Data Mining 2025

Classification III

Dept. of Computer Science and Information Engineering

National Cheng Kung University

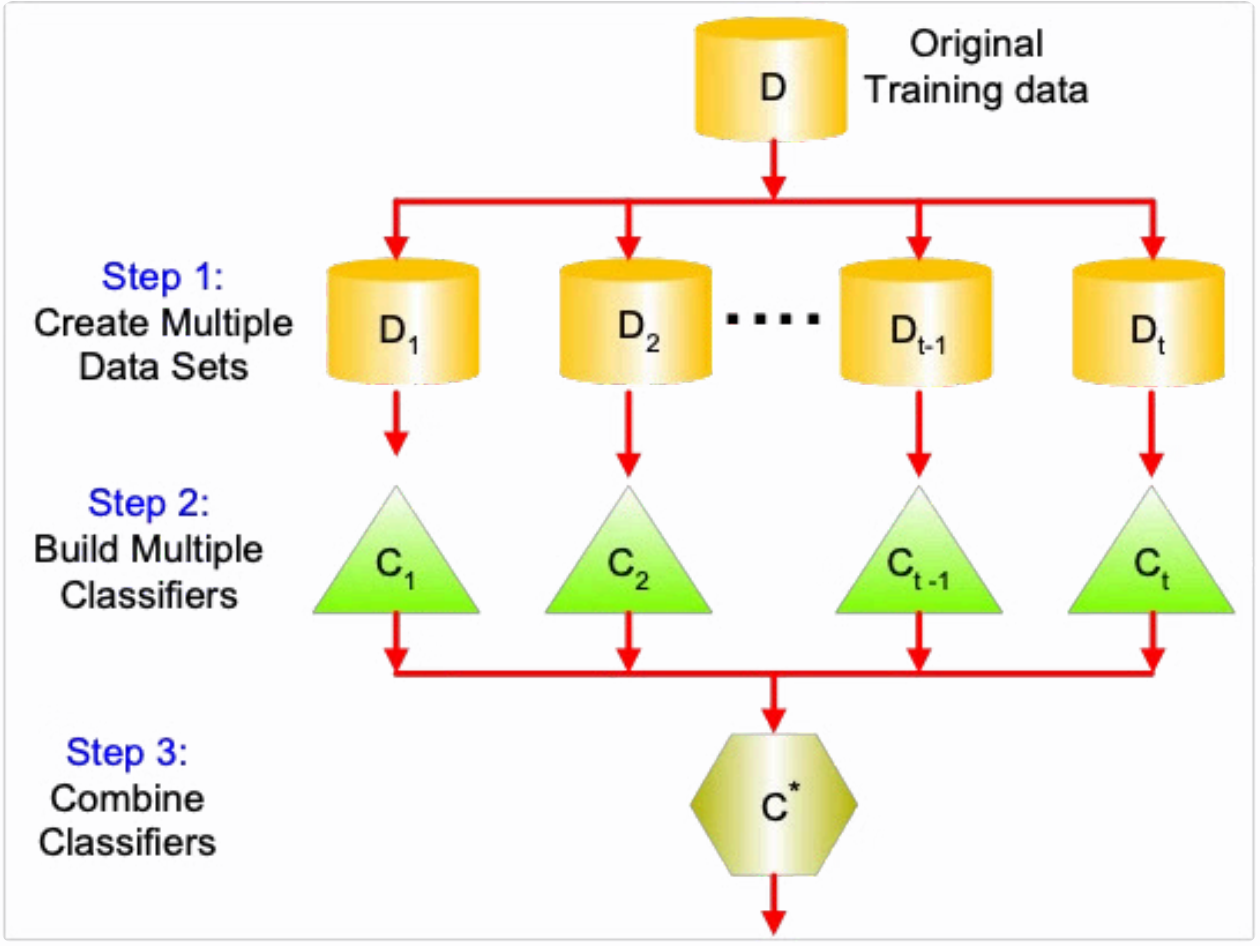
Kun-Ta Chuang

ktchuang@mail.ncku.edu.tw

Ensemble Methods

- Construct a set of classifiers from the training data
- Predict class label of previously unseen records by aggregating predictions made by multiple classifiers

General Idea



Why does it work?

- Suppose there are 25 base classifiers
- Each classifier has error rate, ε = 0.35
- Assume classifiers are independent
- Probability that the ensemble classifier makes a wrong prediction:

$$\sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1-\varepsilon)^{25-i} = 0.06$$

Examples of Ensemble Methods

- How to generate an ensemble of classifiers?
  - Bagging
  - Boosting

Bagging

- Sampling with replacement

Original Data	1	2	3	4	5	6	7	8	9	10
Bagging (Round 1)	7	8	10	8	2	5	10	10	5	9
Bagging (Round 2)	1	4	9	1	2	3	2	7	3	2
Bagging (Round 3)	1	8	5	10	5	5	9	6	3	7

- Build classifier on each bootstrap sample
- Each sample has probability p of being selected

$$p = (1–1/n)^n$$

Boosting

- An iterative procedure to adaptively change distribution of training data by focusing more on previously misclassified records
  - Initially, all N records are assigned equal weights
  - Unlike bagging, weights may change at the end of boosting round

Boosting

- Records that are wrongly classified will have their weights increased
- Records that are classified correctly will have their weights decreased

Original Data	1	2	3	4	5	6	7	8	9	10
Boosting (Round 1)	7	3	2	8	7	9	4	10	6	3
Boosting (Round 2)	5	4	9	4	2	5	1	7	4	2
Boosting (Round 3)	4	4	8	10	4	5	4	6	3	4

a. Example 4 is hard to classify

b. Its weight is increased, therefore it is more likely to be chosen again in subsequent rounds

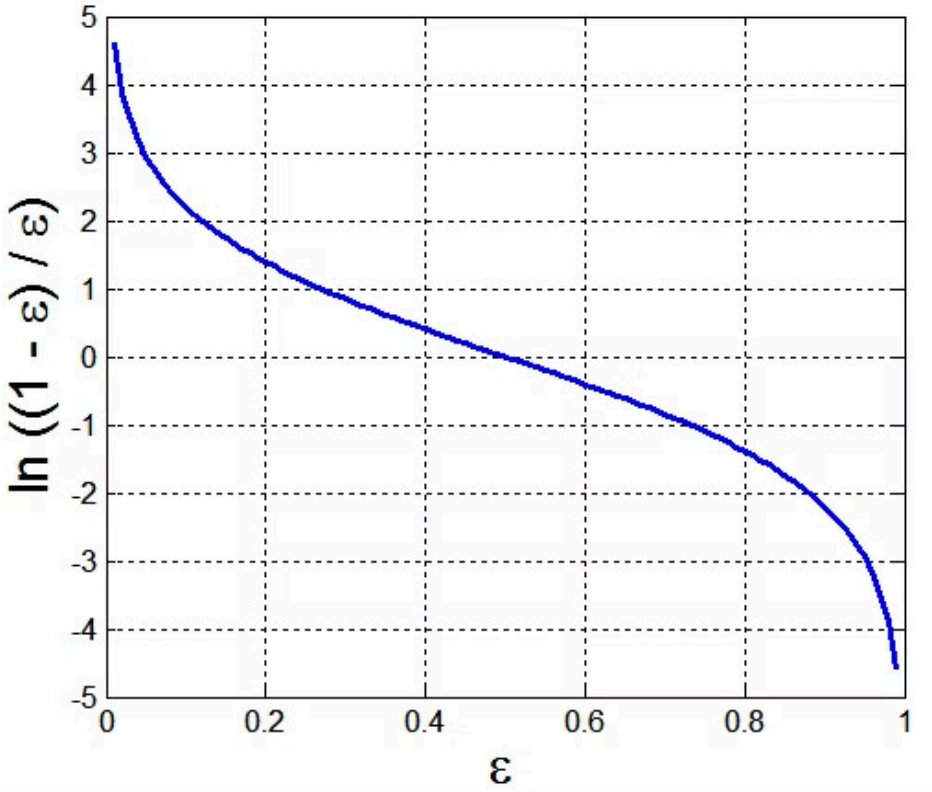
Example: AdaBoost

- Base classifiers: C1, C2, …, CT
- Error rate:

$$\varepsilon_i = \frac{1}{N}\sum_{j=1}^{N} w_j \delta\left(C_i(x_j) \neq y_j\right)$$

- Importance of a classifier:

$$\alpha_i = \frac{1}{2}\ln\left(\frac{1-\varepsilon_i}{\varepsilon_i}\right)$$



Example: AdaBoost

- Weight update:

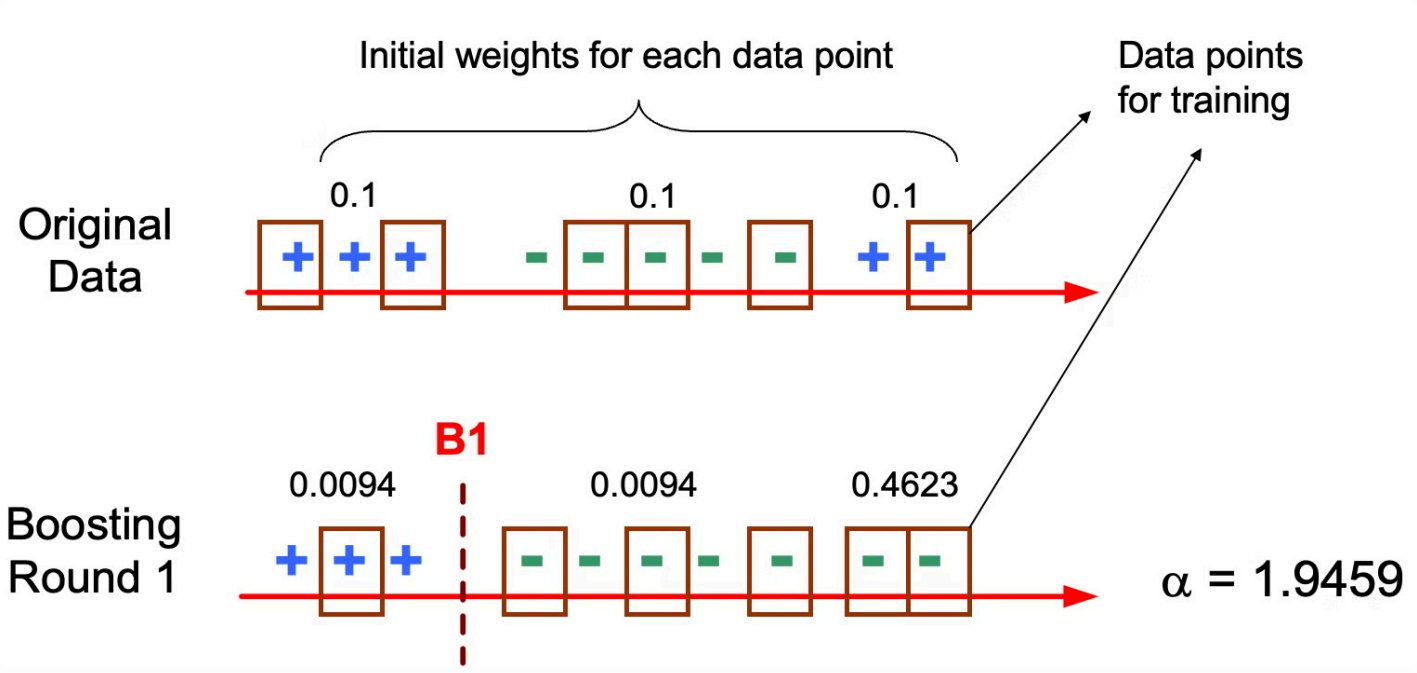
$$w_i^{(j+1)} = \frac{w_i^{(j)}}{Z_j} \begin{cases} \exp^{-\alpha_j} & \text{if } C_j(x_i) = y_i \\ \exp^{\alpha_j} & \text{if } C_j(x_i) \neq y_i \end{cases}$$

where Z_i is the normalization factor

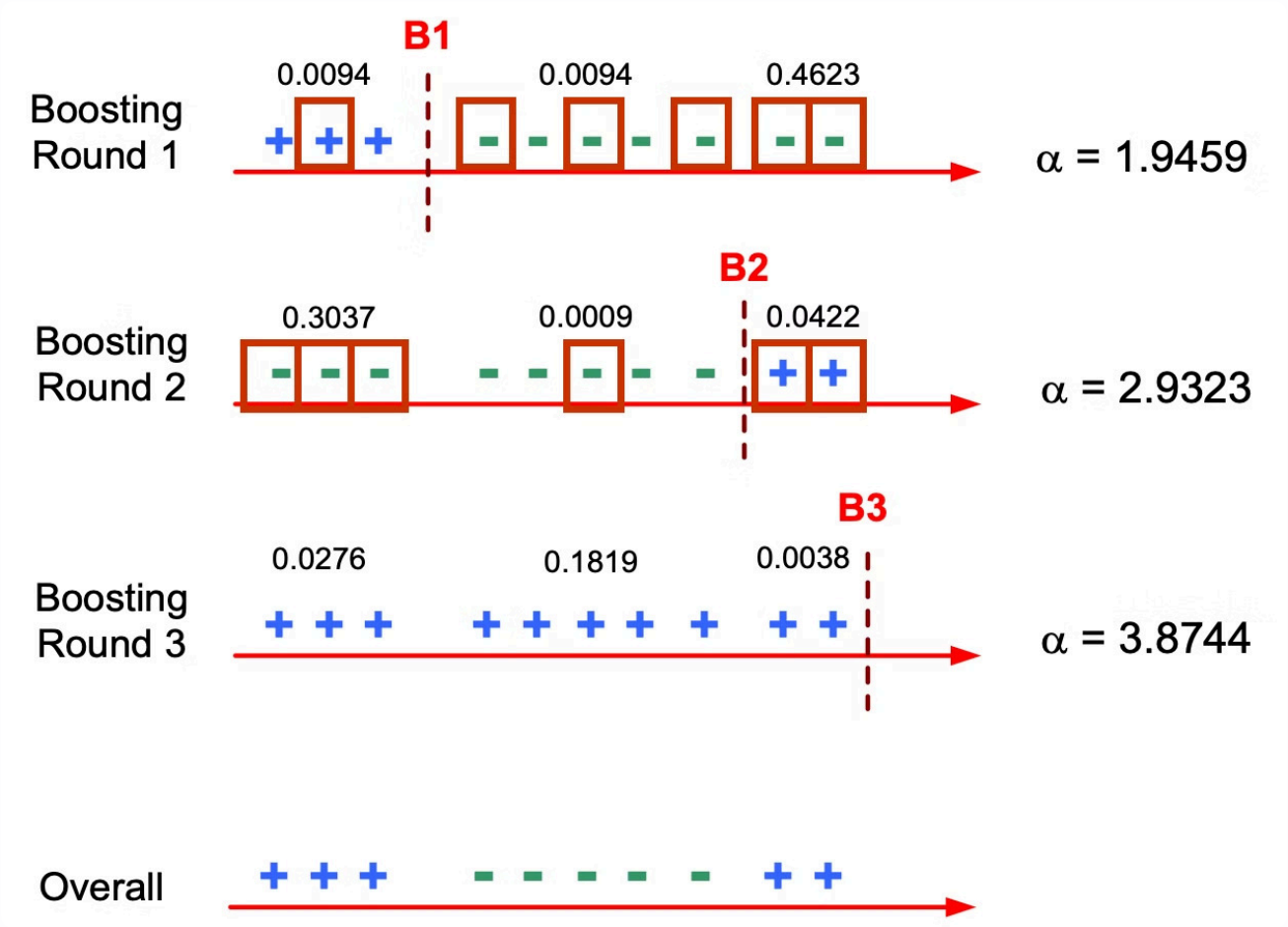
- If any intermediate rounds produce error rate higher than 50%, the weights are reverted back to 1/n and the resampling procedure is repeated
- Classification:

$$C^*(x) = \arg\max_y \sum_{j=1}^{T} \alpha_j \delta\left(C_j(x) = y\right)$$

Illustrating AdaBoost



Illustrating AdaBoost



SOTA Classification Packages

主題：State-of-the-Art (SOTA) Machine Learning Methods for Classification

Tree-based Methods		
Decision Tree	sklearn.tree.DecisionTreeClassifier	最基本決策樹，易於視覺化與解釋
Random Forest	sklearn.ensemble.RandomForestClassifier	多棵樹 bagging，提升穩定性與準確率
Extra Trees	sklearn.ensemble.ExtraTreesClassifier	高隨機性版本，速度快但解釋性較差
Kernel Methods		
SVM (Support Vector Machine)	sklearn.svm.SVC	利用 kernel trick 處理非線性分類問題
Kernel Ridge Classifier	sklearn.kernel_ridge.KernelRidge	結合 ridge regression 與 kernel 方法
Ensemble Methods		
Bagging	sklearn.ensemble.BaggingClassifier	多模型隨機訓練 → 平均結果，減少變異
Stacking	sklearn.ensemble.StackingClassifier	使用第二層模型融合多模型的預測，表現穩定提升
Boosting Methods		
AdaBoost	sklearn.AdaBoostClassifier	傳統 boosting，調整樣本權重，適合弱分類器
Gradient Boosting (GBM)	sklearn.GradientBoostingClassifier	最小化殘差逐步學習
XGBoost	xgboost	工程優化版 GBM，處理缺失值與正則化效果好
LightGBM	lightgbm	使用 histogram-based 分裂，適合大資料訓練
CatBoost	catboost	支援類別特徵自動轉換，適合大量類別型欄位的任務

> 適合 Tabular Data 任務選擇建議：
>
> 小型資料集：SVM, Random Forest, AdaBoost
>
> 中型資料集：GradientBoosting, XGBoost
>
> 大型資料集：LightGBM, CatBoost
>
> 高維類別特徵：優先考慮 CatBoost

AutoML - Automated Machine Learning

Automated Machine Learning (AutoML) is a technology that automates the selection, optimization, and deployment of machine learning models, significantly reducing the expertise and time required to build high-performance models.

Improved Efficiency

Automates hyperparameter tuning, feature engineering, and model selection processes, greatly reducing model development time.

Lower Entry Barriers

Enables non-professional data scientists to build high-quality models, reducing dependence on expert resources.

Model Optimization

Systematically explores a larger space of models and parameters, often discovering optimal solutions difficult to achieve through manual tuning.

Standardized Workflow

Provides a consistent model development approach, ensuring reproducible results and adherence to best practices.

Major AutoML platforms include: Google Cloud AutoML, H2O.ai, DataRobot, Azure AutoML, and open-source options like Auto-Sklearn and AutoGluon. These tools are suitable for various application scenarios from initial experiments to enterprise-level applications.

Auto-Sklearn Example

```
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, roc_auc_score
import autosklearn.classification

# Load dataset
X, y = load_breast_cancer(return_X_y=True) #Test Example
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1)

# Initialize Auto-sklearn classifier
automl = autosklearn.classification.AutoSklearnClassifier(
    time_left_for_this_task=120,  # Total time in seconds
    per_run_time_limit=30        # Time limit per model
    #metric=autosklearn.metrics.roc_auc
)

# Train the model
automl.fit(X_train, y_train)

# Make predictions
y_pred = automl.predict(X_test)

# Evaluate accuracy and ROC AUC
print("Accuracy:", accuracy_score(y_test, y_pred))
print("ROC AUC:", roc_auc_score(y_test, y_pred))
```

You can also use `autosklearn.metrics.roc_auc` metric to evaluate the performance of the Auto-Sklearn classifier.