Hidden Decision Trees to Design Predictive Scores – Application to Fraud Detection

Vincent Granville, Ph.D.
AnalyticBridge

October 27, 2009

Potential Applications

- Fraud detection, spam detection
- Web analytics
  - Keyword scoring/bidding (ad networks, paid search)
  - Transaction scoring (click, impression, conversion, action)
  - Click fraud detection
  - Web site scoring, ad scoring, landing page / advertiser scoring
  - Collective filtering (social network analytics)
  - Relevancy algorithms
- Text mining
  - Scoring and ranking algorithms
  - Infringement detection
  - User feedback: automated clustering

General Model

- Predictive scores: score = f(response)
- Response:
  - Odds of converting (Internet traffic data – hard/soft conversions)
  - CR (conversion rate)
  - Probability that transaction is fraudulent
- Independent variables:
  - Rules
  - Highly correlated
- Traditional models:
  - Logistic regression, decision trees, naïve Bayes

Hidden Decision Trees (HDT)

- One-to-one mapping between scores and features
- Feature = rule combination = node (in DT terminology)
- HDT fundamentals:
  - 40 binary rules → 2^{40} potential features
  - Training set with 10MM transactions → 10MM features at most
  - 500 out of 10MM features account to 80% of all transactions
  - The top 500 features have strong CR predictive power
  - Alternate algorithm required to classify the 20% remaining transactions
  - Using neighboring top features to score the 20% remaining transactions creates bias

HDT Implementation

- Each top node (or feature) is a final node from an hidden decision tree
  - No need for tree pruning / splitting algorithms and criteria: HDT is straightforward, fast, and can rely on efficient hash tables (where key=feature, value=score)
- Top 500 nodes come from multiple hidden decision trees
- Remaining 20% transactions scored using alternate methodology (typically, logistic regression)
- HDT is an hybrid algorithm
  - Blending multiple, small, easy-to-interpret, invisible decision trees (final nodes only) with logistic regression

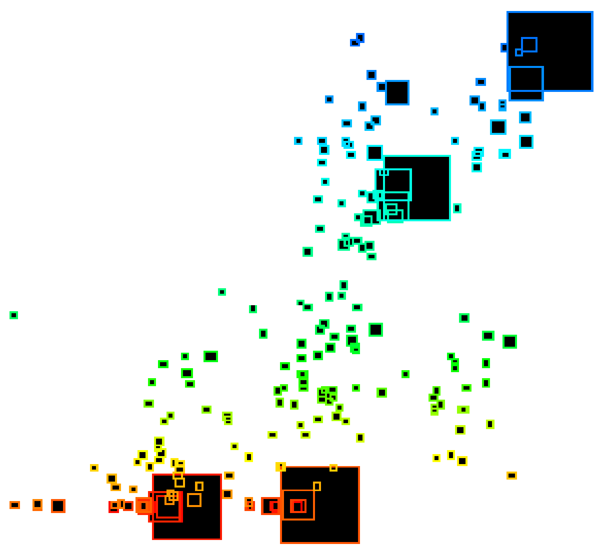
HDT: Score Blending

- The top 500 nodes provide a score S_1 available for 80% of the transactions
- The logistic regression provides a score S_2 available for 100% of the transactions
- Rescale S_2 using the 80% transactions that have two scores S_1 and S_2
  - Make S_1 and S_2 compatible on these transactions
  - Let S_3 be the rescaled S_2
- Transactions that can't be scored with S_1 are scored with S_3

HDT History

- 2003: First version applied to credit card fraud detection
- 2006: Application to click scoring and click fraud detection
- 2008: More advanced versions to handle granular and very large data sets
  - Hidden Forests: multiple HDT's, each one applied to a cluster of correlated rules
  - Hierarchical HDT's: the top structure, not just rule clusters, is modeled using HDT's
  - Non binary rules (naïve Bayes blended with HDT)

HDT Nodes: Example

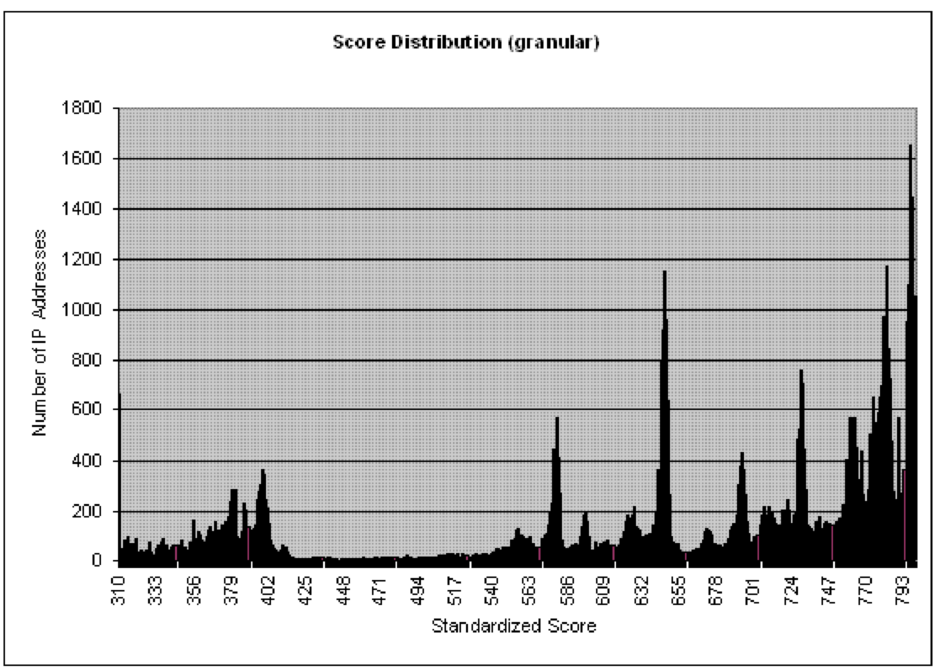


Y-axis = CR, X-axis = Score, Square Size = # observations

HDT Nodes: Nodes = Segments

- 80% of transactions found in 500 top nodes
- Each large square – or segment – corresponds to a specific transaction pattern
- HDT nodes provide an alternate segmentation of the data
- One large, medium-score segment corresponds to neutral traffic (triggering no rule)
- Segments with very low scores correspond to specific fraud cases
- Within each segment, all transactions have the same score
- Usually provides a different segmentation than PCA and other analyses

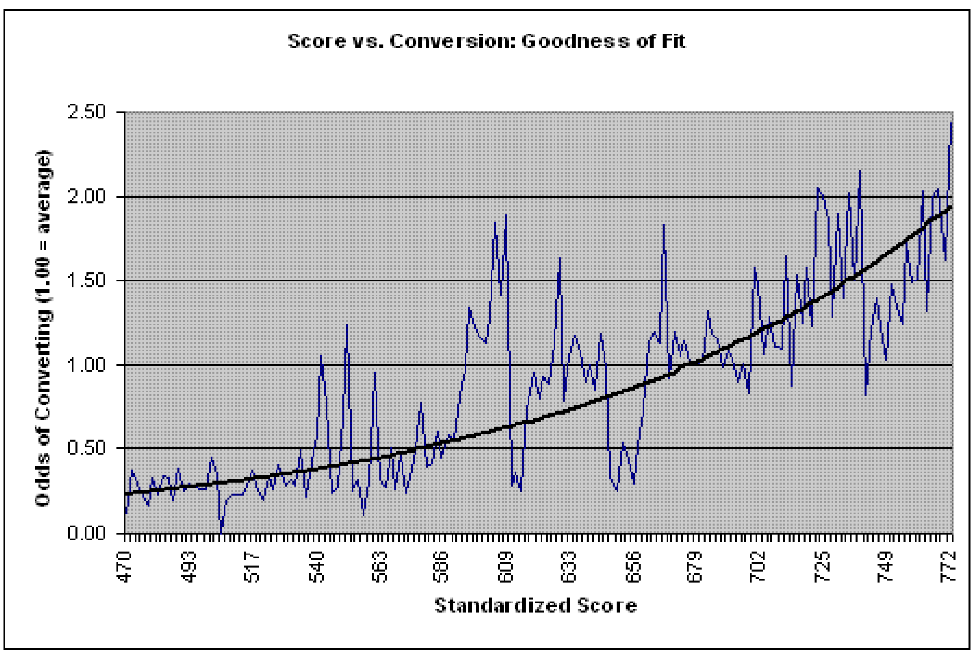
Score Distribution: Example



Score Distribution: Interpretation

- Reverse bell curve
- Scores below 425 correspond to un-billable transactions
- Spike at the very bottom and very top of the score scale
- 50% of all transactions have good scores
- Scorecard parameters
  - A drop of 50 points represents a 50% drop in conversion rate:
  - Average score is 650.
- Model improvement: from reverse bell curve to bell curve
  - Transaction quality vs. fraud detection
  - Anti-rules, score smoothing (will also remove score caking)

GOF Scores vs. CR: Example



GOF: Interpretation

- Overall good fit
- Peaks could mean:
  - Bogus conversions
  - Residual Noise
  - Model needs improvement (incorporate anti-rules)
- Valleys could mean:
  - Undetected conversions (cookie issue, time-to-conversion unusually high)
  - Residual noise
  - Model needs improvement

Conclusions

- Fast algorithm, easy to implement, can handle large data sets efficiently, output is easy to interpret
- Non parametric, robust
  - Risk of over-fitting is small if no more than top 500 nodes are selected and ad-hoc cross validation techniques used to remove unstable nodes
  - Built-in mechanism to compute confidence intervals for scores
- HDT: hybrid algorithm to detect multiple types of structures
  - Linear and non linear structures
- Future directions
  - Hidden forests to handle granular data
  - Hierarchical HDT's