Convolve 3.0

Mathematical Document

Understanding the logic behind the model

Contributors

Shaman Shetty
Aryan Tiwari
Tanay Chaplot

Introduction

The mathematical foundations of modern fraud detection systems reflect the intersection of probability theory, statistical learning, and optimization mathematics. Our analysis explores four key algorithms - Logistic Regression, Random Forest, XGBoost, and LightGBM - each representing different mathematical approaches to the classification problem $P(\text{fraud}|X) \in [0,1]$. While Logistic Regression leverages the properties of sigmoid functions and maximum likelihood estimation, ensemble methods like Random Forest employ bootstrap aggregation and decision tree mathematics. The gradient boosting algorithms (XGBoost and LightGBM) further extend these concepts through Taylor series expansions, gradient optimization, and innovative sampling techniques. Understanding these mathematical principles is crucial for optimizing model performance and interpreting results in practical fraud detection applications.

Mathematical Foundations of Machine Learning Algorithms for Fraud Detection

Logistic Regression

The foundation of our fraud detection system begins with logistic regression, which models the probability of an event occurring based on input features. Let's explore its mathematical framework.

The Logistic Function

At the heart of logistic regression is the sigmoid function $\sigma(z)$:

$$\sigma(z) = 1 / (1 + e^{(-z)})$$

where z is the linear combination of features:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

This creates a probability output between 0 and 1:

$$P(Y=1|X) = \sigma(z)$$

Maximum Likelihood Estimation

The model parameters β are estimated using maximum likelihood estimation. The likelihood function for N observations is:

$L(\beta) = \prod_{i=1}^{N} P(Y=1|X_i)\text{^}y_i * (1-P(Y=1|X_i))\text{^}(1-y_i)$

Taking the log transforms this into:

$\log L(\beta) = \sum_{i=1}^{N} [y_i\log(P(Y=1|X_i)) + (1-y_i)\log(1-P(Y=1|X_i))]$

Gradient Descent

The parameters are optimized using gradient descent:

$\beta\_new = \beta\_old - \alpha * \nabla \log L(\beta)$

where α is the learning rate and the gradient is:

$\nabla \log L(\beta) = \sum_{i=1}^{N} (y_i - P(Y=1|X_i))X_i$

Random Forest

Random Forest builds upon decision trees using ensemble methods. Let's examine its mathematical construction.

Decision Trees

For each split in a tree, we calculate the Gini impurity:

$\text{Gini}(t) = 1 - \sum_{i} p(i|t)^{2}$

where $p(i|t)$ is the proportion of class i observations in node t.

Bootstrap Aggregating (Bagging)

For B trees in the forest, each tree b is trained on a bootstrap sample D_b drawn with replacement from the training data D. The final prediction is:

$P(Y=1|X) = 1/B \sum^{b}{}_{=1}{}^{B} f^{b}(X)$

where $f^{b}(X)$ is the prediction of the b-th tree.

Random Feature Selection

At each split, only a random subset of m features is considered, where:

$m \approx \sqrt{p}$ for classification (p is total number of features)

This decorrelates the trees and reduces variance.

XGBoost (eXtreme Gradient Boosting)

XGBoost builds an additive model using gradient boosting. Let's examine its mathematical framework.

Objective Function

The objective function combines loss and regularization:

$$\text{obj}(\theta) = L(\theta) + \Omega(\theta)$$

where $L(\theta)$ is the training loss and $\Omega(\theta)$ is the regularization term.

Model Building

The prediction for an instance i at the t-th iteration is:

$\hat{y}_i^t = \hat{y}_i^{t-1} + \eta f_t(x_i)$

where η is the learning rate and f_t is the t-th tree.

Taylor Expansion of Loss

XGBoost uses second-order Taylor expansion of the loss function:

$L(\theta) \approx \sum_i [l(y_i, \hat{y}_i^{t-1}) + g_i f_t(x_i) + ½ h_i f_t^2(x_i)]$

where:

- $g_i = \partial/\partial\hat{y}^{t-1}\ l(y_i, \hat{y}^{t-1})$
- $h_i = \partial^2/\partial\hat{y}^{t-12}\ l(y_i, \hat{y}^{t-1})$

Tree Structure Score

For a tree structure q(x), the scoring function is:

$$\text{score}(q) = -\tfrac{1}{2}[\textstyle\sum_{i}\in I_l\ g_i]^2/[\textstyle\sum_{i}\in I_l\ h_i + \lambda] + -\tfrac{1}{2}[\textstyle\sum_{i}\in I_r\ g_i]^2/[\textstyle\sum_{i}\in I_r\ h_i + \lambda]$$

where λ is the L2 regularization term.

LightGBM

LightGBM modifies gradient boosting with efficiency improvements. Let's examine its unique mathematical aspects.

Gradient-based One-Side Sampling (GOSS)

GOSS retains all instances with large gradients and randomly samples instances with small gradients:

a% of instances with largest $|g_i|$

b% of randomly sampled instances from remaining data

Exclusive Feature Bundling (EFB)

For sparse features, the conflict measure between features j and k is:

$\text{conflict}(j,k) = \sum_i I(x_i\square \neq 0 \wedge x_{ik} \neq 0) / \sum_i I(x_i\square \neq 0 \vee x_{ik} \neq 0)$

Features with low conflict are bundled together.

Leaf-wise Growth

The leaf with maximum delta loss reduction is split:

$\text{Gain} = ½[\sum g_i]^2/[\sum h_i + \lambda]_{\text{parent}} - ½[\sum g_i]^2/[\sum h_i + \lambda]_{\text{le}}f_t - ½[\sum g_i]^2/[\sum h_i + \lambda]_{\text{ri}}g_{\text{ht}}$

Histogram-based Algorithm

Instead of finding the exact split points, LightGBM discretizes continuous features into bins:

bin_k = floor((feature_value - min_value) * num_bins / (max_value - min_value))

This reduces memory usage and computation time.

Model Integration for Fraud Detection

In our implementation, these algorithms work together through:

1. Feature Space Transformation

$X \rightarrow \Phi(X)$ where Φ represents our feature engineering pipeline

2. Probability Calibration

For each model M:

$P\_M(Y=1|X) = \text{sigmoid}(\alpha * \text{score\_M}(X) + \beta)$

where α,β are calibration parameters

3. Ensemble Weighting

Final prediction:

$P(Y=1|X) = \sum_i w_i P_i(Y=1|X)$

where w_i are learned weights for each model

Conclusion

This mathematical foundation ensures our fraud detection system has both theoretical soundness and practical effectiveness. The combination of these algorithms, each with its unique mathematical strengths, creates a robust system capable of detecting complex fraud patterns.