【ICDE2016勉強会】

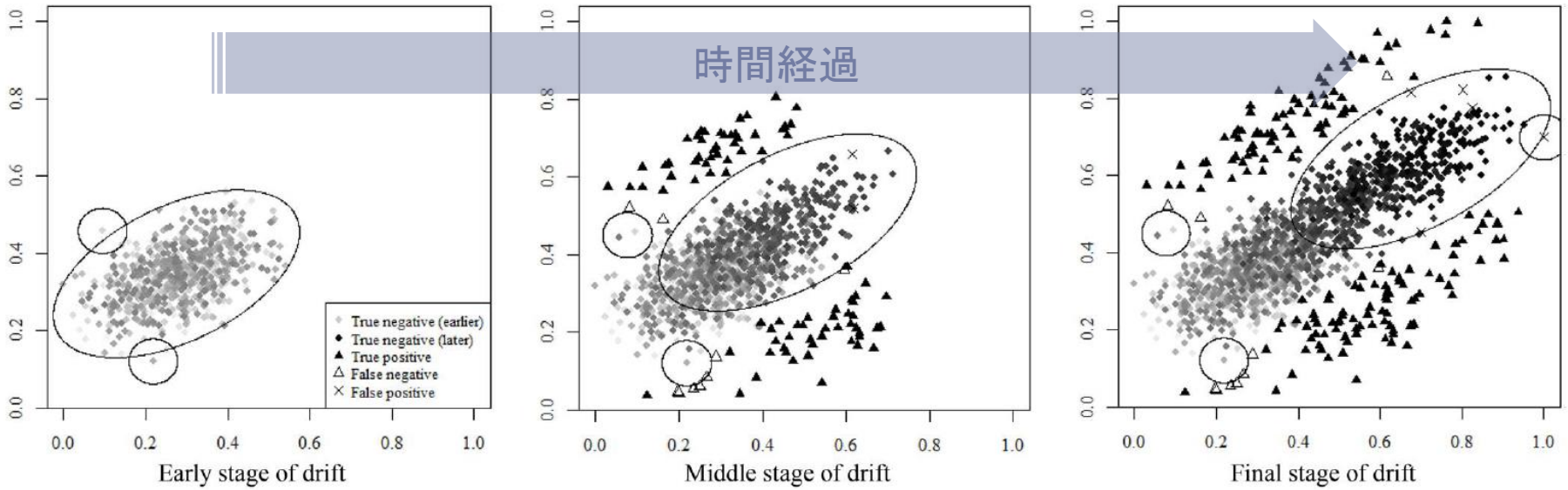
Session C4-1: Efficient Handling of Concept Drift and Concept Evolution over Stream Data

担当：山口晃広（名古屋大学）

明記しない場合，画像は本論文より転用

前置き：Concept drift, concept evolution

- データストリームマイニングの一分野
- Concept drift
  - 学習モデルが時間とともに変化する
  - 徐々に変化する場合や急に変化する場合がある
- Concept evolution
  - 時間が経過してから，新しいクラスが発生する



（別論文より転用）An Incremental Clustering-Based Fault Detection Algorithm for Class-Imbalanced Process Data, IEEE TRANSACTION SEMICONDUCTOR 2015

研究の背景・課題

- 背景
  - データストリームを高速にロバストにクラス分類したい
  - 分野の一例： social networks, online businesses, sensors, military surveillance
- 課題
  - concept drift/evolutionが起きるデータストリームの分類問題では，モデルの定期的な更新が必要
  - 正解ラベルをトレーニングデータに付与するコストが増大
  - 現実的には，新しいクラスが途中で発生する可能性有り
    - 例：侵入検知，テキストの分類，不正利用検知
  - モデルの更新に利用するchunk sizeが固定であったり，忘却による方法では，様々な状況への対応が困難

解決策

- 先の課題は，半教師学習とウインドウベースな著者らのフレームワークSAND※で解決
  - k-nnタイプのモデルから構成されるアンサンブル分類器
    - クラス分類は，ensembleの多数決で決定
  - 入力データからラベル無しで算出可能な信頼度でdriftを検知
  - モデルの学習に必要なラベルは一部（i.e., 半教師学習）
  - Driftだけでなく，新しいクラスが発生するevolutionにも対応
  - Chunk sizeを動的に決定
- しかし，SANDのdriftを検知する計算時間：大 ⇒ 高速化
  - Driftを検知するスコアの計算にDPを適用
  - 信頼度に保証を設けることで，Driftを検知する回数を削減

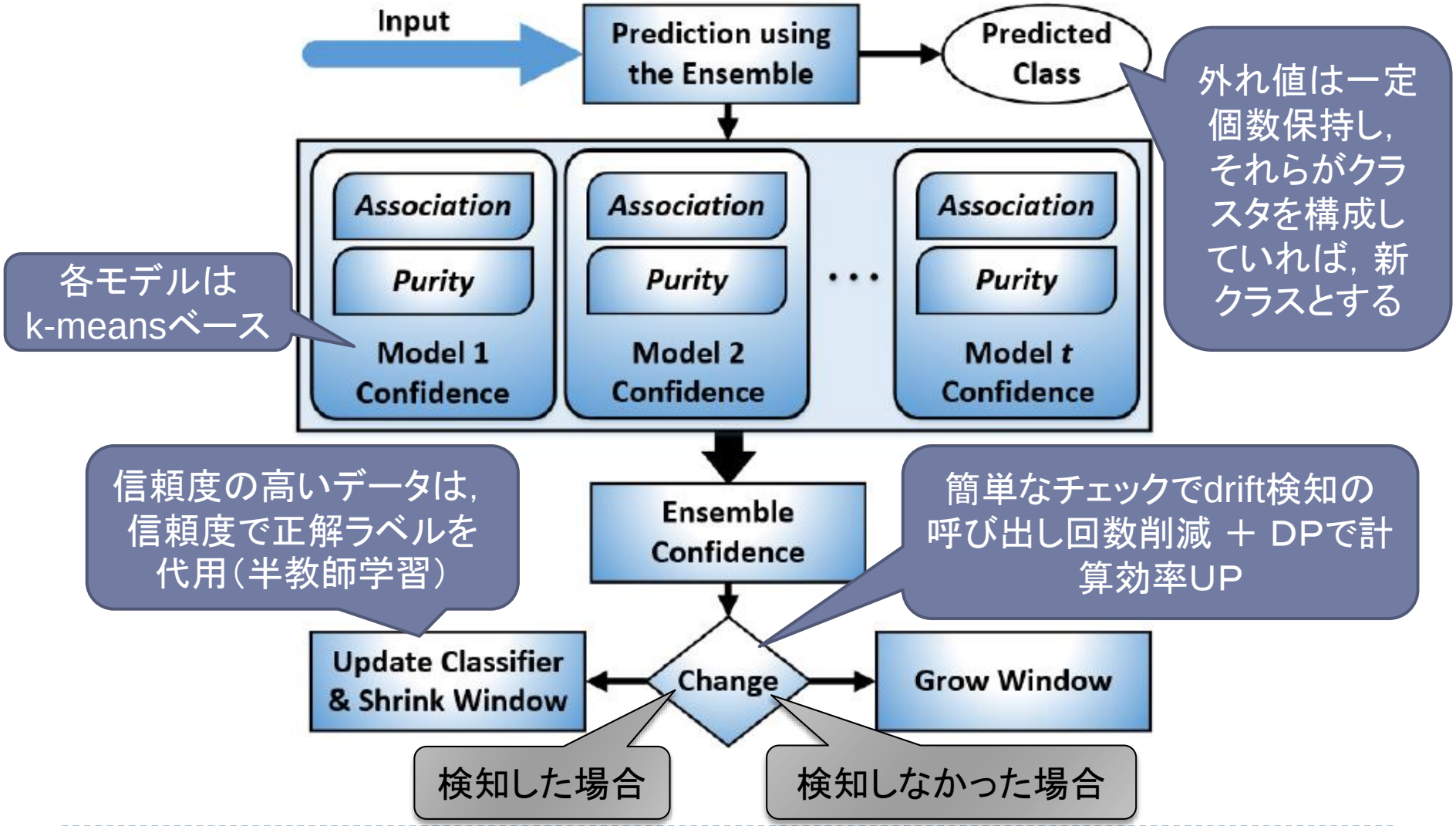
※ A. Haque, L. Khan, and M. Baron, “Sand: Semi-supervised adaptive novel class detection and classification over data stream,” in THIRTIETH AAAI Conference on Artificial Intelligence, Feb 2016.

コントリビューション

1. 他のクラスタリングでも利用可能で，正当性を持つ信頼度の見積もり方法を提供する
   - 正当性とは，信頼度：低下⇒クラスタリングの指標となる論文中の目的関数のコスト：増加
   - つまり，入力データが属するクラスタを探す際に，信頼度の計算のみで，目的関数の直接計算は不要
2. Driftが発生する時，この信頼度が必ず低下することを解析的に示す
   - この事実に基づいて，drift検知の実行時間を削減
3. ラベル付けが必要な入力データは一部だけでよい
   - 著者らの従来研究であるSANDでも既に実現済み
4. 動的にchunkを決定する半教師学習を提供し，DP＋driftの検知が必要なタイミングを選択して時間削減
5. 評価の結果，精度と実行時間が良い

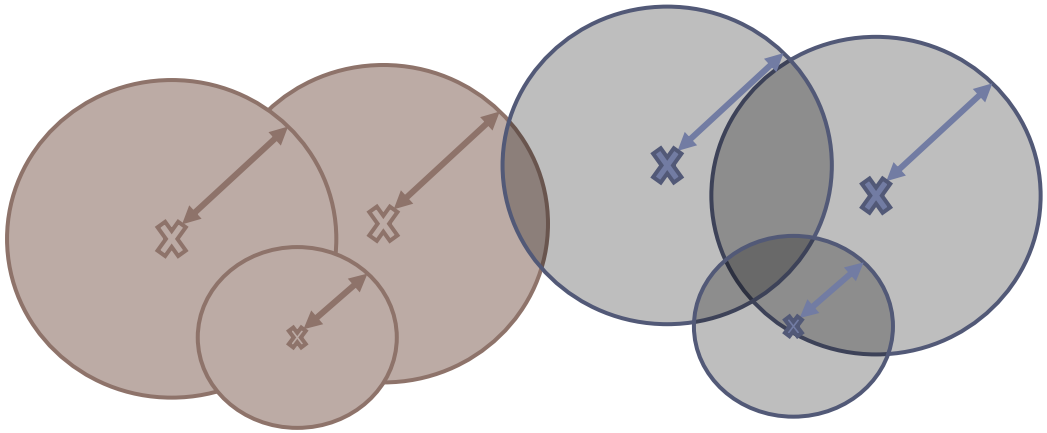
提案方式(ECHO)の概要

※ Efficent Concept Drift and Concept Evolution Handling over Stream Data



分類と学習

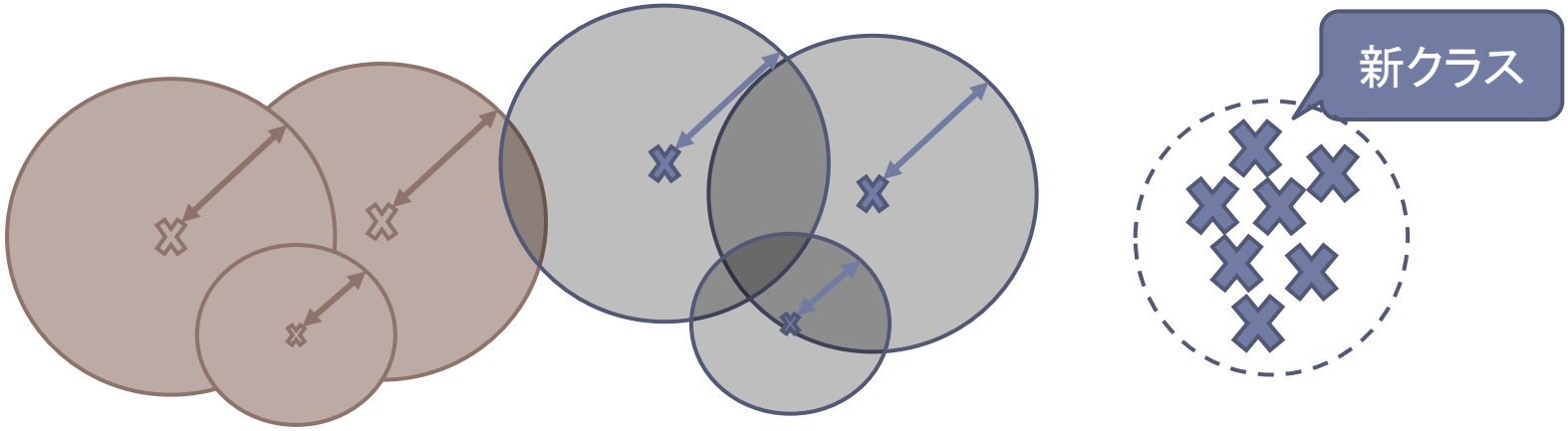
- 各アンサンブルの学習モデル
  - K-meansベース
  - 擬似点（要約情報）のみを管理し，生データを管理しない
    - 中心，半径，頻度，各クラスの頻度
  - 距離: クラスタ内のL2距離にimpurityを加えた目的関数から決定
    - そのクラスタが様々なラベルを持っているとimpureであり，唯一のラベルから構成されるならpure
- クラス分類
  - 各アンサンブルで最近の擬似点で最も頻度の多いクラスを選ぶ
  - アンサンブル全体で多数決



k=3
アンサンブルの数=2

新しく発生したクラスの検知

- 前提：新しいクラスは2個以上発生しない
- アンサンブルモデルの決定境界から外れた入力データを外れ値として，一定個数保持する
- 一定数の外れ値が，お互いに近い（cohesion）が既存クラスとは離れている（separation）場合，新クラスとする
- このメトリクス（q-NSC）は既存研究 ECSMiner※と同様



※ M. M. Masud, J. Gao, L. Khan, J. Han, and B. M. Thuraisingham, Classification and novel class detection in concept-drifting data streams under time constraints, IEEE Trans. Knowl. Data Eng., 2011

信頼度を表すスコア

- SANDと同様に，入力データ(x)に対して2つのヒューリスティックな信頼度を定義

A) Association
  - 値が大きいほど，既存クラスとの関連性が高い
    - 「xと最近の擬似点の半径」-「xとその擬似点の中心とのL2距離」

B) Purity
  - 値が大きいほど，最近の擬似点が複数のクラスを持たない
    - 「xと最近の擬似点で尤もらしいクラスの頻度」÷「その擬似点の全クラスの頻度」

- 各アンサンブルで出した信頼度を正規化・平均してアンサンブル全体の信頼度とする
  - ラベル付けされた結果と予測値との相関をみて，予測があたっているアンサンブルの信頼度を用いる

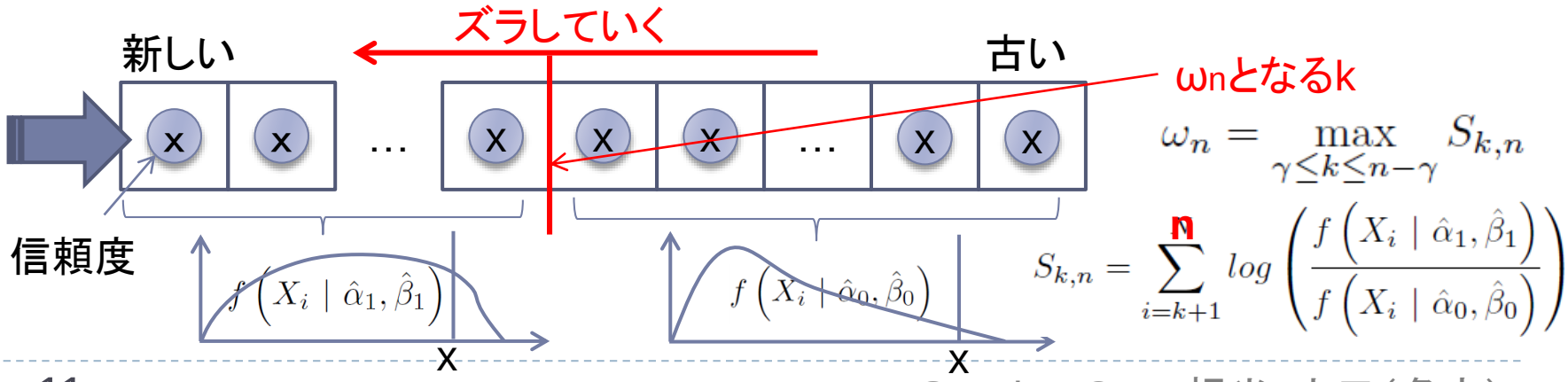
信頼度の正当性確認／driftに対する影響解析

- 信頼度の正当性を確認
  - クラスタリングの目的：以下の重み付き和の最小化
    - クラスタ内の入力データの散らばり（距離：L2）
    - 各クラスタに含まれるラベルの混入度（impurity）
  - Associaion/Purityが増加 ⇒ 目的関数が減少 を確認
    - 入力データの分類では，信頼度が高いクラスタに含めれば良い

- Driftに対する信頼度の影響を解析
  - Driftをすると，Associationが減少することを確認
    - Association = (定数) -「クラスタ内のバラつき」の関係あり 式(9)
    - Driftが起こると，「クラスタ内のバラつき」が増加 式(10,11)

Concept driftの検知

- ウインドウを前後に分割
- 前/後のウインドウに保持された信頼度をβ分布で推定
  - 各β分布の推定には最低でも100個のサンプルを使用
- 分割点をスライドさせて，古い方のβ分布よりも新しい方のβ分布の確率値が十分に大きければ，そこでdriftを検知
- 検知したら，ウインドウのデータを削除しモデルを更新（次頁）
- 以下でも同様に，ウインドウのデータを削除しモデルを更新
  - 全体のconfidenceが低い場合
  - Window sizeが最大サイズを超えた場合



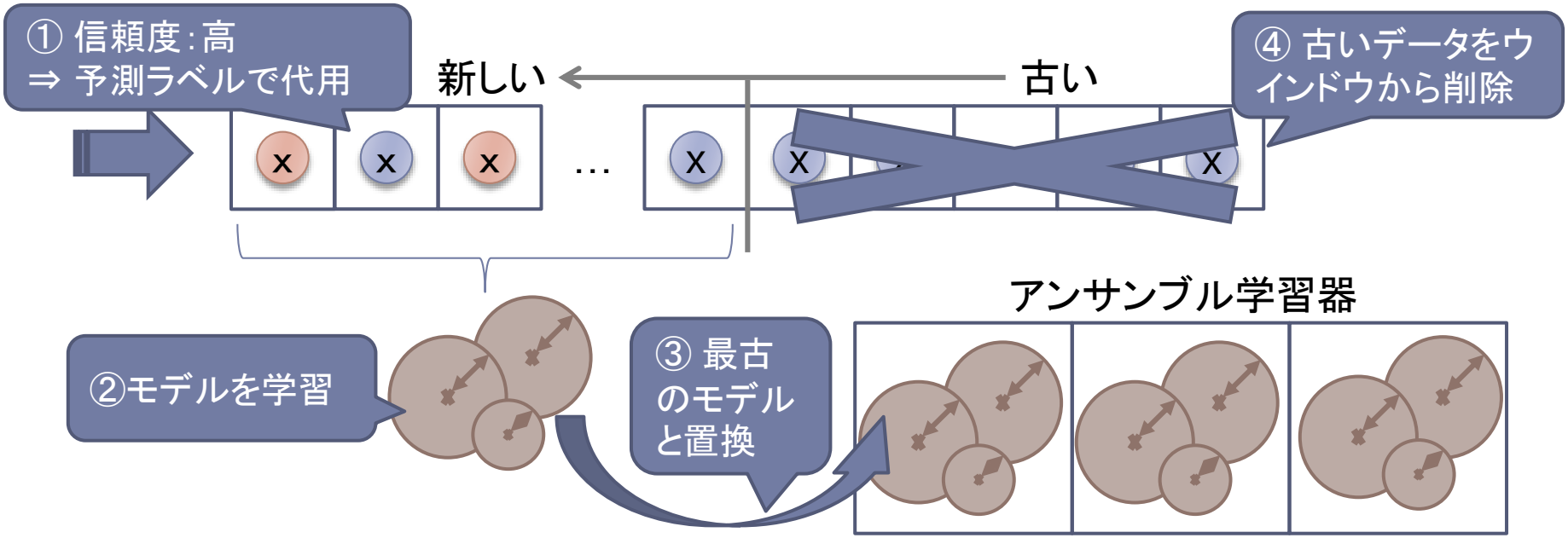
ラベル付け制限有 アンサンブル学習モデルの更新

① 信頼度が閾値よりも低ければ，ラベル付けを要求してそれを使い，高ければ，予測したラベルを使用

② そのラベルとウインドウ内データから，モデルを学習

③ アンサンブルの中で最古のモデルと学習したモデルを置換

④ 変化点から最新のデータだけを含むようウインドウを更新



実行時間の性能改善

- 速度性能のボトルネックとなるdrift検知をA～Cで改善
  - A, Cでは，Driftの発生時に必ず信頼度が低下することを利用
- A. Sporadic Execution
  - ウインドウの前後で信頼度が一定割合以上減少した場合のみ，正確なDrift検知を実行（i.e., β分布の推定/確率値の算出をサボる）
- B. Recursive Calculation（DP適用）
  - 過去に計算した$S_{k,m}$ (m ≦ n)を再利用して$S_{k,n}$を計算

$$S_{k,n} = S_{k,m} + \sum_{i=m+1}^{n} \log \frac{f(x_i|\hat{\alpha}_1,\hat{\beta}_1)}{f(x_i|\hat{\alpha}_0,\hat{\beta}_0)}$$

- C. Selective Execution
  - Drift検知の呼び出し回数自体を削減
    - ECHO-F：入力データの信頼度が一定値以下の場合のみ呼び出す
    - ECHO-D：信頼度cから，呼び出し確率をexp(-c) と定義する

実験方法

- データセット
  - ForestCover（新クラスが発生するように加工有）
  - Physical Activity Monitoring（PAMAP）
  - PowerSupply

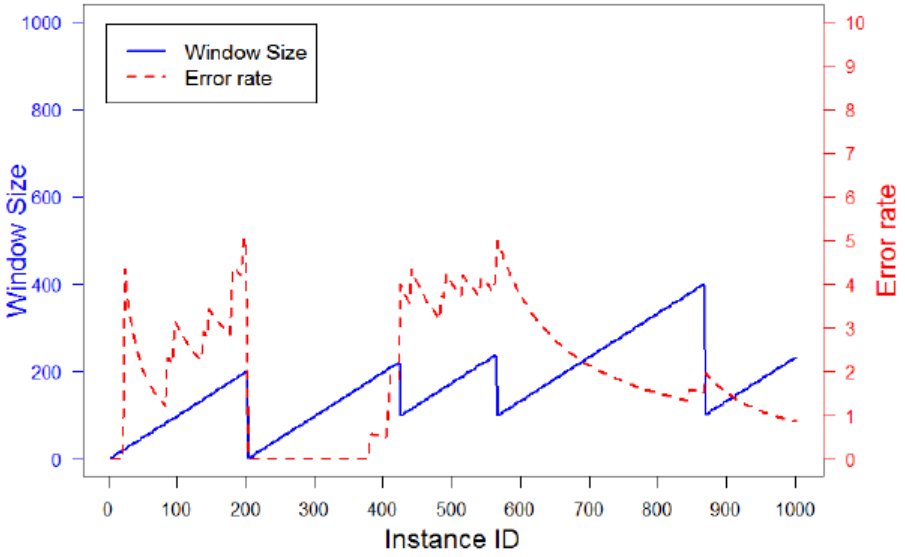
  オープンデータ（UCI）
  - HyperPlane（式より生成）
  - SynRBF@0.002，SynRBF@0.003（MOAより生成）

  人工データ
- 評価方法
  - 提案方式（ECHO-F,D）は，ECSMinerに実装
    - 擬似点50個，アンサンブルのサイズ6，他は交差検定
  - 比較方式（2種類）は，MOAに実装
    - OzaBagAdwin（OBA），Adaptive Hoeffding Tree（AHT）
    - いずれも，driftの方法は，ADWINベース
    - いずれも，新クラスの発生には非対応なため，分類精度のみ比較
    - 提案方法と異なり，100%のラベル付け

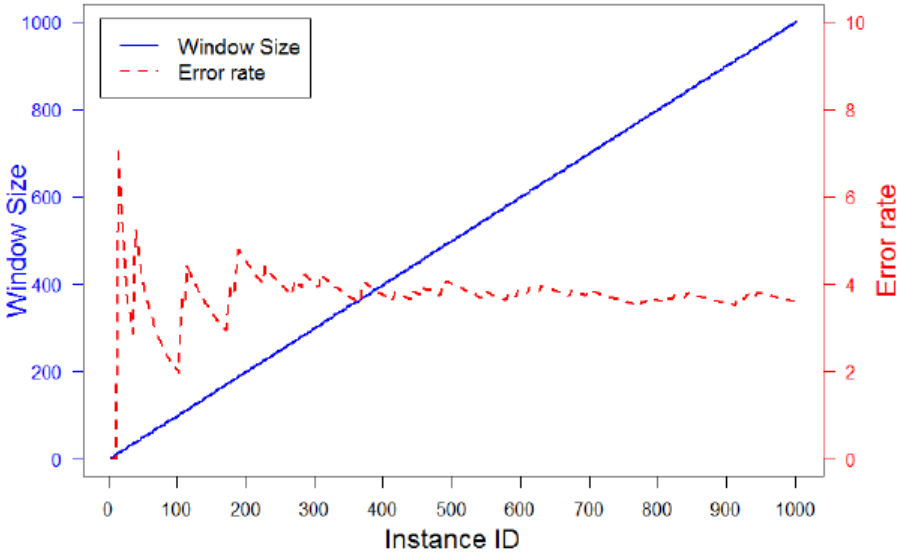
評価：分類精度（ウインドウサイズとエラー率）

- 図（a）より，エラー率※が増加すると，提案方式ではdriftを検知し，モデルを更新するので，エラー率は下がる
- 図（b）のデータでは，concept driftは起こらず，提案方式ではウインドウは成長し続ける

※エラー率: 全データのうち分類をミスした割合



(a) SynRBF@0.002



(b) HyperPlane

評価：分類精度（比較方式との精度比較）

- 信頼度に対するラベル付けの閾値が高いとき，提案方式の結果が大幅に良好
- なお，PowerSupplyデータでは，殆どの入力データの信頼度が高かった

TABLE III: Summary of classification results

Name of Data Set	ECHO-F ($\tau = 0.9$)		ECHO-D ($\tau = 0.9$)		ECSMiner	AHT	OBA
	Error%	% of labeled data	Error%	% of labeled data	Error%	Error%	Error%
ForestCover	3.95	96.54	**3.68**	95.16	4.55	22.89	18.06
PAMAP	4.75	93.64	**3.73**	94.7	35.26	8.76	7.27
Power Supply	0.01	8.93	**0.01**	9.8	0.01	85.59	86.92
HyperPlane	2.79	99.76	**2.18**	100	3.73	46.24	48.55
SynRBF@0.002	**23.98**	99.79	34.01	99.93	63.43	38.75	37.04
SynRBF@0.003	**22.37**	99.51	36.91	99.8	65.39	48.65	46.86

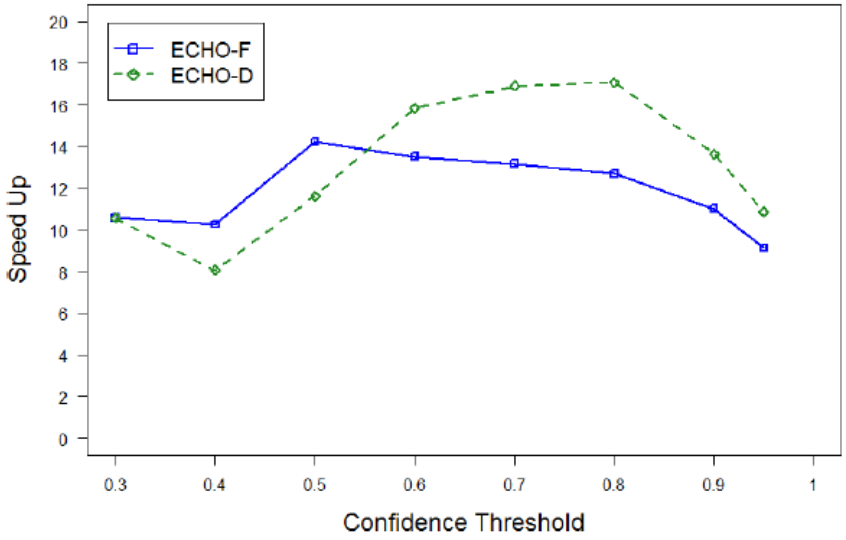
- ラベル付けが少なくても，比較方式と良い勝負
- ECHOはラベル付けが高価な場合でも利用可能

TABLE IV: Comparison of classification performance using limited amount of labeled data

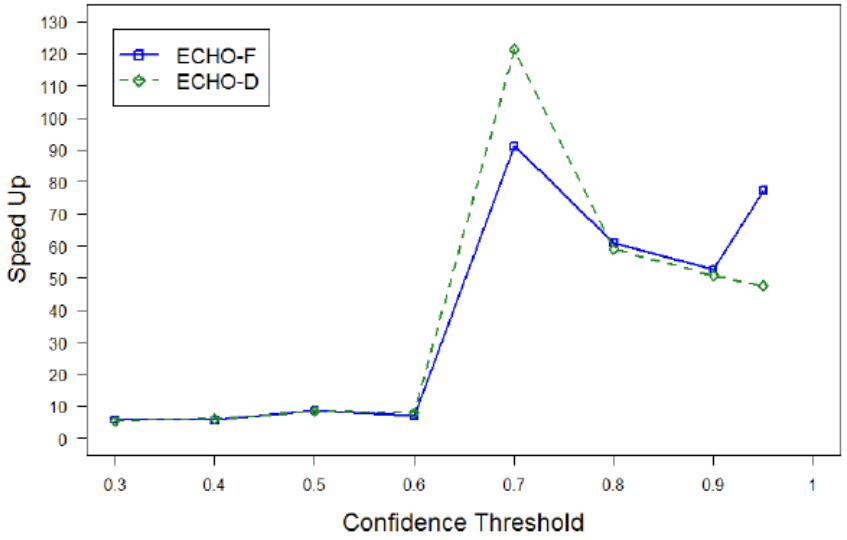
Name of Data Set	ECHO-F ($\tau = 0.4$)		ECHO-D ($\tau = 0.4$)		ECSMiner	AHT	OBA
	Error%	% of labeled data	Error%	% of labeled data	Error%	Error%	Error%
ForestCover	7.96	39.26	**3.88**	35.33	4.55	22.89	18.06
PAMAP	4.77	69.56	**4.73**	68.62	35.26	8.76	7.27
Power Supply	0.01	0.1	**0.01**	0.14	0.05	85.59	86.92
HyperPlane	2.99	32.6	**2.36**	26.9	3.73	46.24	48.55
SynRBF@0.002	53.65	38.4	44.99	28.17	63.43	38.75	**37.04**
SynRBF@0.003	50.51	36.51	**40.65**	54.48	65.39	48.65	46.86

評価：高速化

- 実行時間の性能改善をしないECHOとの速度比で比較
- SynRBF@Xのデータでラベル付けの閾値を変えて測定
  - X=0.002より0.003の方がdriftの頻度が高い
- SynRBF@0.002では18倍，SynRBF@0.003では120倍，それぞれ最大時に高速化



(a) SynRBF@0.002



(b) SynRBF@0.003

Fig. 4: Confidence threshold (τ) vs Speed Up

評価：新クラス発生時の精度

- 新クラス発生に対応するECSMinerと比較
- 最大400インスタンスまで保持して，新クラスを検知
- 新クラスが発生するForestCover, PAMAPデータを使用
- 結果は，ECSMinerと良い勝負

Data set	Method	M_{new}	F_{new}	F_2
ForestCover	ECHO-F	11.37	2.27	0.72
	ECHO-D	14.51	**2.11**	0.71
	ECSMiner	**8.42**	2.13	**0.88**
PAMAP	ECHO-F	0.05	4.22	0.78
	ECHO-D	**0.05**	**3.39**	**0.82**
	ECSMiner	0.05	37.53	0.45

- 全体評価では，ECHOの性能が明らかに良い

まとめ

- Evolvingなストリームを分類する半教師学習フレームワークECHOを提案
- Chunkの境界を動的に変えてconcept driftを検知
- 限られたラベル付きデータのみで分類器を更新するために，信頼度を使用
- 信頼度について理論的に解析，driftへの影響を正当化
- 幾つかの高速化を提案
- 実験で有効性を確認