

機械学習

2-7

工学院情報通信系

中原 啓貴

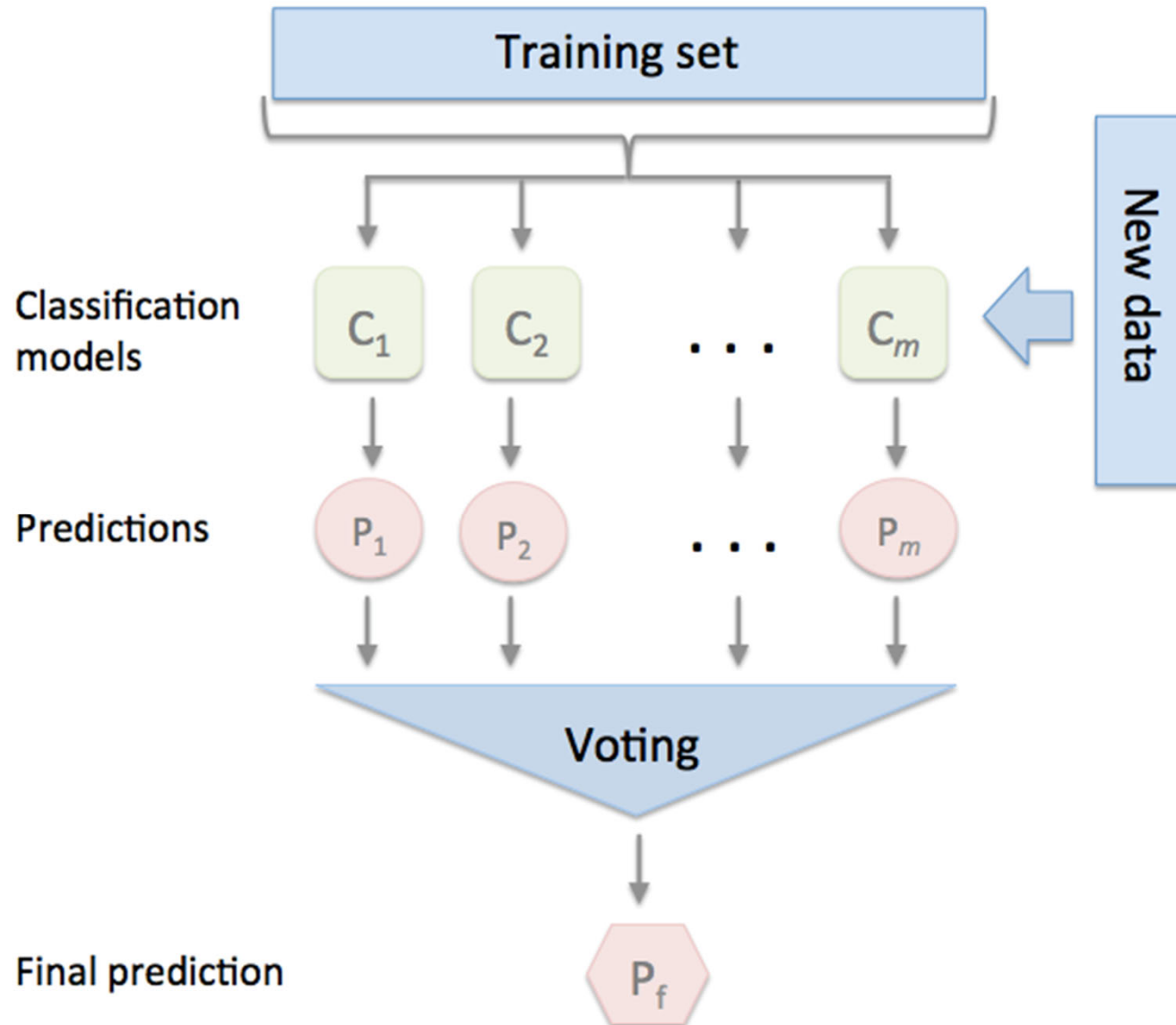
今日の内容

- アンサンブル学習
 - 多数決に基づく予測
 - トレーニングセットの抽出法
 - 弱学習器による強力なモデルの構築
- バギング
- アダブースト
- + 上記内容の演習

アンサンブル学習とは

- 様々な分類器（回帰も可能）を1つのメタ分類器として組み合わせる
- 各分類器よりも高い汎化性能
 - 3人寄れば文殊の知恵、4人集まれば烏合の衆
- ランダムフォレスト、ブースティング
- 機械学習のコンペティションで主流の手法

アンサンブル学習のイメージ



多数決の原理

- 2値分類問題：多数決(majority voting)
- 多クラス分類問題：相対多数決(plurality voting)
- 平均値をとれば回帰問題にも適用可能
- 実装方法
 - 分類器の組み合わせ＋多数決器
 - 異なる分類器の組み合わせ
 - 同じ分類器に異なるデータセットで学習

アンサンブル法で期待できる効果

- (興味があれば板書を)
- 前提条件として
 1. 各分類器が独立していること (誤分類率に相関がない)
 2. 分類器の個数が奇数
 3. かつ、半分以上の分類器の誤認識率が0.5未満
- このとき個々の分類器よりも誤分類率が低くなる (演習で確認)

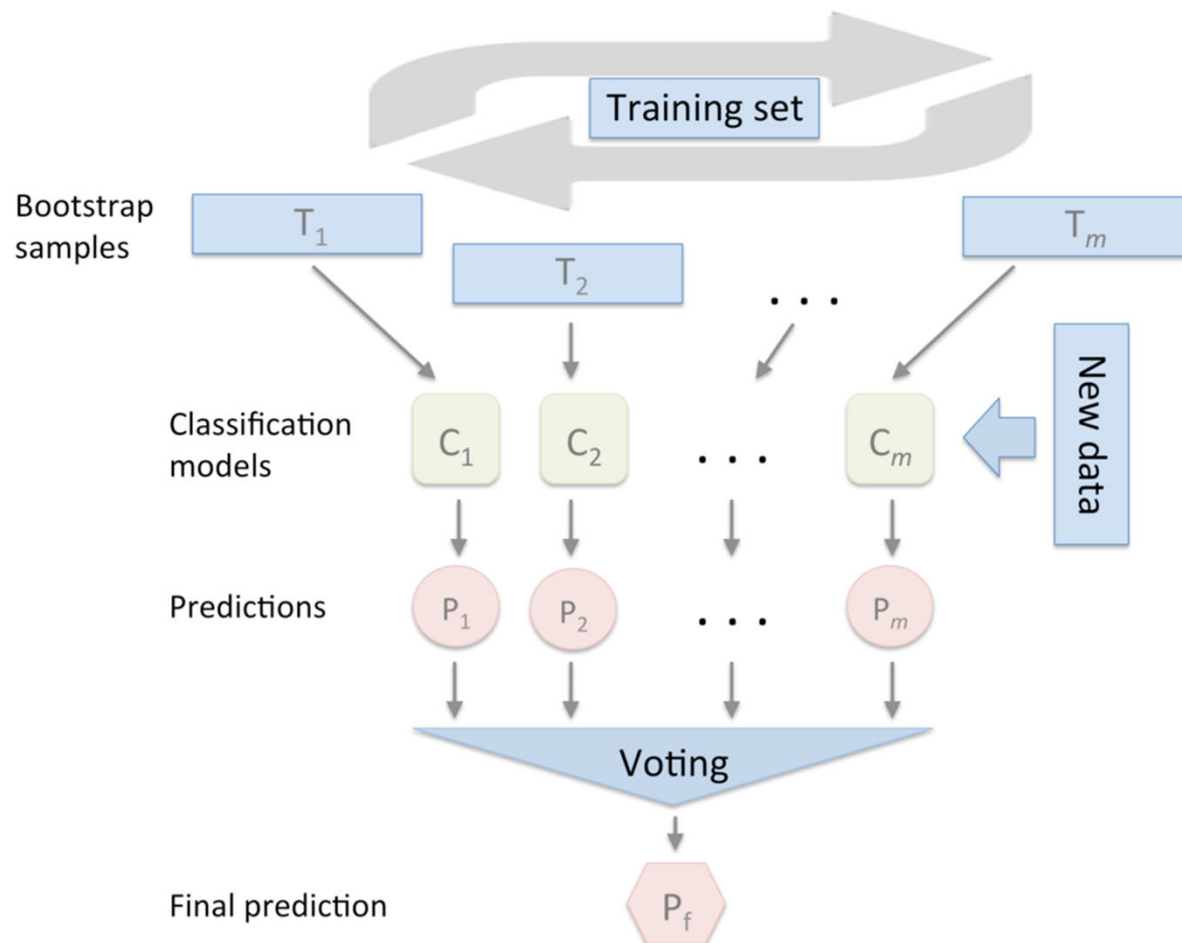
多数決方式の分類アルゴリズム

- 3種類の分類器をアンサンブル
 - ロジスティック回帰分類器
 - 決定木分類器
 - k-means分類器
- Scikit-learnのMajorityVotingClassifierで簡単に実装可能
(演習で確認)

バギング

- ブートストラップ標本を使った分類アンサンブル
- Bootstrap AGGREGatING (→BAGGING)
- 個々の分類器に同じ訓練データを用いず、ブートストラップ標本 (ランダムな復元抽出) を用いる
- 分類器として決定木が用いられることが多い
- 予測精度の向上と過学習の抑制が可能
 - 詳しくは L. Breiman, "Gagging predictors," Machine learning, No. 24, Vol. 2, pp.123-140, 1996.

バギングのイメージ



Sample indices	Bagging round 1	Bagging round 2	...
1	2	7	...
2	2	3	...
3	1	2	...
4	3	1	...
5	7	1	...
6	2	7	...
7	4	7	...

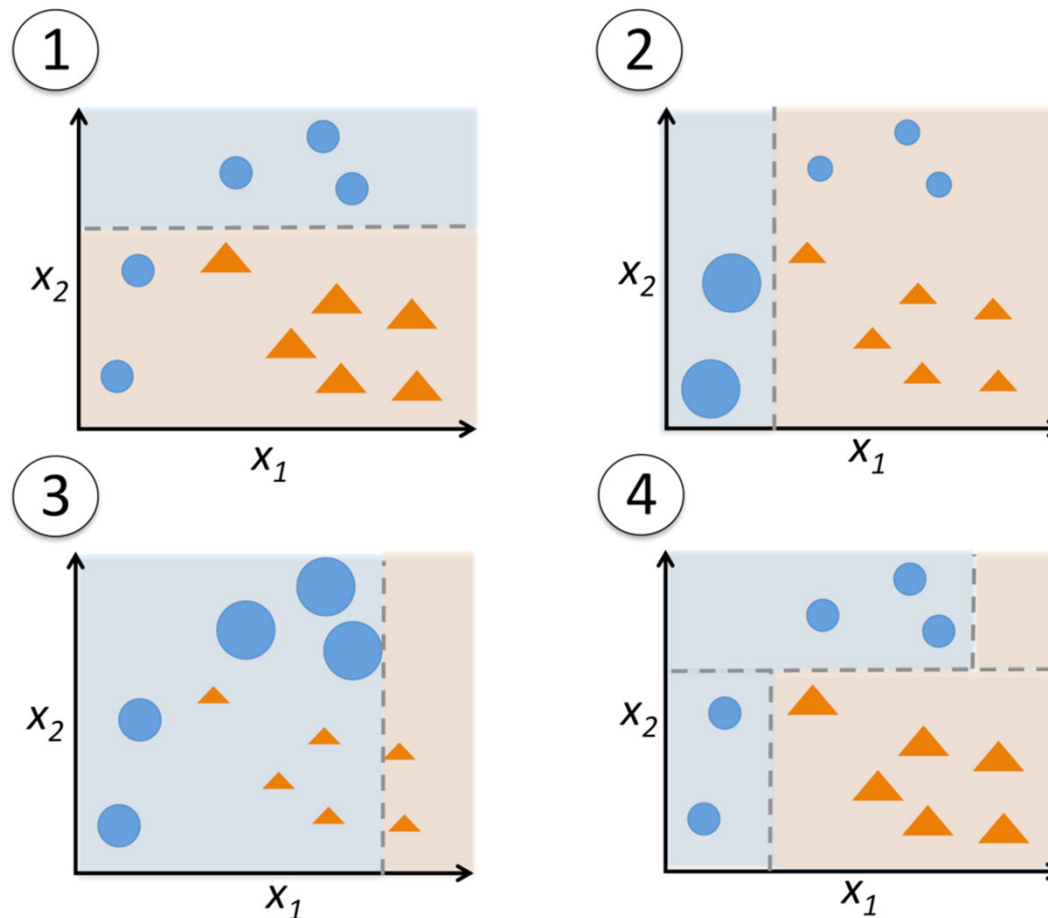
$\downarrow$ C_1 $\downarrow$ C_2 $\downarrow$ C_m

アダブースト (Adaptive Boosting: AdaBoost)

- ブースティングの一般的な実装方法
- 弱学習器のアンサンブルで構成
 - 決定株（深さ1の決定木）など
- 誤分類された訓練データをあとから学習させる
 - 訓練データからランダムに非復元抽出する
- バギングと比べてバイアスが低く、バリエーションが高い（過学習しがち…）

アダブーストの概要

- 誤判定されたサンプルに重みを付けて繰り返し学習



アダブーストのアルゴリズム

1. トレーニングサンプルの重みベクトル w を設定する
2. 全体で m 回実行するブースティングのうち j 回目のステップで3~8を繰り返す
3. 重み付けされた弱学習器をトレーニング
4. クラスラベルの予測
5. 重み付けされた誤分類率を計算
6. 重みの更新に用いる係数の計算
7. 重みの更新
8. 重みを正規化して合計が1になるようにする
9. 入力された特徴行列に対する最終予測を計算する

演習

- 多数決方式の効果を確認
- 単純な多数決分類器
- バギングによる分類
- アダブーストによる分類

まとめ

- **アンサンブル学習**
 - 多数決に基づく予測
 - トレーニングセットの抽出法
 - 弱学習器による強力なモデルの構築
- **一見協力に見えて重大な欠陥が**
 - 性能（認識精度）向上に比べて計算コストが増加

レポート課題

- どんな方法を用いてもよい
- 評価はスコア（認識精度）
- 提出点＋上位10名に順位に応じた加点
- 課題 1, 課題 2 それぞれ分類器を設計すること
- Jupyter notebookとスコア（認識精度）を Googleフォームに提出（ただし 1 度限り）
- 詳細はgithubのreport内にあるREADME.txt
- 提出〆切: 2018年11月30日(JST 23:59)