Data Mining in Credit Card Fraud Detection

Xuebin He
xhe2@wpi.edu

CS548 Showcase
April 22 2014

Resources

- http://cs.fit.edu/~pkc/papers/ieee-is99.pdf
- http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.218.7317
- http://en.wikipedia.org/wiki/Credit_card_fraud
- http://www.statisticbrain.com/credit-card-fraud-statistics/

Statistics

- About 10,000 credit card transactions are processed each second worldwide.
- 10% of Americans have been victims of credit card fraud
- 40% of all financial fraud related to credit card
- $5.5 Billion lose worldwide

Challenging For Data Mining

- Data from transactions grows too fast
- Limited time to find fraud
- Banks are unable to share data

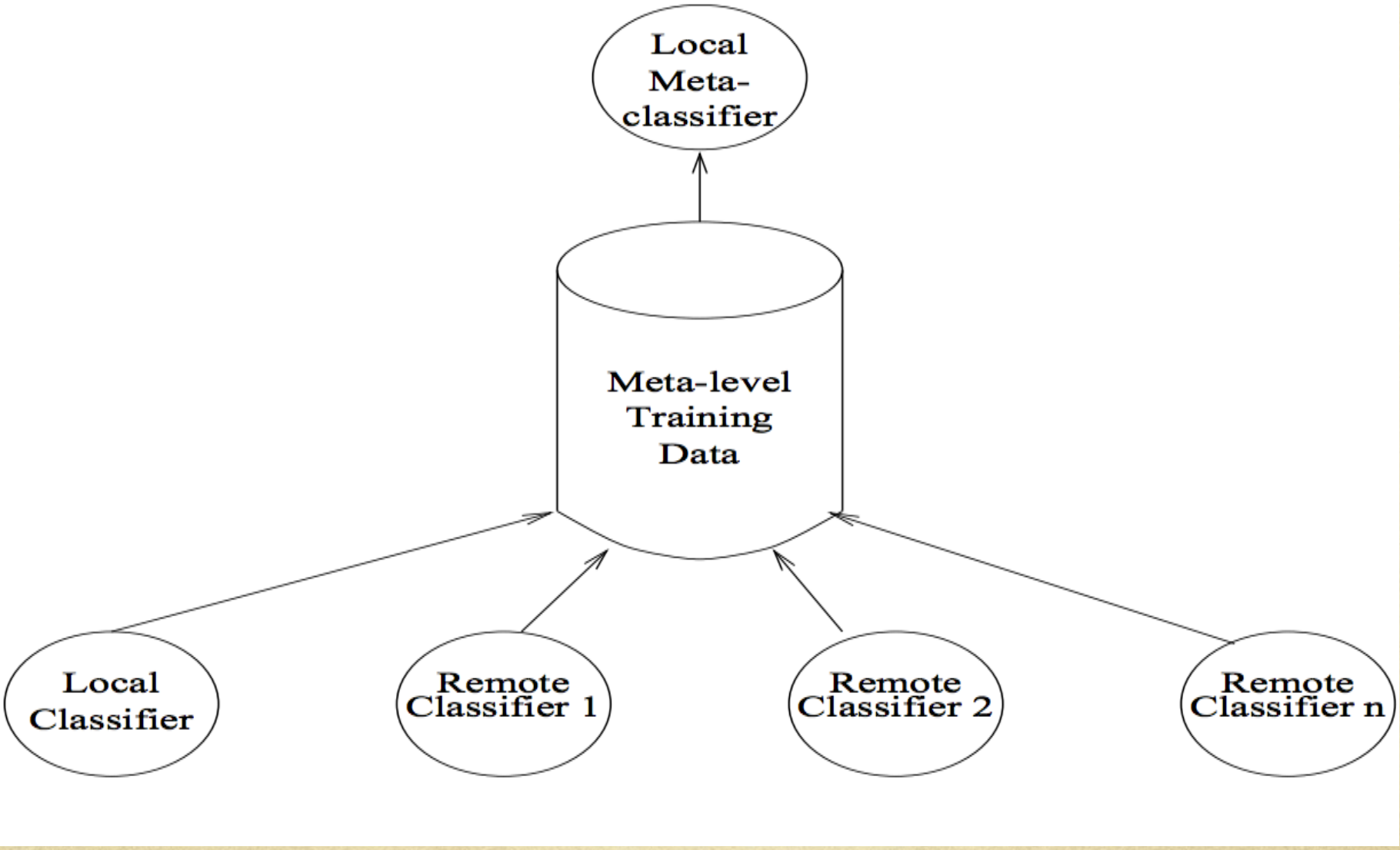
Meta-Learning

- A subfield of machine learning
- Main idea:

  Learn from classifiers generated by a number of (different) learning algorithms
- Process:

  1. Generate meta-data by the predictions of those algorithms
  2. Use another Algorithm to learn from the meta-data

Structure of Meta-Learning



*Taken from Figure 4 of reference #2

Distribute Approach

- Inner Bank
  1. Divide origin data set of transactions into smaller subsets
  2. Apply mining algorithms to generate classifiers in parallel
  3. Combine the classifiers to generate meta-classifier
- Inter Banks
  1. Define proprietary fields shared by all banks
  2. Generate classifier using its own data with those fields
  3. Combine the classifier with other banks' classifiers to generate a more reliable one

Proprietary Fields

- A hashed credit card account number
- Scores produced by a commercial authorization/detection system
- The date and time of each transaction
- Past payment information of the transactor
- The amount of the transaction
- Geographic information
- Codes for the validity and the manner of entry of the transaction
- An industry standard code for the type of merchant
- A code for other recent “non-monetary" transaction types
- The age of the account and the card
- Other credit card account information
- Condential and Proprietary Fields
- The fraud label

*Taken from Page 10 of reference #2

Java Agents for Meta-Learning (JAM)

- It is a distributed meta-learning system that supports the launching of learning and meta- learning agents to distributed database sites
- Each datasite contains:

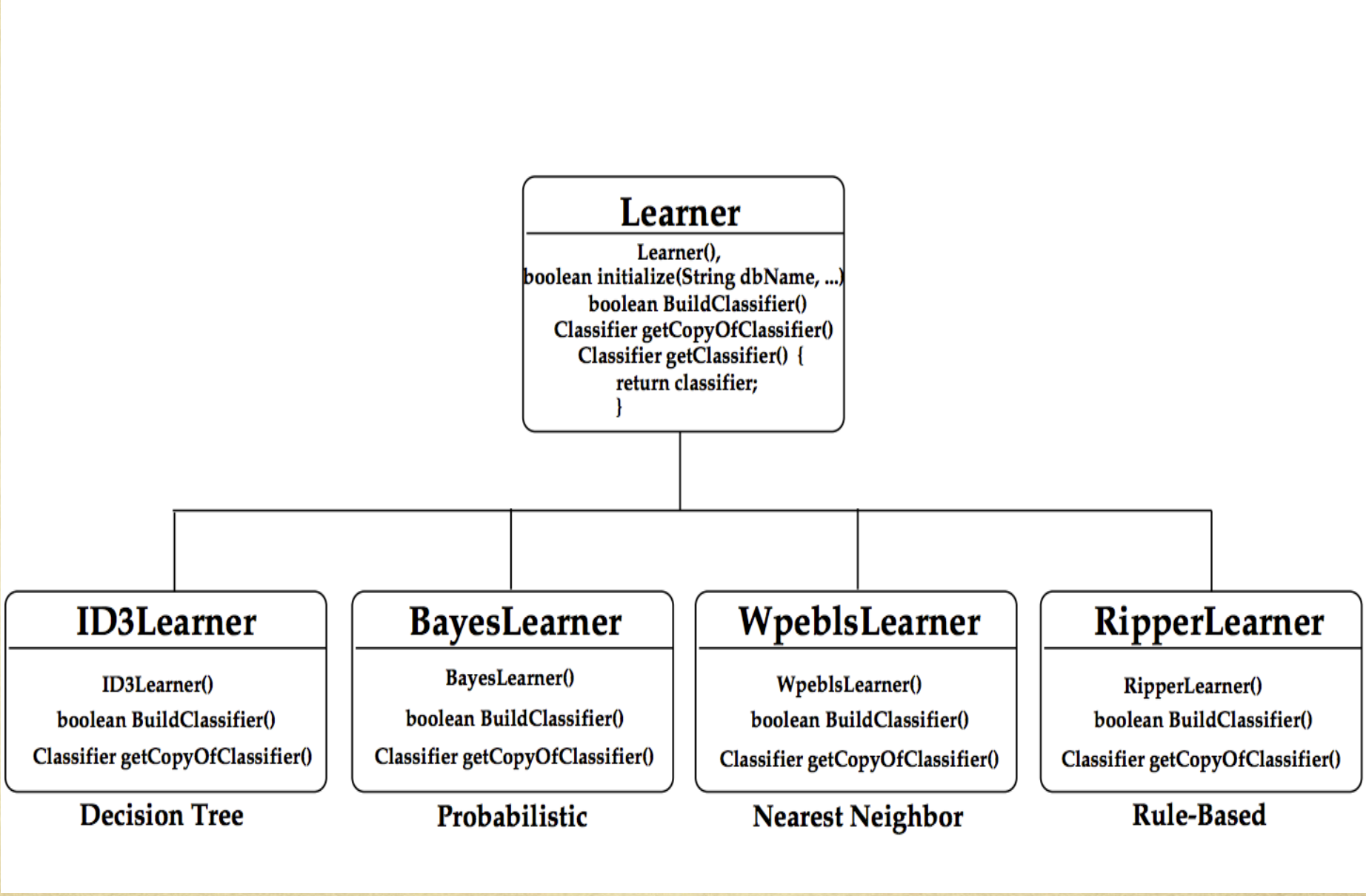
  1. A local database

  2. A learning agent

  3. A meta-learning agent

*refer from Page 2 of reference #2

Learning Agent API



*Taken from Figure 3 of reference #2

AdaCost Algorithm

Given: $(x1, c1, y1), \ldots, (x_m, c_m, y_m): x_i \in \mathcal{X}, c_i \in R+, y_i \in \{-1,+1\}$
Initialize $D1(i)$ (such as $D_1(i) =$ (such as $D_1(i) = c_i / \sum_j^m c_j$)
For $t = 1, \ldots, T$:

1. Train weak learner using distribution D_t.
2. Compute weak hypothesis $h_t: \mathcal{X} \rightarrow R$.
3. Choose $\alpha_t \in R$ and $\beta(i) \in R^+$
4. Update

$$D_{t+1}(i) = \frac{D_t(i) \exp\left(-\alpha_t y_i h_t(x_i) \beta(\text{sign}(y_i h_t(x_i)), c_i)\right)}{Z_t}$$

where $\beta(\text{sign}(y_i h_t(x_i)), c_i)$ is a cost-adjustment function. Z_t is a normalization factor chosen so that D_{t+1} will be a distribution.

Output the final hypothesis:

$$H(x) = \text{sign}(f(x)) \text{ where } f(x) = \left(\sum_{t=1}^{T} \alpha_t h_t(x) \right)$$

*Taken from Figure A of reference #1

Thank you