データストリームの分類

Savong Bou
計算科学研究センター
筑波大学

コンテンツ

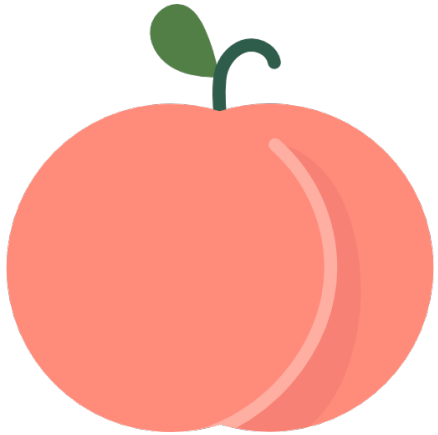
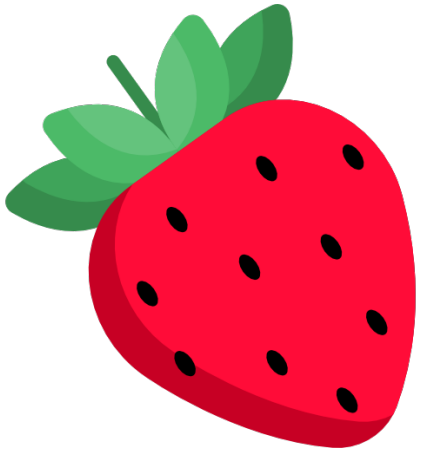
- 分類の概要
- 静的データ分類のレビュー
  - Decision Tree
- データストリームにおける分類
  - Hoeffding Tree
  - VFDT
  - CVFDT
  - Ensemble Method

https://www.flaticon.com/

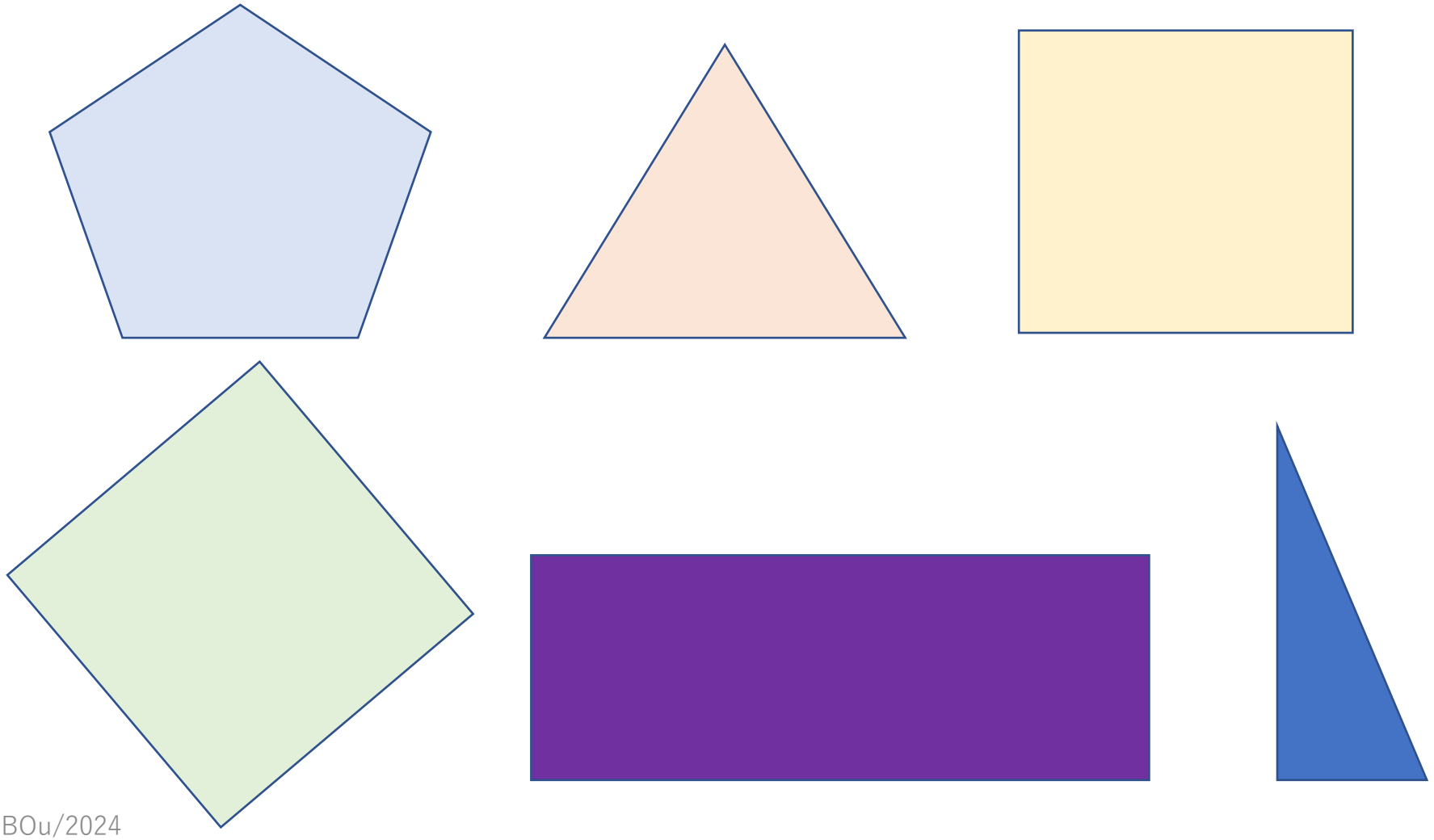
分類と数える

https://www.flaticon.com/

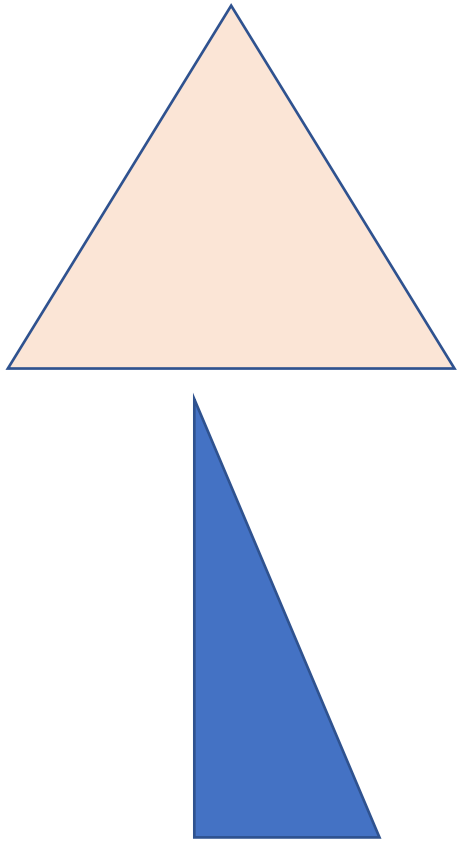
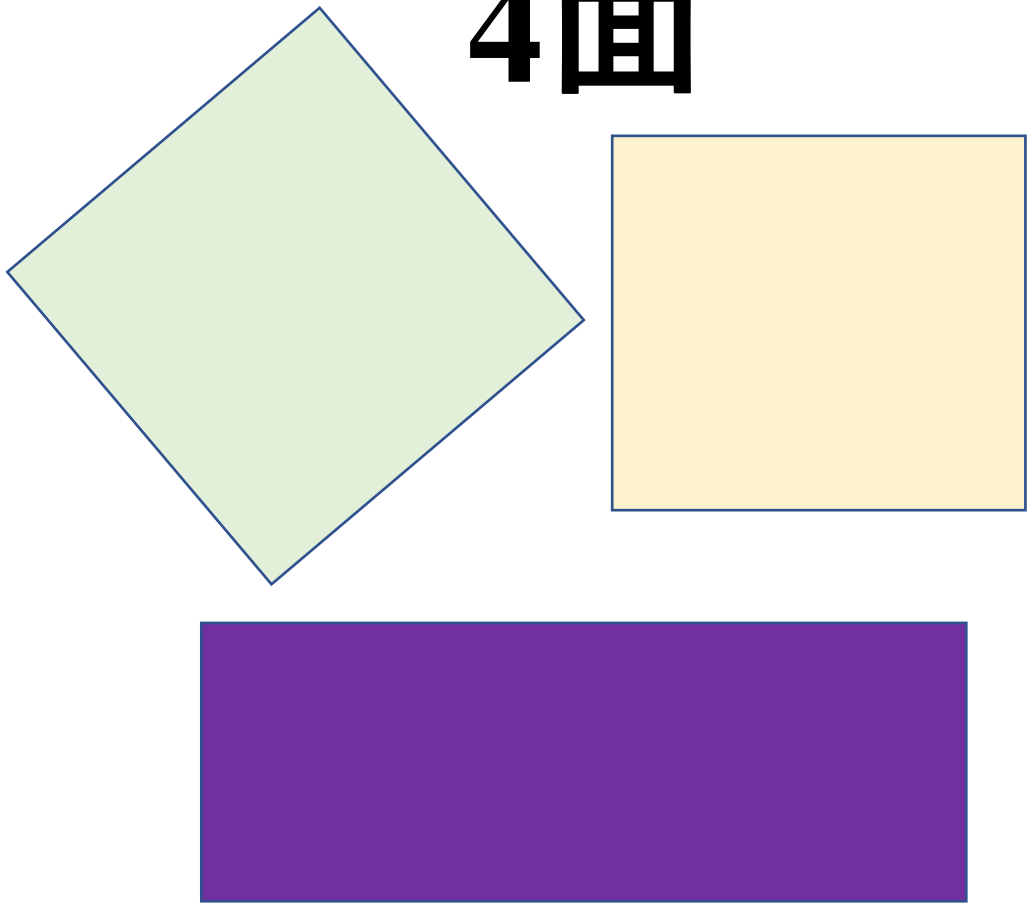
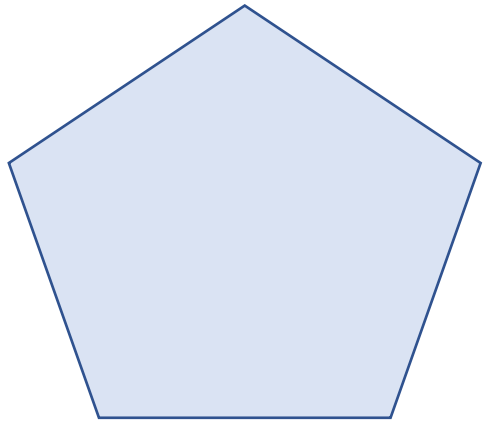
分類と数える

 4	 3	 5
 6	 3	 7

形状による分類



形状による分類

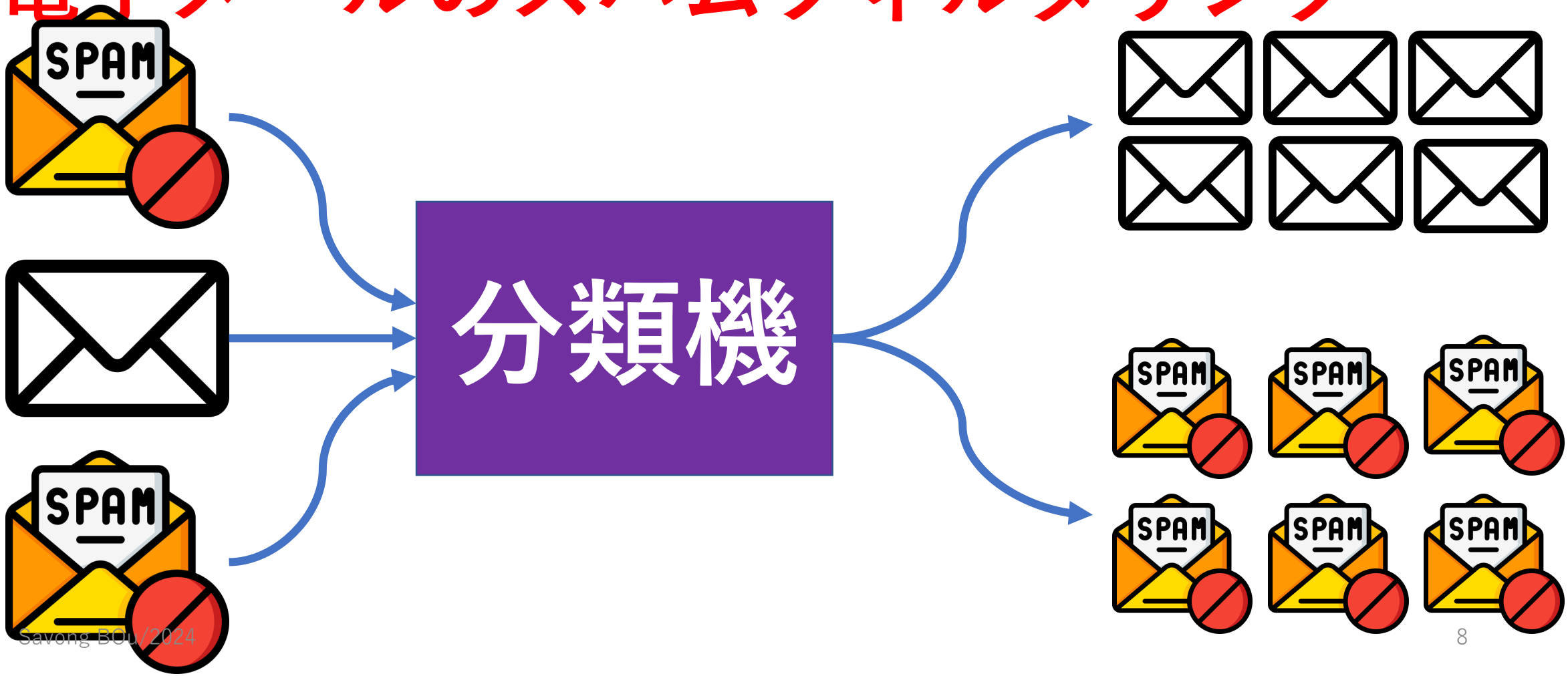
3面	4面	5面
		

分類

- オブジェクトのラベル/クラスを予測するためのデータ マイニング タスク
- 1 つ以上の数値/カテゴリ変数に基づいてモデルを作成
- 教師あり学習
  - 事前定義されたクラスを使用

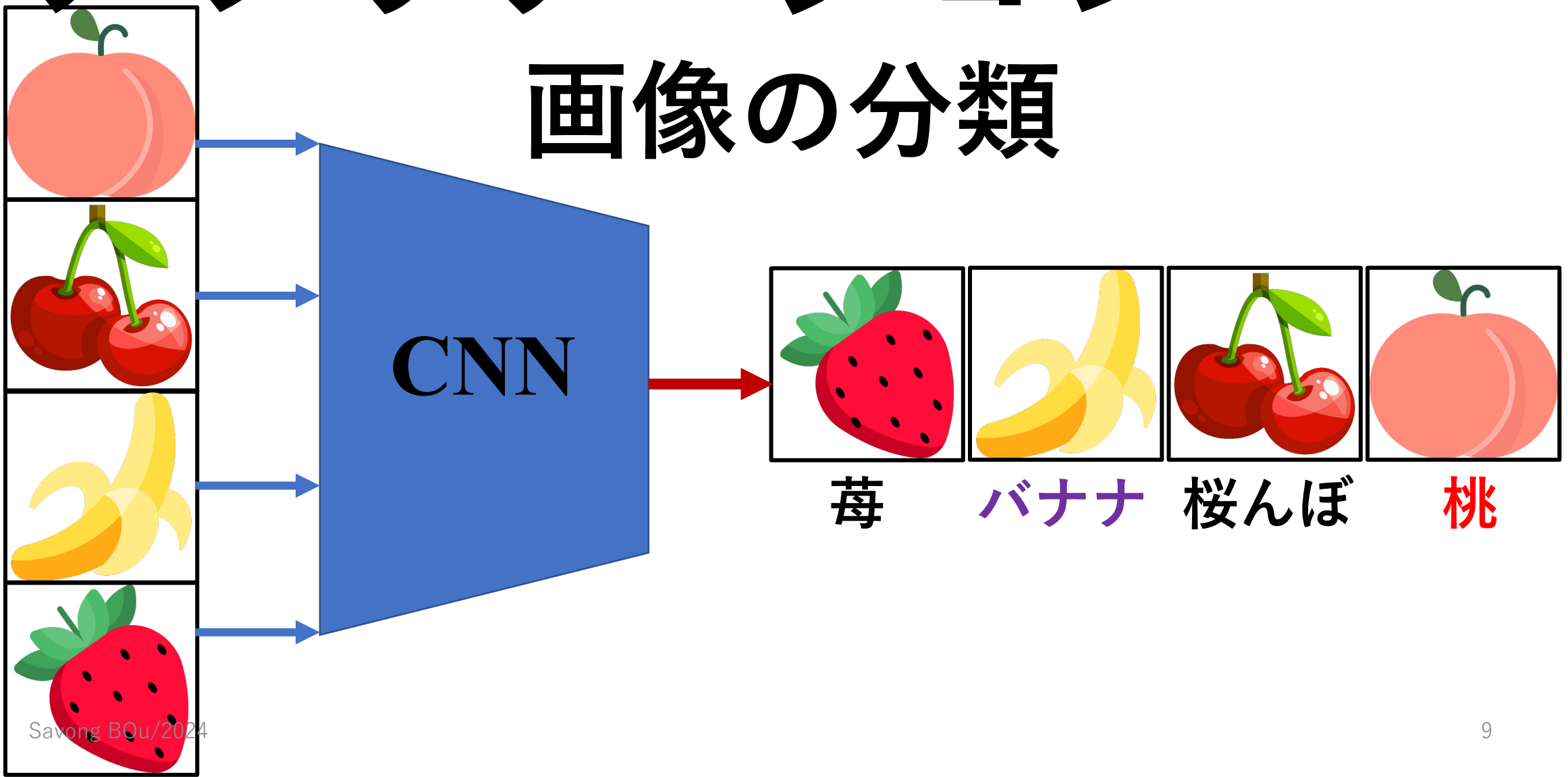
アプリケーション

電子メールのスパムフィルタリング



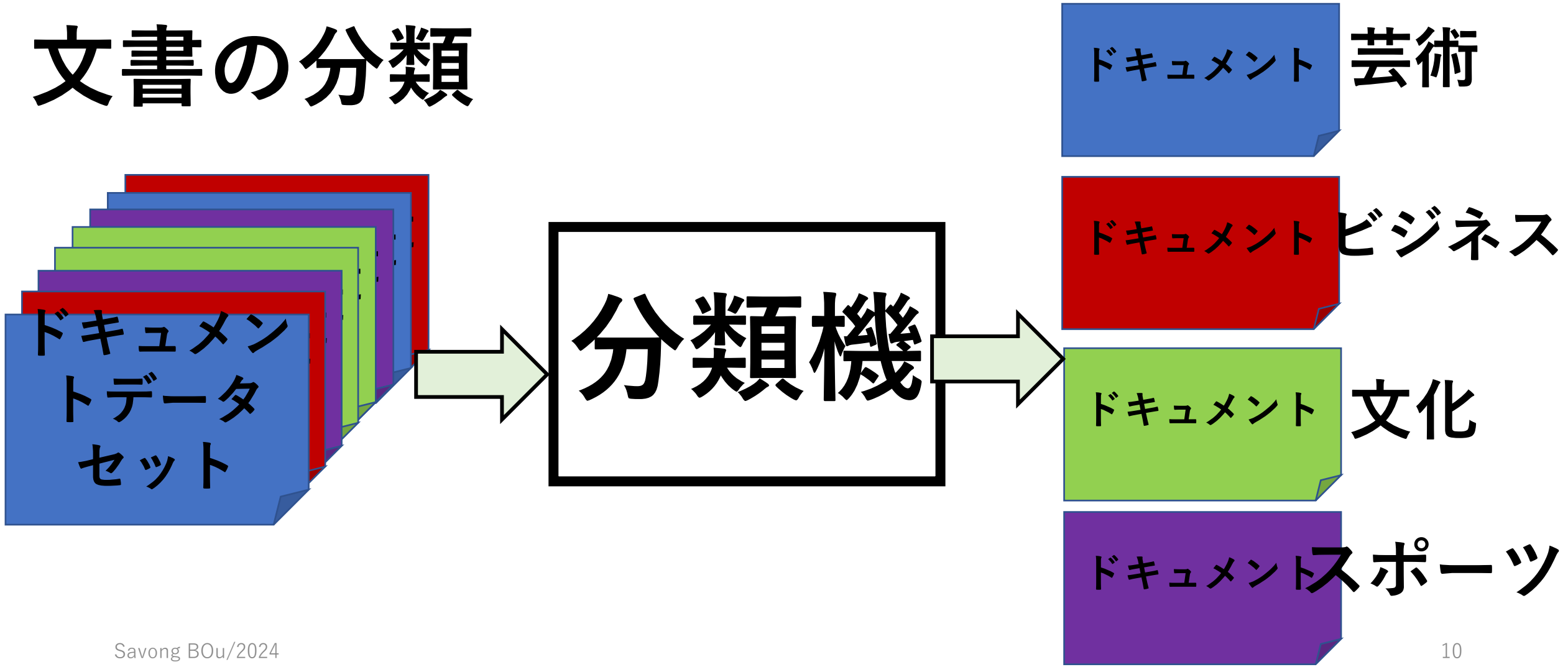
アプリケーション

画像の分類



アプリケーション

文書の分類



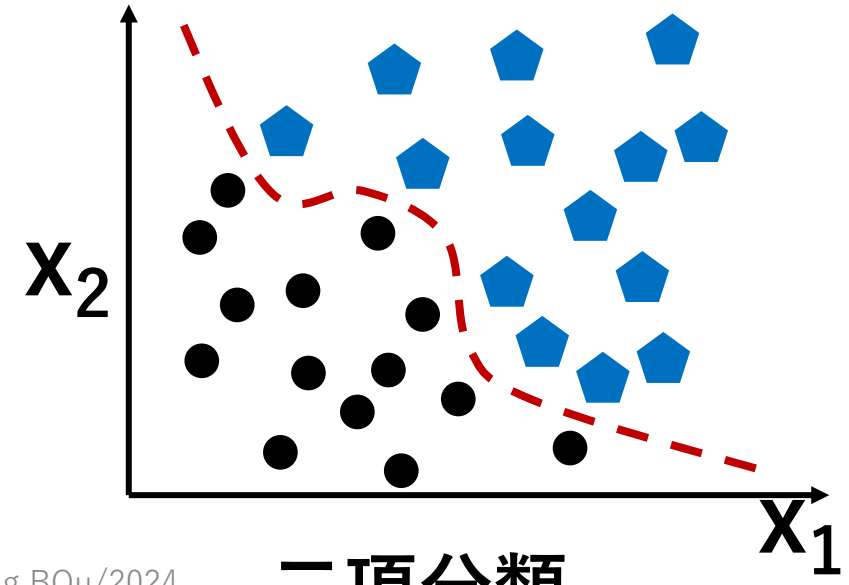
アプリケーション

- 文書の分類
- 広告のクリック率予測
- 製品の分類
- マルウェアの分類
- 画像感情分析
- 顧客離れの予測
- プロモーション特典に対する顧客行動の評価
- クレジットカード不正検知
- 感情分析

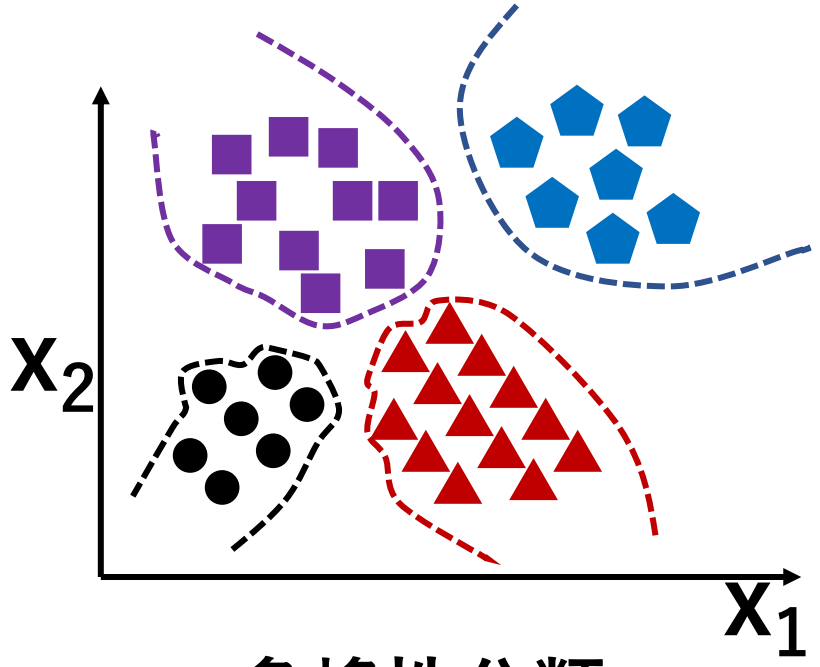
https://vitalflux.com/classification-problems-real-world-examples/

分類の種類

- 二項分類 (Binary classification)
- 多峰性分類 (Multimodal classification)



二項分類



多峰性分類

人気のあるアルゴリズム

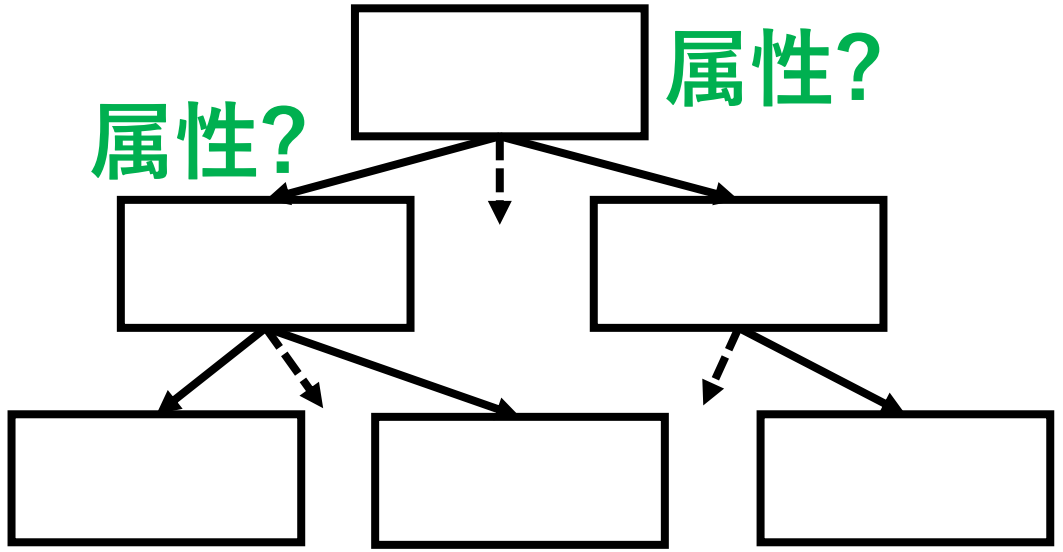
- Logistic regression
- Decision trees
- Random forest
- XGBoost
- Light GBM
- Voting classifiers
- Artificial neural networks

https://www.xoriant.com/blog/decision-trees-for-classification-a-machine-learning-algorithm

Decision Treeの例

その日にゴルフが行われるかどうかのDecision Treeを構築します

Day	Outlook	Temperature	Humidity	Wind	Play Golf
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



パーティションに最適な属性を選択する必要

Entropy & IG

https://www.xoriant.com/blog/decision-trees-for-classification-a-machine-learning-algorithm

Decision Treeの例

Entropy

$$H(S) = \sum_{x \epsilon X} p(x) log_2 \frac{1}{p(x)}$$

Day	Outlook	Temperature	Humidity	Wind	Play Golf
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Information Gain (IG)

$$IG(S, A) = H(S) - \sum_{i=0}^{n} p(x) * H(x)$$

“Play Golf”

$$H(S) = \sum_{x \epsilon X} p(x) log_2 \frac{1}{p(x)}$$

$$= -\left(\frac{9}{14}\right) log_2 \left(\frac{9}{14}\right) - \left(\frac{5}{14}\right) log_2 \left(\frac{5}{14}\right) = 0.940$$

可能な限り最高のInformation Gainを与えるルート ノードの最適な属性を選択

https://www.xoriant.com/blog/decision-trees-for-classification-a-machine-learning-algorithm

Decision Treeの例

「Wind」属性をはじめとする全属性のIGを計算

Day	Outlook	Temperature	Humidity	Wind	Play Golf
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$IG(S, Wind) = H(S) - \sum_{i=0}^{n} p(x) * H(x)$$

$$= H(S) - P(S_{Weak}) * H(S_{weak}) - P(S_{Strong}) * H(S_{Strong})$$

$$= 0.940 - \left(\frac{8}{14}\right)(0.811) - \left(\frac{6}{14}\right)(1.00) = 0.048$$

$$P(S_{Weak}) = \frac{\#\ of\ Weak}{Total} = \frac{8}{14}$$

$$P(S_{Strong}) = \frac{\#\ of\ Strong}{Total} = \frac{6}{14}$$

$$H(S_{Weak}) = -\left(\frac{6}{8}\right) log_2 \left(\frac{6}{8}\right) - \left(\frac{2}{8}\right) log_2 \left(\frac{2}{8}\right) = 0.811$$

$$H(S_{Strong}) = -\left(\frac{3}{6}\right) log_2 \left(\frac{3}{6}\right) - \left(\frac{3}{6}\right) log_2 \left(\frac{3}{6}\right) = 1.000$$

https://www.xoriant.com/blog/decision-trees-for-classification-a-machine-learning-algorithm

Decision Treeの例

残りの属性の IG を計算する

Day	Outlook	Temperature	Humidity	Wind	Play Golf
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$IG(S, Wind) = 0.048$$

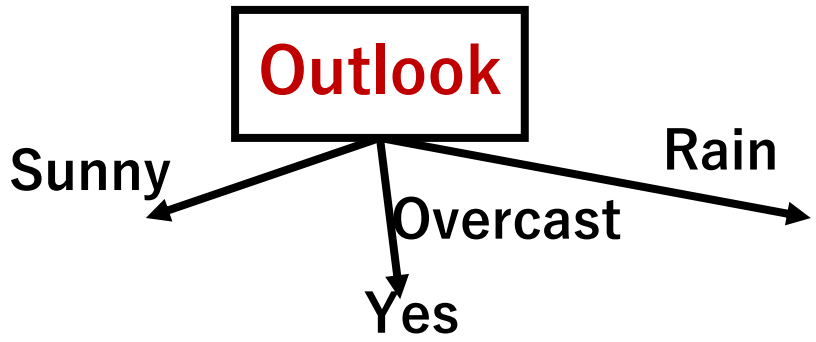
$$IG(S, Outlook) = 0.246$$

$$IG(S, Temperature) = 0.029$$

$$IG(S, Humidity) = 0.151$$

最大

ルートノードとして“Outlook”を選択



https://www.xoriant.com/blog/decision-trees-for-classification-a-machine-learning-algorithm

Day	Outlook	Temperature	Humidity	Wind	Play Golf
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Table where the value of Outlood is Sunny

Temperature	Humidity	Wind	Play Golf
Hot	High	Weak	No
Hot	High	Strong	No
Mild	High	Weak	No
Cool	Normal	Weak	Yes
Mild	Normal	Strong	Yes

“Sunny”から、“Sunny”に関する残りのすべての属性（Humidity, Temperature 、 Wind）のエントロピーと IG を計算

$$H(S_{sunny}) == -\left(\frac{3}{5}\right) log_2 \left(\frac{3}{5}\right) - \left(\frac{2}{5}\right) log_2 \left(\frac{2}{5}\right)$$
$$= 0.96$$

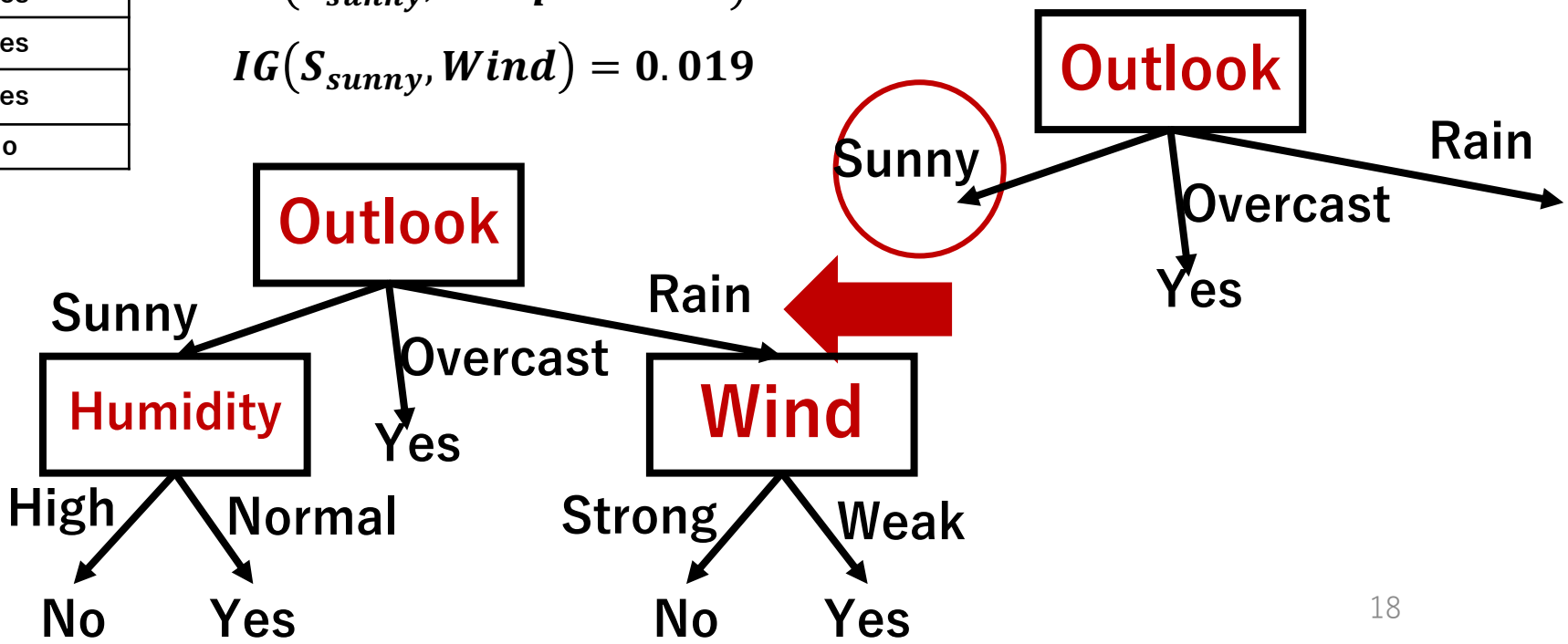
Biggest

$$IG(S_{sunny}, Humidity) = 0.96$$

Partition Humidity attribute

$$IG(S_{sunny}, Temperature) = 0.57$$

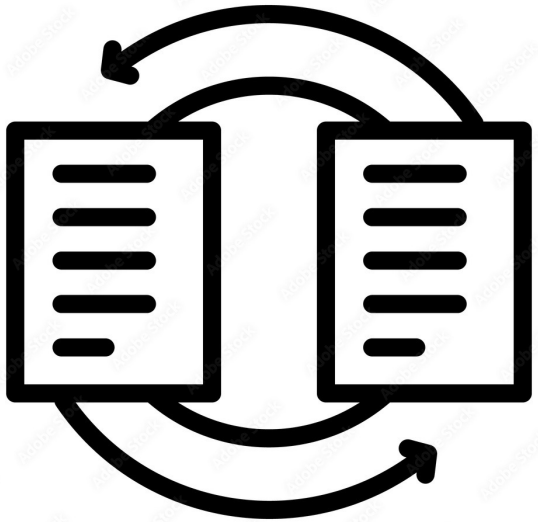
$$IG(S_{sunny}, Wind) = 0.019$$



データストリームでの分類

- 問題は?

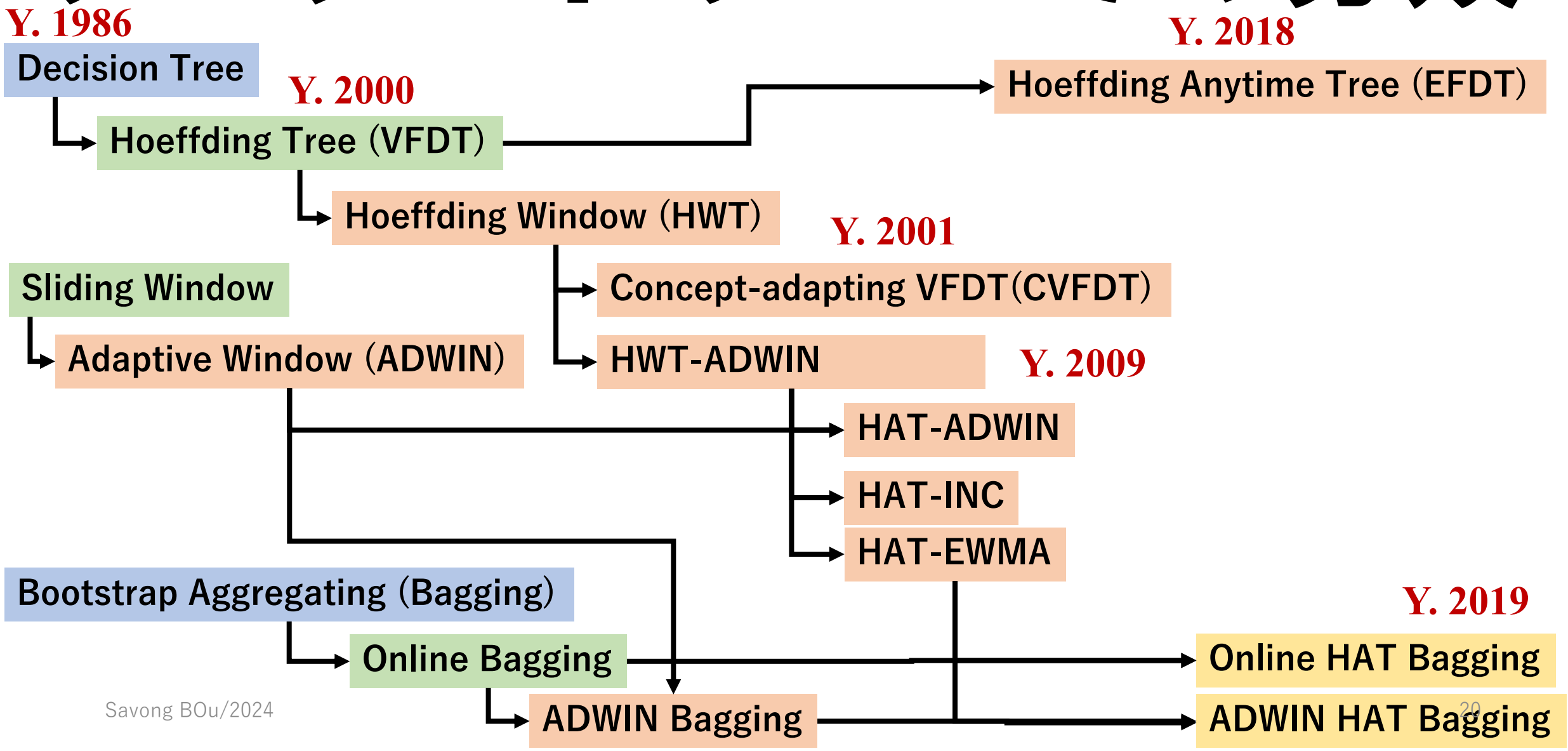
すべてのデータを保存することは不可能



複数回のスキャンは不可能

時間とともに変化、コンセプトのドリフト

データストリームでの分類



Hoeffding Tree Algorithm

- 最適な分割属性を選択するには、小さなサンプルで十分
- **Hoeffding bound**を使用

$$\varepsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}}$$

r: random variable representing the attribute selection method

R: range of r

n: # independent observations

δ: user-defined parameter

Hoeffding Tree Algorithm

- **Input:**
  - **S: sequence of examples**
  - **X: attributes**
  - **G(): evaluation function, i.e., Information Gain**
  - **δ: desired accuracy**

For each example in S

Retrieve $G(X_a)$ and $G(X_b)$ //two highest $G(X_i)$

if $(G(X_a) - G(X_b) > \varepsilon)$

split on X_a

recurse to next node

break

Hoeffding Tree Algorithm

Hoeffding bound

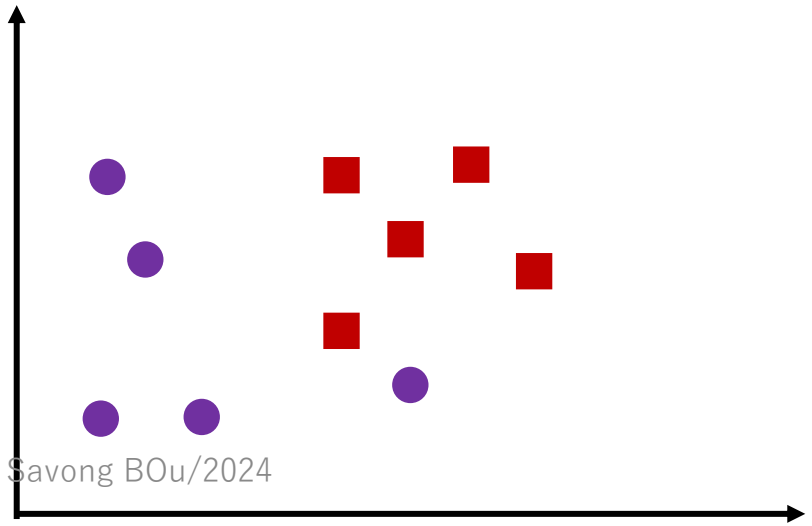
$$G(X_a) - G(X_b) > \varepsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}}$$

- １つのリーフ ノードから開始
- 受信した記録を蓄積

Hoeffding Tree Algorithm

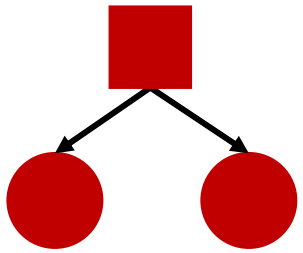
Hoeffding bound

$$G(X_a) - G(X_b) > \varepsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}}$$



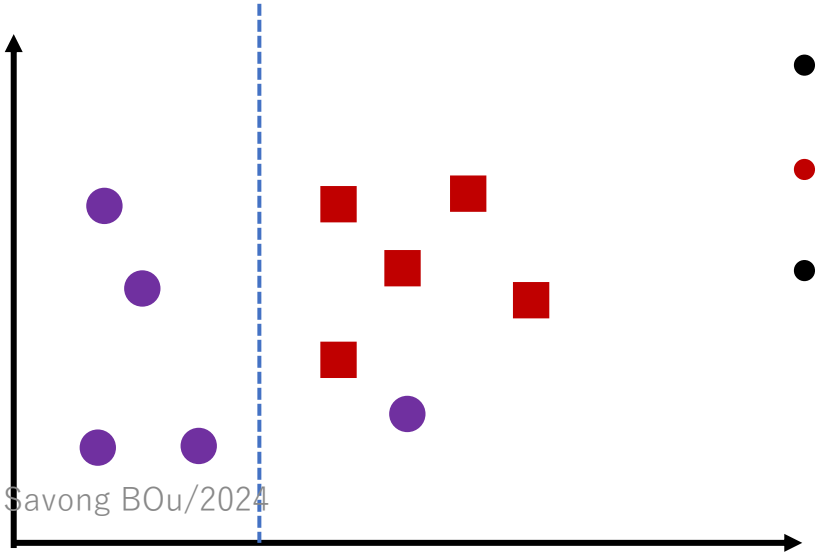
- １つのリーフ ノードから開始
- 受信した記録を蓄積

Hoeffding Tree Algorithm



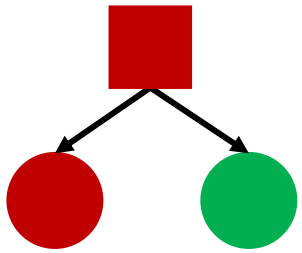
Hoeffding bound

$$G(X_a) - G(X_b) > \varepsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}}$$



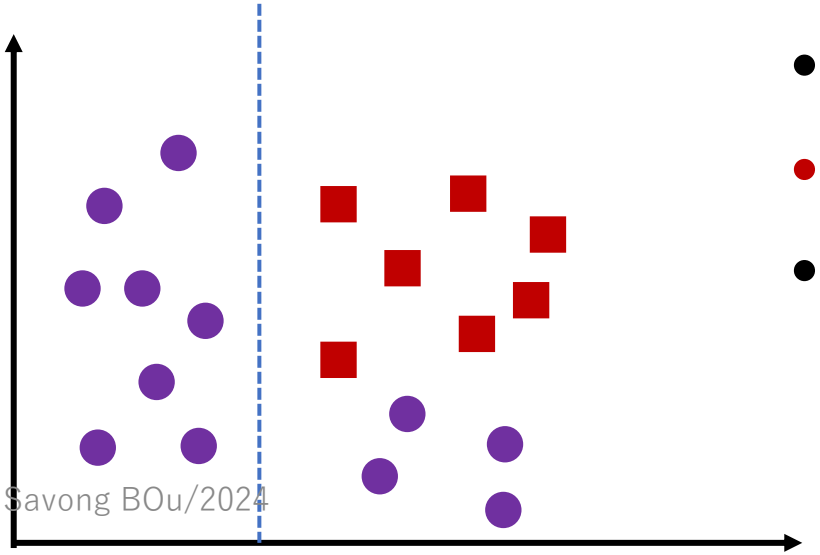
- 十分な記録を受け取った後
- Hoeffding boundは満たされ
- 最適な属性ごとにリーフ ノードを分割

Hoeffding Tree Algorithm



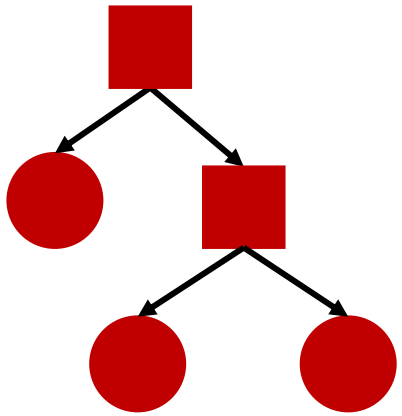
Hoeffding bound

$$G(X_a) - G(X_b) > \varepsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}}$$



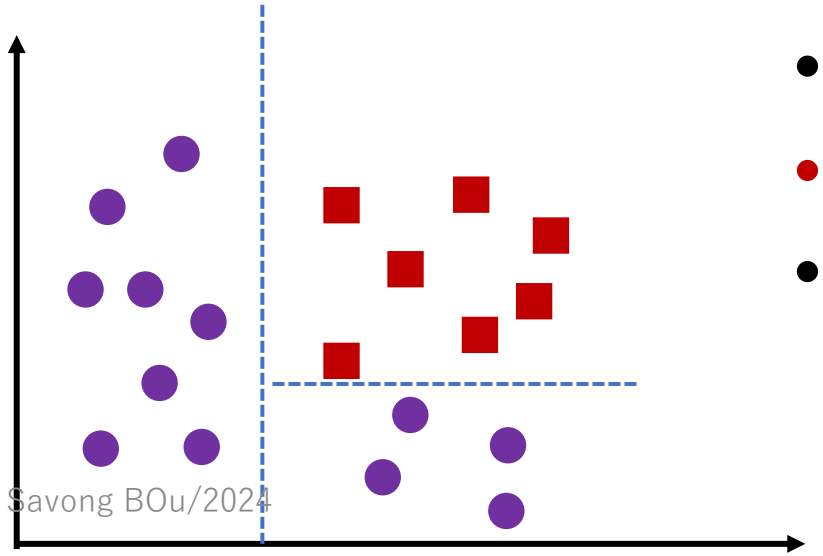
- 十分な記録を受け取った後
- Hoeffding boundは満たされ
- 最適な属性ごとにリーフ ノードを分割

Hoeffding Tree Algorithm



Hoeffding bound

$$G(X_a) - G(X_b) > \varepsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}}$$



- 十分な記録を受け取った後
- Hoeffding boundは満たされ
- 最適な属性ごとにリーフ ノードを分割

Hoeffding Tree Algorithm

- **Computational Complexity:**

$$O(ldvc)$$

l : maximum depth
d: # of attributes
v: maximum # of values for any attribute
c: # of class

Hoeffding Tree Algorithm

- **利点:**
  - 従来の方法よりも拡張性が高い
    - サンプリングによるサブリニア
    - メモリ使用量が非常に少ない
  - 増分
    - クラス予測を並行して行う
    - 新しい例は随時追加され
- **弱点:**
  - 限界を満たすには多くの時間
  - ツリー展開で使用されるメモリ
  - 候補属性の数

P. Domingos and G. Hulten, “Mining high-speed data streams,” KDD 2000

VFDT (Very Fast Decision Tree)

- **Hoeffding Treeを改善:**
  - Near-tiesをより積極的に破る
  - G() は、ユーザー定義のサンプル数の後に計算
  - 非アクティブなリーフを非アクティブ化
  - 有望でない初期の属性を削除
  - 従来の学習者によるブートストラップ Hoeffding Tree よりも優れた時間とメモリ
- **従来のDecision treeを比較**
  - 同様の精度
  - 161万例による実行時間の向上
    - VFDT の場合は 21 分
    - 従来のdecision treeでは 24 時間
- コンセプトのドリフトを処理できない

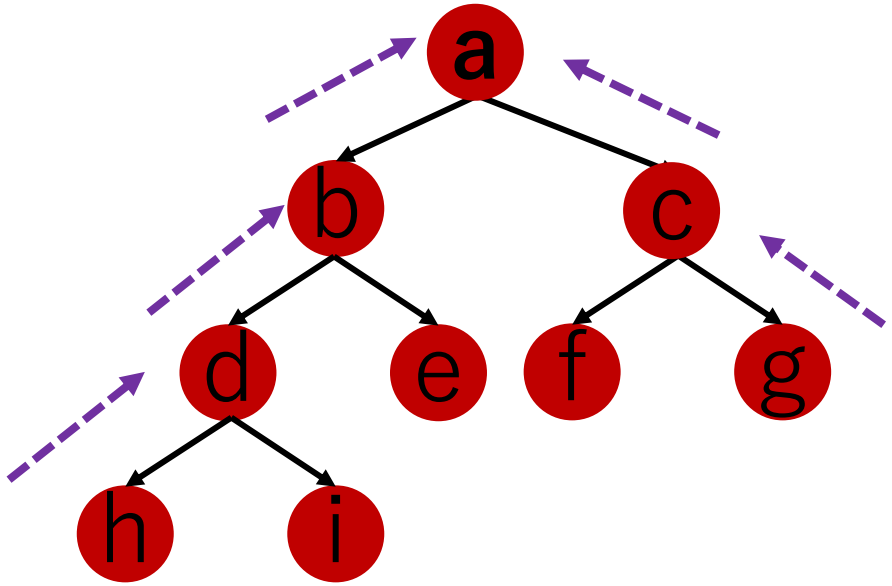
P. Domingos and G. Hulten, “Mining high-speed data streams,” KDD 2000

CVFDT (Concept-adapted VFDT)

- **コンセプトドリフト:**
  - 時間とともに変化するデータ ストリーム
  - 新しいものを取り入れ、古いものを排除
- **CVFDT:**
  - 新しい例の数を増やす
  - 代替のサブツリーを成長させ、適切な場合は古いサブツリーを置き換え
    - T_0 **examples** =>すべてのノードを走査し、現在の属性が最適な分割ではなくなった場合は代替ツリーを作成
    - T_1 **examples** =>より良い属性で代替ツリーを成長させ
    - T_2 **examples** =>代替ツリーをテスト.代替ツリーの方が正確な場合は、古いツリーを代替ツリーに置き換え.

G. Hulten, L. Spencer, and P. Domingos, “Mining time-changing data streams,” KDD2001

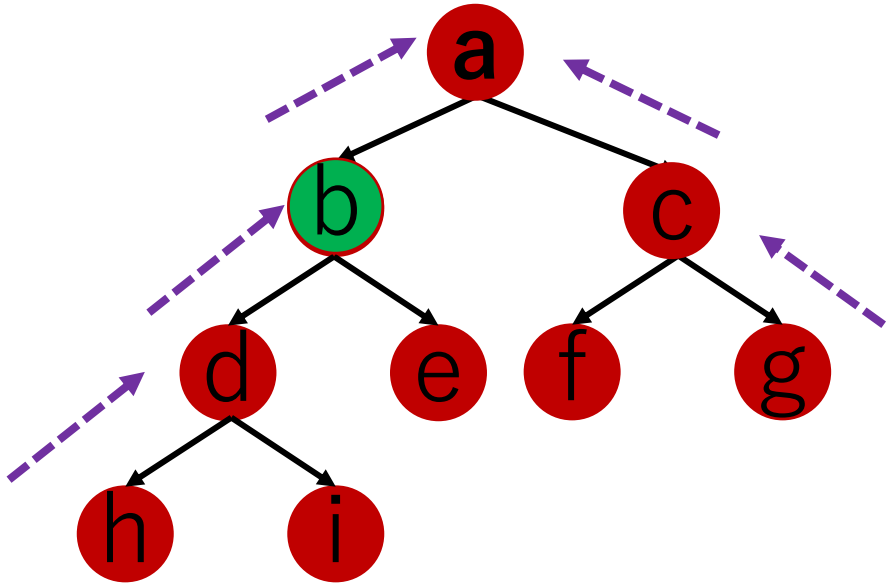
CVFDT (Concept-adapted VFDT)



- T_0サンプルを受信した後、すべてのノードを走査
- 現在の属性が最適ではなくなった場合 (i.e., “b”)、新しい最適な属性 (i.e., “d”) をルートとする代替ツリーを成長させ

G. Hulten, L. Spencer, and P. Domingos, “Mining time-changing data streams,” KDD2001

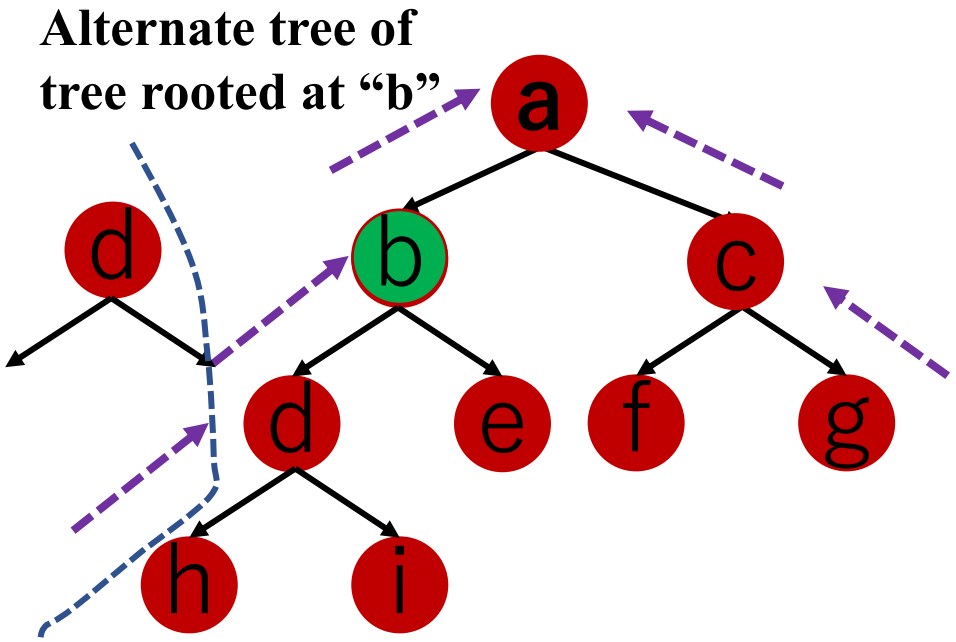
CVFDT (Concept-adapted VFDT)



- T_0サンプルを受信した後、すべてのノードを走査
- 現在の属性が最適ではなくなった場合 (i.e., “b”)、新しい最適な属性 (i.e., “d”) をルートとする代替ツリーを成長させ

G. Hulten, L. Spencer, and P. Domingos, “Mining time-changing data streams,” KDD2001

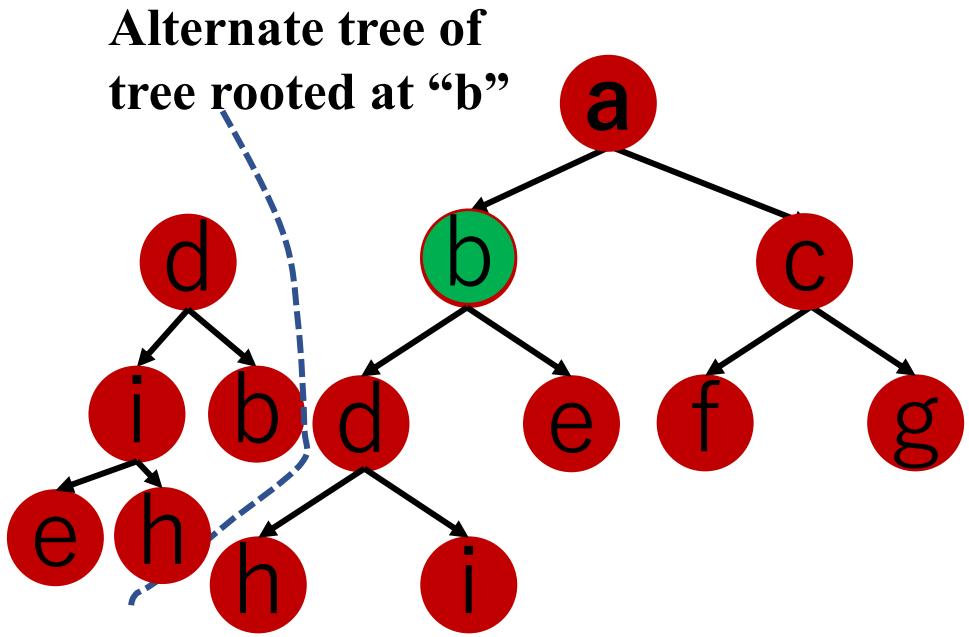
CVFDT (Concept-adapted VFDT)



- T_0サンプルを受信した後、すべてのノードを走査
- 現在の属性が最適ではなくなった場合 (i.e., “b”)、新しい最適な属性 (i.e., “d”) をルートとする代替ツリーを成長させ

G. Hulten, L. Spencer, and P. Domingos, “Mining time-changing data streams,” KDD2001

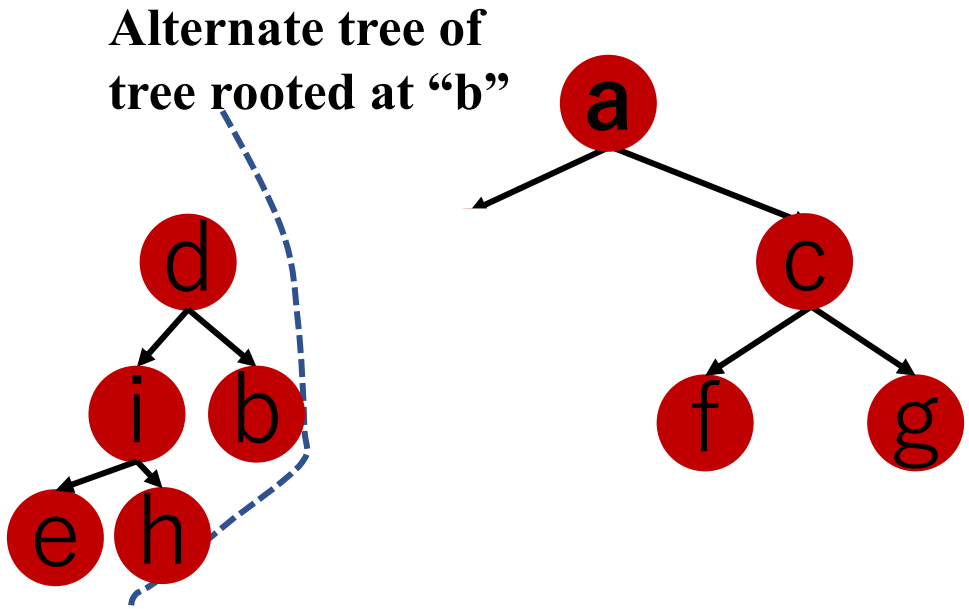
CVFDT (Concept-adapted VFDT)



- T_1の例を受け取った後、
- 代替ツリーを成長させます（i.e., root at “d”）

G. Hulten, L. Spencer, and P. Domingos, “Mining time-changing data streams,” KDD2001

CVFDT (Concept-adapted VFDT)



- T_2の例を受け取った後、
- 代替ツリーの方が正しい場合は、現在のツリーを代替ツリーに置き換え
  - “b”をルートとするサブツリーを“d”をルートとする代替ツリーに置き換え

G. Hulten, L. Spencer, and P. Domingos, “Mining time-changing data streams,” KDD2001

Ensemble of Classifiers Algorithm

- 古い例を破棄する代わりに、最も精度の低い分類子が破棄され

Train K classifiers from K chunks
For each subsequent chunk
    train a new classifier
    test other classifiers against the chunk
    assign weight to each classifier
    select top K classifiers

H. Wang, W. Fan, P. S. Yu, and J. Han, “Mining concept-drifting data streams using ensemble classifiers”, KDD2003

Conclusion

- 分類の概要
- 静的データ分類のレビュー
  - Decision Tree
- データストリームにおける分類
  - Hoeffding Tree
  - VFDT
  - CVFDT
  - Ensemble Method

H. Wang, W. Fan, P. S. Yu, and J. Han, “Mining concept-drifting data streams using ensemble classifiers”, KDD2003