



# ランダムフォレストの 基礎と最新動向

2013年6月12日  
近畿大学工学部情報学科  
波部 齊 (Hitoshi HABE)

[habe@kindai.ac.jp](mailto:habe@kindai.ac.jp)

# Real-Time Human Pose Recognition for Kinect

---

[ Shotton et al., CVPR 2011 ]


CVPR 2011 Best Paper Winner



# Behind-the-scenes Story about Kinect

[http://www.eng.cam.ac.uk/news/stories/2011/Xbox\\_Kinect/](http://www.eng.cam.ac.uk/news/stories/2011/Xbox_Kinect/)

[search](#) [contact](#)

 **UNIVERSITY OF CAMBRIDGE**

**Department of Engineering**

[University of Cambridge](#) > [Engineering Department](#) > [News & Features](#)


## Alumnus Dr Jamie Shotton and the development of Kinect for Xbox 360

*20 January 2011*


Dr Jamie Shotton completed his PhD in the Machine Intelligence Lab with Professor Cipolla, here at the Department of Engineering from 2003-2007. Jamie now works for Microsoft at their Cambridge research laboratory, where he has been intimately involved in the development of Kinect for Xbox 360. Kinect makes you the controller, allowing you to jump in and play games using your whole body, without holding or wearing anything special. Jamie came back to the Department to lecture the 4th year undergraduate students on this in November 2010. Below, he tells his behind-the-scenes story about Kinect.

"I joined the Machine Learning & Perception group at Microsoft Research Cambridge (MSRC) in June 2008 as a post-doc to continue my PhD research in computer vision. In this, I had focused on automatic visual object recognition: teaching computers how to recognise different types of object in photographs such as cars, sheep, trees, etc. Little did I know at that point how quickly I would get pulled into the frenzy of research and development around Kinect, and how this blue-skies research could be applied to such a practical problem.

"I had taken a machine learning approach to visual object recognition in photos, which works as follows. First, you build up a varied training set of images where you label each pixel with a colour, according to which object category it belongs to. So, for example, you hand label all 'cow' pixels in blue, and all 'tree' pixels in green, using a



Dr Jamie Shotton



# ランダムフォレストとは？

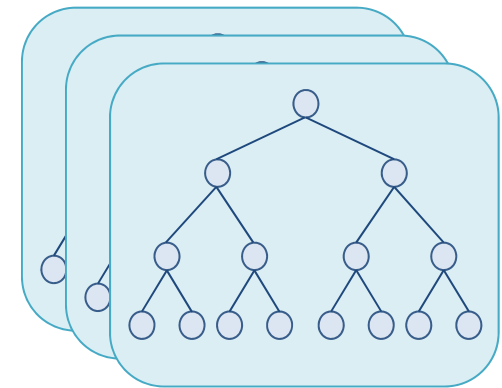
---

- アンサンブル学習

- 複数の決定木（弱識別器）の結果をあわせて識別・回帰・クラスタリングを行う

- バギング

- 入力学習データからのランダムサンプリングを繰り返して弱識別器を構成



ランダムフォレスト

- 高速・シンプル・わかりやすい・精度も良い
- 大量データを用いた学習に適している

# 歴史

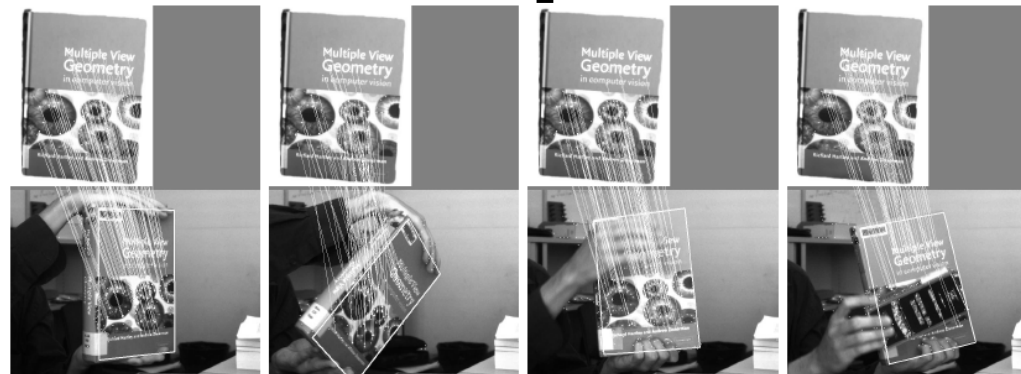
---

- Ho, T. K. 1995
  - Random decision forests, ICDAR 1995
  - ランダムに抽出した部分空間を使って学習した決定木を統合
- Amit & Geman 1997
  - Shape Quantization and Recognition with Randomized Trees, Neural Computation, 1997
- Leo Breiman 2001
  - Random Forests , Machine Learning, 2001
  - バギングによって決定木を学習
  - 理論的考察
    - Randomized Decision Trees
    - Randomized Decision Forests など様々な名前では呼ばれている

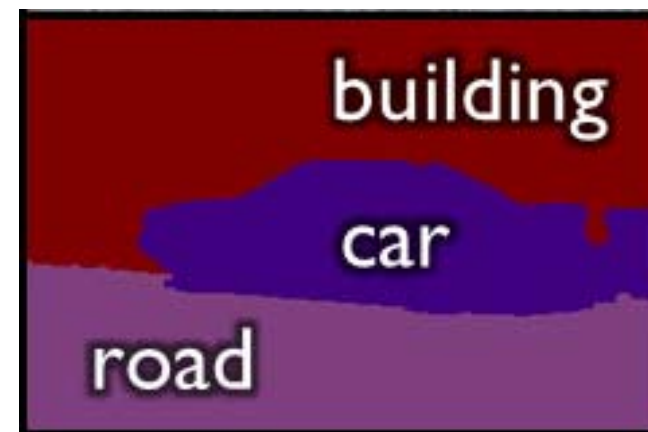
# コンピュータビジョンへの応用

---

- 特徴点認識 [Lepetit *et al.* 2005]



- 画像分割 & 認識 [Shotton *et al.* 2008]



# 入手可能な参考資料（1）

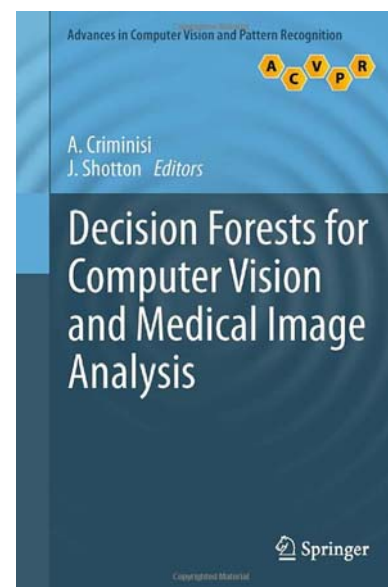
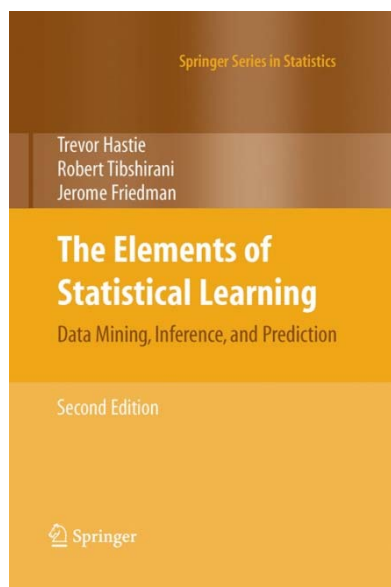
---

- ICCV 2009 tutorial by Jamie Shotton
  - [http://www.iis.ee.ic.ac.uk/~tkkim/iccv09\\_tutorial](http://www.iis.ee.ic.ac.uk/~tkkim/iccv09_tutorial)
- ICCV 2011 tutorial by Antonio Criminisi
  - <http://research.microsoft.com/en-us/projects/decisionforests/>
- 中部大学藤吉研究室
  - <http://www.vision.cs.chubu.ac.jp/RTs/>

# 入手可能な参考資料 (2)

---

- Elements of Statistical Learning (T. Hastie et al., 2009)
- Decision Forests for Computer Vision and Medical Image Analysis (Antonio Criminisi et al., 2013)





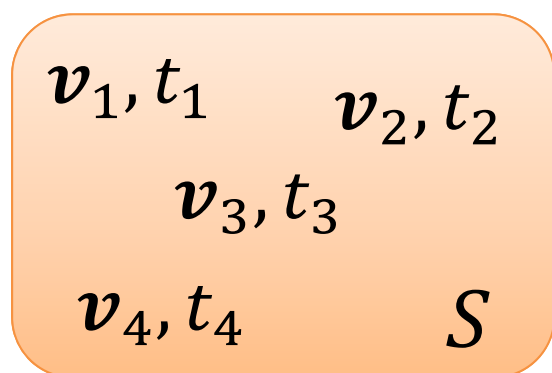
# アウトライン

---

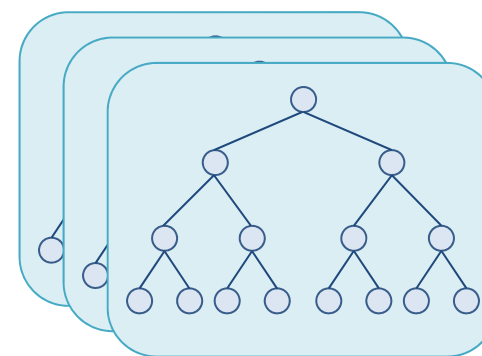
- ランダムフォレストの概要
  - 学習
  - テスト
  - 性質
- コンピュータビジョンへの応用
  - 利用法
  - パターン識別
  - 回帰
  - クラスタリング
- まとめ
- 応用例

# 準備

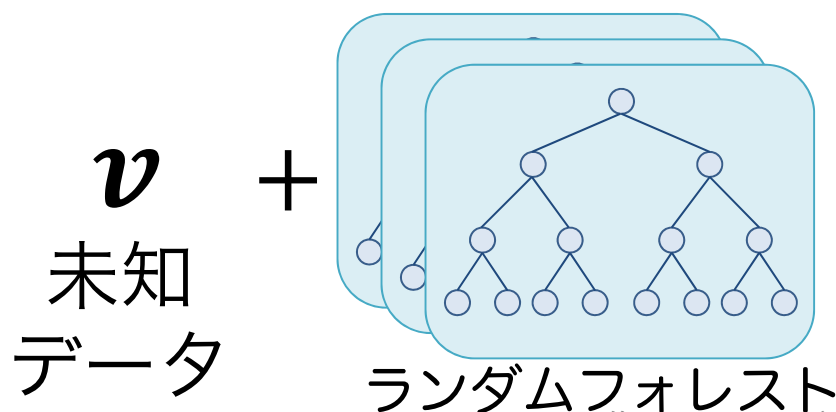
- 対象とするデータ  $\boldsymbol{v} = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d$
- 目標  $t$
- 教師データ集合  $S = \{(\boldsymbol{v}, t)\}$



学習



ランダムフォレスト

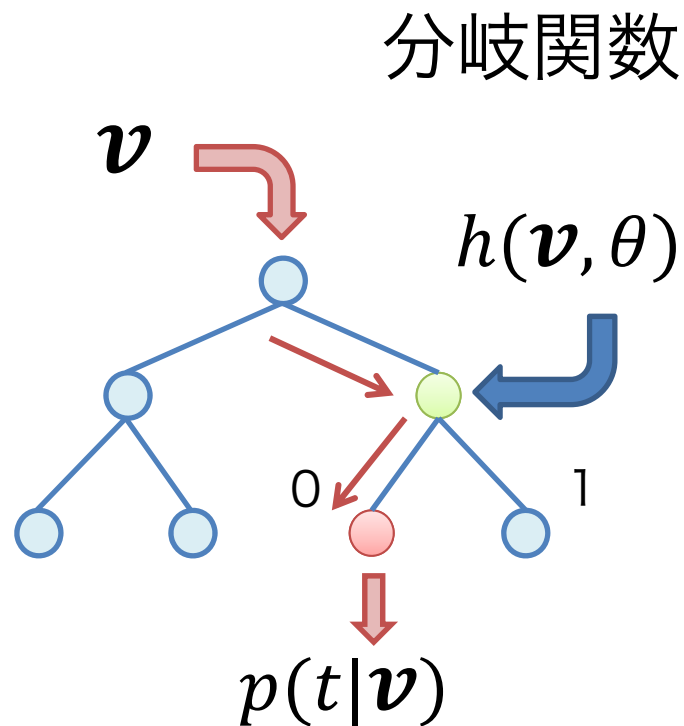


テスト

$t$

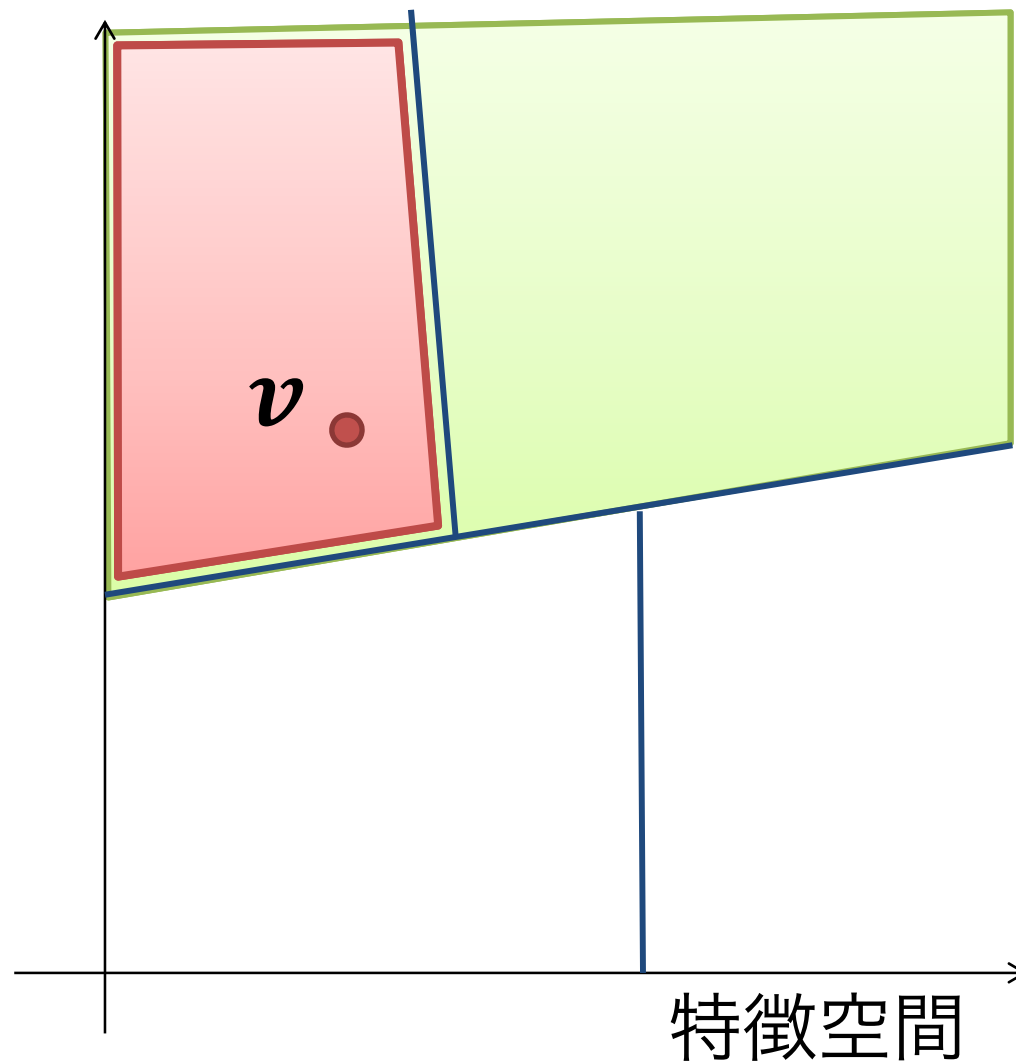
- クラス識別:  $c \in \mathbb{N}$
- 回帰:  $y \in \mathbb{R}^n$

# 決定木 (二分決定木)

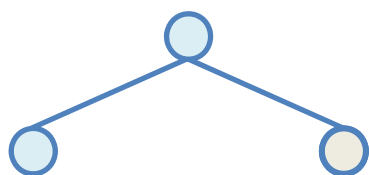


$$h(v, \theta) = \{0, 1\}$$

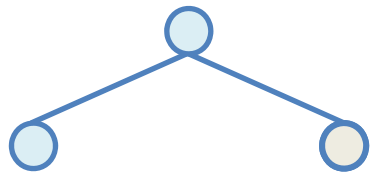
パラメータ



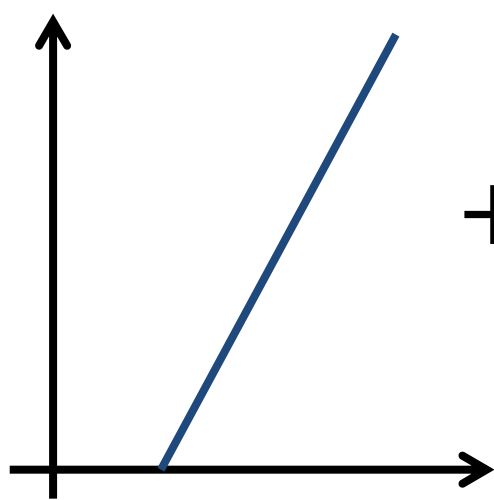
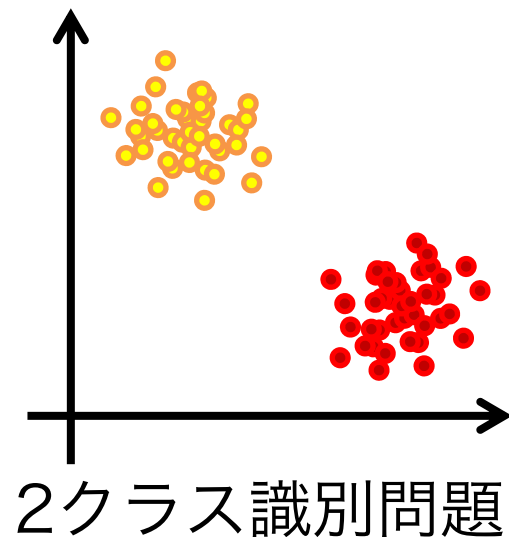
# 決定木判定結果の統合 (パターン識別)



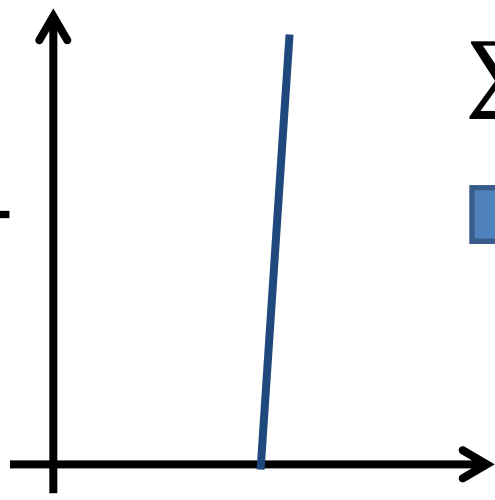
↓  
 $p_t(c|\mathbf{v})$



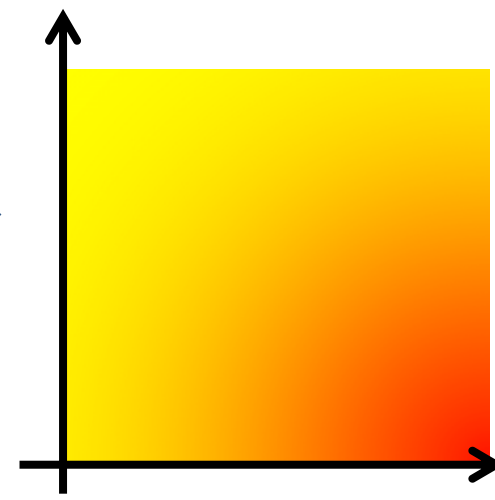
↓  
 $p_t(c|\mathbf{v})$



+



$\Sigma p_t(c|\mathbf{v})$



# アウトライン

---

- ランダムフォレストの概要
  - 学習
  - テスト
  - 性質
- コンピュータビジョンへの応用
  - 利用法
  - パターン識別
  - 回帰
  - クラスタリング
- まとめ
- 応用例

# 学習アルゴリズムの流れ

並列化可能！！

- 入力：教師データ集合  $S = \{v, t\}$
- $T$ 個の決定木  $T_1, T_2, \dots, T_T$  それぞれについて
  - 教師データ集合からサンプリング： $S \Rightarrow S_0$
  - サンプリングしたデータを根ノード ( $j = 0$ ) に与え以下を再帰的に繰り返す
    - ノード  $j$  での最適な分岐関数  $h^*(v, \theta^*)$  を求める
    - $h^*$ に従って  $S_j$  を分割して  $S_j^L$  と  $S_j^R$  を得る
    - 終了条件をみたす？ No  $\rightarrow S_j^L$  と  $S_j^R$  を子ノードに与えて学習継続, Yes  $\rightarrow$  それより深くはしない
  - すべての葉ノードが終了条件をみたす  $\rightarrow$  木の学習終了
- すべての木の学習終了  $\rightarrow$  Forestの学習終了

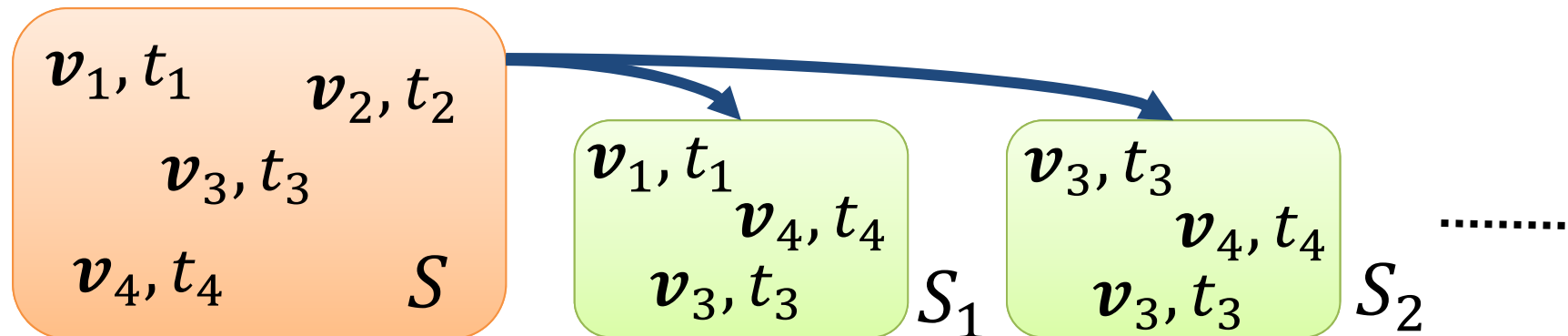
# 学習アルゴリズムの流れ

---

- 入力：教師データ集合  $S = \{v, t\}$
- $T$ 個の決定木  $T_1, T_2, \dots, T_T$  それぞれについて
  - 教師データ集合からサンプリング： $S \Rightarrow S_0$
  - サンプリングしたデータを根ノード ( $j = 0$ ) に与え以下を再帰的に繰り返す
    - ノード  $j$  での最適な分岐関数  $h^*(v, \theta^*)$  を求める
    - $h^*$ に従って  $S_j$  を分割して  $S_j^L$  と  $S_j^R$  を得る
    - 終了条件をみたす？ No  $\rightarrow S_j^L$  と  $S_j^R$  を子ノードに与えて学習継続, Yes  $\rightarrow$  それより深くはしない
  - すべての葉ノードが終了条件をみたす  $\rightarrow$  木の学習終了
- すべての木の学習終了  $\rightarrow$  Forestの学習終了

# 教師データ集合からのサンプリング

- 教師データ集合全体から個々の決定木の学習に利用する教師データをランダムに選ぶ
  - 重複を許す (w/ replacement)
  - 使われないデータがあっても良い



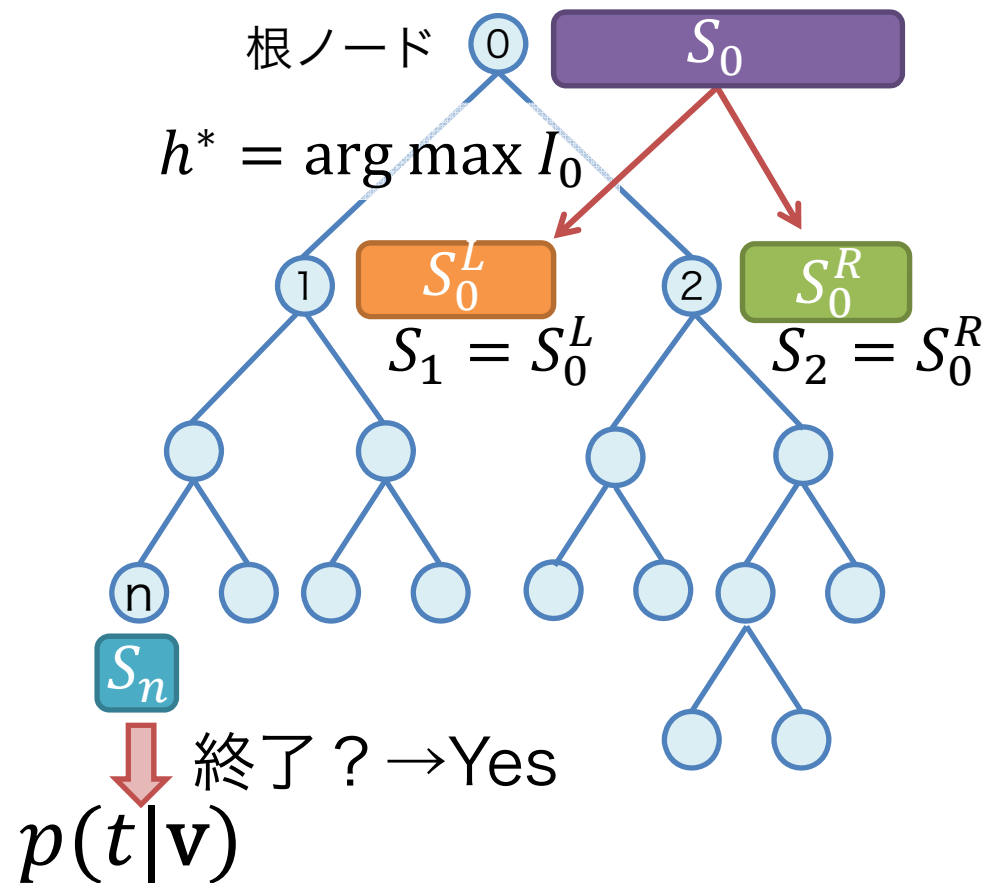
- ブートストラップサンプル
- バギング (bootstrap aggregating)



# 学習アルゴリズムの流れ

- 入力：教師データ集合  $S = \{v, t\}$
- $T$ 個の決定木  $T_1, T_2, \dots, T_T$  それぞれについて
  - 教師データ集合からサンプリング： $S \Rightarrow S_0$
  - サンプリングしたデータを根ノード ( $j = 0$ ) に与え以下を再帰的に繰り返す
    - ノード  $j$  での最適な分岐関数  $h^*(v, \theta^*)$  を求める
    - $h^*$ に従って  $S_j$  を分割した  $S_j^L$  と  $S_j^R$  を与えて子ノードを生成
    - 終了条件をみたす？ No→学習継続, Yes→それより深くはしない
  - すべての葉ノードが終了条件をみたす→木の学習終了
- すべての木の学習終了→Forestの学習終了

# 決定木学習の流れ



1. 根ノードから学習開始
2. 再帰的に以下を実行

A) ノード  $j$  での最適な分岐関数  $h^*(v, \theta^*)$  を求める

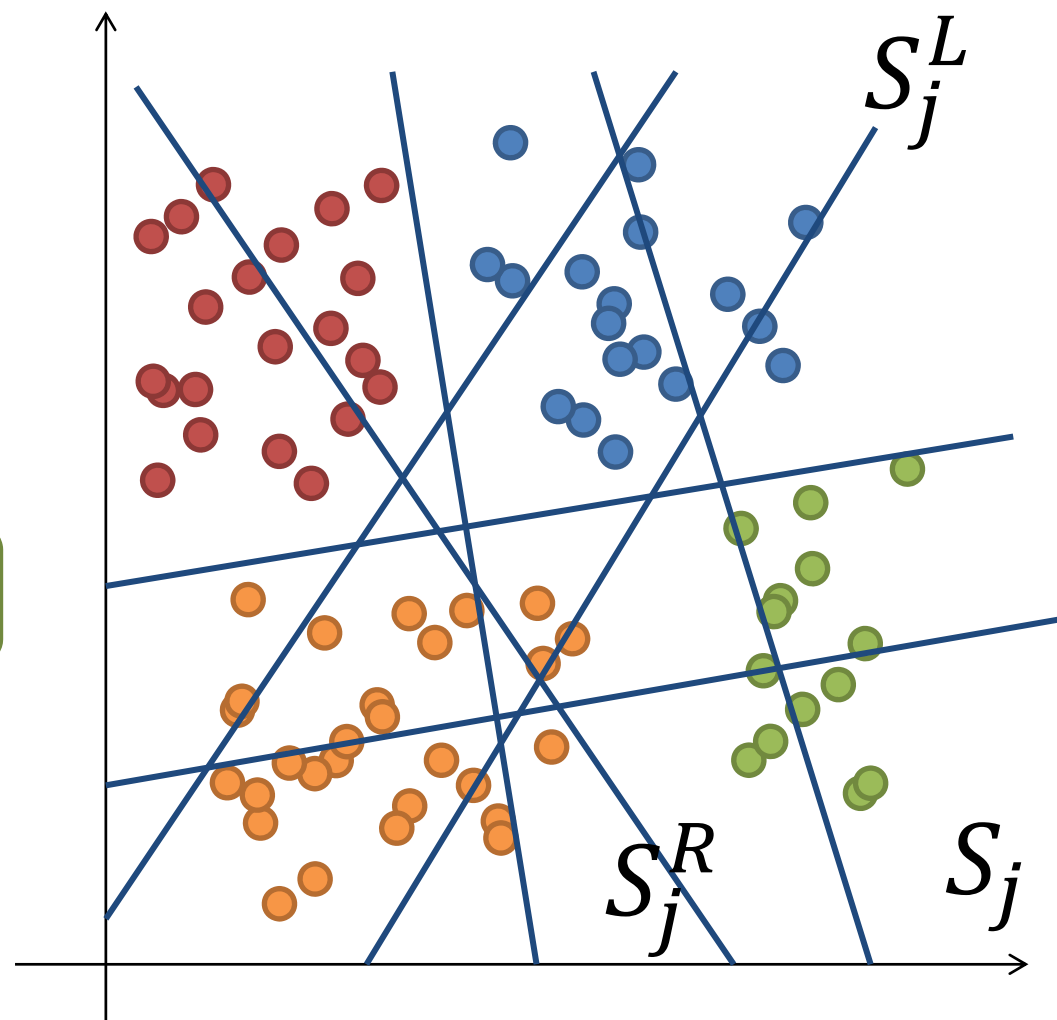
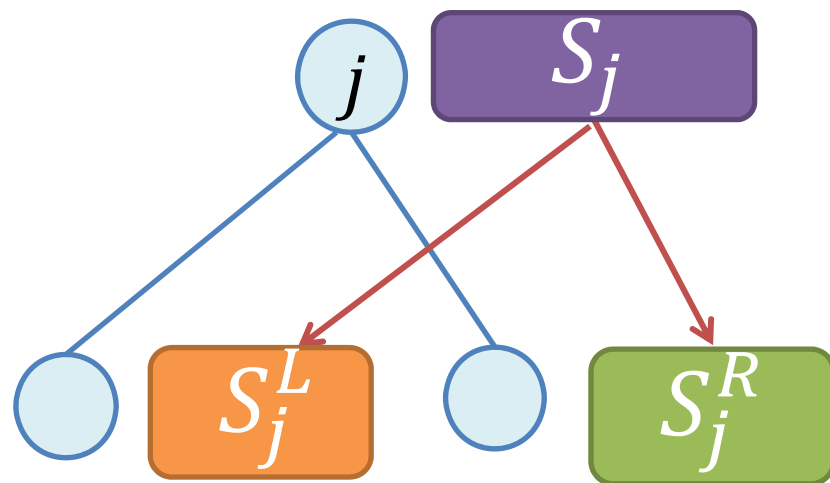
B)  $h^*$  に従って  $S_j$  を分割した  $S_j^L$  と  $S_j^R$  を与えて子ノードを生成

C) 終了条件をみたす?  
No  $\rightarrow$  子ノードで学習継続, Yes  $\rightarrow$  終了

3. 全葉ノードで終了  $\rightarrow$  終了

# 分岐関数の学習(1)：イメージ

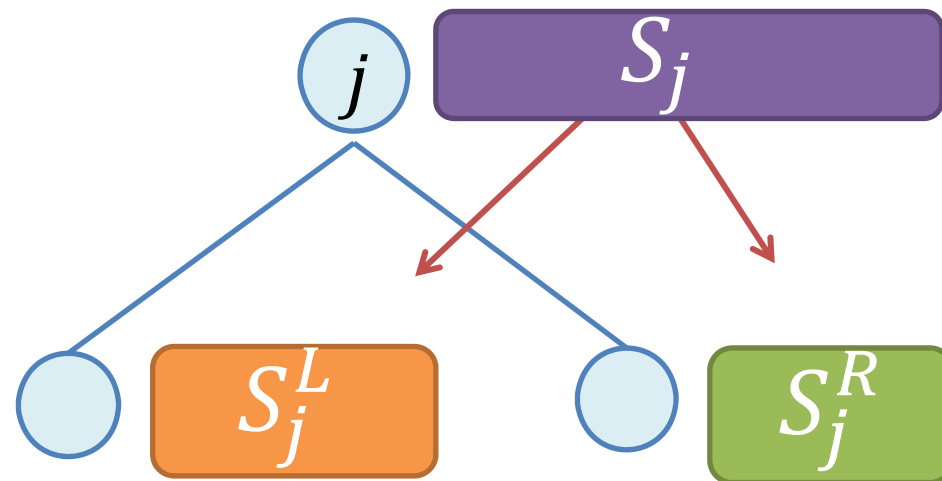
[藤吉先生のチュートリアル資料を参考]



# 分岐関数の学習(2) : 定式化

$$h(\mathbf{v}, \theta_j) = \{0, 1\}$$

$\mathbf{v}$  : 入力データ  
 $\theta_j$  : パラメータ



最適なパラメータ

$$\theta^* = \arg \max_{\theta_j \in \mathfrak{S}_j} I_j$$

目的関数 :  $I_j = I(S_j, S_j^L, S_j^R, \theta_j)$

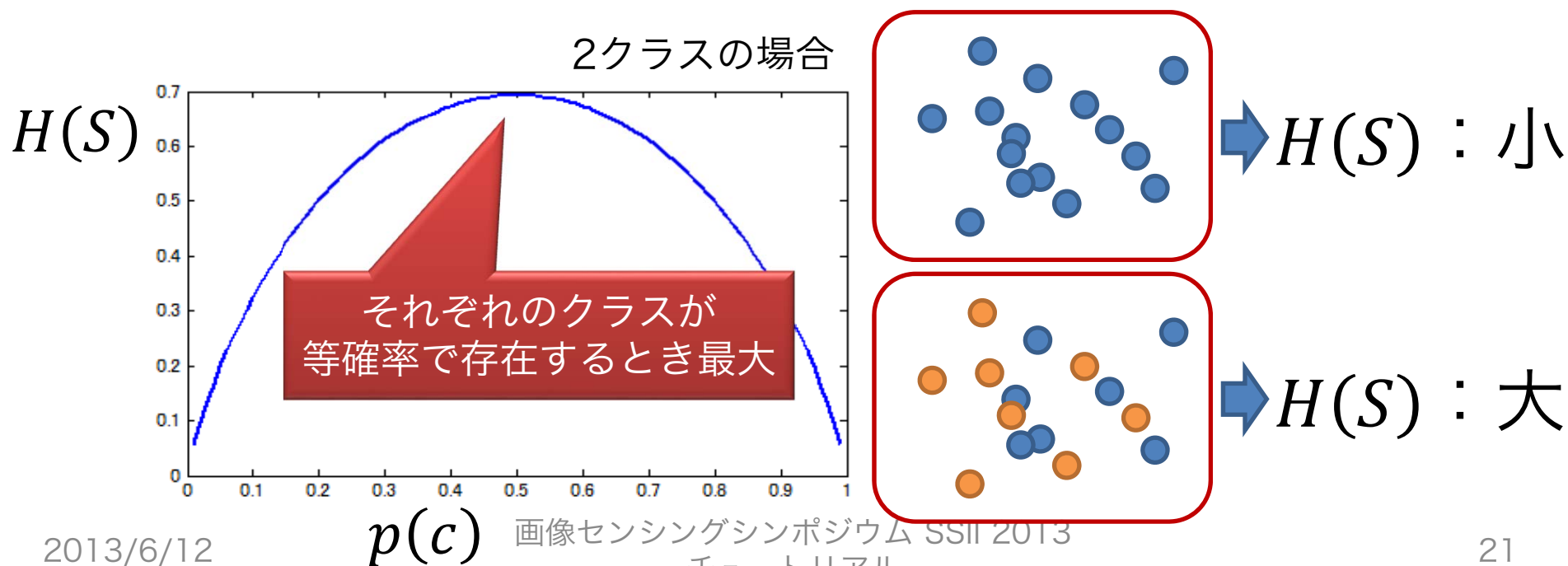
パラメータ候補 :  $\mathfrak{S}_j \in \mathfrak{S}$

# 分岐関数の学習(3) : 目的関数(1)

パターン識別の場合 (クラスラベル集合 $C$ )

- エントロピー (平均情報量)

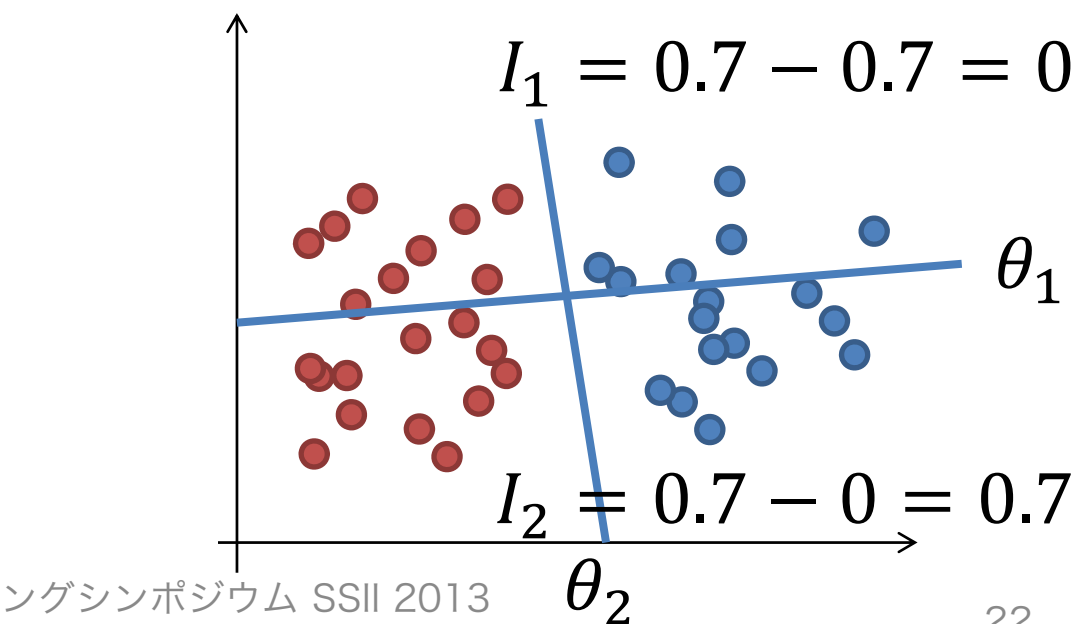
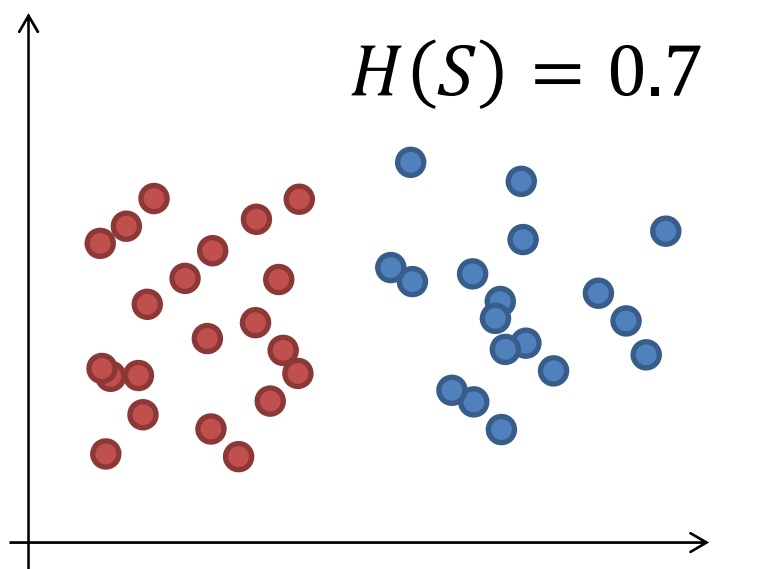
$$H(S) = - \sum_{c \in C} p(c) \log(p(c))$$



# 分岐関数の学習(3) : 目的関数(2)

- 情報利得 : 分割前後のエントロピーの差

$$I = \underbrace{H(S)}_{\text{分割前}} - \underbrace{\sum_{i \in \{L,R\}} \frac{|S^i|}{|S|} H(S^i)}_{\text{分割後 (重み付き平均)}}$$



# 分岐関数の学習(4)：パラメータ候補

---

- すべてのパラメータ候補を試すのは非現実的



- ランダムに抽出したものを利用

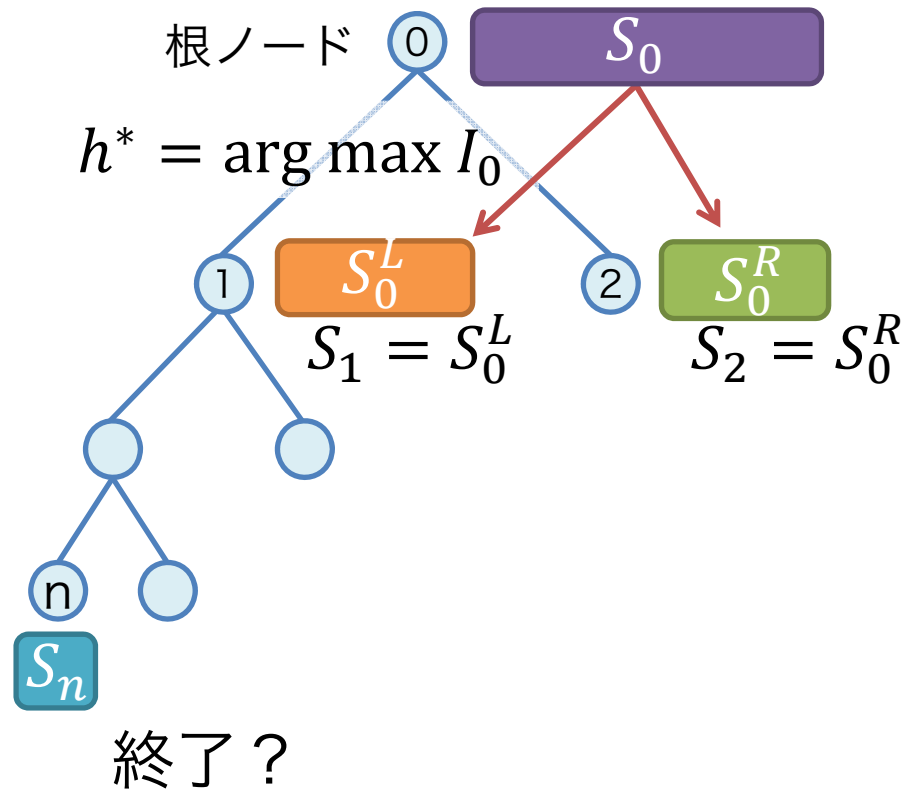
ノード $j$ の学習に  
利用する  
パラメータ

$$\tilde{\mathfrak{J}}_j \in \tilde{\mathfrak{J}}$$

全パラメータ

$$\text{ランダム度合い} : \rho = \frac{|\tilde{\mathfrak{J}}_j|}{|\tilde{\mathfrak{J}}|}$$

# 決定木学習の流れ



1. 根ノードから学習開始
2. 再帰的に以下を実行

A) ノード  $j$  での最適な分岐関数  $h^*(v, \theta^*)$  を求める

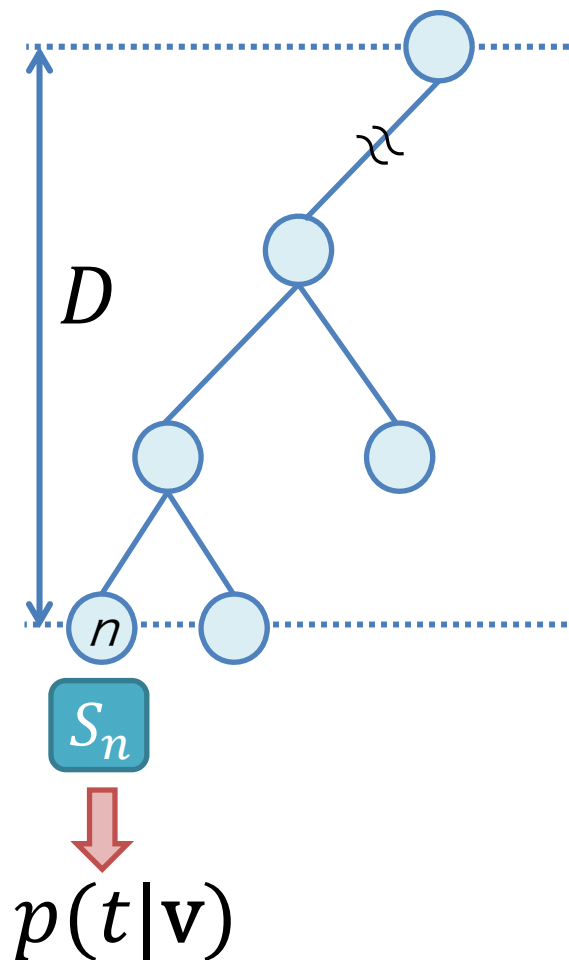
B)  $h^*$  に従って  $S_j$  を分割した  $S_j^L$  と  $S_j^R$  を与えて子ノードを生成

C) 終了条件をみたます?  
No → 子ノードで学習継続, Yes → 終了

3. 全葉ノードで終了 → 終了



# 決定木学習の終了



## 終了条件

1. あらかじめ定めた深さ  $D$  に達した場合
2. ノードに割り当てられた学習データの個数  $|S_n|$  が一定値以下になったとき
3. 分割による情報利得  $I_n$  が一定値以下になったとき

終了すると  $S_n$  から事後確率  $p(t|v)$  の推定値を計算して保持

# 学習アルゴリズムの流れ

---

- 入力：教師データ集合  $S = \{v, t\}$
- $T$ 個の決定木  $T_1, T_2, \dots, T_T$  それぞれについて
  - 教師データ集合からサンプリング： $S \Rightarrow S_0$
  - サンプリングしたデータを根ノード ( $j = 0$ ) に与え以下を再帰的に繰り返す
    - ノード  $j$  での最適な分岐関数  $h^*(v, \theta^*)$  を求める
    - $h^*$ に従って  $S_j$  を分割して  $S_j^L$  と  $S_j^R$  を得る
    - 終了条件をみたす？ No  $\rightarrow S_j^L$  と  $S_j^R$  を子ノードに与えて学習継続, Yes  $\rightarrow$  それより深くはしない
  - すべての葉ノードが終了条件をみたす  $\rightarrow$  木の学習終了
- すべての木の学習終了  $\rightarrow$  Forestの学習終了

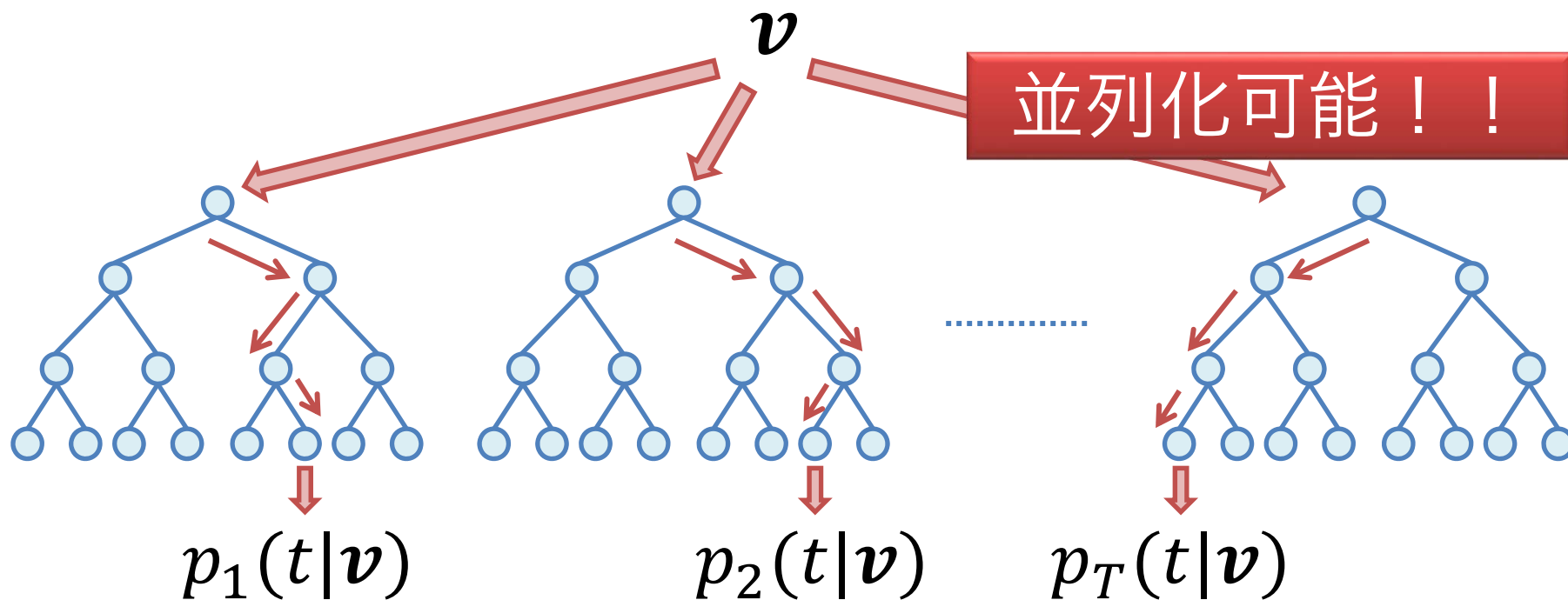
# アウトライン

---

- ランダムフォレストの概要
  - 学習
  - テスト
  - 性質
- コンピュータビジョンへの応用
  - 利用法
  - パターン識別
  - 回帰
  - クラスタリング
- まとめ
- 応用例

# テスト

- 与えられた入力を全ての決定木に入力して統合



$$p(t|\mathbf{v}) = \frac{1}{T} \sum_{t=1}^T p_t(t|\mathbf{v})$$

- 多数決
- 積をとる場合もある

# ランダムフォレストの性質(1)

---

- 目標：バイアスと分散の低減
- 決定木を使った識別
  - 決定木をある程度深くするとバイアスは低くできる
  - 分散は大きくなる
- 複数の決定木の組み合わせ
  - 木の間の変動が小さければ平均をとることで分散を低減できる←ランダム性

cf. ブースティング

# ランダムフォレストの性質(2) [Breiman 2001]

---

- 汎化誤差の収束性
  - 木の個数  $T$  を増大させると汎化誤差は収束する
- 汎化誤差の上限值

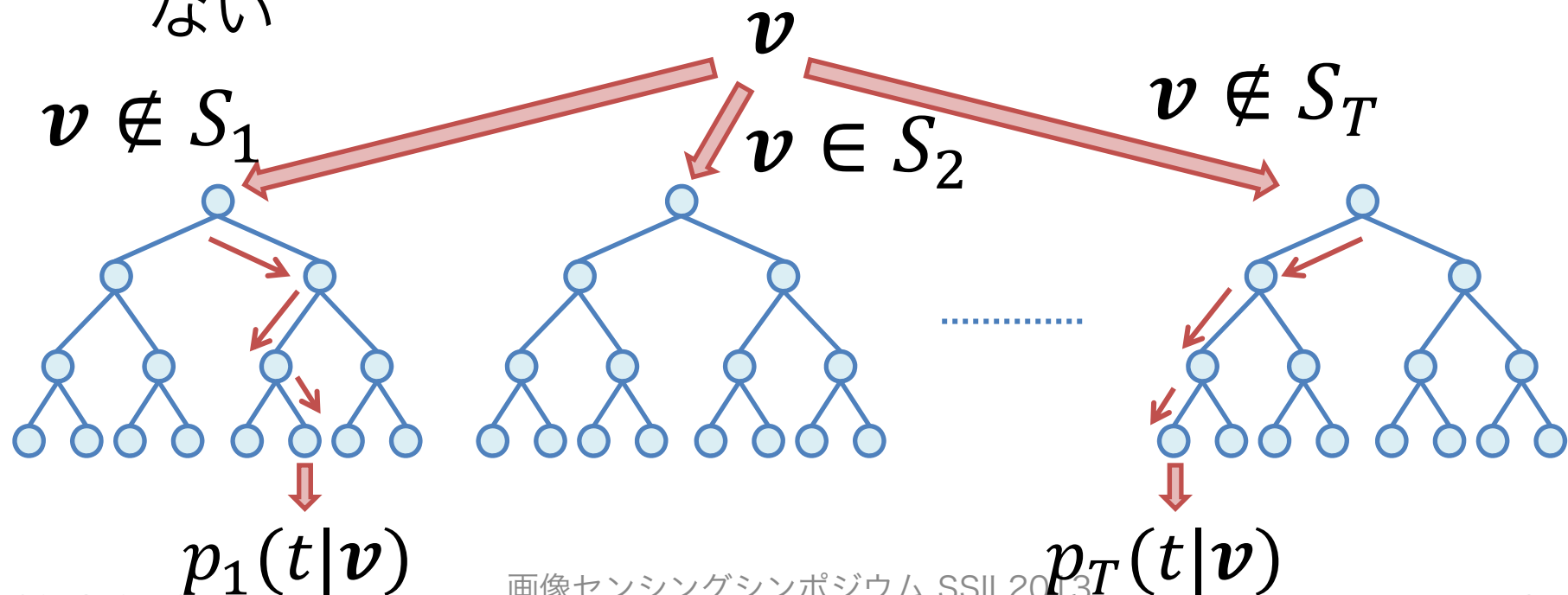
$$PE^* \leq \frac{\bar{\rho}(1 - s^2)}{s^2}$$

- $\bar{\rho}$  : 木同士の相関
- $s$  : 木の「強さ」 (マージンの期待値)

ランダム性によって相関を小さくする

# ランダムフォレストの性質(3) [Breiman 2001]

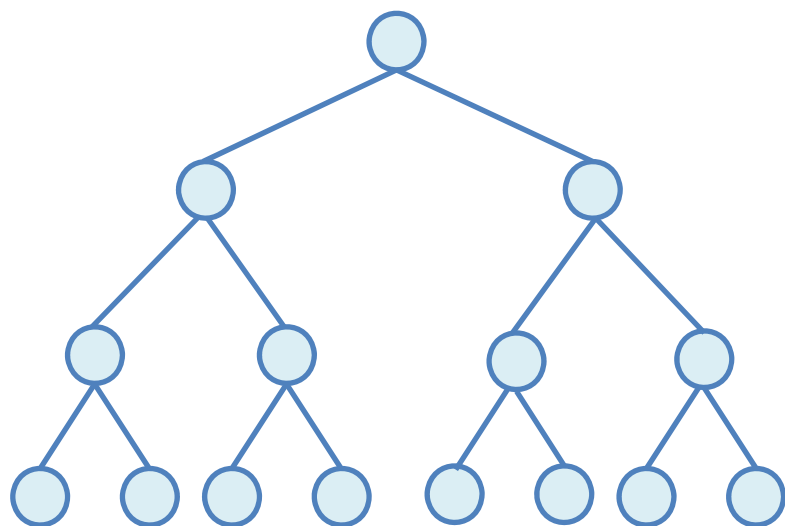
- Out-of-Bag Estimates
  - ブートストラップサンプルに含まれなかったサンプルを使って汎化誤差を推定
  - 交差検定 (cross-validation) を実行する必要はない



# ランダムフォレストの性質(4)

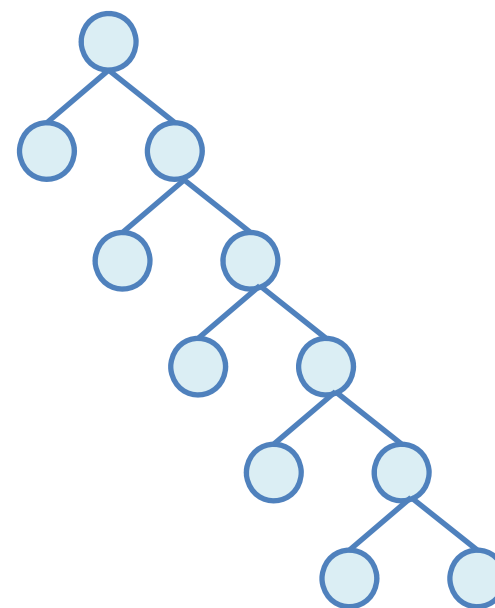
---

- RFとClassifier Cascade [Viola & Jones 2001]



ランダムフォレスト

複数の事例・仮説を扱う構造



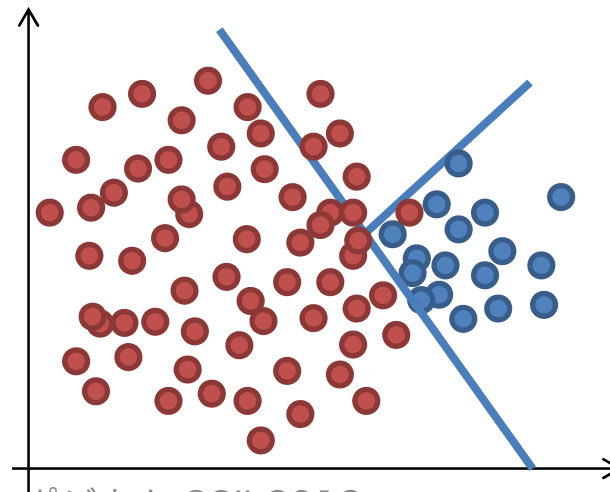
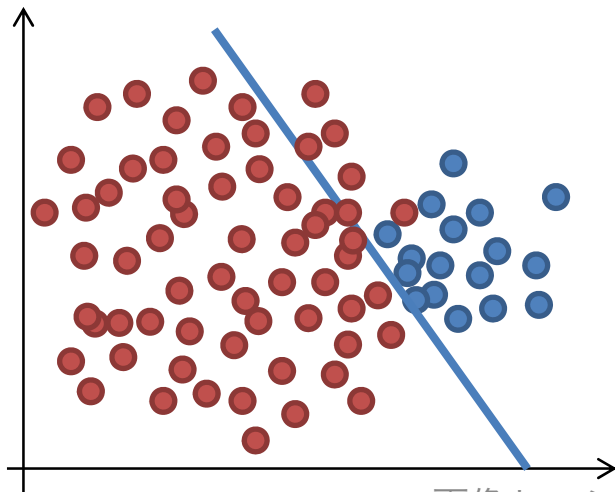
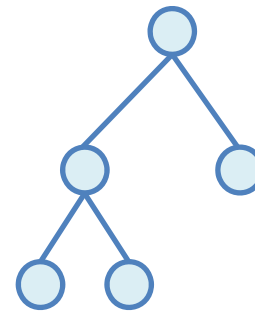
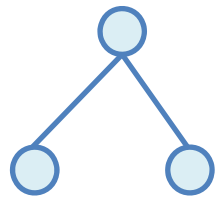
Classifier Cascade

「1 vs. 他」の判定を行う構造

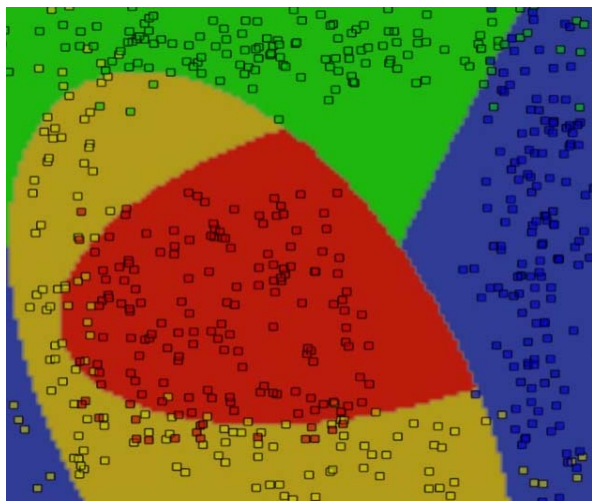
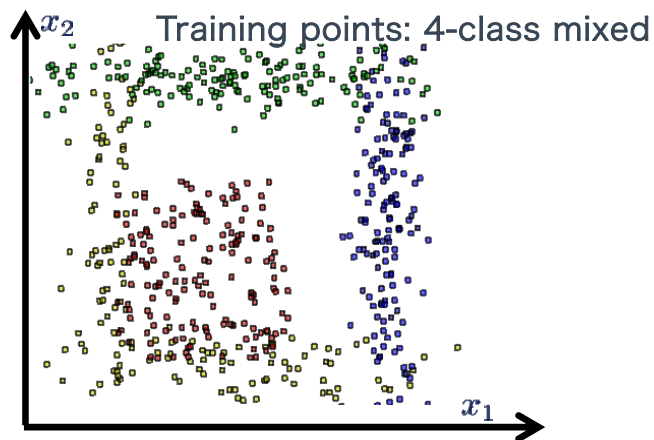


# 木の深さと過学習

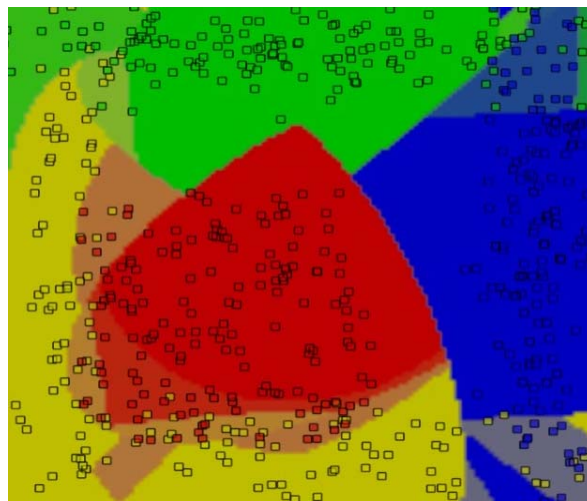
- 木の深さ  $D$ 
  - あまり深くしすぎると過学習を起こす



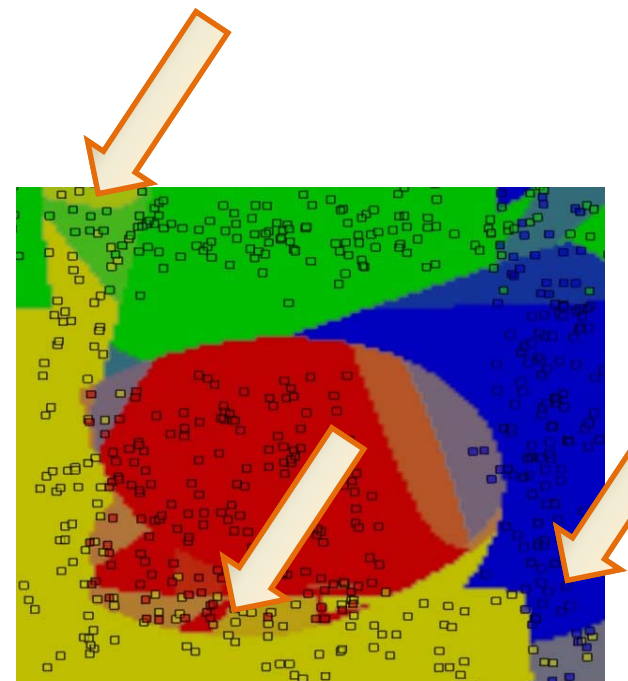
# 木の深さと過学習：実例 [Criminisi 2011]



T=200, D=3, w. l. = conic



T=200, D=6, w. l. = conic



T=200, D=15, w. l. = conic

max tree depth, D

underfitting

2013/6/12

画像センシングシンポジウム SSII 2013  
チュートリアル

overfitting

34

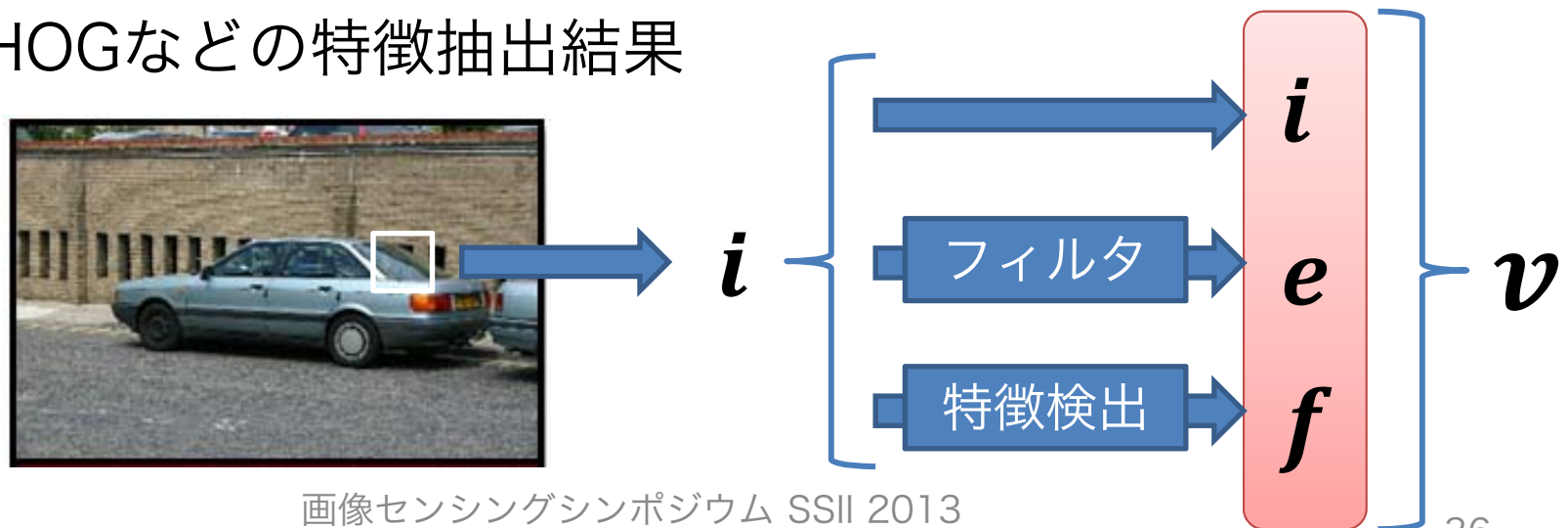
# アウトライン

---

- ランダムフォレストの概要
  - 学習
  - テスト
  - 性質
- コンピュータビジョンへの応用
  - 利用法
  - パターン識別
  - 回帰
  - クラスタリング
- まとめ
- 応用例

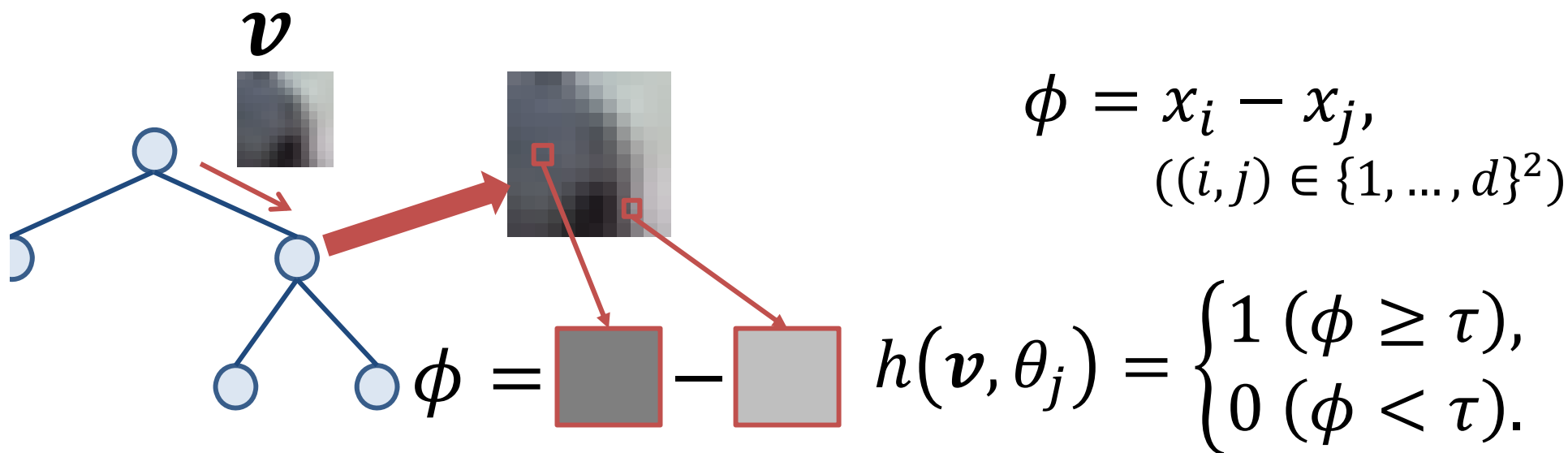
# CV問題への適用：入力データ

- 入力データ  $v = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d$ 
  - 画像／デプスマップ／3次元ボリュームの局所領域を単位とする
  - 局所領域における以下の特徴量で  $v$  を構成（例）
    - 輝度／色成分そのまま
    - フィルタリング結果（微分など）
    - HOGなどの特徴抽出結果



# CV問題への適用：分岐関数

- 分岐関数の定義  $h(\mathbf{v}, \theta_j) = \{0, 1\}$ 
  - データ  $\mathbf{v}$  の2要素の差分をとって閾値処理
  - 一様な明度変化などに対してロバスト  
( cf. Haar-like 特徴)



# リアルタイム特徴点認識

[Lepetit et al., 2006]

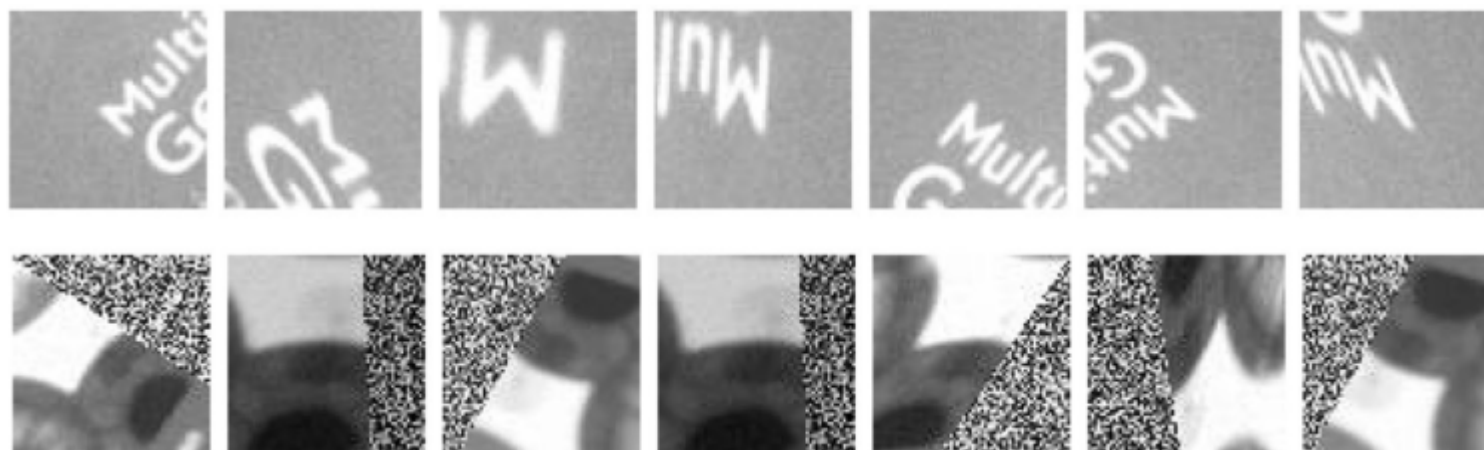
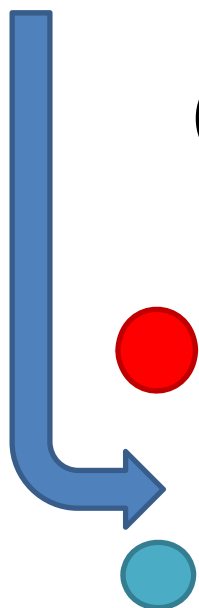
- 姿勢・照明変化などにロバストな特徴点認識を実現したい
- ランダムフォレストを利用
  - マルチクラス
  - 高速
- 学習：  
様々な変形を加えて大量の学習データを生成

# 学習データの生成

[Lepetit et al., 2006]

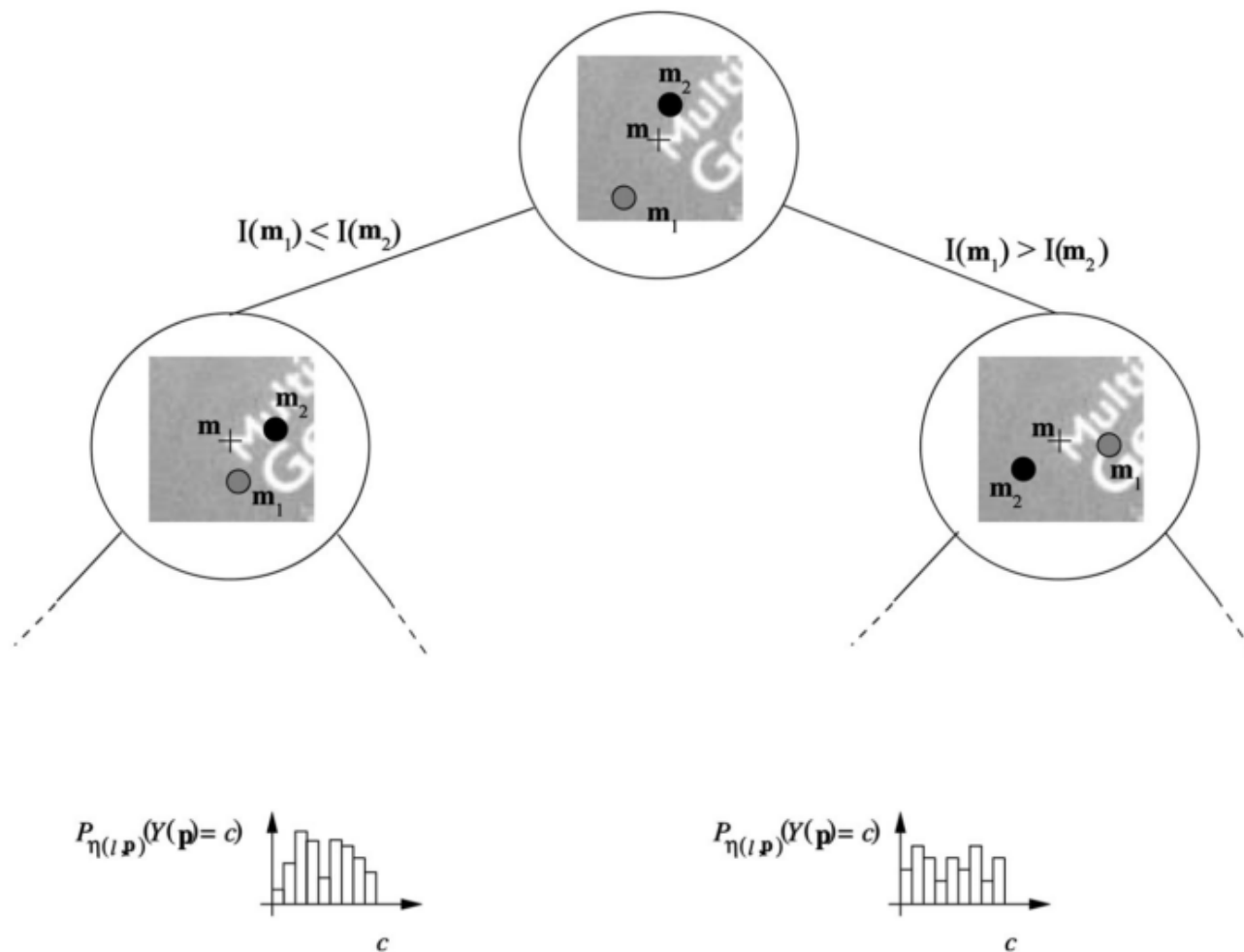


(アフィン変換 + ノイズ)



# テスト

[Lepetit et al., 2006]



局所勾配ヒストグラムを使った場合との比較も



# 特徴点認識の結果

[Lepetit et al., 2006]



→視点推定と特徴点認識を別のRFで実現 [西村 et al., MIRU 2010]

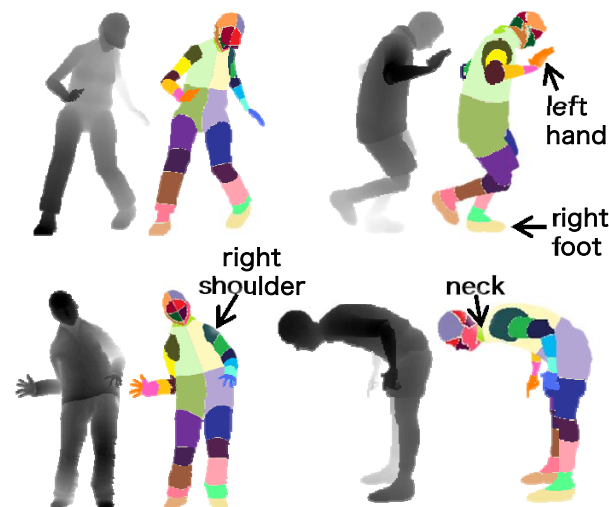
画像センシングシンポジウム SSII 2013

チュートリアル

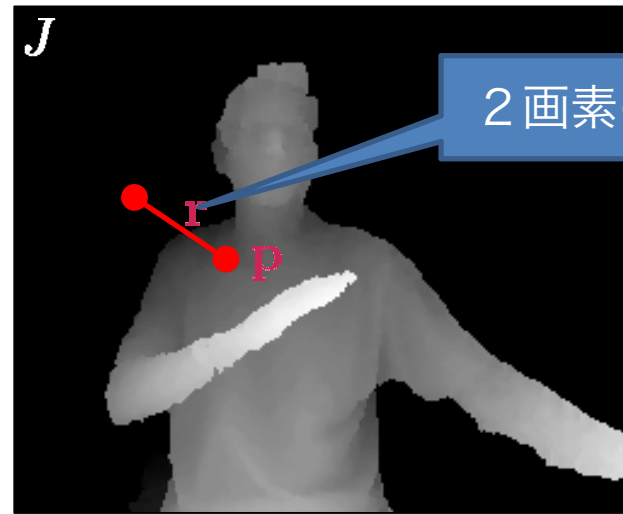
# デプスマップからの身体部位推定

[Sutton et al. 2011]

- 人間の身体を 31 個の部位に分割
- 入力：デプスマップ
- 出力：部位ラベル



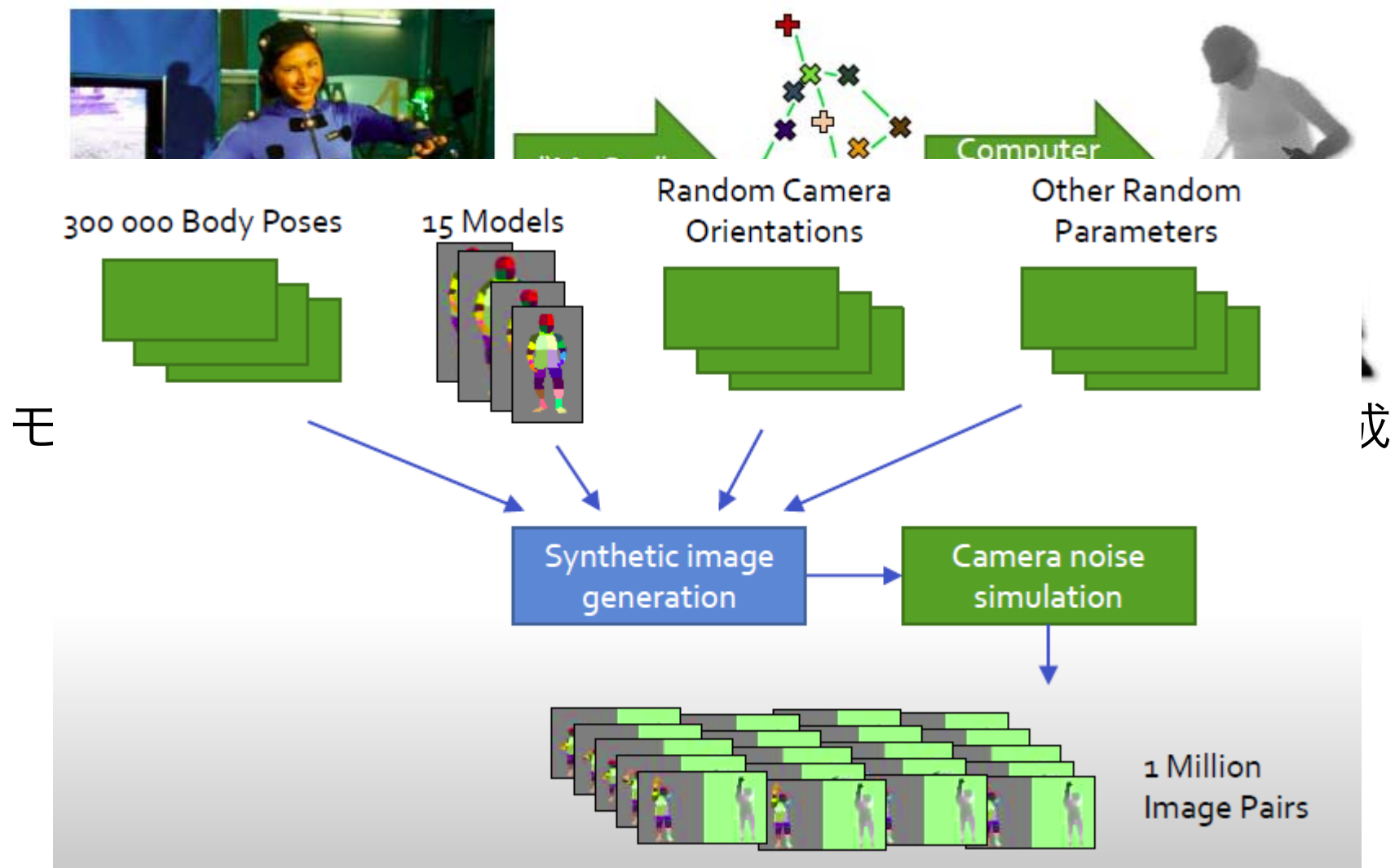
Input depth image



Visual features

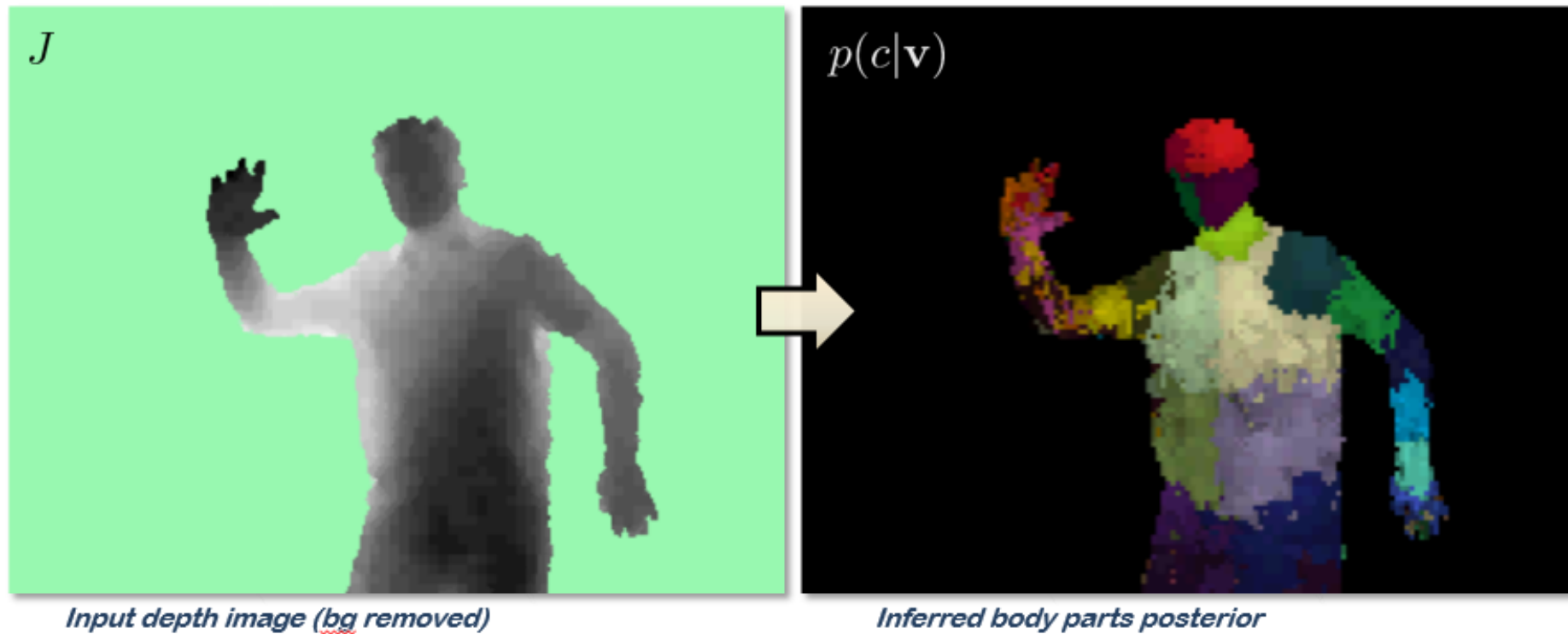
# 学習データの生成

[Shotton *et al.* 2011]



# 身体部位推定結果

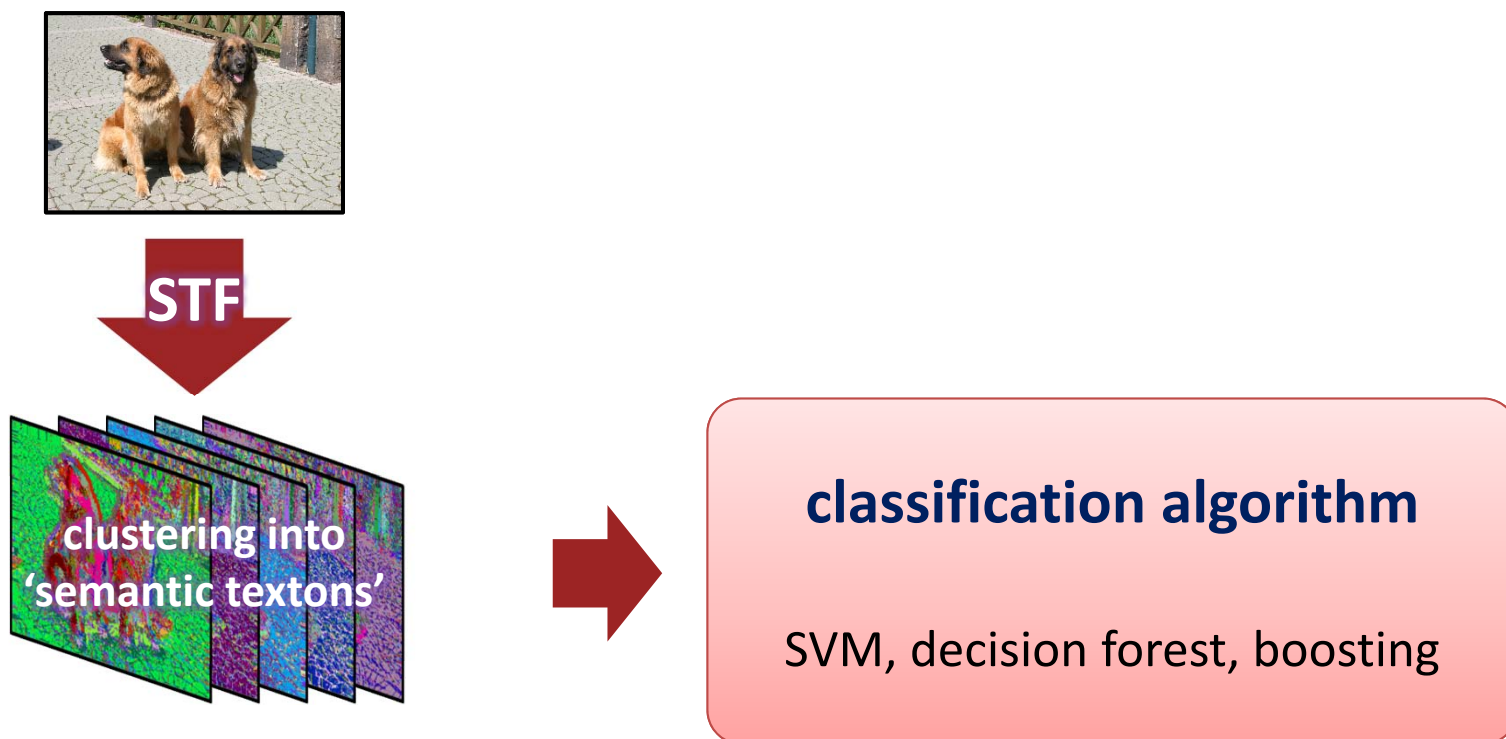
[Shotton *et al.* 2011]



# 画像分類とセグメンテーション

[Shotton *et al.* 2008]

- 局所領域からの特徴抽出にRFを利用  
(Semantic Texton Forests)

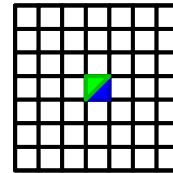


# Semantic Texton Forests

[Shotton *et al.* 2008]



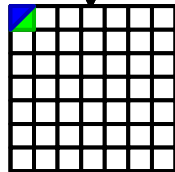
Input Image



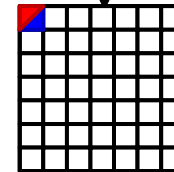
$$A[g] - B[b] > 28$$



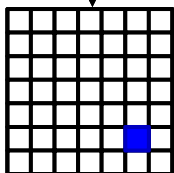
Ground Truth



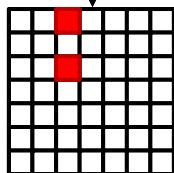
$$|A[b] - B[g]| > 37$$



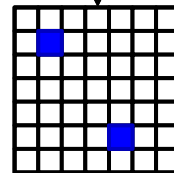
$$|A[r] - B[b]| > 21$$



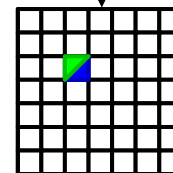
$$A[b] > 98$$



$$A[r] + B[r] > 363$$



$$A[b] + B[b] > 284$$



$$A[g] - B[b] > 13$$

$P(c|l)$



Example Patches

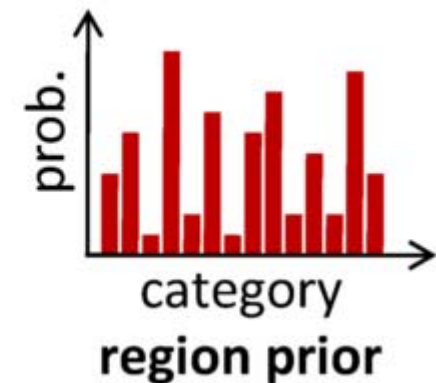
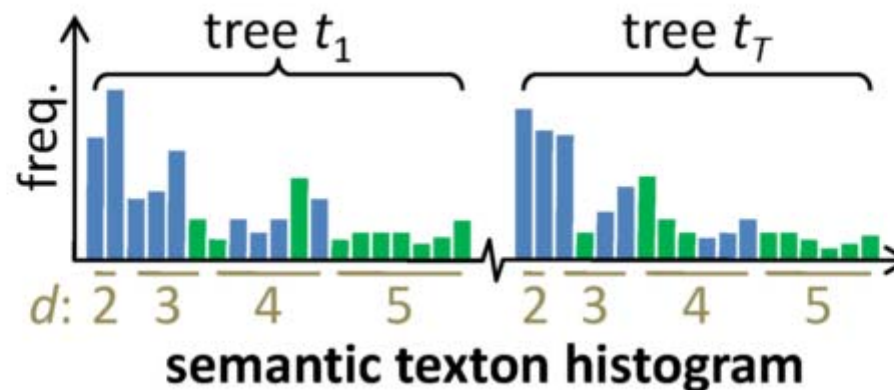
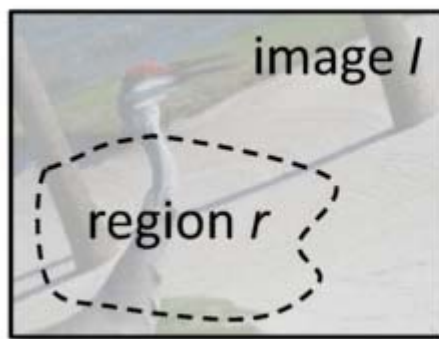
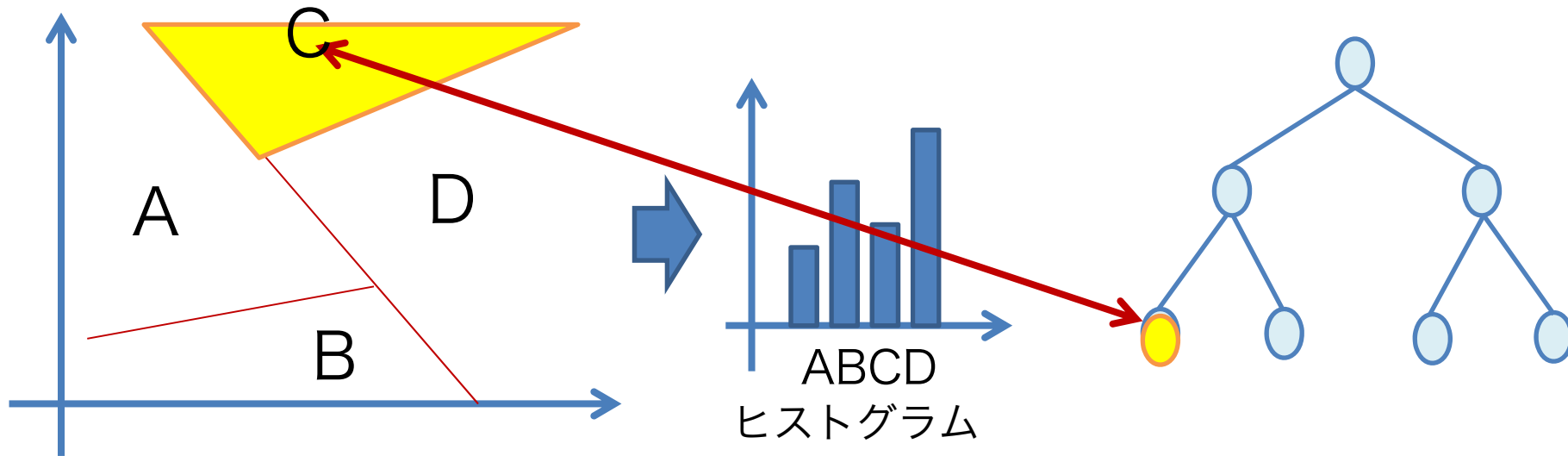


2013/6/12

# Bag of Semantic Textons

[Shotton *et al.* 2008]

- Bag of words



# セグメンテーション結果

[Shotton *et al.* 2008]

---

## Real-Time Semantic Segmentation

Jamie Shotton  
Matthew Johnson  
Roberto Cipolla



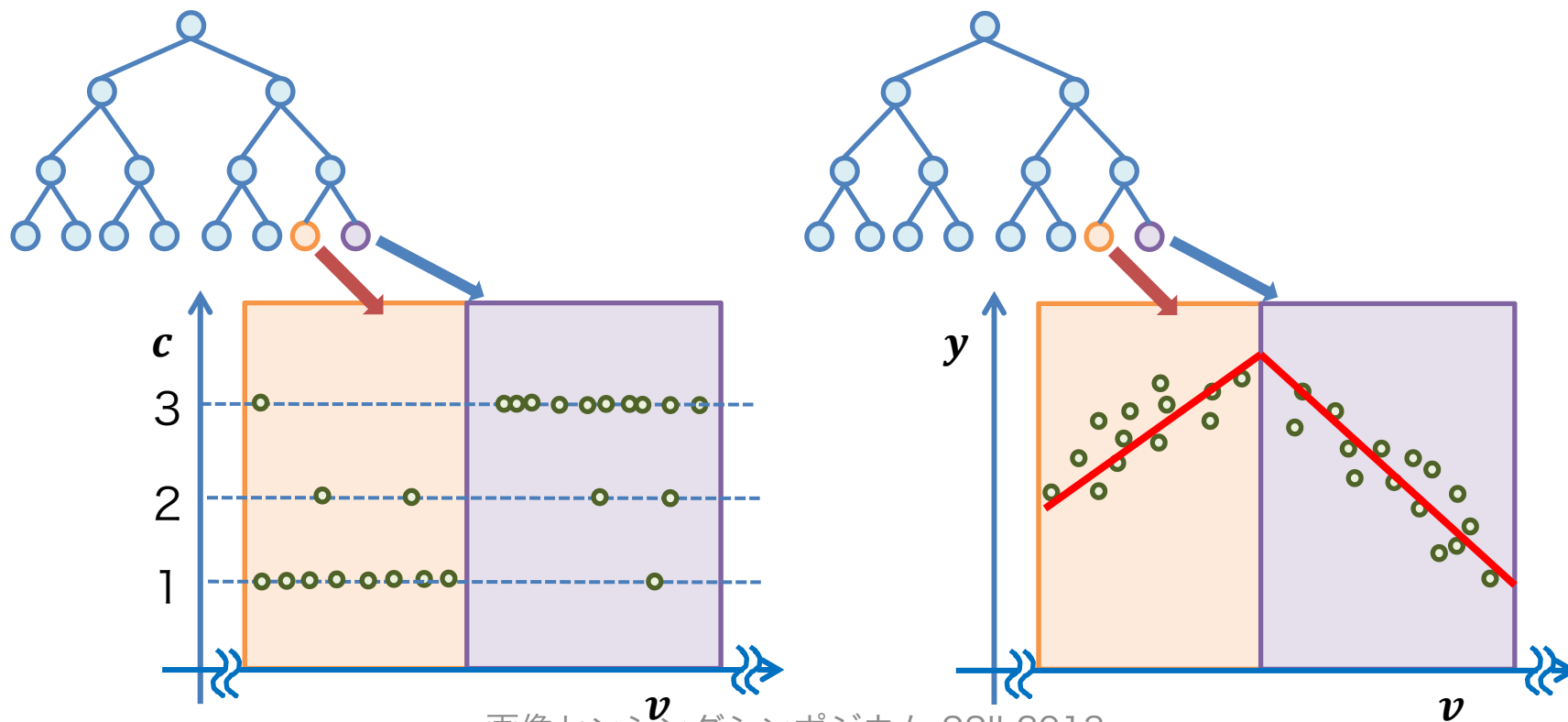
# アウトライン

---

- ランダムフォレストの概要
  - 学習
  - テスト
  - 性質
- コンピュータビジョンへの応用
  - 利用法
  - パターン識別
  - 回帰
  - クラスタリング
- まとめ
- 応用例

# ランダムフォレストによる回帰

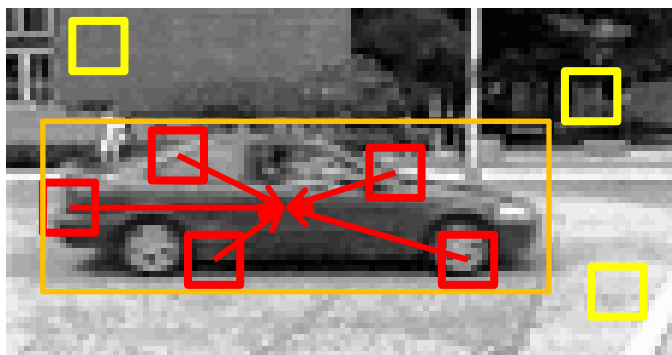
- 出力
  - パターン識別問題:  $c \in \mathbb{N}$
  - 回帰問題:  $y \in \mathbb{R}^n$



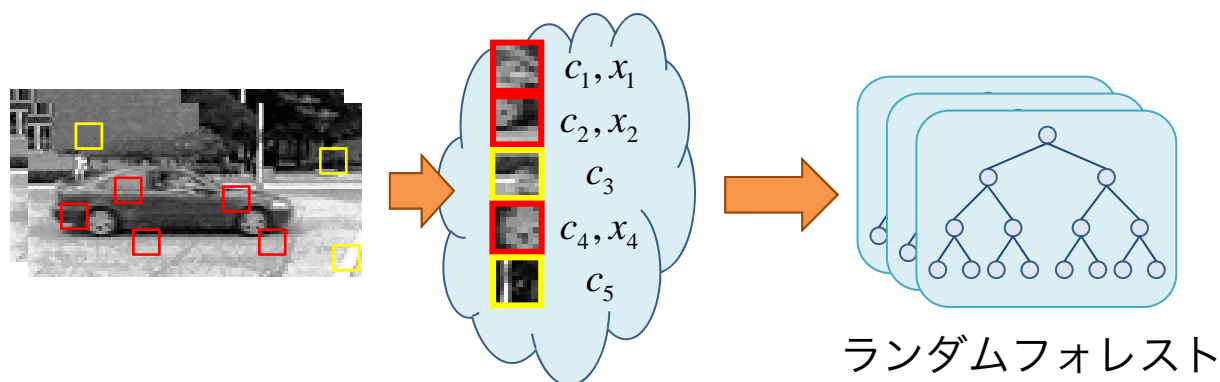
# Hough Forest

[Gall *et al.* 2009]  
[Okada 2009]

- 局所パッチ & 投票ベースの物体検出
  - Implicit Shape Model [Leibe *et al.* 2003]



- ランダムフォレストで識別・回帰を行う



入力：

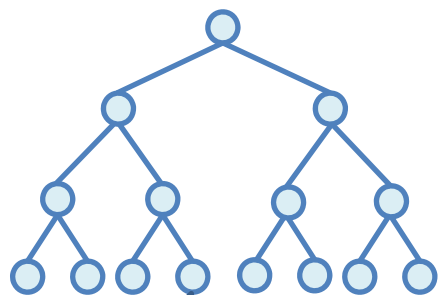
- 局所パッチ

出力：

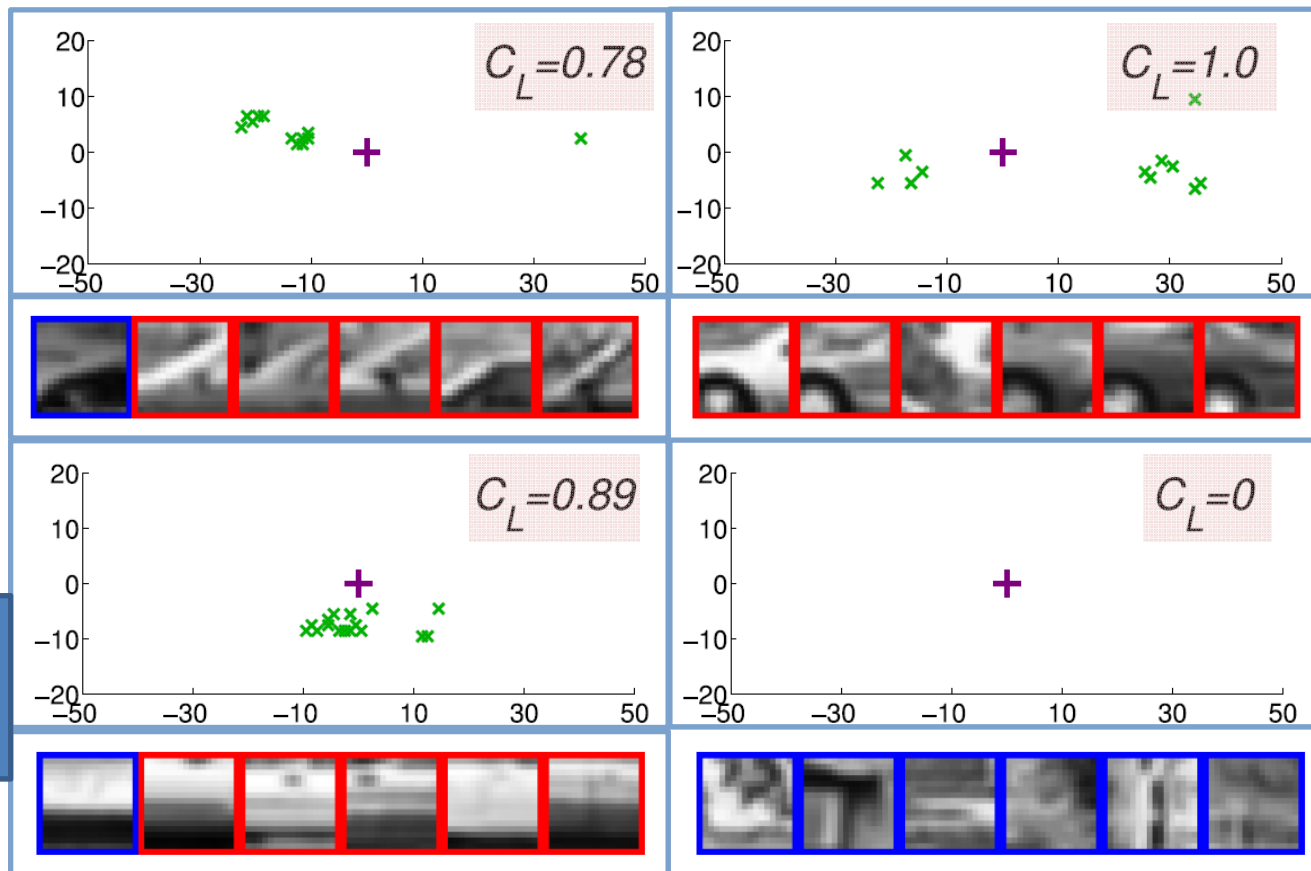
- クラスラベル
- オフセット

# Hough Forest の学習結果

[Gall *et al.* 2009]

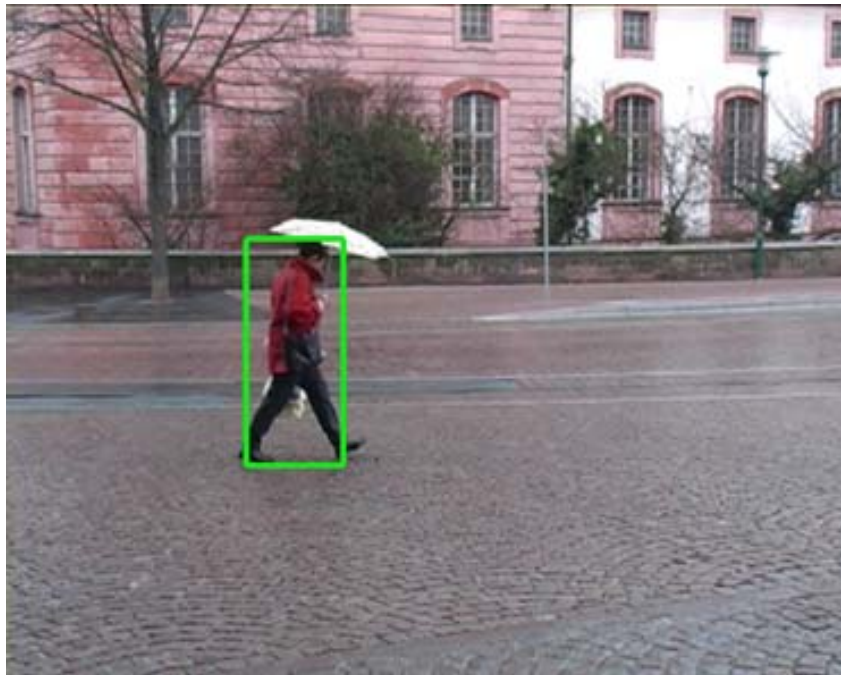


葉ノードに対応づいた  
クラス分布と  
オフセット量



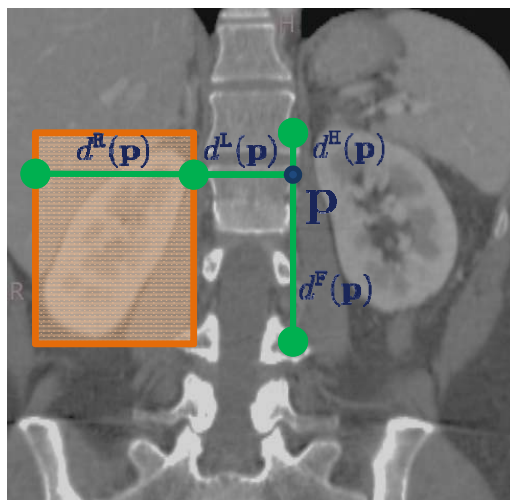
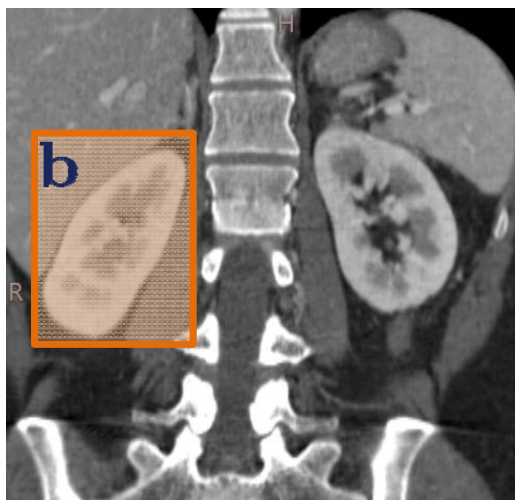
# Hough Forests の検出結果

[Gall *et al.* 2009]

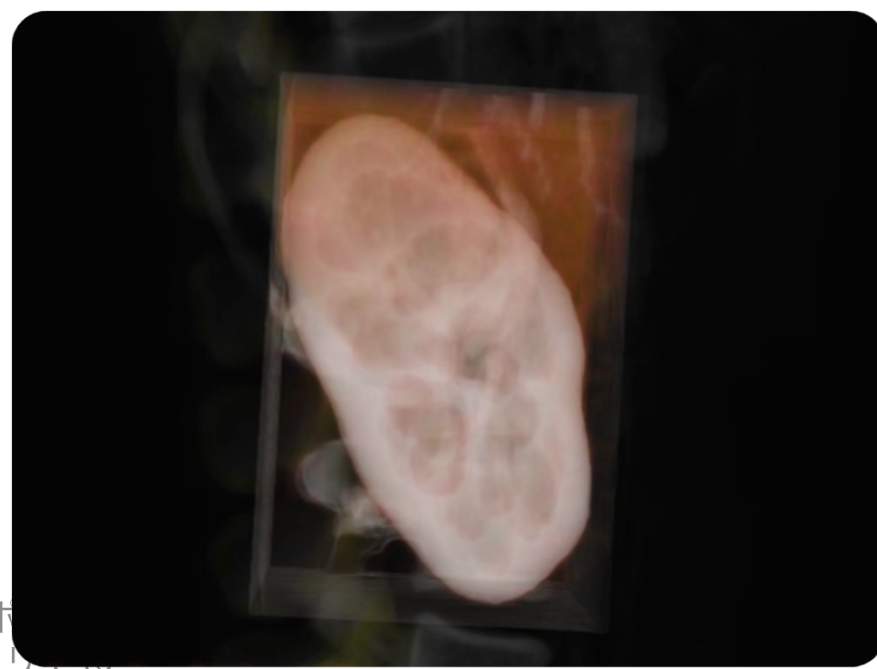


# CT画像中の臓器検出

[ Criminisi et al. 2010 ]



- 入力：  
・局所ボクセル
- 出力：  
・臓器の位置

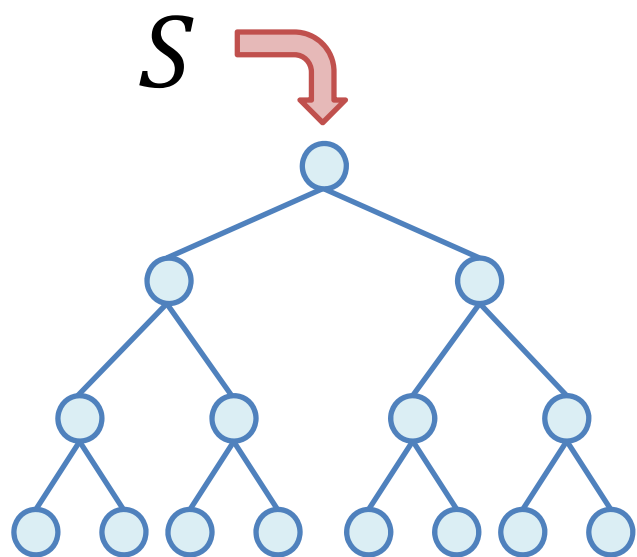


# アウトライン

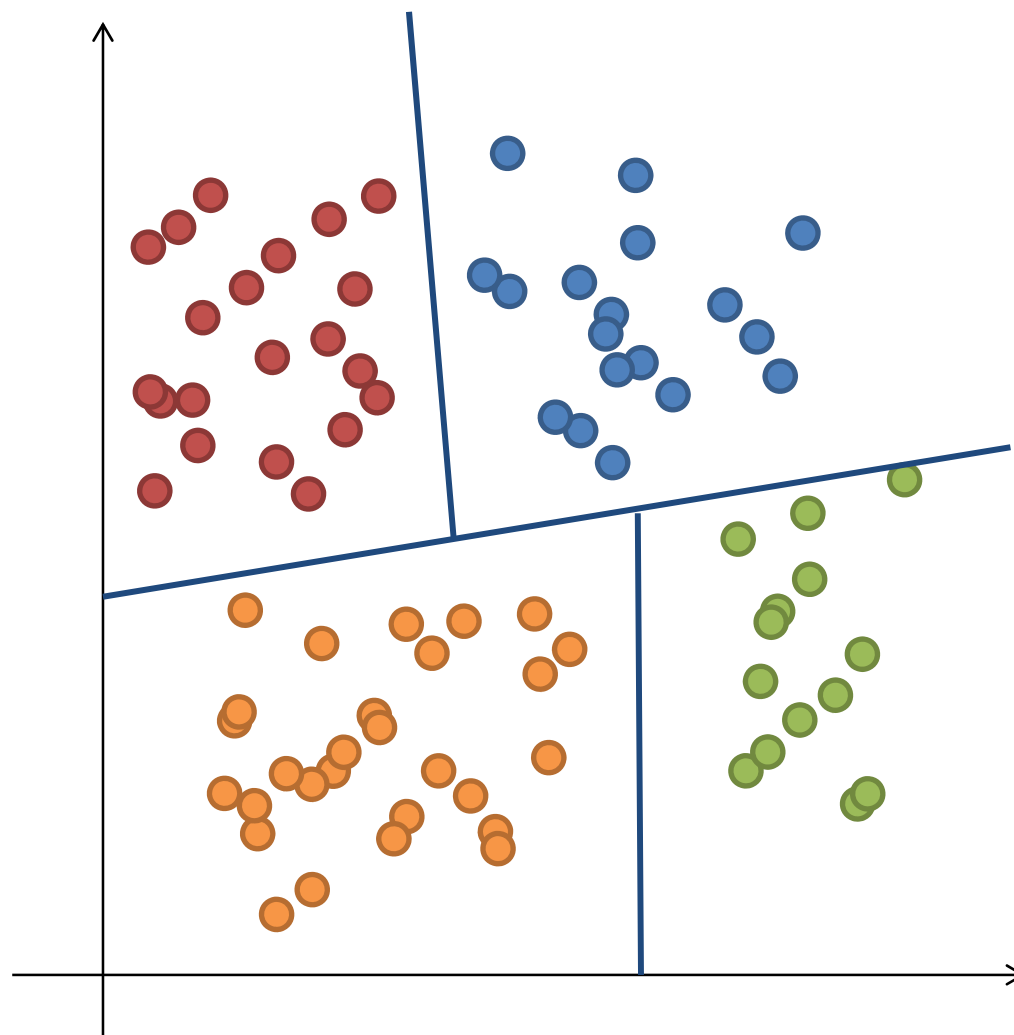
---

- ランダムフォレストの概要
  - 学習
  - テスト
  - 性質
- コンピュータビジョンへの応用
  - 利用法
  - パターン識別
  - 回帰
  - クラスタリング
- まとめ
- 応用例

# 決定木構築→クラスタリング



教師なしの場合  
距離尺度で分割してく

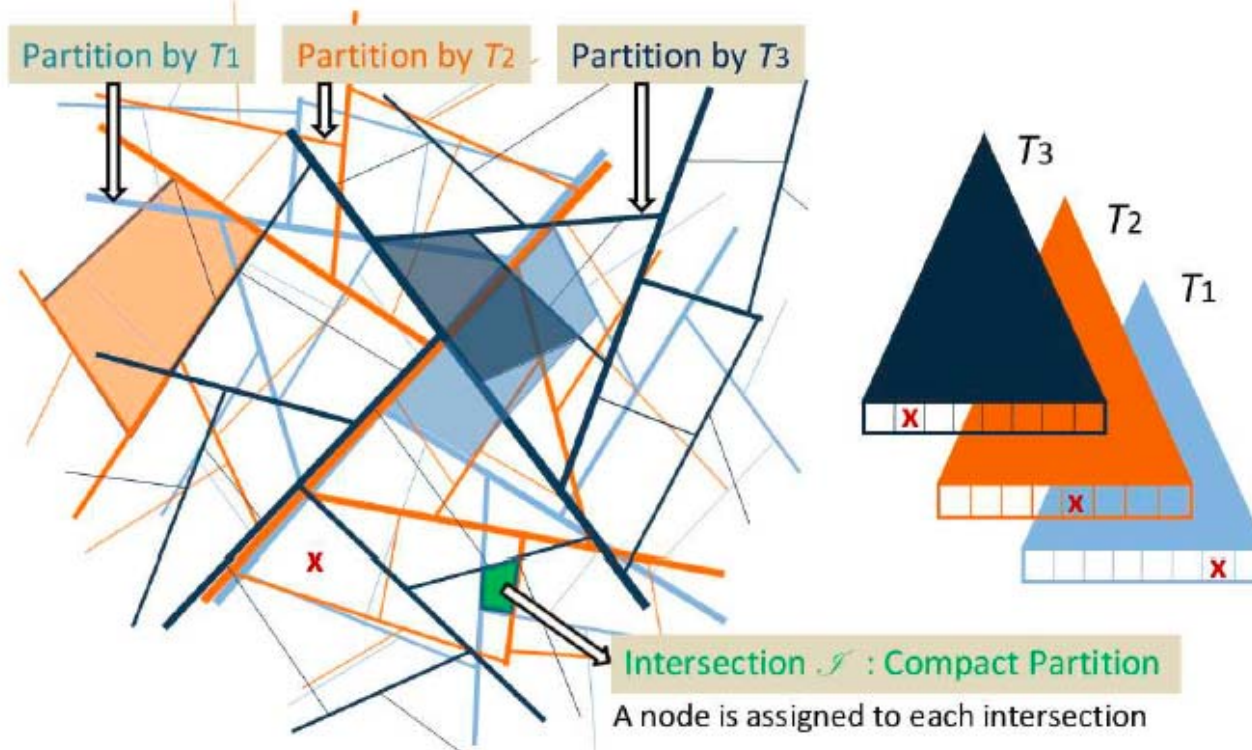




# ビデオセグメンテーション

[Perbet *et al.* 2009]

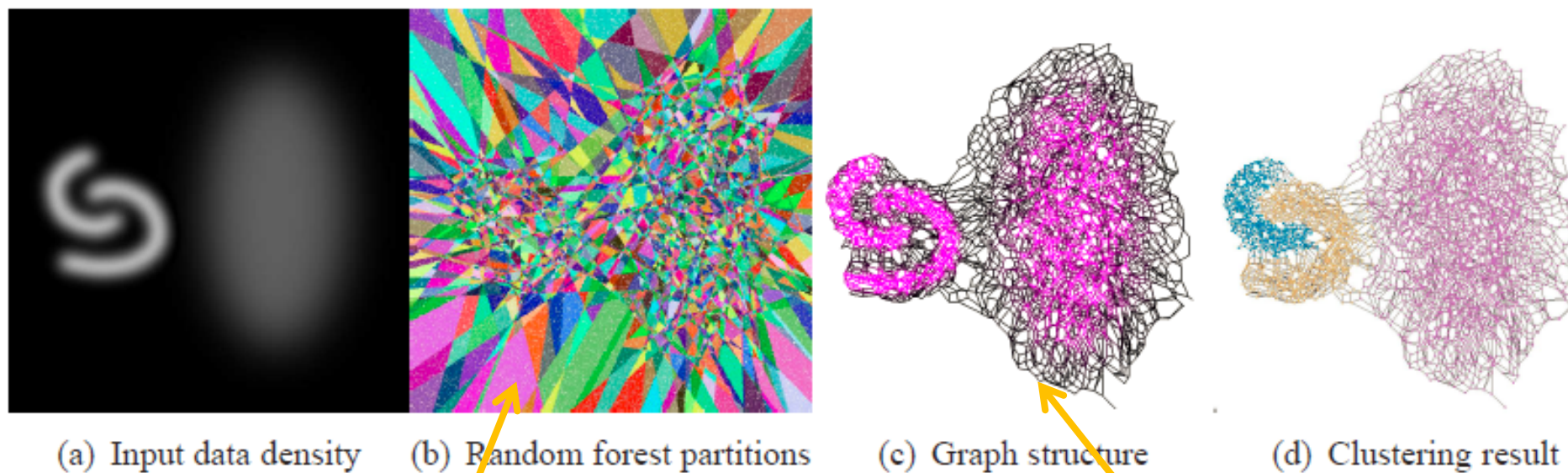
- ランダムフォレストを用いて特徴空間内でデータを細分化



- グラフクラスタリングで領域統合

# 細分化結果のグラフ表現

[Perbet *et al.* 2009]

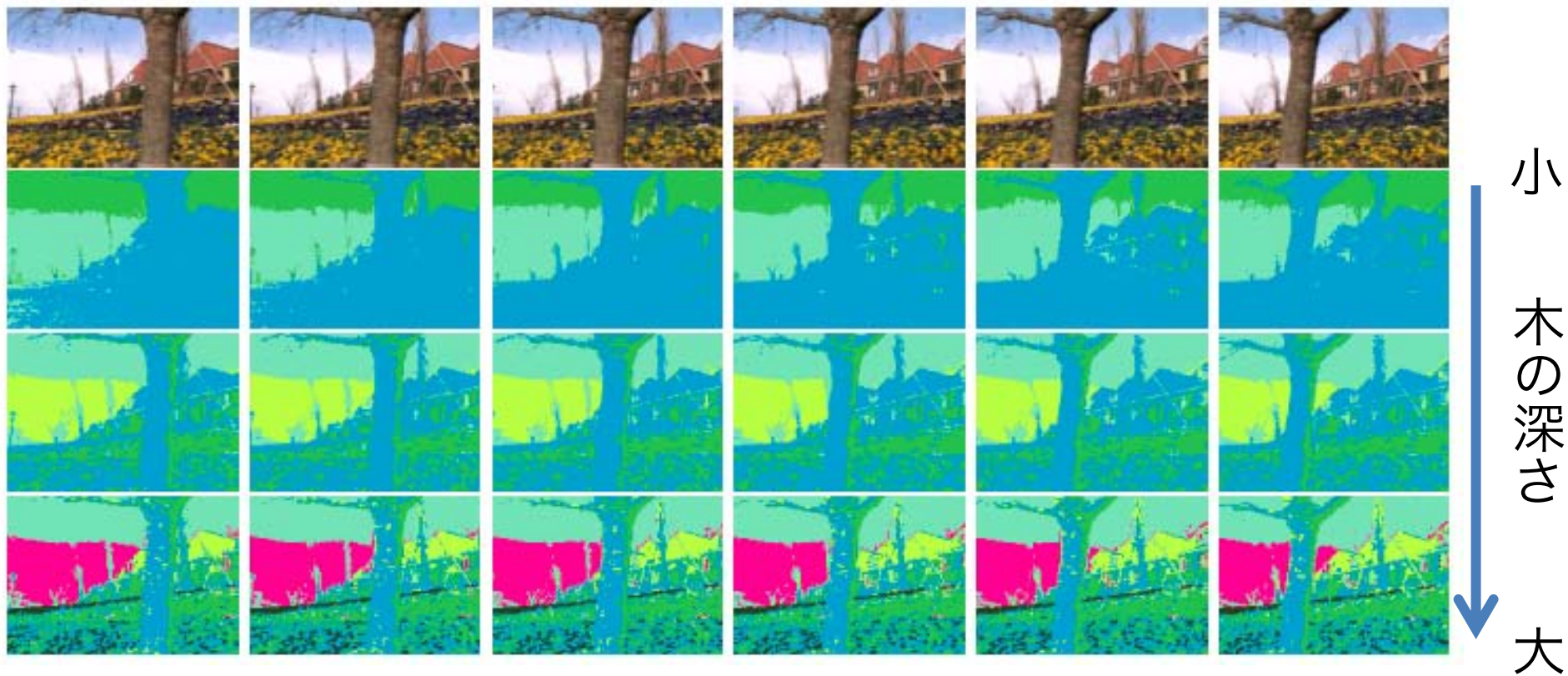


インデックスベクトル  
 $l(\mathbf{x}) = (l_1, l_2, \dots, l_T)$

$l(\mathbf{x})$ のハミング距離が  
1のものを結合

# セグメンテーション結果

[Perbet *et al.* 2009]



# アウトライン

---

- ランダムフォレストの概要
  - 学習
  - テスト
  - 性質
- コンピュータビジョンへの応用
  - 利用法
  - パターン識別
  - 回帰
  - クラスタリング
- まとめ
- 応用例

# まとめ

---

- ランダムフォレストの特長
  - 高速（並列化容易）・簡単・わかりやすい
  - 大量データに適している
- 利用の上でのポイント
  - 大量データが必要（人工的な合成）
  - 木の深さなどのパラメータ調整（あまり敏感でもないが）
  - メモリ使用量が多くなる

# 利用可能なライブラリ

---

- Open CV
  - CvRTrees
- Python
  - scikit-learn
  - <http://scikit-learn.org/stable/>
- Waffles
  - <http://waffles.sourceforge.net/>
- Randomforest-matlab
  - <http://code.google.com/p/randomforest-matlab/>
- Leo Breiman
  - [http://www.stat.berkeley.edu/~breiman/RandomForests/cc\\_home.htm](http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm)

# アウトライン

---

- ランダムフォレストの概要
  - 学習
  - テスト
  - 性質
- コンピュータビジョンへの応用
  - 利用法
  - パターン識別
  - 回帰
  - クラスタリング
- まとめ
- 応用例

# Random Fern

[Ozuysal *et al.* 2010]

- ナイーブベイズ

- $P(f_1, f_2, \dots, f_N | C = c_i) = \prod_{k=1}^N P(f_k | C = c_i)$



画像特徴は完全に独立ではない

- Semi-naïve Bayesian Approach

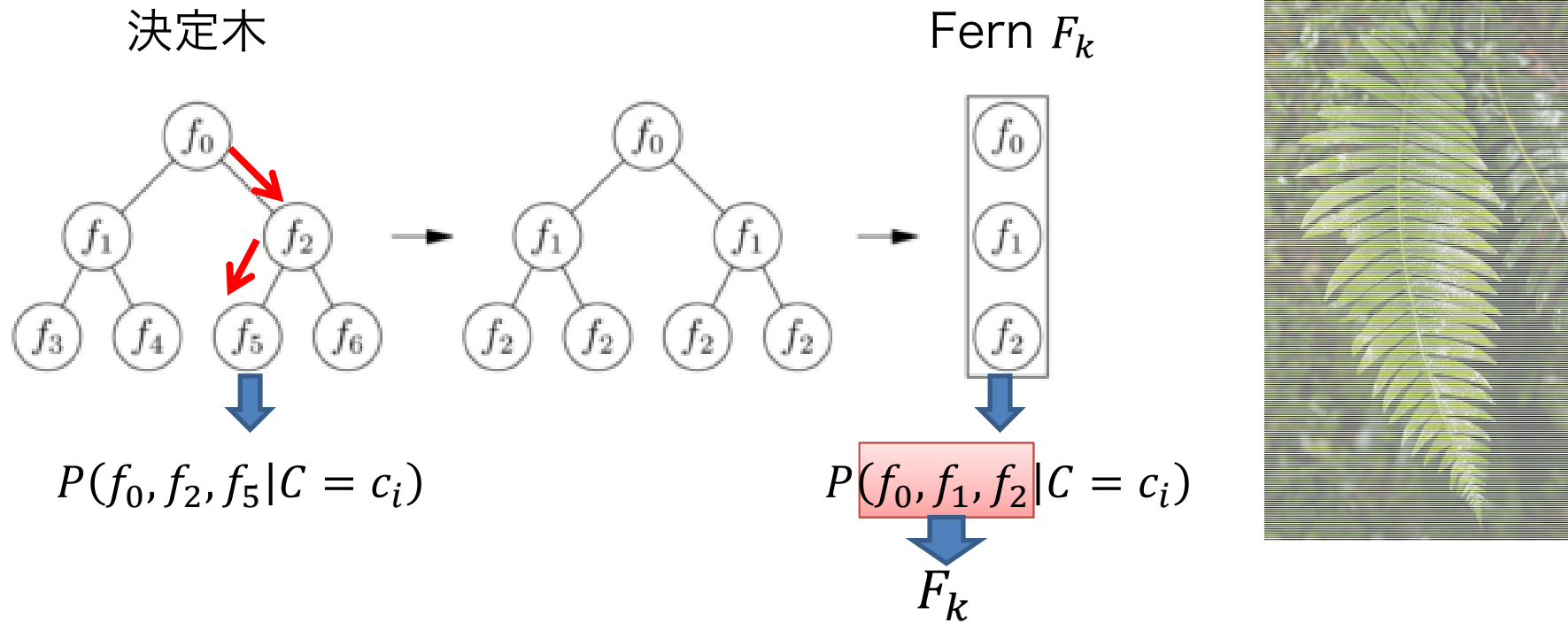
- $P(f_1, f_2, \dots, f_N | C = c_i) = \prod_{k=1}^M P(F_k | C = c_i)$

- $F_k = \{f_{\sigma(k,1)}, f_{\sigma(k,2)}, \dots, f_{\sigma(k,S)}\}, k = 1, \dots, M$



# RF vs. Random Fern

[Ozuysal *et al.* 2010]

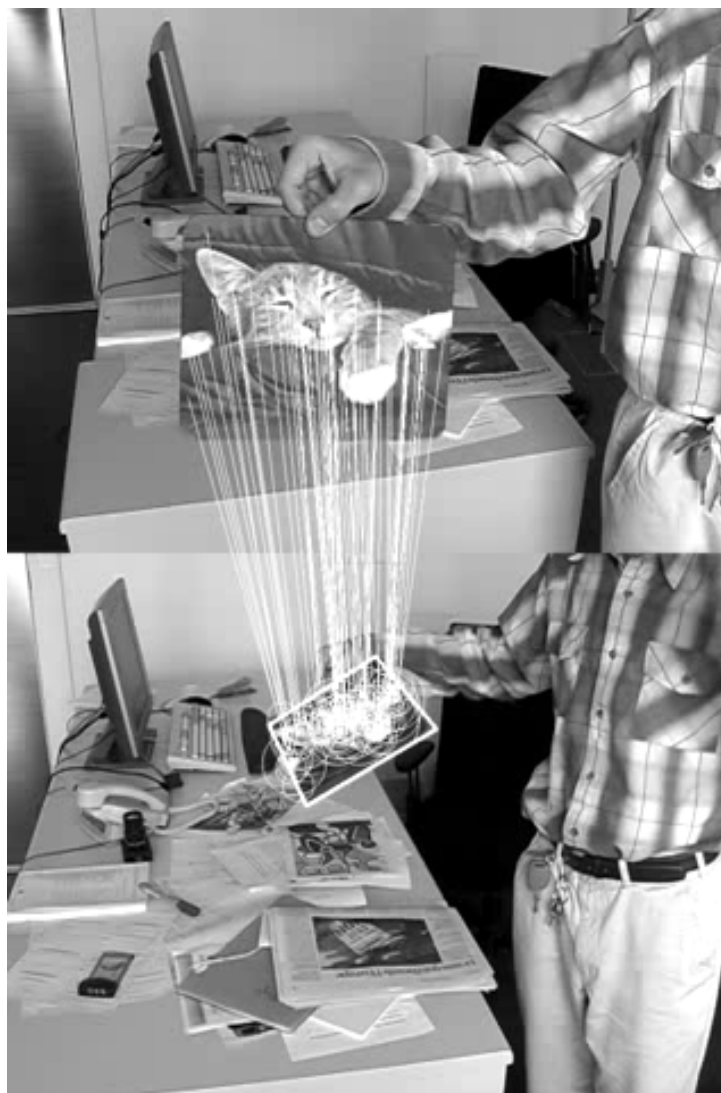


和ではなく積

$$P(f_1, f_2, \dots, f_N | C = c_i) = \prod_{k=1}^M P(F_k | C = c_i)$$

# 特徴点認識結果

[Ozuysal *et al.* 2010]



2013/6/12

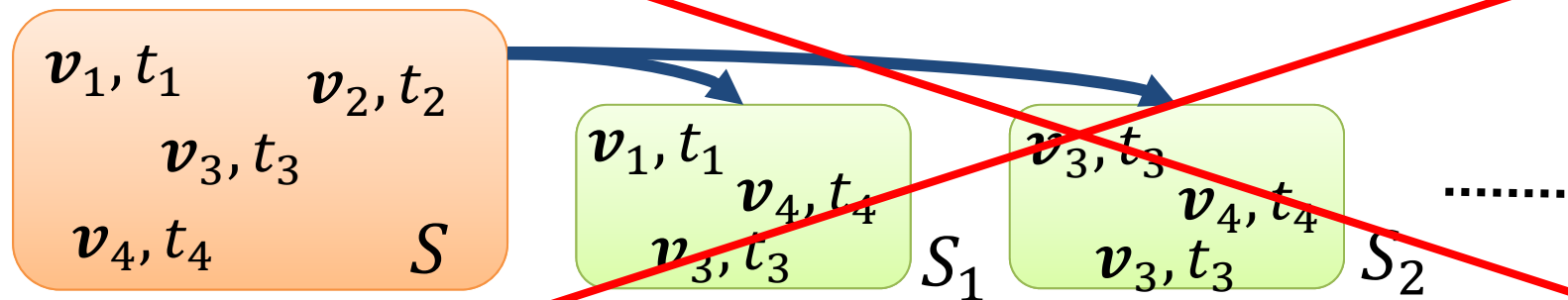
画像センシングシンポジウム SSII 2013  
チュートリアル

66

# Extremely Randomized Trees

[Geurts *et al.* 2006]

- ランダムフォレストとの違い
  - ブートストラップサンプルではなく学習サンプル全体を利用



- 分岐関数の決定を完全にランダムに

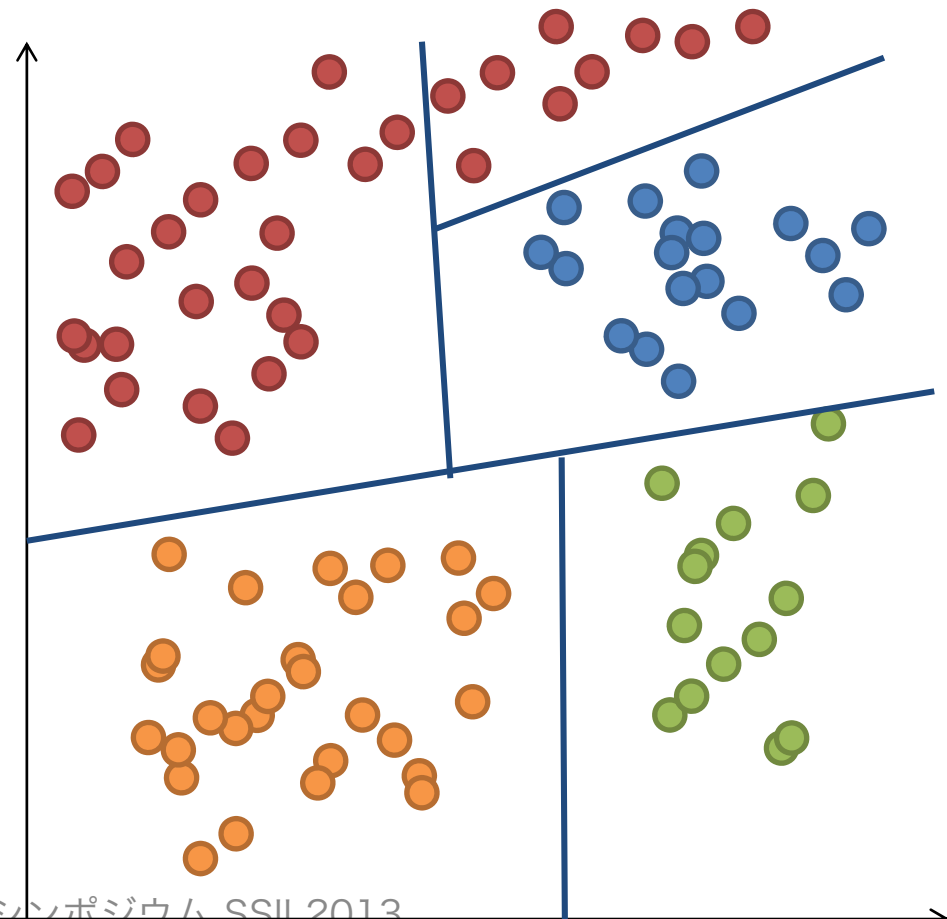
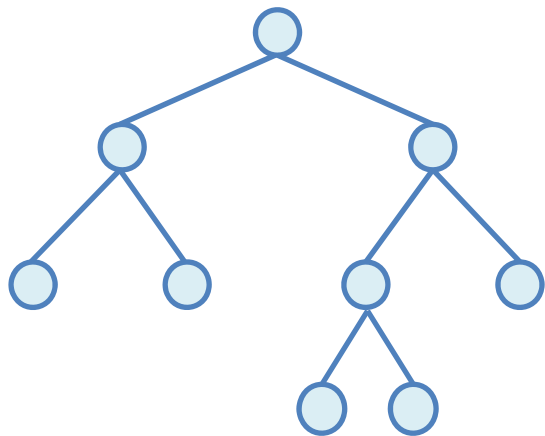
最適なパラメータ

$$\theta^* = \arg \max_{\theta \in \mathcal{S}_J} l_j$$

# On-line Random Forests

[Saffari et al. 2009]

- 初期モデル (Forest) 構築後に得られたデータを利用して Forest を逐次更新する



# 更新アルゴリズム(1)

[Saffari et al. 2009]

入力：学習済みのForest, 新規データ  $(\mathbf{v}_i, t_i)$

1. それぞれの決定木について以下を実行

- a. ポアソン分布に従って  $(\mathbf{v}_i, t_i)$  が発生していると仮定し  $k = \text{Poisson}(1)$  個だけ  $(\mathbf{v}_i, t_i)$  を生成
- b. 生成したデータを決定木に入力
- c. 葉ノードを分割可能か判断
- d. Out-of-bag による汎化誤差推定 → 不要な決定木の削除 & 新しい決定木の追加

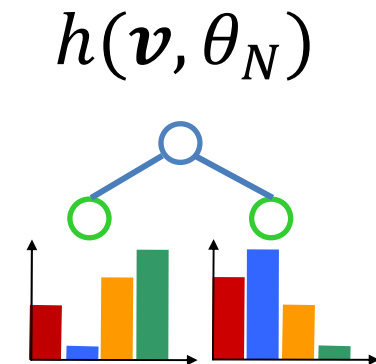
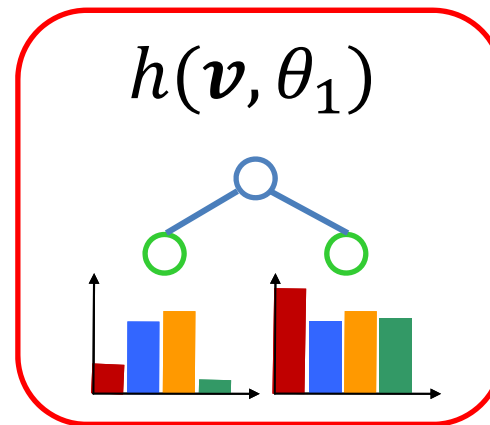
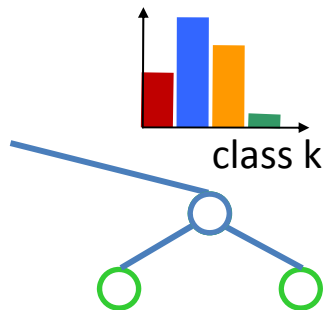
2. 更新終了

# 更新アルゴリズム(2)

[Saffari et al. 2009]

## 葉ノードにおける分割

現在の葉ノード 分岐関数の候補



すべての候補について分割した場合の  
情報利得を計算

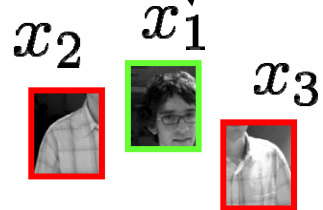
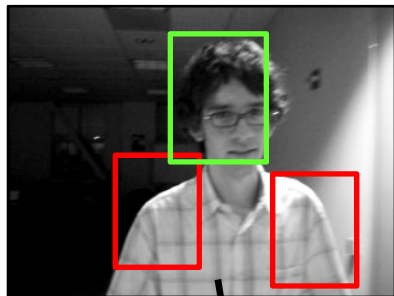


「子ノードのデータ数」と「情報利得」が  
それぞれある閾値を超えた場合に分割

# Multiple Instance Learning

[Babenko et al. 09]

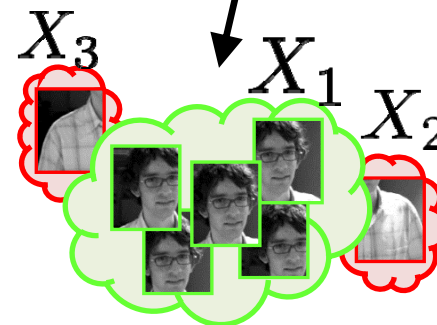
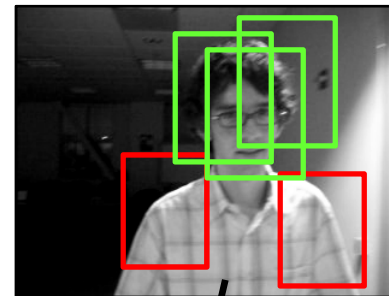
通常の学習



$\{(x_1, 1), (x_2, 0), (x_3, 0)\}$

Classifier

MILでの学習



$\{(X_1, 1), (X_2, 0), (X_3, 0)\}$

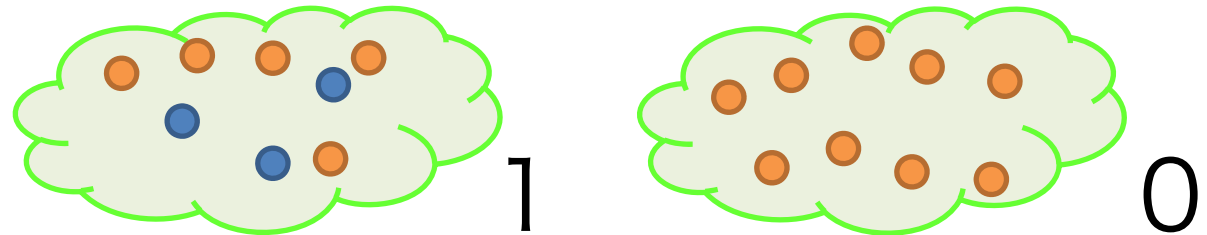
MIL  
Classifier

# MIForest

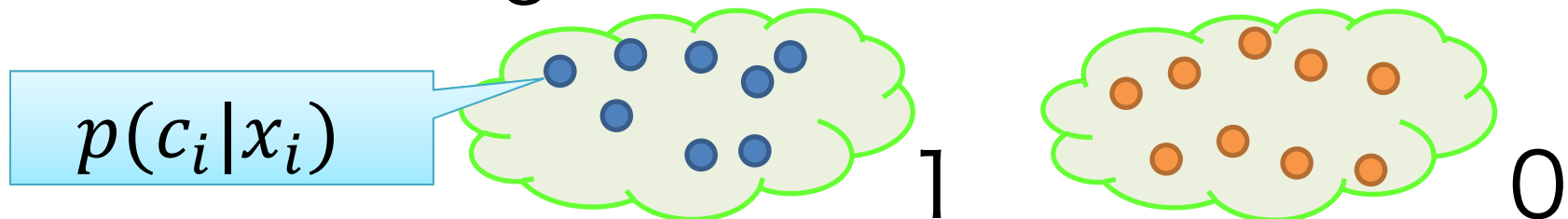
[Leistner et al. 2010]

繰り返しによって各データのラベルを更新

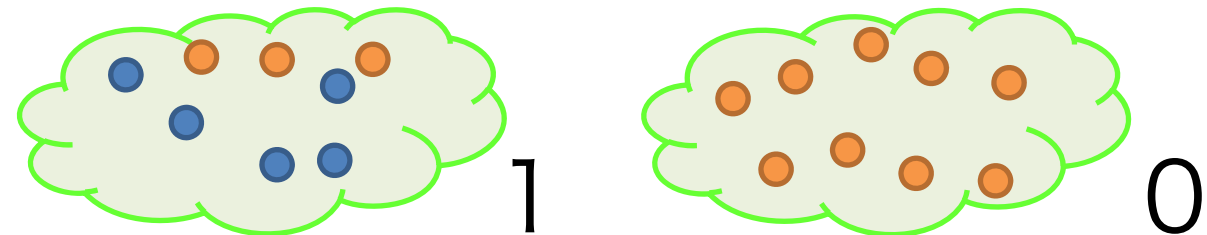
- 入力データ :



- 初期値 : 1 のbagのデータはすべて 1



- 推定された事後確率に応じてラベルを付与



- 繰り返し



# Conditional Regression Forests <sup>Modified</sup>

---

- Real-time Facial Feature Detection using **Conditional Regression Forests**
- **Conditional Regression Forests** for Human Pose Estimation

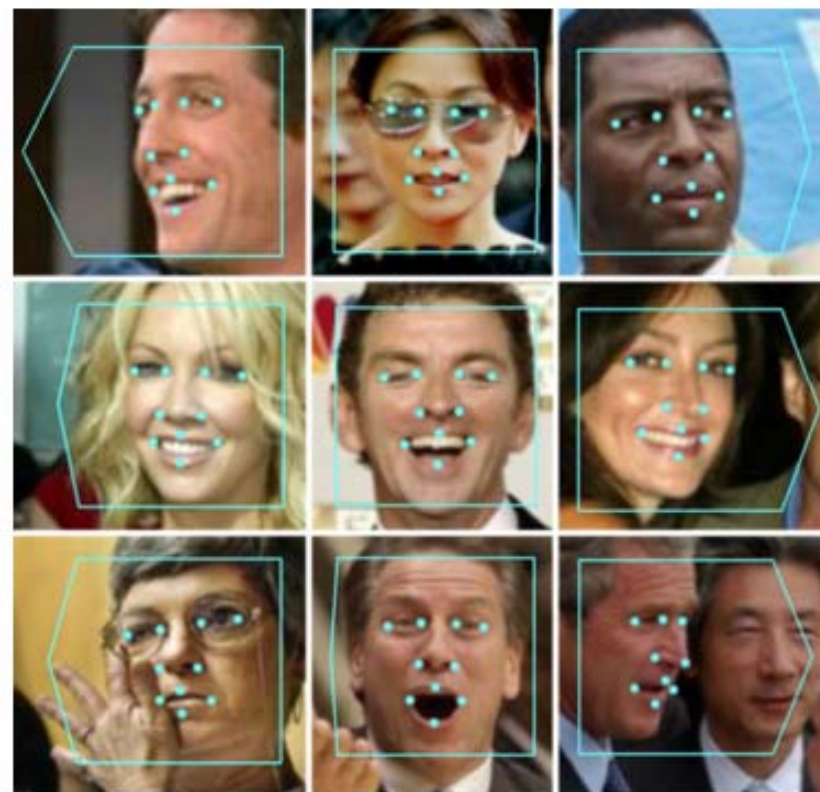
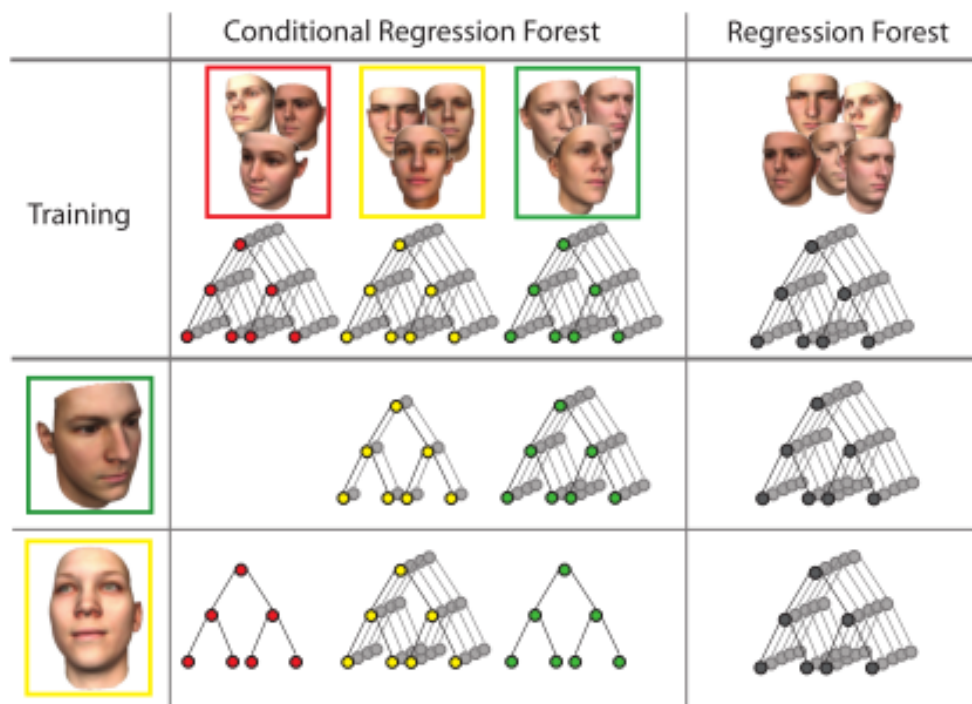
条件付き

回帰

$$\begin{aligned} S(x_j|I) &= \sum_{l \in L} \sum_{c_l \in C} p(x_j, c_l | I_l) \\ &= \sum_{l \in L} \sum_{c_l \in C} p(x_j | c_l) p(c_l | I_l) \end{aligned}$$

# CRF による顔特徴検出

[Dantone et al. 2012]



# CRF による人物姿勢推定

[Sun et al. 2012]

- グローバルな姿勢パラメータの条件付確率を表現



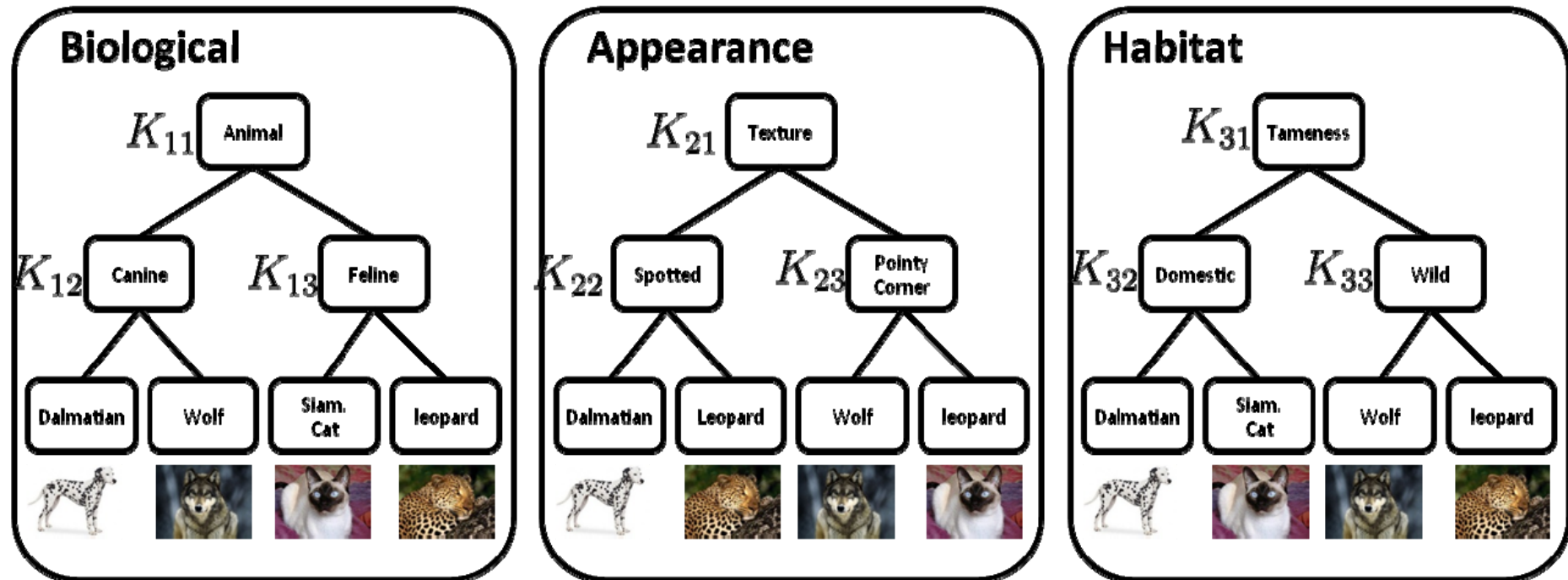
# 最近の事例 – NIPS2012

---

Modified

- Semantic Kernel Forests from Multiple Taxonomies [Hwang et al. NIPS 2012]
- Context-Sensitive Decision Forests for Object Detection [Kontschieder et al. NIPS 2012]

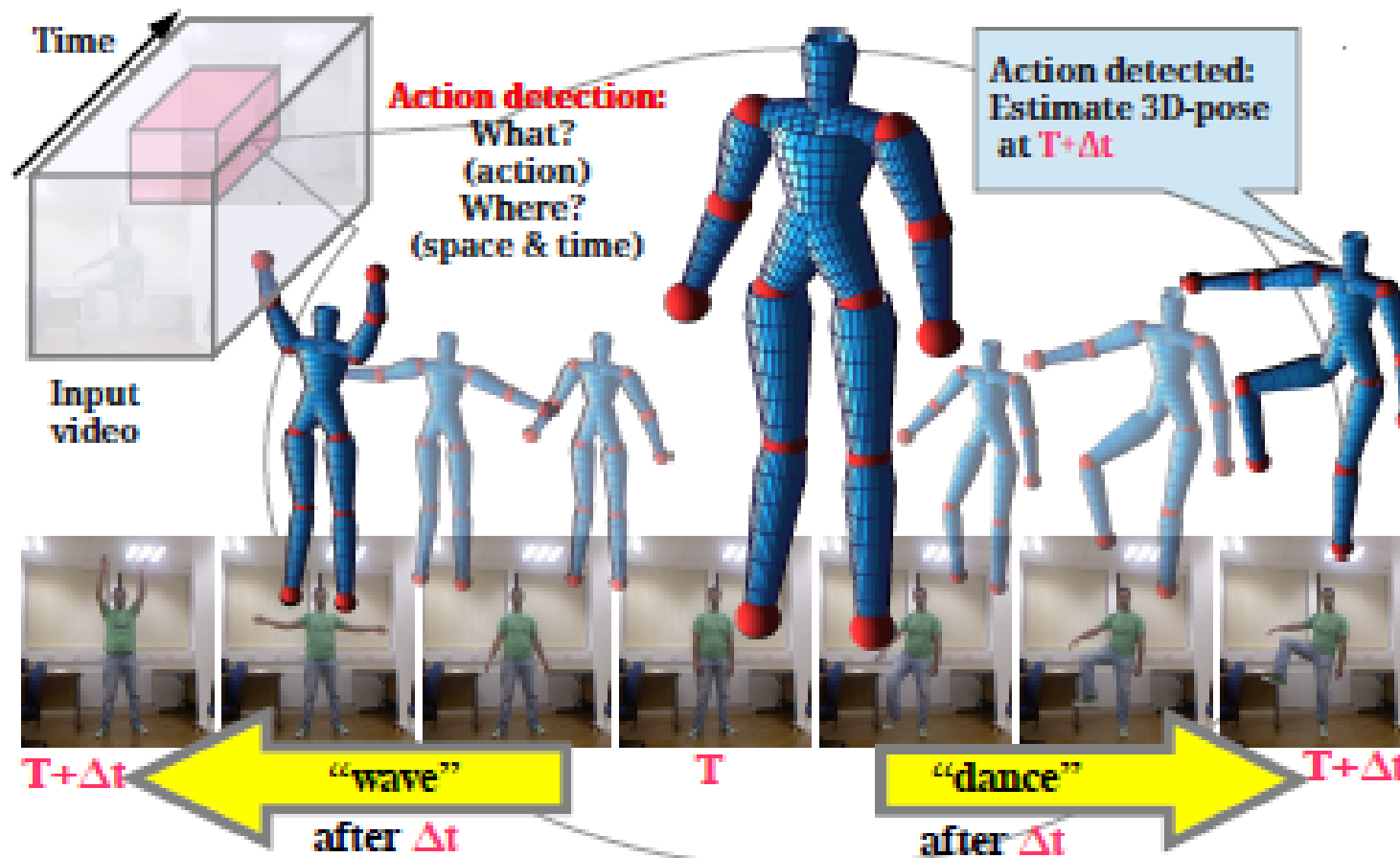
# Semantic Kernel Forests from Multiple Taxonomies



- 
- Semi-supervised Node Splitting for Random Forest Construction
  - GeoF: Geodesic Forests for Learning Coupled Predictors
  - Unconstrained Monocular 3D Human Pose Estimation by Action Detection and Cross-modality Regression Forest
  - Alternating Decision Forests
  - Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images

# 単一のビデオからの姿勢推定

Modified

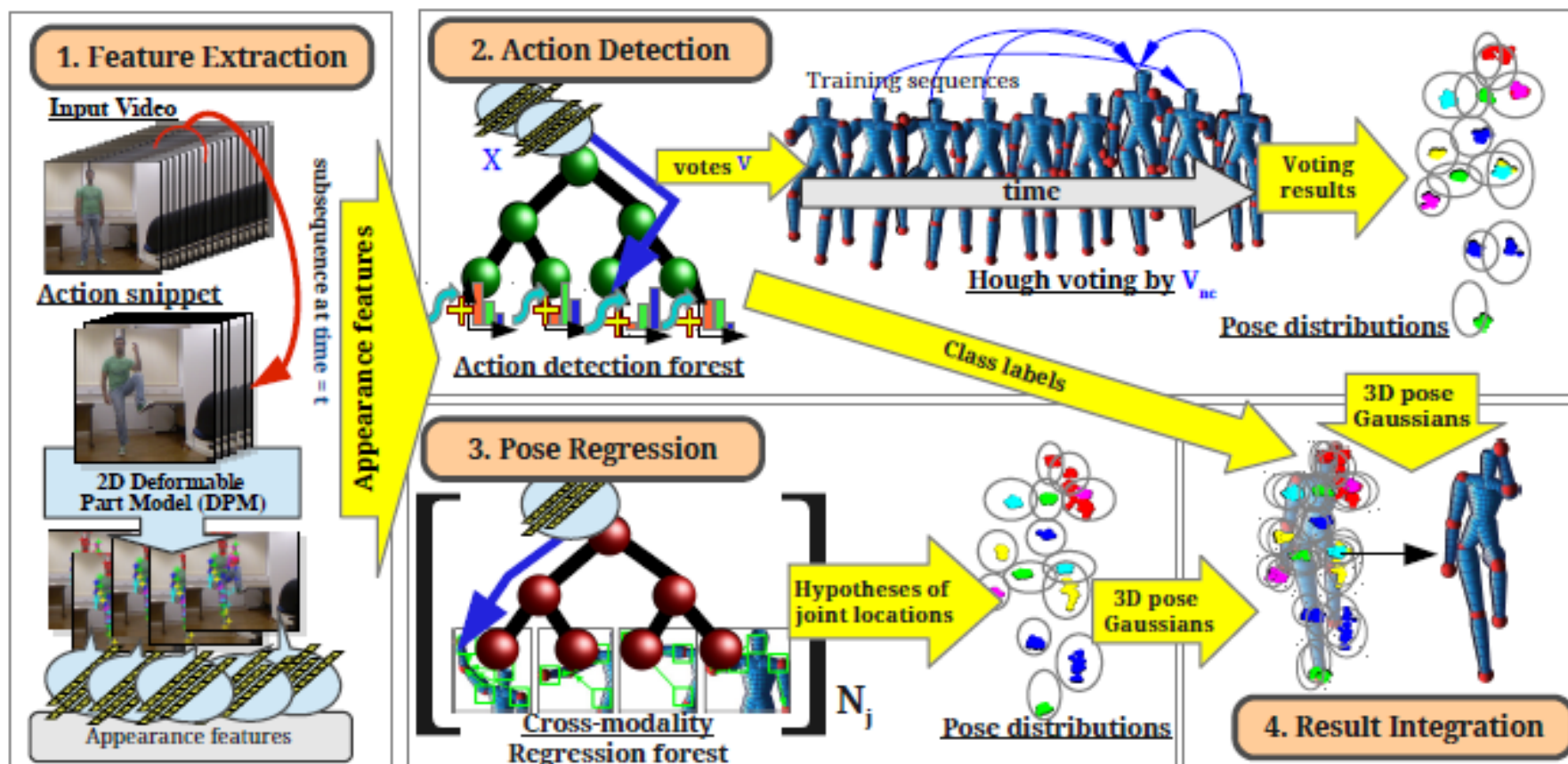


T.H. Yu, T.K. Kim, R. Cipolla,  
Unconstrained Monocular 3D Human Pose Estimation by Action  
Detection and Cross-modality Regression Forest

2013/6/12

# RF による動作検出と姿勢推定を統合

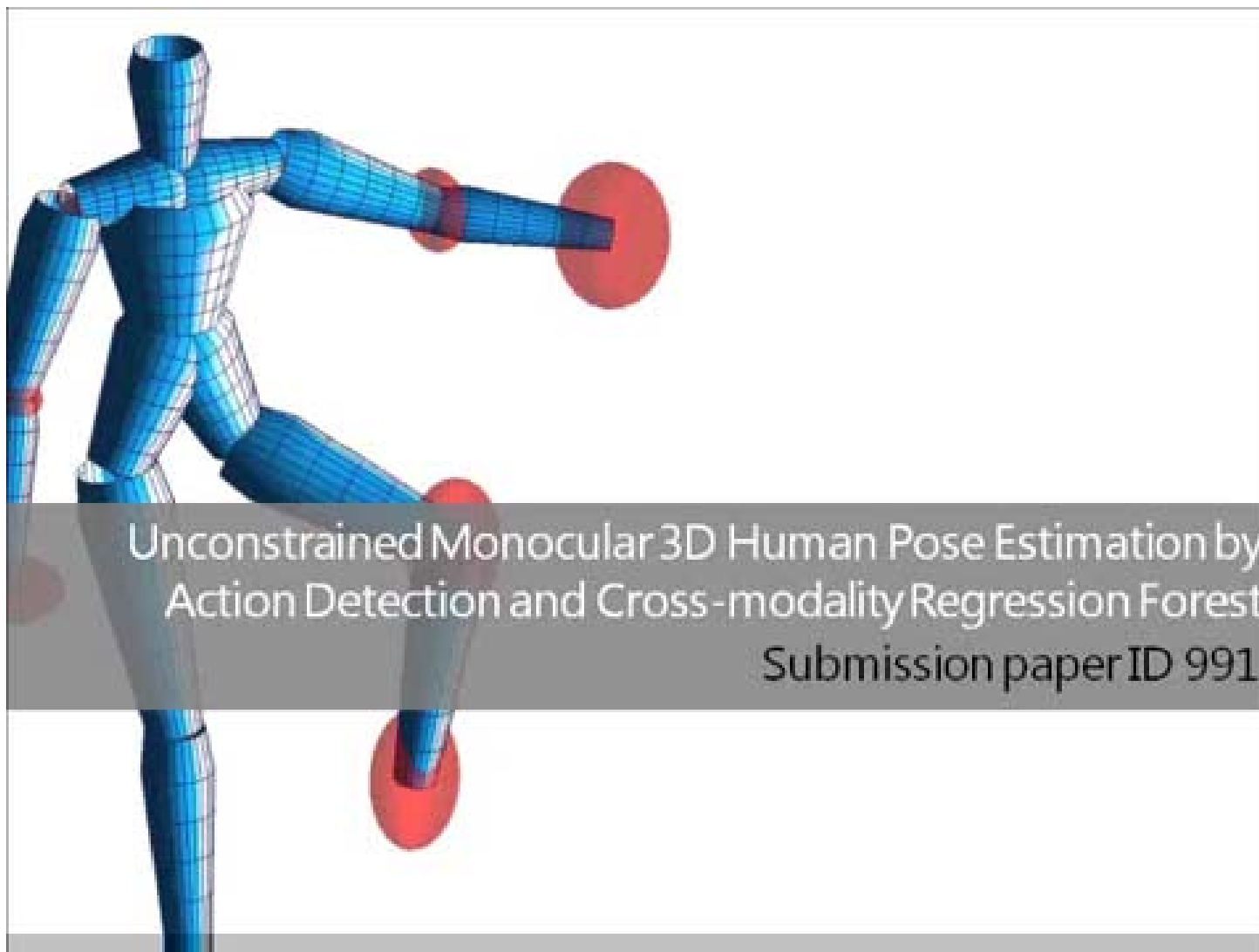
Modified





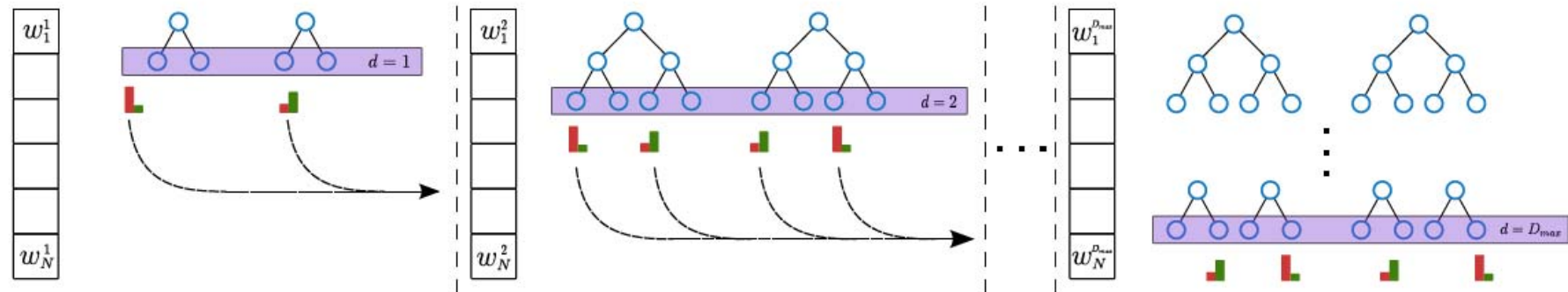
# デモビデオ

Modified



# Alternating Decision Forests Modified

[Schulter et al. CVPR 2013]



- 深さ優先ではなく，幅優先で木を構築する
- 損失関数を大域的に最小化
- 通常の Random Forests よりも良い結果

# Follow-ups

---

- 本講演の元になった情報処理学会CVIM研究会での発表スライドを以下で公開中

<http://www.habe-lab.org/habe/RFtutorial>

- 本講演についても補足情報があれば上記アドレスで公開予定
- ご意見などコメントいただければ幸いです

- Prof. Roberto Cipolla at Cambridge University and his group at CUED
- 牧 淳人 先生（スウェーデン王立工科大学）
- Prof. Tae-Kun Kim at Imperial College London
- 奈良先端大情報科学研究科 『若手研究者海外武者修行プログラム』
- 所属研究室の皆様
  - 奈良先端大・木戸出研究室
  - 奈良先端大・横矢研究室
  - 大阪大学・八木研究室

順不同

# 謝辞

Modified

- 
- 画像センシングシンポジウム実行委員会の皆様
  - インターネットなどで情報を公開してくださっている皆様
  - ご参加の皆様

順不同

明日（6月13日）の  
10時～11時にデモブースにおります

明日まではSSIIに参加しておりますので  
他の時間帯にも何かございましたら  
お声がけください

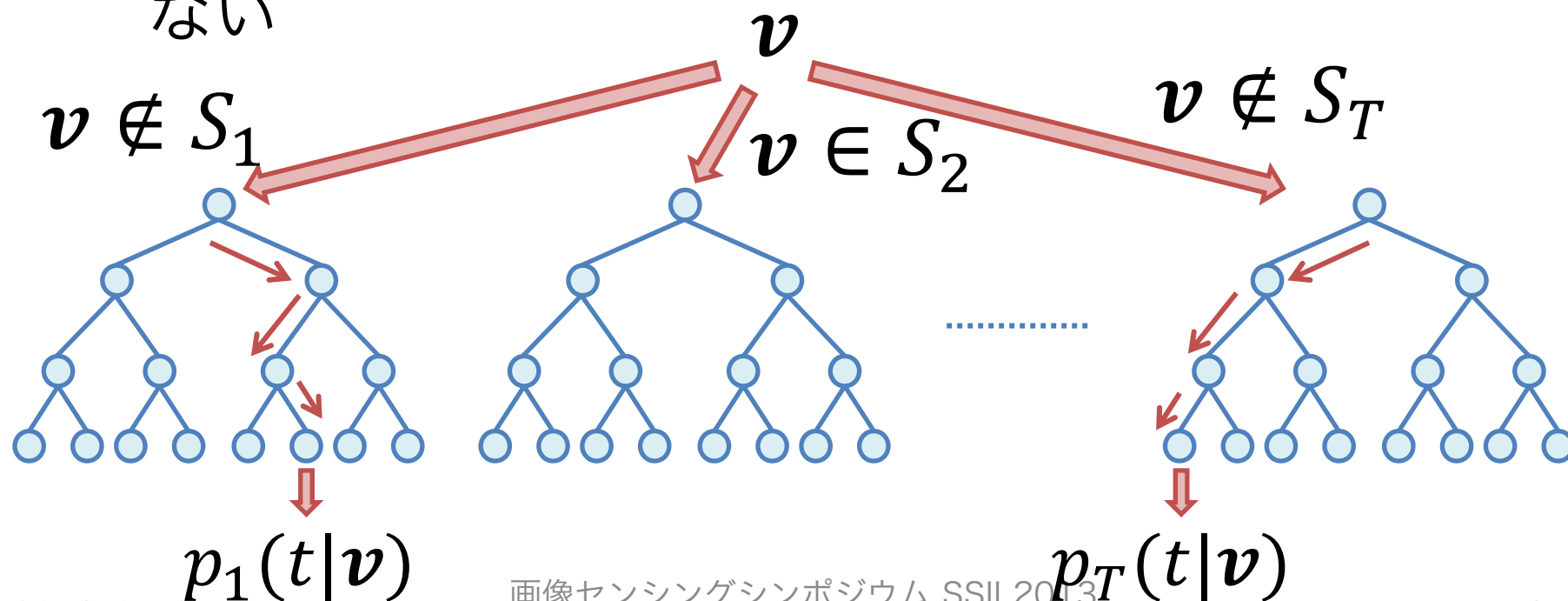


**ご静聴ありがとうございました。**

# ランダムフォレストの性質(3) [Breiman 2001]

- Adaboost is Random Forests

- ブートストラップサンプルに含まれなかったサンプルを使って汎化誤差を推定
- 交差検定 (cross-validation) を実行する必要はない



# Rotation Forests

[Rodriguez *et al.* 2006]

- 各決定木の“多様性”を増すためにPCAを利用して特徴空間を“回転”させる

