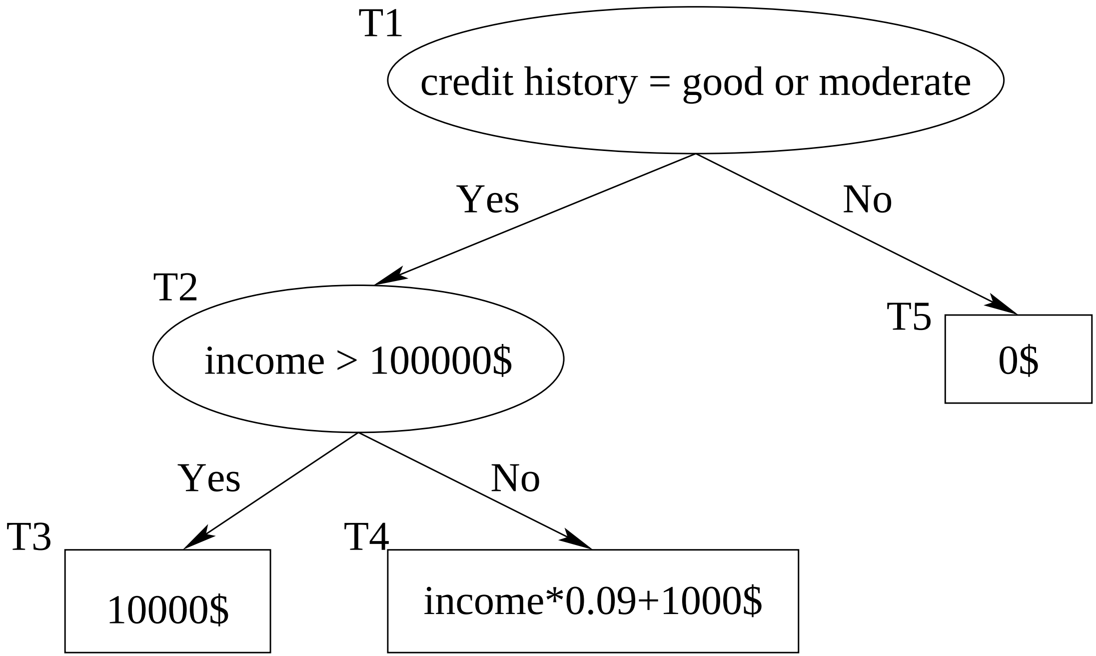
# SECRET: A Scalable Linear Regression Tree Algorithm

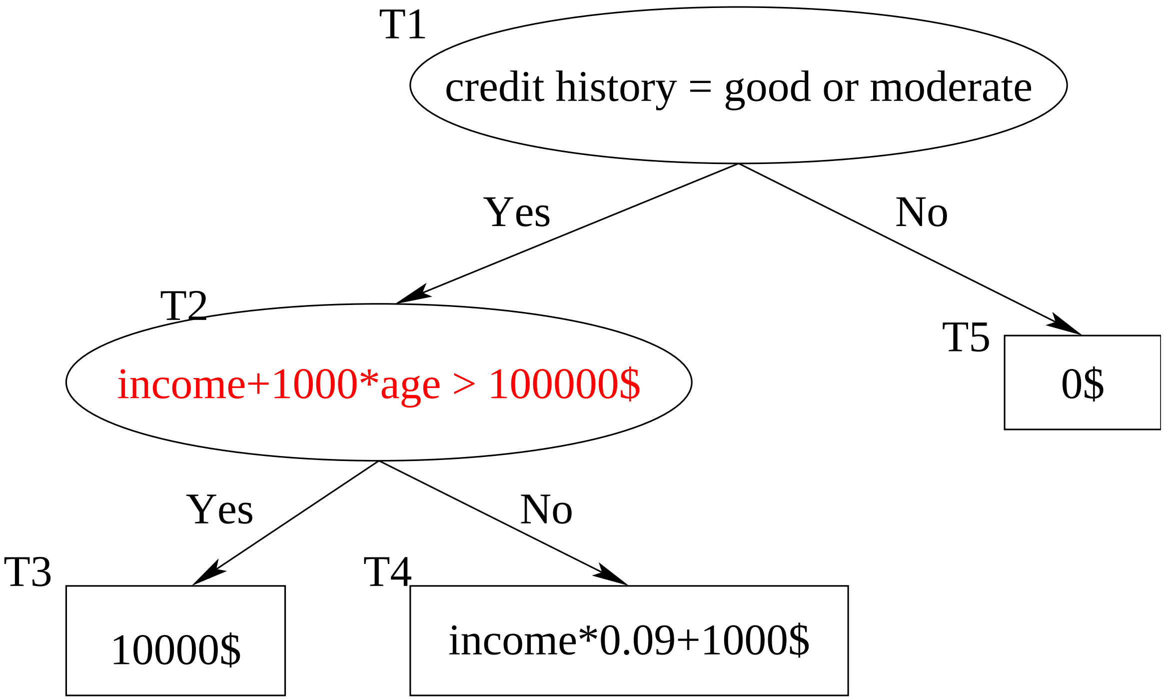
SIGKDD-2002

Alin Dobra
Johannes Gehrke
Cornell University

## Linear Regression Trees



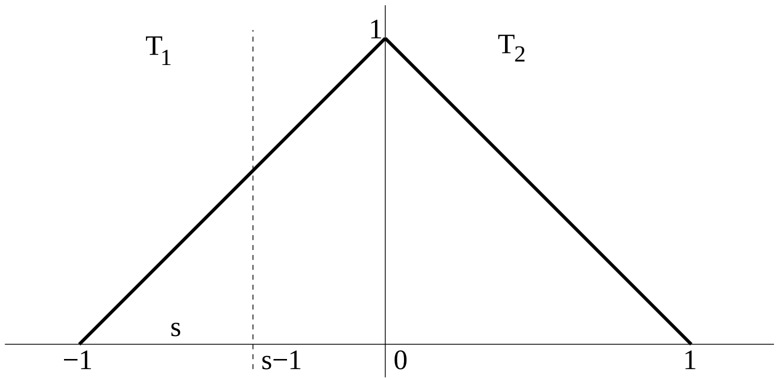
# Linear Regression Trees with Orthogonal Splits



# Previous Approaches

**Quinlan 1992:** *Pretend* that a regression tree with constant models in leaves is built using variance as impurity and find linear models for leaves only after growth phase

**Problems:**




The split-point chosen is $-(\sqrt{5}-1)/2 = -0.618$ which is very far from 0. Such split criteria produce unnecessary *fragmentation* and unbalanced trees.

# Previous Approaches (cont.)

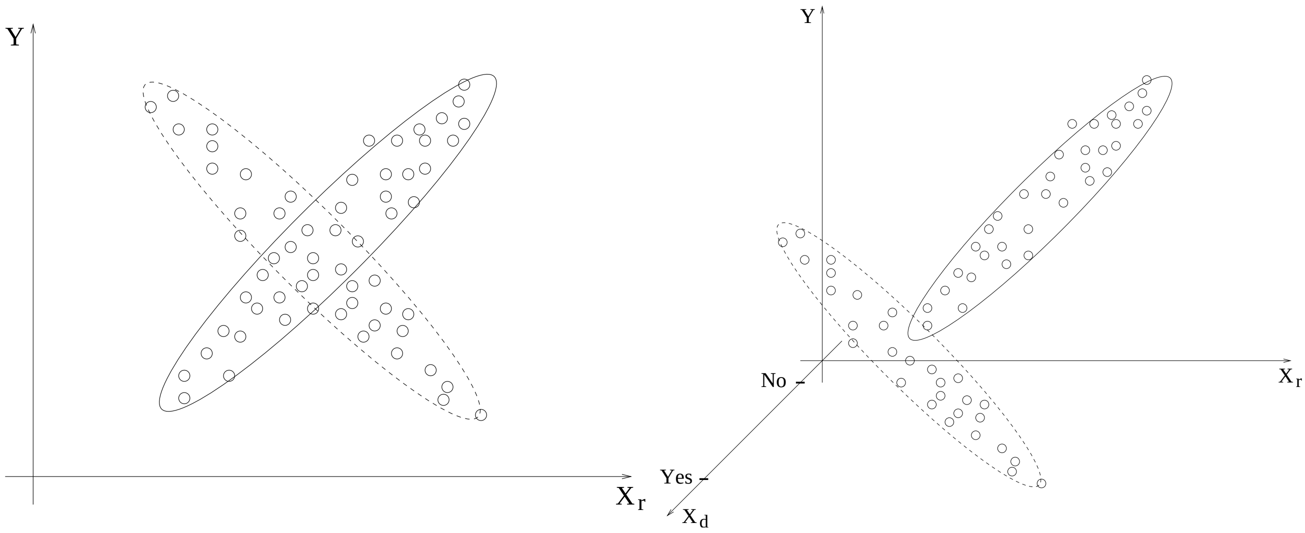
**Karalic 1992**

- Use error with respect to the linear model as goodness metric not variance (fixes the problem of Quinlan's algorithm)
- Exhaustive search used to find split attribute and split point
  - For every possible value of a continuous attribute a linear system has to be formed and solved
  - For discrete attributes an exponential number of linear systems have to be formed and solved since Theorem 9.4 in Breiman 1994 does not apply

**Chaudhuri et al. 1994**

- Avoids building many linear systems by locally classifying the data-points based on the sign of the residual w.r.t. the best linear regressor
- Usually the negative residuals surround the positive ones so the separation in classes does not provide a useful separation w.r.t. the regression problem

# Main Idea




- Find two Gaussian distributions in the data
- Classify points based on closeness w.r.t. these distributions
- Find best split attribute and corresponding split point using *gini gain* criterion in the resulting classification problem

# SECRET Algorithm

**Input**: node $T$, data-partition $D$
**Output**: regression tree $\mathcal{T}$ for $D$ rooted at $T$

**BuildTree**(node $T$, data-partition $D$)
(1) normalize data-points to unitary sphere
(2) find two Gaussian clusters in regressor–output space (EM)
(3) label data-points based on closeness to these clusters
(4) **foreach** split attribute
(5) find best split point and determine its gini gain
(6) **endforeach**
(7) let $X$ be the attribute with the greatest gini gain and $Q$ the corresponding best split predicate set
(8) **if** ($T$ splits)
(9) partition $D$ into $D_1, D_2$ based on $Q$ and label node $T$ with split attribute $X$
(10) create children nodes $T_1, T_2$ of $T$ and label the edge $(T, T_i)$ with predicate $q_{(T,T_i)}$
(11) BuildTree($T_1$, $D_1$); BuildTree($T_2$, $D_2$)
(12) **else**
(13) label $T$ with the least square linear regressor of $D$
(14) **endif**

# Split Point and Attribute Selection

- *Gini gain* used as split attribute selection criterion for all types of attributes
- For *discrete* attributes the best split point is found by finding the partition of the values into two sets in order to minimize *gini gain*
- For *continuous* attributes use Quadratic Discriminant Analysis

$$\alpha_1 \frac{1}{\sigma_1\sqrt{2\pi}} e^{-(\eta_1-\eta)^2/2\sigma_1^2} = \alpha_2 \frac{1}{\sigma_2\sqrt{2\pi}} e^{-(\eta_2-\eta)^2/2\sigma_2^2}$$

To compute *gini gain* is enough to compute:

$$\begin{aligned} P[x \in C_1 \mid x \le \eta] &= \int_{x\le\eta} \frac{1}{\sigma_1\sqrt{2\pi}} e^{-(x-\eta_1)^2/2\sigma_1^2} dx \\ &= \frac{1}{2}\left(1 + \mathsf{Erf}\left(\frac{\eta_1-\eta}{\sigma_1\sqrt{2}}\right)\right) \end{aligned}$$

and $P[x \in C_2 \mid x \le \eta]$ by a similar equation.

# Oblique Splits

- If the distribution of the data-points with the same class label (closer to the same Gaussian) is approximated with a Gaussian distribution, a good *oblique split* can be found by finding the hyperplane that best separates the two distributions

- Minimizing *gini gain* is hard. Fisher's separability criterion

$$J(\mathbf{n}) = \frac{\mathbf{n}^T \Sigma_w \mathbf{n}}{\mathbf{n}^T \Sigma_b \mathbf{n}}$$

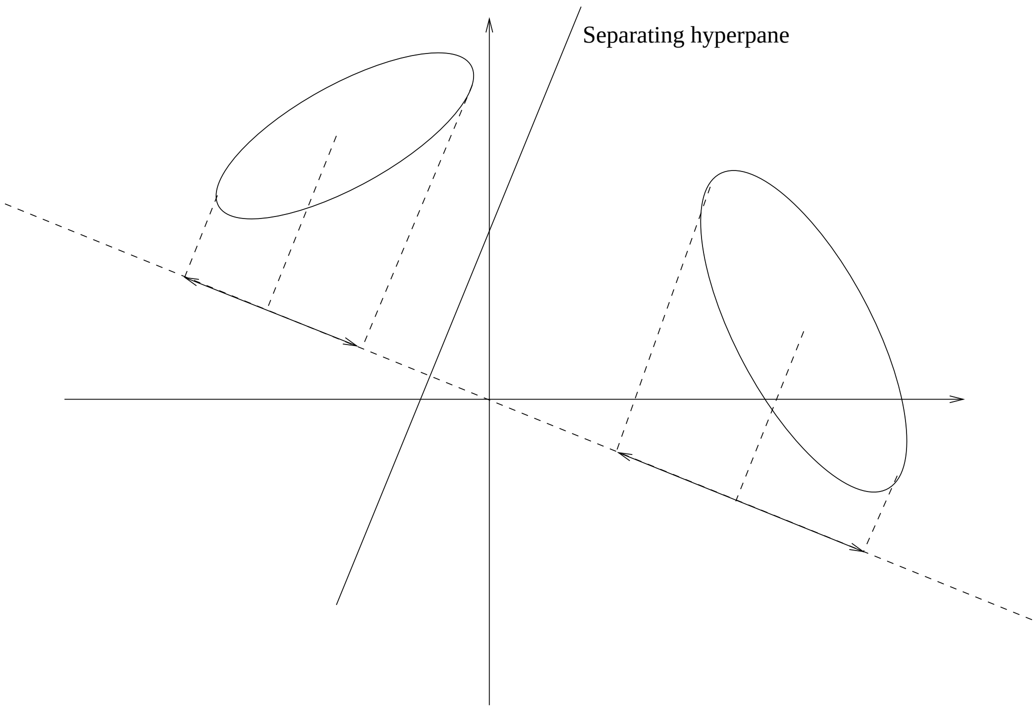
  with

$$\Sigma_w = \sum_{i=1,2} \alpha_i (\mu - \mu_i)(\mu - \mu_i)^T, \quad \mu = \sum_{i=1,2} \alpha_i \mu_i$$

$$\Sigma_b = \sum_{i=1,2} \alpha_i \Sigma_i$$

  is minimized instead. A point contained in the separating hyperplane is found using unidimensional QDA on the line given by $\mathbf{n}$ and the origin. This means setting $\eta_i = \mathbf{n}^T \mu_i$ and $\sigma_i^2 = \mathbf{n}^T \Sigma_i \mathbf{n}$ in previous equations

# Oblique Splits Example



# Experimental Evaluation

Datasets used

| Name | Source | # cases | # nominal | # continuous |
|---|---|---|---|---|
| Abalone | UCI | 4177 | 1 | 7 |
| Basball | UCI | 261 | 3 | 17 |
| Kin8nm | DVELVE | 8192 | 0 | 8 |
| Mpg | UCI | 392 | 3 | 5 |
| Mumps | `SatLib` | 1523 | 0 | 4 |
| Stock | `SatLib` | 950 | 0 | 10 |
| TA | UCI | 151 | 4 | 2 |
| Tecator | `SatLib` | 240 | 0 | 11 |
| Cart | Breiman et al. | – | 10 | 1 |
| Fried | Friedman | – | 0 | 11 |
| 3DSin | $3\sin(X_1)\sin(X_2)$ | – | 0 | 3 |

- Compared with GUIDE [Loh 2002], state-of-the-art regression tree construction algorithm
- GUIDE uses exhaustive search, GUIDE(S) uses 1% sample
- Experiments performed on a Pentium III 933MHz running Redhat Linux 7.2
- Each experiment repeated 100 times

- For accuracy experiments 50% of data for training, 30% for pruning and 20% for testing
- Quinlan's resubstitution error pruning used

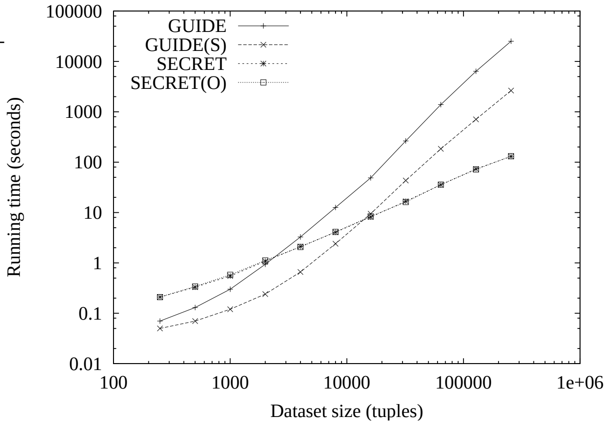
# Accuracy Results

| | Constant Regressors | | | Linear Regressors | | |
|---|---|---|---|---|---|---|
| | GUIDE | SECRET | SECRET(O) | GUIDE | SECRET | SECRET(O) |
| Abalone | 5.32±0.05 | 5.50±0.10 | 5.41±0.10 | *4.63±0.04* | 4.67±0.04 | 4.76±0.05 |
| Baseball | 0.224±0.009 | 0.200±0.008 | 0.289±0.012 | **0.173±0.005** | 0.243±0.011 | 0.280±0.009 |
| Boston | **23.34±0.72** | 28.00±0.92 | 30.91±0.94 | 40.63±6.63 | 24.01±0.69 | 26.11±0.66 |
| Kin8nm | 0.0419±0.0002 | 0.0437±0.0002 | 0.0301±0.0003 | 0.0235±0.0002 | 0.0222±0.0002 | **0.0162±0.0001** |
| Mpg | **12.94±0.33** | 30.09±2.28 | 26.26±2.45 | 34.92±21.92 | 15.88±0.68 | 16.76±0.74 |
| Mumps | 1.34±0.02 | 1.59±0.02 | 1.56±0.02 | **1.02±0.02** | 1.23±0.02 | 1.32±0.04 |
| Stock | 2.23±0.06 | 2.20±0.06 | 2.18±0.07 | 1.49±0.09 | 1.35±0.05 | **1.03±0.03** |
| TA | 0.74±0.02 | 0.69±0.01 | *0.69±0.01* | 0.81±0.04 | 0.72±0.01 | 0.79±0.08 |
| Tecator | 57.59±2.40 | 49.72±1.72 | 28.21±1.75 | 13.46±0.72 | 12.08±0.53 | **7.80±0.53** |
| 3DSin | 0.1435±0.0020 | 0.4110±0.0006 | 0.2864±0.0077 | 0.0448±0.0018 | 0.0384±0.0026 | **0.0209±0.0004** |
| Cart | 1.506±0.005 | **1.171±0.001** | N/A | N/A | N/A | N/A |
| Fried | 7.29±0.01 | 7.45±0.01 | 6.43±0.03 | **1.21±0.00** | 1.26±0.01 | 1.50±0.01 |

- GUIDE and SECRET have comparable accuracy
- Oblique splits sometimes make a big difference

# Scalability Results: 3DSin

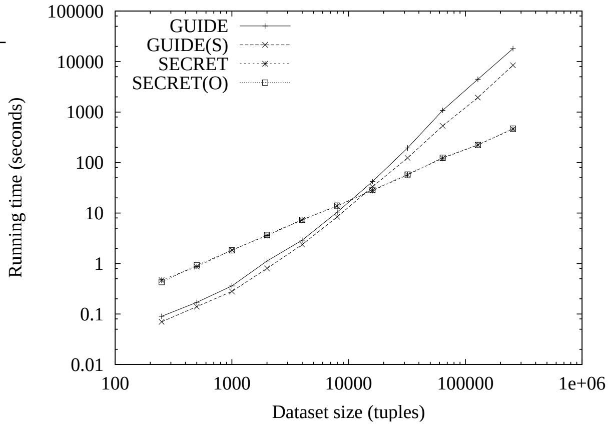
| Size | GUIDE | GUIDE(S) | SECRET | SECRET(O) |
|---|---|---|---|---|
| 250 | 0.07 | 0.05 | 0.21 | 0.21 |
| 500 | 0.13 | 0.07 | 0.33 | 0.34 |
| 1000 | 0.30 | 0.12 | 0.55 | 0.58 |
| 2000 | 0.94 | 0.24 | 1.08 | 1.12 |
| 4000 | 3.28 | 0.66 | 2.11 | 2.07 |
| 8000 | 12.58 | 2.40 | 4.07 | 4.12 |
| 16000 | 48.93 | 9.48 | 8.16 | 8.37 |
| 32000 | 264.50 | 43.25 | 16.71 | 16.19 |
| 64000 | 1389.88 | 184.50 | 35.62 | 35.91 |
| 128000 | 6369.94 | 708.73 | 73.35 | 71.67 |
| 256000 | 25224.02 | 2637.94 | 129.95 | 131.70 |




- Only tree growth time reported (pruning much faster)
- SECRET and SECRET(O) have comparable performance
- GUIDE and GUIDE(S) have quadratic (in the number of tuples) running time
- SECRET and SECRET(O) have linear running time

# Scalability Results: Fried

| Size | GUIDE | GUIDE(S) | SECRET | SECRET(O) |
|---|---|---|---|---|
| 250 | 0.09 | 0.07 | 0.47 | 0.43 |
| 500 | 0.17 | 0.14 | 0.87 | 0.92 |
| 1000 | 0.36 | 0.28 | 1.85 | 1.83 |
| 2000 | 1.12 | 0.80 | 3.58 | 3.69 |
| 4000 | 2.90 | 2.38 | 7.33 | 7.36 |
| 8000 | 10.46 | 8.43 | 13.77 | 14.05 |
| 16000 | 42.16 | 33.09 | 27.80 | 28.68 |
| 32000 | 194.63 | 123.63 | 56.87 | 58.01 |
| 64000 | 1082.70 | 533.16 | 122.26 | 124.60 |
| 128000 | 4464.88 | 1937.94 | 223.42 | 222.75 |
| 256000 | 18052.16 | 8434.33 | 460.12 | 470.68 |




- The increase of the number of attributes to 11 (was 3 before) results in slow-downs of about 3.5 for GUIDE(S), SECRET and SECRET(O) but GUIDE slightly faster
- For large datasets SECRET two orders of magnitude faster than GUIDE and one order of magnitude faster than GUIDE(S)

# Conclusions

- Main idea: locally transform the regression problem into a classification problem
  - First identify two Gaussian distributions in the data
  - Classify the points based on closeness w.r.t. these Gaussian
  - Find best split attribute and best split point for resulting classification problem
  - Find best predictors using linear regression
- SECRET is comparably accurate but much faster than GUIDE
- Oblique splits are easy to obtain and give sometimes 45% accuracy increase
- Most of the running time of SECRET spent in EM. Sampling or scalable EM versions should give significantly speed up

## References


[1] P. S. Bradley, U. M. Fayyad, and C. Reina. Scaling clustering algorithms to large databases. In *Knowledge Discovery and Data Mining*, pages 9–15, 1998.

[2] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, 1984.

[3] P. Chaudhuri, M.-C. Huang, W.-Y. Loh, and R. Yao. Piecewise-polynomial regression trees. *Statistica Sinica*, 4:143–167, 1994.

[4] J. Gehrke, R. Ramakrishnan, and V. Ganti. Rainforest – a framework for fast decision tree construction of large datasets. In *Proceedings of the 24th International Conference on Very Large Databases*, pages 416–427. Morgan Kaufmann, August 1998.

[5] A. Karalic. Linear regression in regression tree leaves. In *International School for Synthesis of Expert Knowledge, Bled,Slovenia*, 1992.

[6] W.-Y. Loh. Regression trees with unbiased variable selection and interaction detection. *Statistica Sinica*, 2002. in press.

[7] J. R. Quinlan. Learning with Continuous Classes. In *5th Australian Joint Conference on Artificial Intelligence*, pages 343–348, 1992.