

Search Data Science Central [Search](#)

- [Ramiro Arce](#)
- [Sign Out](#)



# Data Science Central™

THE ONLINE RESOURCE FOR BIG DATA PRACTITIONERS

• [HOME](#) [DATAVIZ](#) [HADOOP](#) [BIG DATA](#) [ANALYTICS](#) [WEBINARS](#) [DEEP LEARNING](#) [AI](#) [JOBS](#) [MEMBERSHIP](#) [SEARCH](#) [CLASSIFIEDS](#) [CONTACT](#)

[Subscribe to DSC Newsletter](#)

- All Blog Posts
- My Blog
- Edit Blog Posts
- Add



## Hidden decision trees revisited

- Posted by Vincent Granville on November 13, 2013 at 8:30pm
- [Send Message](#) [View Blog](#)

*This is a revised version of an earlier article posted on [AnalyticBridge](#). The most recent article on this topic can be found [here](#).*

Hidden decision trees (HDT) is a technique patented by Dr. Granville, to score large volumes of transaction data. It blends robust logistic regression with hundreds small decision trees (each one representing for instance a specific type of fraudulent transaction) and offers significant advantages over both logistic regression and decision trees: robustness, ease of interpretation, and no tree pruning, no node splitting criteria. It makes this methodology powerful and easy to implement even for someone with no statistical background.

*Hidden Decision Trees* is a statistical and data mining methodology (just like logistic regression, SVM, neural networks or decision trees) to handle problems with large amounts of data, non-linearity and strongly correlated independent variables.

The technique is easy to implement in any programming language. It is more robust than decision trees or logistic regression, and helps detect natural final nodes. Implementations typically rely heavily on large, granular hash tables.

No decision tree is actually built (thus the name hidden decision trees), but the final output of a hidden decision tree procedure consists of a few hundred nodes from multiple non-overlapping small decision trees. Each of these parent (invisible) decision trees corresponds e.g. to a particular type of fraud, in fraud detection models. Interpretation is straightforward, in contrast with traditional decision trees.

The methodology was first invented in the context of credit card fraud detection, back in 2003. It is not implemented in any statistical package at this time. Frequently, hidden decision trees are combined with logistic regression in an hybrid scoring algorithm, where 80% of the transactions are scored via hidden decision trees, while the remaining 20% are scored using a compatible logistic regression type of scoring.

Hidden decision trees take advantage of the structure of large multivariate features typically observed when scoring a large number of transactions, e.g. for fraud detection. The technique is not connected with hidden Markov fields.

### 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

### Implementation

The model presented here is used in the context of click scoring. The purpose is to create predictive scores, where  $score = f(response)$ , that is, score is a function of the response. The response is sometimes referred to as the *dependent variable* in statistical and predictive models.

- Examples of Response:
  - Odds of converting (Internet traffic data – hard/soft conversions)
  - CR (conversion rate)
  - Probability that transaction is fraudulent
- Independent variables: Called *features* or rules. They are highly correlated

## Embrace the Modern Analytics Lifecycle - June 26



Join Radiant Advisors for a discussion surrounding new research that explores solutions and reference architectures for data science platforms on Azure – all in the context of a modern analytics lifecycle.

[Register Now](#)

HDT fundamentals, based on typical data set:

- If we use 40 binary rules, we have 2 at power 40 potential multivariate features
- If training set has 10 MM transactions, we will obviously observe 10MM multivariate features at most, a number much smaller than 2 at power 40
- 500 out of 10MM features account to 80% of all transactions
- The top 500 multivariate features have strong predictive power
- An alternate algorithm is required to classify the 20% remaining transactions
- Using neighboring top multivariate features to score the 20% remaining transactions creates bias, as rare multivariate features (sometimes not found in the training set) corresponds to observation that are worse than average, with a low score (because they trigger many fraud detection rules).

#### Implementation details

Each top node (or multivariate feature) is a final node from a hidden decision tree. There is 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). The top 500 nodes, used to classify (that is, score) 80% of transactions, come from multiple hidden decision trees - hidden because you never used a decision tree algorithm to produce them.

The remaining 20% transactions scored using alternate methodology (typically, logistic regression). Thus HDT is a hybrid algorithm, blending multiple, small, easy-to-interpret, invisible decision trees (final nodes only) with logistic regression.

Note that in the logistic regression, we use constrained regression coefficients. These coefficients depend on 2 or 3 top parameters and have the same sign as the correlation between the rule they represent, and the response or score. This make the regression non-sensitive to high cross correlations among the "independent" variables (rules) which are indeed not independent in this case. This approach is similar to *ridge regression*, *logic regression* or *Lasso regression*. The regression is used to fine tune the top parameters associated with regression coefficients. I will later in this book show that approximate solutions (we are doing approximate logistic regression here) are - if well designed - almost as accurate as exact solutions, but can be far more robust.

#### Score blending

We are dealing with two types of scores:

- 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

To blend the scores,

- *Rescale*  $S_2$  using the 80% transactions that have two scores  $S_1$  and  $S_2$ . Rescaling means apply a linear transformation so that both scores have same mean and same variance. Let  $S_3$  be the rescaled  $S_2$ .
- Transactions that can't be scored with  $S_1$  are scored with  $S_3$

HDT nodes provide an alternate *segmentation* of the data. One large, medium-score segment corresponds to neutral transactions (triggering no rule). Segments with very low scores correspond to specific fraud cases. Within each segment, all transactions have the same score. HDT's provide a different type of segmentation than PCA (*principal component analysis*) and other analyses.

#### 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)

#### Example: Scoring Internet Traffic

The figure below shows the score distribution with a system based on 20 rules, each one having a low triggering frequency. It has the following features:

- 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
  - Add anti-rules, perform score smoothing (will also remove score caking)

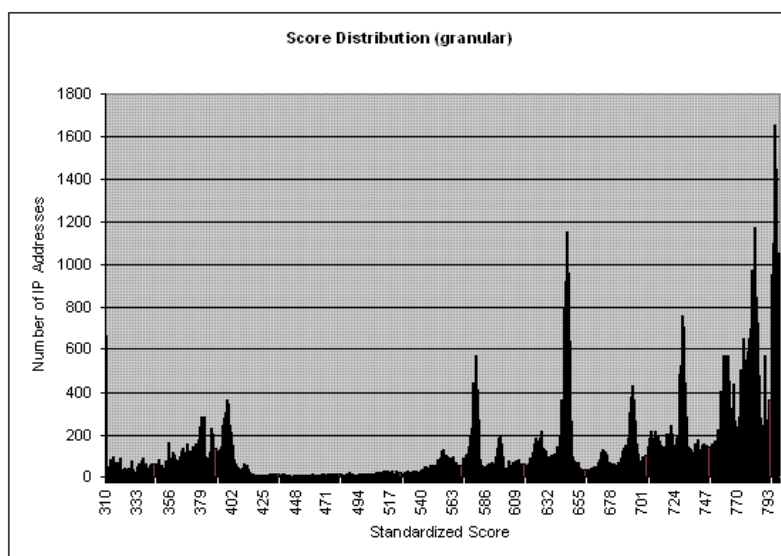


Figure 5.1: Example of score distribution based on HDT's

The figure below compares scores with conversion rates (CR). HDT's were applied to Internet data, scoring clicks with a score used to predict chances of conversion (a conversion being a purchase, a click out, sign-up on some landing page). Overall, we have a rather good fit.

Peaks in the figure below could mean:

- Bogus conversions (happens a lot if conversion is simply a click out)
- 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

Peaks and valleys can also be cause if you blend together multiple types of conversions: traffic with 0.5% CTR together with traffic with 10% CTR.

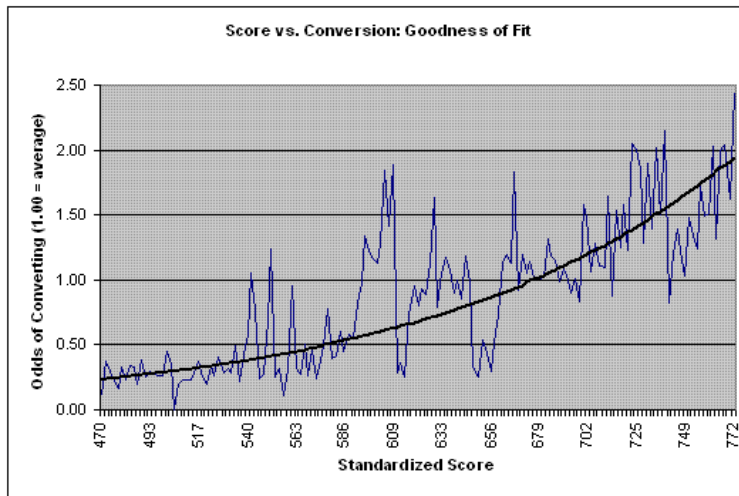


Figure 5.2: HDT scores to predict conversions

## Conclusions

HDT is a fast algorithm, easy to implement, can handle large data sets efficiently, and the output is easy to interpret.

It is non parametric and robust. The 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. It offers built-in, simple mechanism to compute confidence intervals for scores. See also next section.

HDT is hybrid algorithm to detect multiple types of structures: linear structures via the regression, and non linear structures via the top nodes.

## Future directions

- Hidden forests to handle granular data
- Hierarchical HDT's

## Related article

- Fast Combinatorial Feature Selection with New Definition of Predict...

Views: 25592

Like  
12 members like this

Share Tweet Facebook

Like 0

- < Previous Post
- Next Post >

## Comment



Visual Mode HTML Editor



10pt **B** *I* ~~S~~ U [List Icon] [Indent Icon] [Link Icon]

Follow – Email me when people comment

Add Comment



Comment by Richard Kenyon on March 6, 2018 at 2:43pm

Sounds a bit like <https://www.ne.anl.gov/codes/mset/patents/>



Comment by Sander Stepanov on June 20, 2016 at 8:20am

Thanks a lot, very interesting poat.  
Only from another post  
Black-box Confidence Intervals: Excel and Perl Implementation  
<http://www.datasciencecentral.com/profiles/blogs/black-box-confiden...>  
I see  
The next one to be detailed (with source code) will be our Hidden Decision Trees.  
It would be great share this code,  
Thank you very much in advance  
Sander



Comment by Dr. Z on November 18, 2013 at 2:09am

Very elegant and functional. A truly ingenious method with lots of applications. Good work!



Comment by Gary D. Miner, Ph.D. on November 15, 2013 at 12:09pm

An "Absolutely Great Explanation" !!!! .....You and I have been talking about HDT since about 2003, and now I finally understand it !!!!! .... When a person can "get through" to my "left-brained mind" they have really accomplished something - and you, Vincent, have done it !!!!!

Thanks,  
Gary

[RSS](#)

**RAMIRO ARCE**


[Sign Out](#)


- [Inbox](#)
- [Alerts](#)
- [Friends – Invite](#)
- [Settings](#)


FOLLOW US


@DataScienceCtrl | RSS Feeds


TOP CONTENT


- 

1  
Free Book: Applied Stochastic Processes
- 

2  
Will GDPR kill business in Ireland?
- 

3  
R, Python or SAS: Which one should you learn first?
- 

4  
Big Data to Pave the Way for Smart Cities
- 

5  
Debunking the 68 Most Common Myths About Big Data – Part 1
- 

6  
PostgreSQL, MonetDB, and Too-Big-for-Memory Data in R – Part I
- [RSS](#)
  - [View All](#)

ANNOUNCEMENTS

- Customer Analytics with Wharton Executive Education
- Whitepaper: Delivering data-informed insights
- Analytics Reimagined – Virtual Conference
- Join NYU Stern's MS in Business Analytics Program
- 10 Keys to Data Science Success – Whitepaper
- [eBook] A Guide To Data Science At Scale
- Data Prep: Gartner Guide
- Strengthen Your Skills - Advance Your Career
- Find the Right Accelerator for Your Deep Learning Needs
- 4 Ways to Fix Common Data Prep Issues

VIDEOS



•

DSC Webinar Series: Modern Data Science Workflows

Added by Tim Matteson 0 Comments 0 Likes +1



•

DSC Webinar Series: An Expert's Guide to Apache Spark™

Added by Tim Matteson 1 Comment 0 Likes +1



•

DSC Webinar Series: Top 4 Excel Functions Done Better with Data Wrangling

Added by Tim Matteson 0 Comments 0 Likes +1

- Add Videos
- View All

RESOURCES

- Migrating an Excel Spreadsheet to MySQL and to Spark 2.0.1 (Part 1)
- Introduction to Programming in Stata
- Benchmarking 20 Machine Learning Models Accuracy and Speed
- Stata Cheat Sheet
- Selection of best articles from our past weekly digests
- Statistical Analysis Advisor Chart
- Selection of best articles from our past weekly digests
- Free Online Book: Forecasting, Principles and Practice
- 38 Seminal Articles Every Data Scientist Should Read
- Black-box Confidence Intervals: Excel and Perl Implementation

TOP CATEGORIES

- Machine Learning
- R Programming
- Python for Data Science
- Visualization, Dashboards
- NoSQL and NewSQL
- Big Data
- Cheat Sheets
- Internet of Things
- Excel

