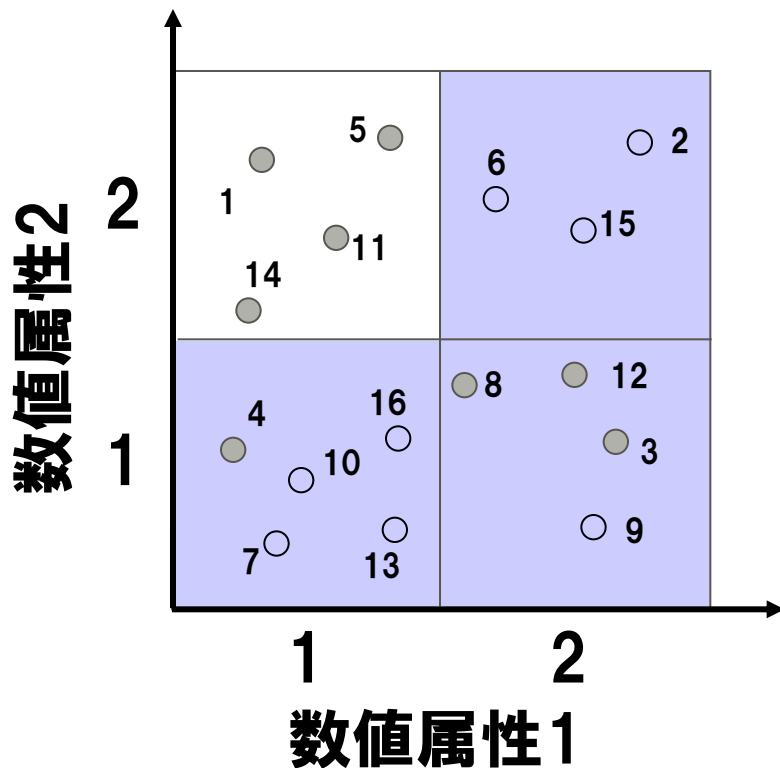
ゲイン最大の領域を抽出する。

最適領域抽出問題(続き)

2×2 (=4ピクセル) のメッシュで考えた場合の最適領域

ゲインは、「yesデータ数」8 - 「noデータ数」4 = 4

○ = yesデータ ● = noデータ

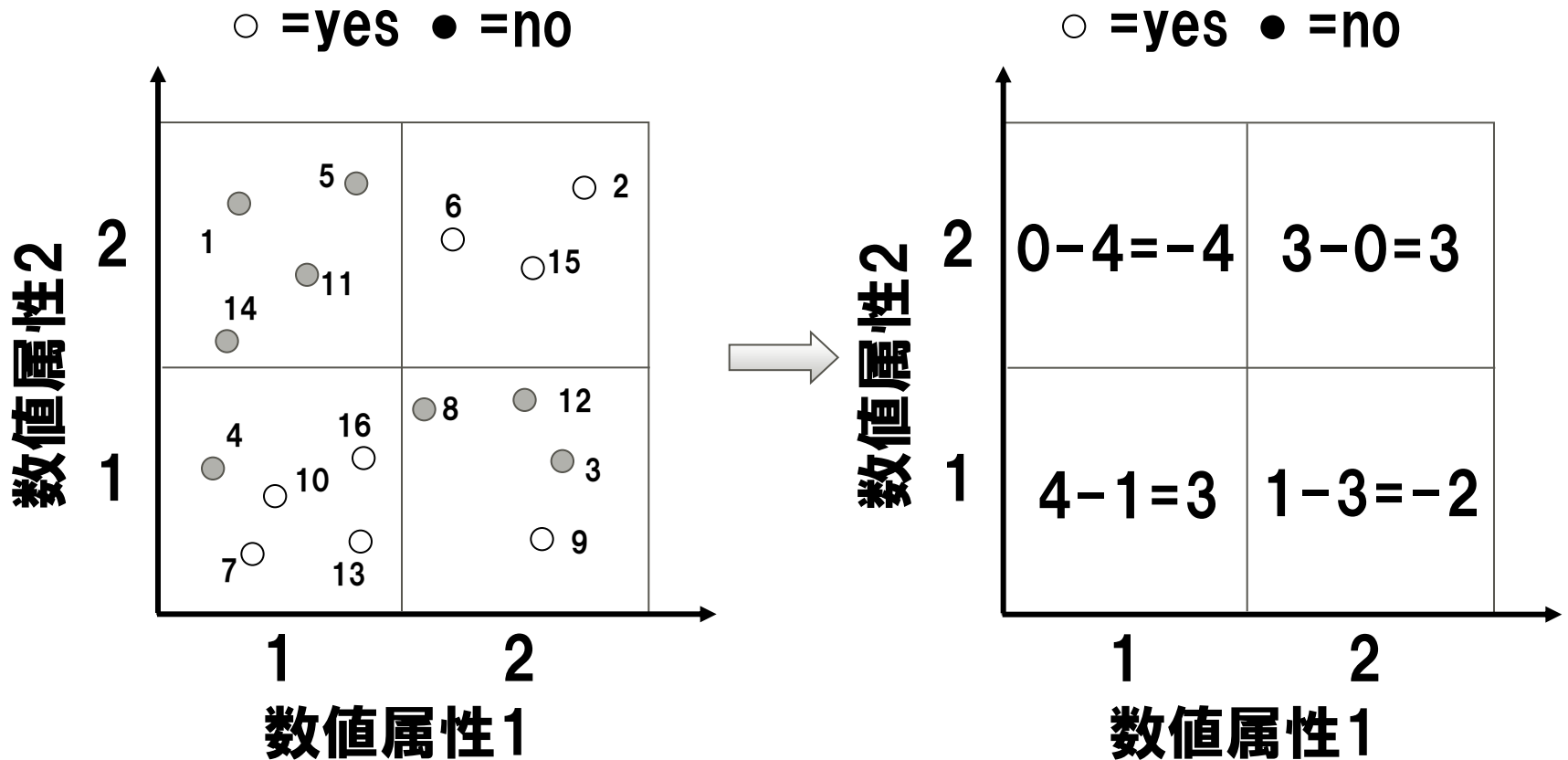


従来法と課題

- 1次元の場合, 区間数 N に対し, $O(N)$ アルゴリズム [福田01,加藤08]
- 課題
 - n 次元の場合, 2次元以上の部分は組合せ全てを試す必要あり.
 - ・ 各次元の区間数 N に対し, $O(N^{2(n-1)+1})$.
 - 矩形領域しか考慮できない.

従来法

①各ピクセルのゲインをそれぞれ計算する。



従来法

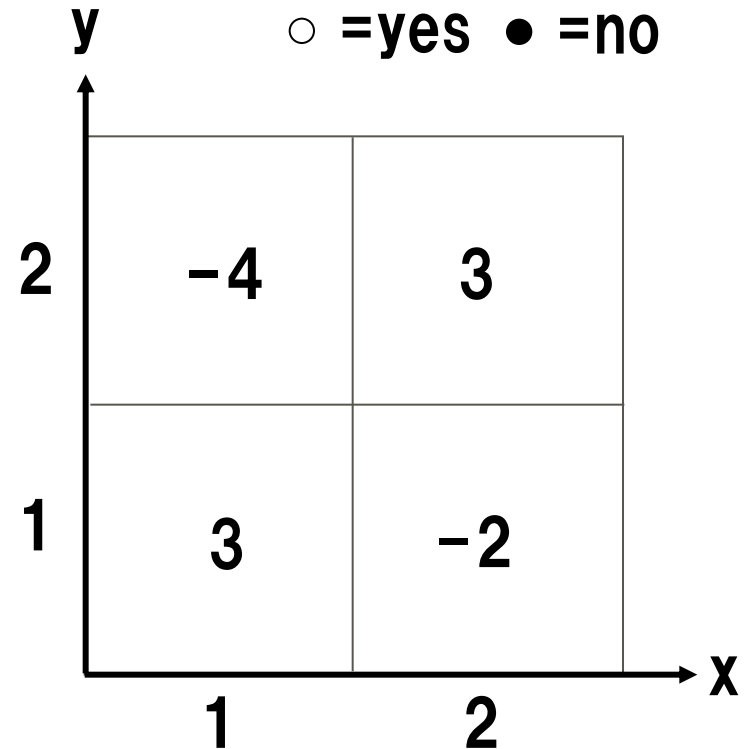
■ ②ゲインが最大の領域を求める。

- y方向の区間を $[y1, y2]$ に固定して, x方向の最適区間を一次元のO(N)アルゴリズムで解く. $[y1, y2]$ を任意の区間について調べて, 最適な矩形領域を選ぶ

(i) $y = [1, 1]$ 固定のとき

(ii) $y = [2, 2]$ 固定のとき

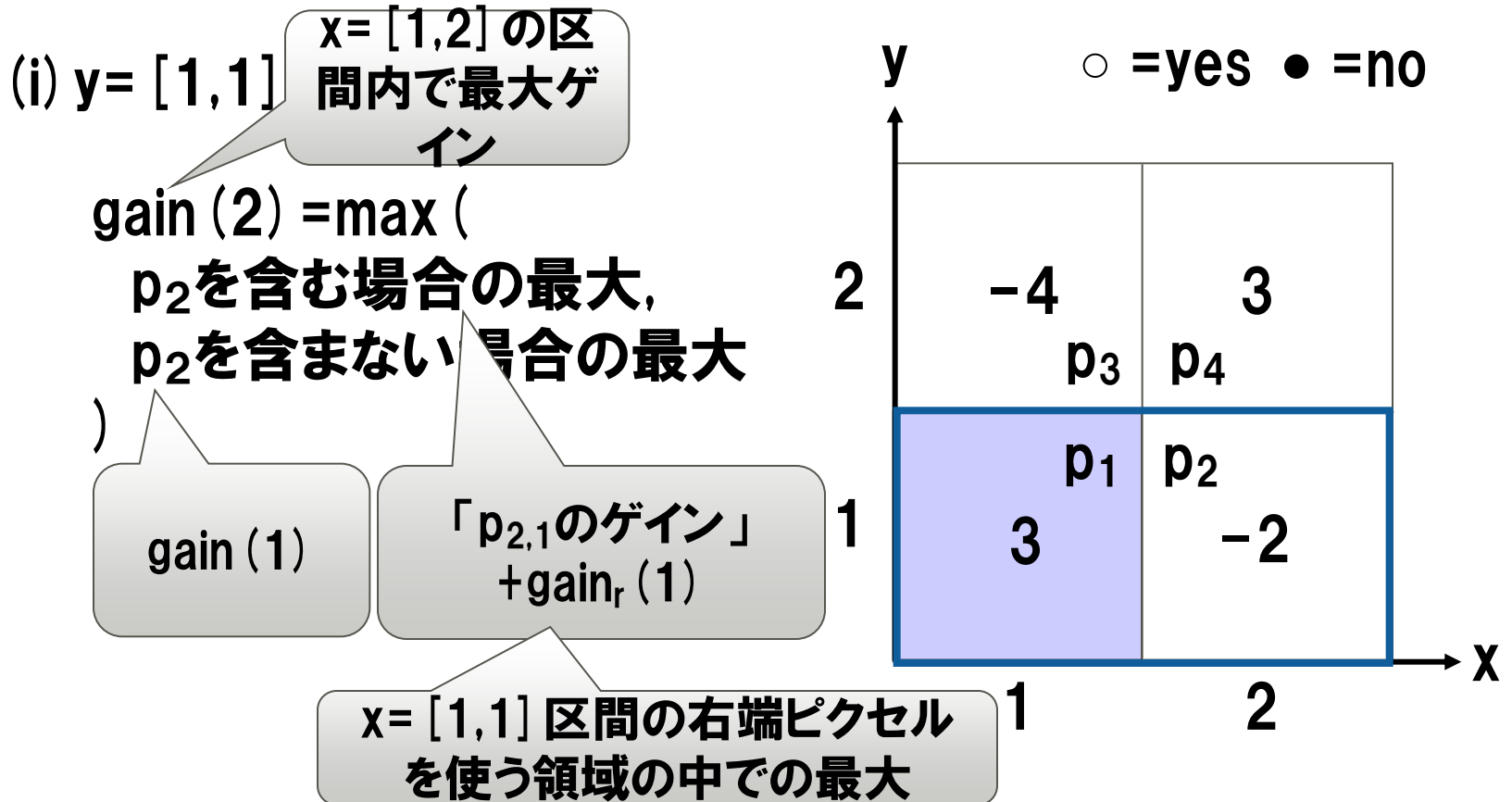
(iii) $y = [1, 2]$ 固定のとき



従来法

■ ②ゲインが最大の領域を求める。

- y方向の区間を $[y1, y2]$ に固定して, x方向の最適区間を一次元のO(N)アルゴリズムで解く. $[y1, y2]$ を任意の区間について調べて, 最適な矩形領域を選ぶ



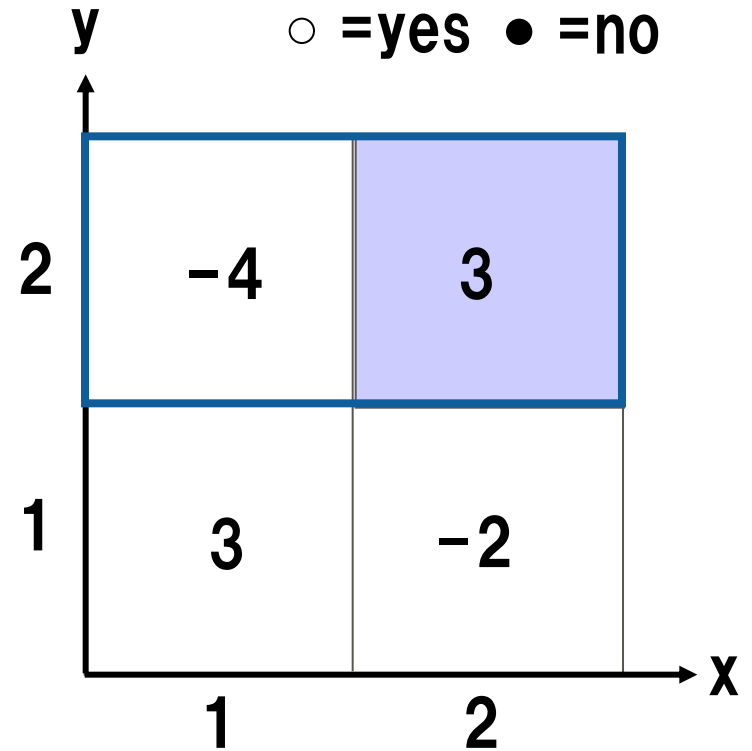
従来法

■ ②ゲインが最大の領域を求める。

- y方向の区間を $[y1, y2]$ に固定して, x方向の最適区間を一次元のO(N)アルゴリズムで解く. $[y1, y2]$ を任意の区間について調べて, 最適な矩形領域を選ぶ

(i) $y = [1, 1]$ 固定のとき

(ii) $y = [2, 2]$ 固定のとき



従来法

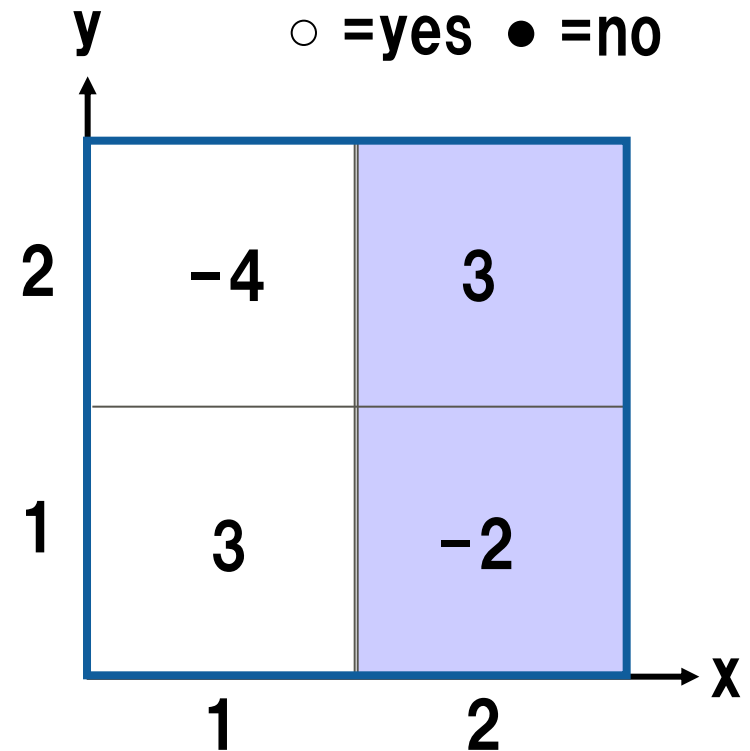
■ ②ゲインが最大の領域を求める。

- y方向の区間を $[y1, y2]$ に固定して, x方向の最適区間を一次元のO(N)アルゴリズムで解く. $[y1, y2]$ を任意の区間について調べて, 最適な矩形領域を選ぶ

(i) $y = [1, 1]$ 固定のとき

(ii) $y = [2, 2]$ 固定のとき

(iii) $y = [1, 2]$ 固定のとき



従来法

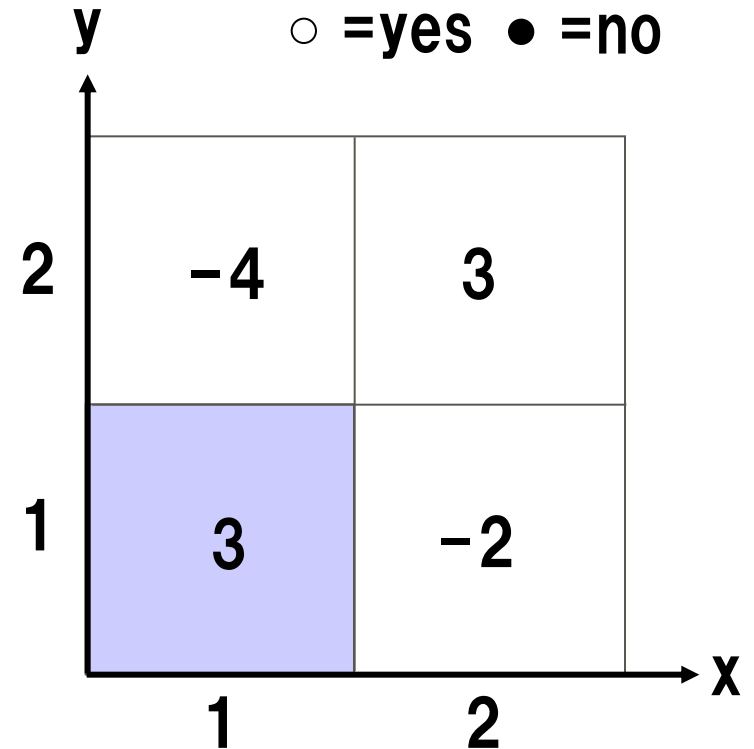
■ ②ゲインが最大の領域を求める.

- y方向の区間を $[y1,y2]$ に固定して, x方向の最適区間を一次元のO(N)アルゴリズムで解く. $[y1,y2]$ を任意の区間について調べて, 最適な矩形領域を選ぶ

(i) $y = [1,1]$ のとき ゲイン3

(i) $y = [2,2]$ のとき ゲイン3

(ii) $y = [1,2]$ のとき ゲイン1

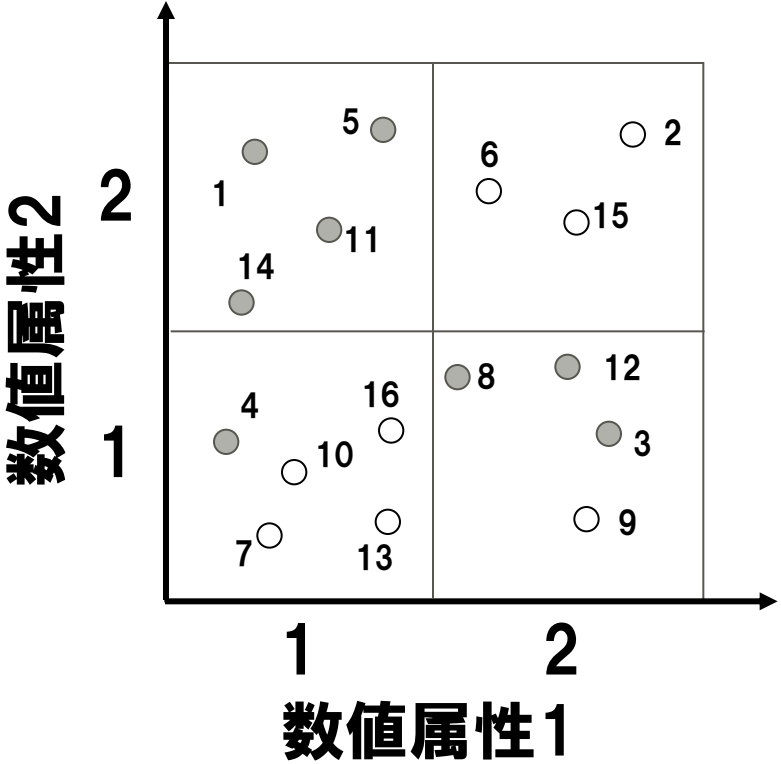


本手法の狙い

- 問題の難しさは考慮する領域の数が組合せ的に増える点にある.
- 領域 = ピクセルの集合
- 領域の集合 → ピクセルの集合の集合 → BDD
- (領域, ゲイン値) 対の集合 → BDDベクトル
- 最大ゲインの領域の集合をBDD処理により抽出する.

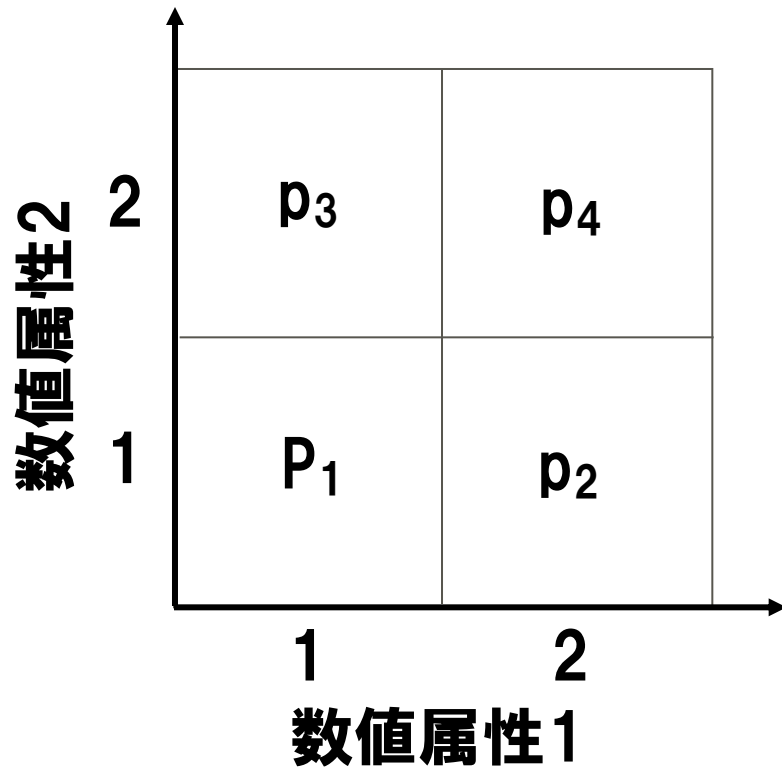
本手法(例による説明)

○ =yesデータ ● =noデータ



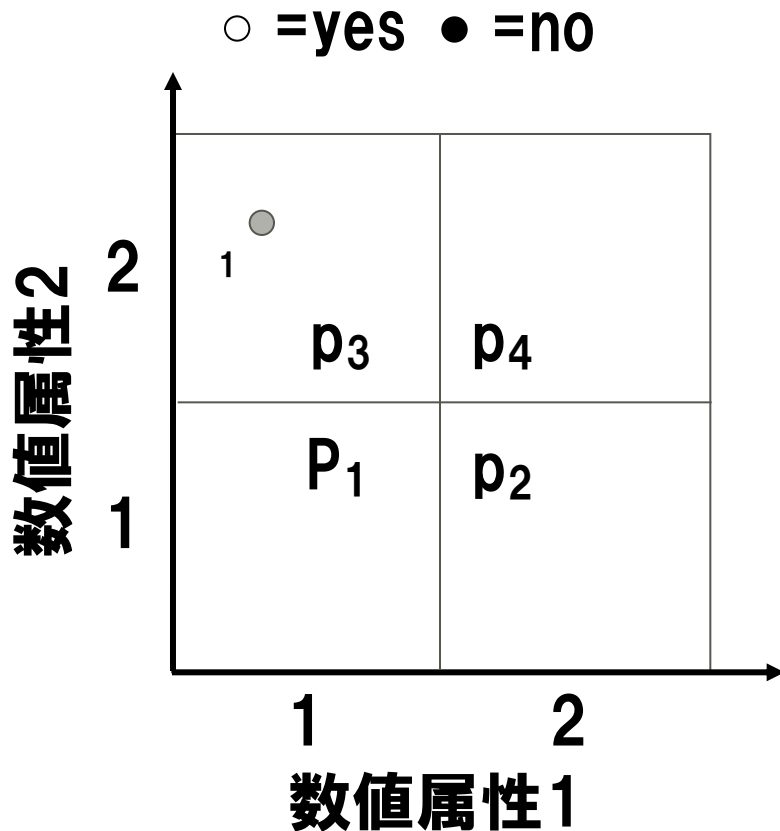
本手法(変数割当)

①各ピクセルに論理変数を割り当てる。



本手法(ゲイン計算)

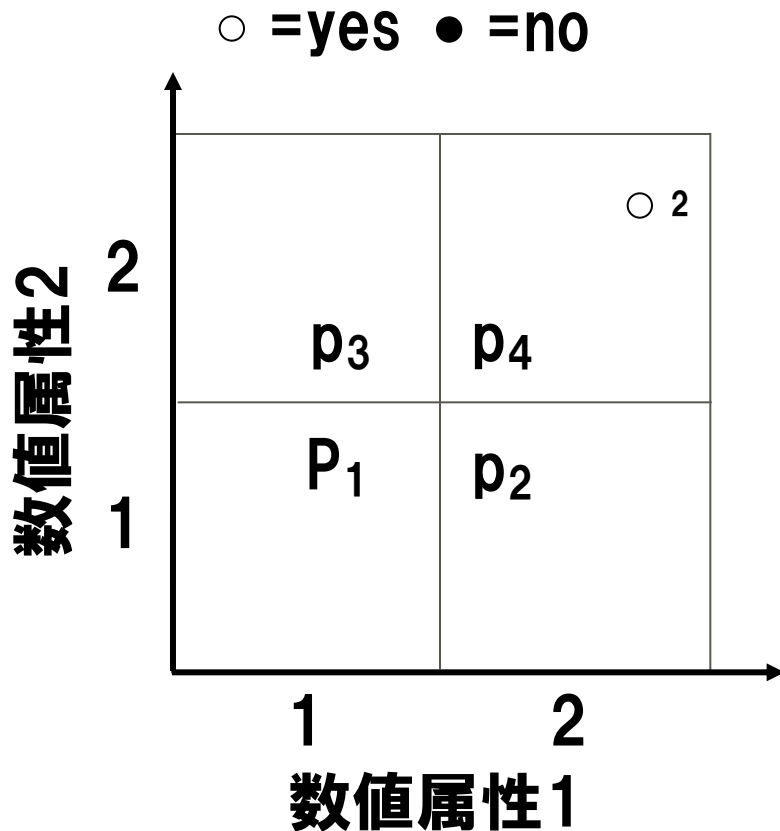
- ②各データを順に処理して、各領域のゲインをBDDの演算で一括して行う。



| input | old | cur. | new | (new) |
|-------------------|-----------|-------------|-------------|-------|
| $p_1 p_2 p_3 p_4$ | $f_1 f_2$ | $+ f_1 f_2$ | $= f_1 f_2$ | (f) |
| 0000 | 00 | 00 | 00 | (0) |
| 0001 | 00 | 00 | 00 | (0) |
| 0010 | 00 | 11 | 11 | (-1) |
| 0011 | 00 | 11 | 11 | (-1) |
| 0100 | 00 | 00 | 00 | (0) |
| 0101 | 00 | 00 | 00 | (0) |
| 0110 | 00 | 11 | 11 | (-1) |
| 0111 | 00 | 11 | 11 | (-1) |
| 1000 | 00 | 00 | 00 | (0) |
| 1001 | 00 | 00 | 00 | (0) |
| 1010 | 00 | 11 | 11 | (-1) |
| 1011 | 00 | 11 | 11 | (-1) |
| 1100 | 00 | 00 | 00 | (0) |
| 1101 | 00 | 00 | 00 | (0) |
| 1110 | 00 | 11 | 11 | (-1) |
| 1111 | 00 | 11 | 11 | (-1) |

本手法(ゲイン計算)

- ②各データを順に処理して、各矩形領域のゲインをBDDの演算で一括して行う。

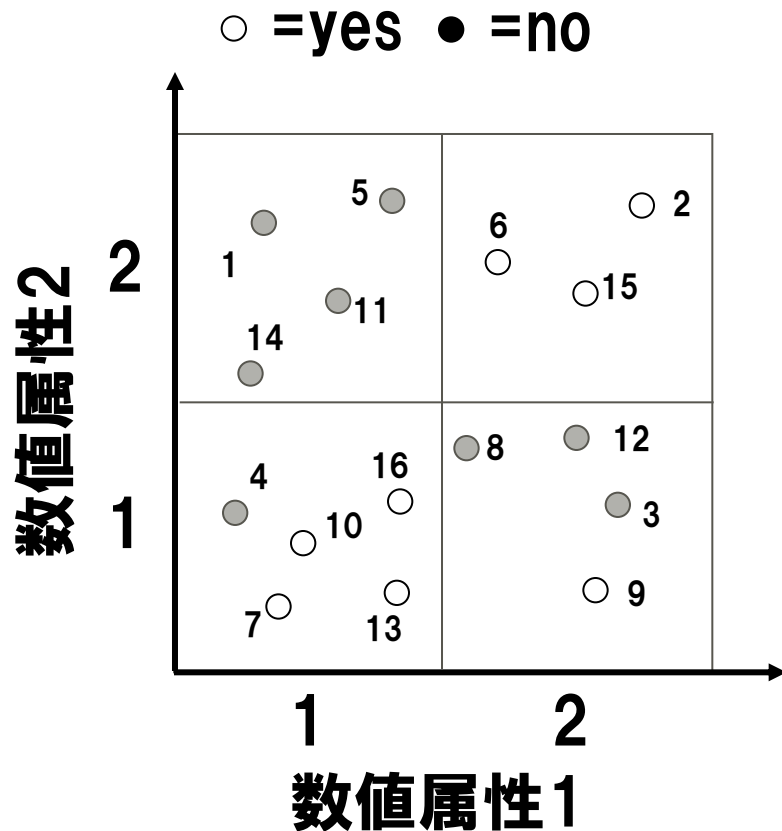


| input $p_1 p_2 p_3 p_4$ | old $f_1 f_2$ | cur. $+ f_1 f_2$ | new $= f_1 f_2$ | (new) (f) |
|----------------------------|------------------|---------------------|--------------------|--------------|
| 0000 | 00 | 00 | 00 | (0) |
| 0001 | 00 | 01 | 01 | (1) |
| 0010 | 11 | 00 | 11 | (-1) |
| 0011 | 11 | 01 | 00 | (0) |
| 0100 | 00 | 00 | 00 | (0) |
| 0101 | 00 | 01 | 01 | (1) |
| 0110 | 11 | 00 | 11 | (-1) |
| 0111 | 11 | 01 | 00 | (0) |
| 1000 | 00 | 00 | 00 | (0) |
| 1001 | 00 | 01 | 01 | (1) |
| 1010 | 11 | 00 | 11 | (-1) |
| 1011 | 11 | 01 | 00 | (0) |
| 1100 | 00 | 00 | 00 | (0) |
| 1101 | 00 | 01 | 01 | (1) |
| 1110 | 11 | 00 | 11 | (-1) |
| 1111 | 11 | 01 | 00 | (0) |

本手法(ゲイン計算)

- ②各データを順に処理して, 各矩形領域のゲインをBDDの演算で一括して行う.

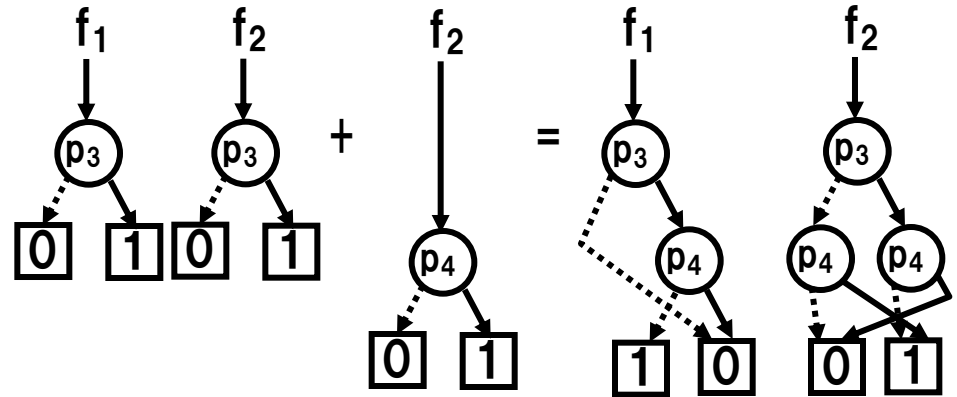
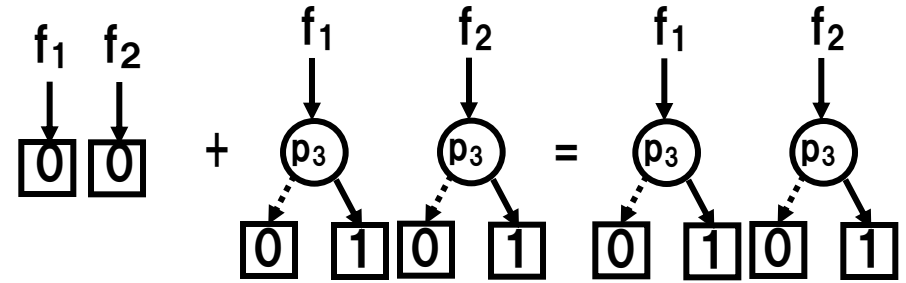
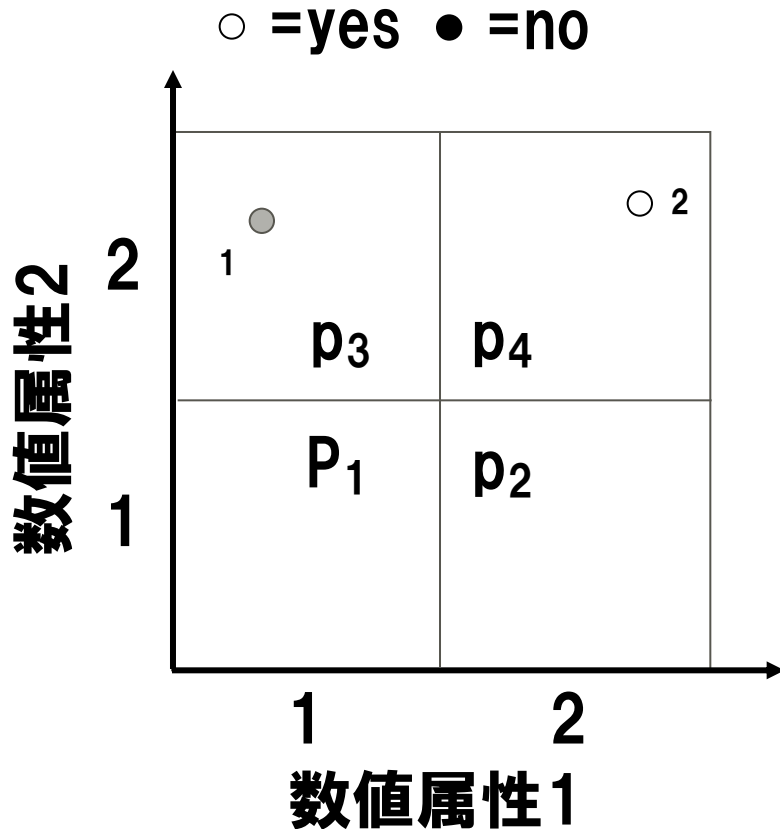
ゲイン計算完了時



| input | | | | output | | | | (f) |
|------------------|------------------|------------------|------------------|----------------|----------------|----------------|----------------|------|
| p _{1,1} | p _{2,1} | p _{1,2} | p _{2,2} | f ₁ | f ₂ | f ₃ | f ₄ | (f) |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | (0) |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | (3) |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | (-3) |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | (0) |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | (-2) |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | (-1) |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | (-1) |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | (-2) |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | (3) |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | (6) |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | (0) |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | (3) |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | (1) |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | (4) |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | (-2) |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | (0) |

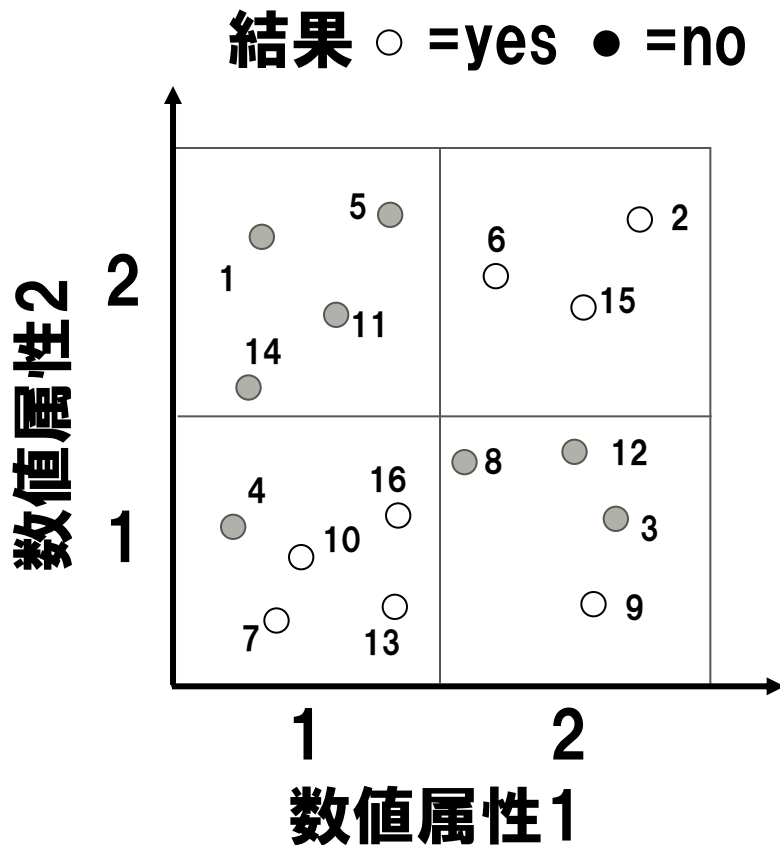
本手法(ゲイン計算 (BDDで書くと))

- ②各データを順に処理して、各矩形領域のゲインをBDDの演算で一括して行う。



本手法(制約付加と最適領域抽出)

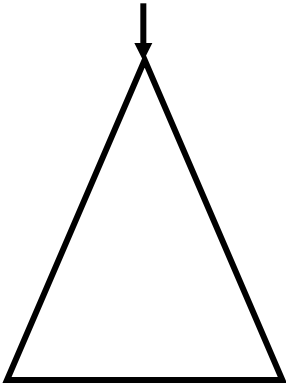
- ③領域として考慮しないものの削除と全ての最適ゲイン領域の抽出をBDDの論理演算により行う。



| input | output | 連続制約 | |
|-------------------------------------------------------------|-------------------------------------------------------------|----------|-----------------|
| p ₁ p ₂ p ₃ p ₄ | f ₁ f ₂ f ₃ f ₄ | c | (f) |
| 0000 | 0000 | 1 | (0) |
| 0001 | 0011 | 1 | (3) |
| 0010 | 1100 | 1 | (-4) |
| 0011 | 1111 | 1 | (-1) |
| 0100 | 1110 | 1 | (-2) |
| 0101 | 0001 | 1 | (1) |
| 0110 | 1010 | 0 | (-6) |
| 0111 | 1101 | 1 | (-3) |
| 1000 | 0011 | 1 | (3) |
| 1001 | 0110 | 0 | (6) |
| 1010 | 1111 | 1 | (-1) |
| 1011 | 0010 | 0 | (2) |
| 1100 | 0001 | 1 | (1) |
| 1101 | 0100 | 1 | (4) |
| 1110 | 1101 | 1 | (-3) |
| 1111 | 0000 | 1 | (0) |

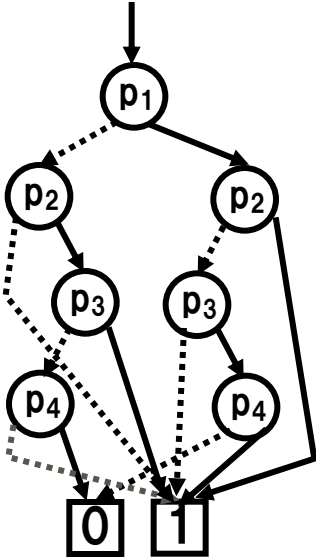
本手法（領域連続の制約付加）

領域集合のBDD



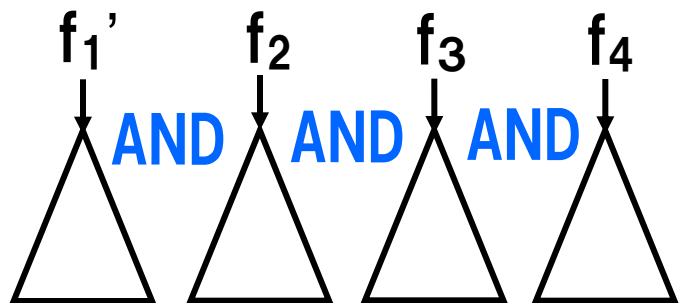
AND演算

連続制約を表すBDD



本技術（最適領域抽出）

(符号) 重み4 重み2 重み1



恒偽でない場合のみ
結果を更新していく

| input | output | 連続制約 | 最適領域抽出 | | | |
|-------------------|-------------------|----------|--------------------------|-------------------------|----------|---------------|
| $p_1 p_2 p_3 p_4$ | $f_1 f_2 f_3 f_4$ | c | $f_1' * c \rightarrow z$ | $z * f_2 \rightarrow z$ | z | $z * f_3 = 0$ |
| 0000 | 0000 | 1 | 1 | 1 | 1 | 0 |
| 0001 | 0011 | 1 | 1 | 1 | 1 | 0 |
| 0010 | 1100 | 1 | 0 | 1 | 0 | 1 |
| 0011 | 1111 | 1 | 0 | 1 | 0 | 1 |
| 0100 | 1110 | 1 | 0 | 1 | 0 | 1 |
| 0101 | 0001 | 1 | 1 | 1 | 1 | 0 |
| 0110 | 1010 | 0 | 0 | 0 | 0 | 0 |
| 0111 | 1101 | 1 | 0 | 1 | 0 | 1 |
| 1000 | 0011 | 1 | 1 | 1 | 1 | 0 |
| 1001 | 0110 | 0 | 1 | 0 | 0 | 1 |
| 1010 | 1111 | 1 | 0 | 1 | 0 | 1 |
| 1011 | 0010 | 0 | 1 | 0 | 1 | 0 |
| 1100 | 0001 | 1 | 1 | 1 | 1 | 0 |
| 1101 | 0100 | 1 | 1 | 1 | 1 | 1 |
| 1110 | 1101 | 1 | 0 | 1 | 0 | 1 |
| 1111 | 0000 | 1 | 1 | 1 | 1 | 0 |

$z * f_4 = 0$
 なので
 更新せず

まとめと今後の予定

■ まとめ

- BDDを用いた最適領域抽出手法「に向けて」
- 領域の集合 \Rightarrow ピクセルの集合の集合 \Rightarrow BDD
 - ・ 領域とゲインの対応が全て求まる.
 - ・ 考慮する領域の制約が自由に設定できる.

■ 今後の予定

- 実験・評価
- 最適な領域のみ求めることに特化して大規模問題に適用できる枠組みの検討
 - ・ 変数割当て？
 - ・ フロンティア法？