Transaction Anomaly Detection
CMPS 242 Project Talk

Maria Daltayanni

University of California, Santa Cruz

Last update: December 11, 2009
1 Problem Setting

2 Logistic Regression

3 AdaBoost

4 Boosting Trees

5 Conclusion
Outline

1. Problem Setting
2. Logistic Regression
3. AdaBoost
4. Boosting Trees
5. Conclusion
Anomaly Detection Problem

- Detect Transaction Anomalies
- Input:
  - Set of 19-feature data
  - target feature: Class
    - Class = 1, if record contains anomaly
    - Class = 0, otherwise
- Goal:
  - Detect if test set records contain anomaly
Dataset

- Records
  - \( \sim 92000 \) records
  - 70% training set, 30% test set

- Features
  - 19 Original Features
    - amount
    - total
    - hour1
    - hour2
    - field1
    - field2
    - field3
    - field4
    - field5
    - flag1
    - flag2
    - flag3
    - flag4
    - flag5
    - indicator1
    - indicator2
    - state
    - zip
    - domain
    - class
  - 1028 features after selection/creation:
Data Preprocessing - Nominal Features

- Turn each nominal features to multiple binary features
- Example:
  - Feature: "State", Domain: \{AL, AK, AZ, AR, CA, CO, ..., WY\}
  - New feature: "S", Domain: \{0,1\}
  - \( S = \{\text{State} = 'AL', ..., \text{State} = 'CA', ..., \text{State} = 'WY'\} \)

  1 feature replaced by 52 features
  \[
  x(\text{State}) = CA \\
  x(\text{State} == CA) = 1 \\
  x(\text{State} == AZ) = 0 \\
  ... \\
  x(\text{State} == WY) = 0
  \]
Data Preprocessing - Feature Selection

- Some features carry more information about anomaly than others.
- Identify *informative* features:
  - Isolate & plot rows of *(class 1)*
  - Plot all rows *(class 1 and class 0)*
  - Compare histograms

*Sample histograms in Weka*
Outline

1. Problem Setting
2. Logistic Regression
3. AdaBoost
4. Boosting Trees
5. Conclusion
- 2-norm regularization
- Logistic Loss Gradient Descent
- add weight $\beta = 0.7$ for class-1 records and $\beta = 0.3$ for class-0 records

Initialization: $n = \# \text{ features}, T = \# \text{ transactions}, w_i = 0, 1 \leq i \leq n$

while $\text{norm}_1 > 10^{-4}$ do

$$\hat{y}_t = \text{sigmoid}(w \cdot \beta(y_t)x_t), 1 \leq t \leq T$$

where $\beta(y_t) = \begin{cases} 
0.7, & \text{class}(y_t) = 1 \\
0.3, & \text{class}(y_t) = 0 
\end{cases}$

$$\text{gradient} = 2\lambda w + \frac{1}{T} \sum_{t=1}^{T}(\hat{y}_t - y_t)x_t$$

$$w_{t+1} = w_t - \eta \cdot \text{gradient}$$

Update $\text{norm}_1(\text{gradient})$

end while
Informative Features

- t-statistic metric

![Informative predictor features graph](image)
Model Evaluation

- Data too skewed
  - Accuracy 99% would predict all as 0 leaving out 1s
- Use $\text{Lift} = \frac{a}{b}$
  - extract and sort probabilities $\text{Pr}[\text{class}(\text{record}) = 1]$
  - $a$: rate of 1s on top 20% probabilities
  - $b$: rate of 1s in test set
  - high lift - good prediction
Evaluation - Lift & Error

- Lift:
  - $b = 4.6\%$ (rate of Class-1 records in test set)
  - $a = 14.3\%$ (rate of Class-1 records in top 20% records)
  - $Lift = 3.54$

- $Error = 0.0328$
Outline

1. Problem Setting
2. Logistic Regression
3. AdaBoost
4. Boosting Trees
5. Conclusion
Algorithm

- Weak Learners: features
- Class column in $\pm 1$
- $a = \frac{1}{2} \log \left( \frac{1+r}{1-r} \right)$

Given: $(x_1, y_1), ..., (x_m, y_m) : x_i \in X, y_i \in \{-1, +1\}$

Initialize $D_t(i) = 1/m$

for $t = 1, ..., T$ do
  - Train weak learner using distribution $D_t$.
  - Get weak hypothesis $h_t : X \rightarrow R$
  - Choose $a_t \in R$
  - Update

$$D_{t+1}(i) = \frac{D_t(i)e^{-a_t y_i h_t(x_i)}}{Z_t}$$

where $Z_t$ is a normalization factor chosen so that $D_{t+1}$ will be a distribution.

end for

Output the final hypothesis:

$$H(x) = \text{sign} \left( \sum_{t=1}^{T} a_t h_t(x) \right)$$
Training

- Error Evaluation on Training Set

![Graph showing error evaluation over iterations](image)
Evaluation - Loss

- Loss Evaluation

![Graph showing loss evaluation over iterations for training and test sets.](image)
Outline

1. Problem Setting
2. Logistic Regression
3. AdaBoost
4. Boosting Trees
5. Conclusion
Method

- Compute sequence of simple trees
- Each successive tree is built for the prediction residuals of the preceding tree
- “Additive Weighted Expansions”: produce excellent fit even though class and features have non linear relationship
Evaluation - Missclassification Error

- $k$: penalty on misclassifying a record which is class 1
Evaluation - Lift

- Based on prob that class = 1, find k that gives higher lift (highest number of correctly classified records with class = 1)

![Graph showing lift as a function of k with Best k = 13 highlighted]
Evaluation - Loss

![Graph showing the evaluation loss over iterations with 'best on iteration 39' highlighted.](image)

- Train (green line)
- Test (red line)

The graph indicates that the loss for both training and testing data decreases as the number of iterations increases, with an improvement observed around iteration 39.
Outline

1. Problem Setting
2. Logistic Regression
3. AdaBoost
4. Boosting Trees
5. Conclusion
Conclusion

- Dataset too skewed ($\text{class} = 1 \sim 1\%$)
- Most classifiers return wrong results unless tuned with "prediction weighting"
- Linear Regression coefficient calculations fail due to big sparsity of class feature, results extracted only based on dataset subset with numerical features
- Boosting Trees outperformed other classifiers (both lift and error metrics)