ONLINE NONLINEAR SYSTEM IDENTIFICATION USING LINEAR MODEL TREES

Duncan Potts * **Claude Sammut** *

* *School of Computer Science and Engineering and ARC Centre of Excellence for Autonomous Systems, University of New South Wales, Australia*

Abstract: A linear model tree is a decision tree with a linear functional model in each leaf that can represent piecewise linear systems. This paper extends recent work in the machine learning literature on the online learning of linear model trees. Both the leaf splitting rule and the pruning rule are improved to take into account more information. The algorithm is demonstrated on two nonlinear system identification problems, including the modelling of complex high dimensional flight dynamics. Results compare favourably with existing online approaches, including an alternative linear model tree algorithm. *Copyright©2005 IFAC.*

Keywords: System Identification, Nonlinear Models, Non-parametric Regression, Decision Trees.

1. INTRODUCTION

A typical approach when identifying systems is to compile a set of example data containing samples of the input and output signals. A batch process then uses this data set to optimise the model parameters. Sometimes, however, it is not possible to obtain this data set in advance, or there is too much data for the batch process, in which case some form of online algorithm is required that can update the system model as more data arrives. For example an autonomous system may have an approximate initial model of its environment, but would benefit from updating this model in an online fashion as new information is received from its sensors.

In this paper it is assumed that a discrete time dynamic environment can be fully described by the time-invariant state-space model

$$\mathbf{z}_{k+1} = f(\mathbf{z}_k, \mathbf{u}_k) \tag{1}$$

where $\mathbf{z}_k$ is the n-dimensional state vector and $\mathbf{u}_k$ is the m-dimensional action vector at time k (a simple extension of continuous time representations such as in (Slotine and Li, 1991) where the input is constant between sampling times). Furthermore it is assumed that the entire state and action vectors can be sensed (with observation noise) at each sampling instant k. The system identification task is to find a sufficiently accurate approximation to the multiple-input multiple-output (MIMO) function f given any available prior knowledge and a sequence of noisy observations of the system's behaviour.

It is often advantageous to decompose the large MIMO problem (1) into n multiple-input single-output (MISO) problems where the outputs are the components of $\mathbf{z}_{k+1}$ (Nelles, 2001). Each of these tasks can be formulated as a regression problem

$$y = f(\mathbf{x}) + \epsilon$$

where $f(\mathbf{x})$ is the corresponding component of $\mathbf{z}_{k+1}$ and $\mathbf{x} = [\mathbf{z}_k^T \ \mathbf{u}_k^T \ 1]^T$ is a $d = n+m+1$ dimensional column vector of regressors (the constant is added to simplify the notation). The observed

values y are corrupted by zero-mean noise ϵ with unknown variance σ^2. The online system identification task is to construct an approximation $\hat{f}(\mathbf{x})$ to $f(\mathbf{x})$ using a sequence of training examples.

This paper examines several non-parametric methods from the system identification and machine learning literature that can construct nonlinear models online. It is shown that each forms a radial basis function representation with linear local models.

The theoretical contribution is to extend one of the online model tree algorithms. Both the leaf splitting rule and the pruning rule are improved to take into account more information. The algorithms are empirically tested in the pendulum and cart domain, and on the identification of complex aircraft dynamics.

2. ONLINE NONLINEAR SYSTEM IDENTIFICATION

This section examines several techniques from both the system identification and machine learning literature.

2.1 Radial Basis Functions with Linear Models

A normalised radial basis function (nRBF) approximation with linear local models is formed by a weighted sum of M linear functions

$$\hat{f}(\mathbf{x}) = \frac{1}{W}\sum_{i=1}^{M} \mathbf{x}^T \boldsymbol{\theta}_i \boldsymbol{\Phi}_i(\mathbf{x}) \text{ and } W = \sum_{i=1}^{M} \boldsymbol{\Phi}_i(\mathbf{x}) \quad (2)$$

where the weights $\boldsymbol{\Phi}_i(\mathbf{x})$ depend on the distance of $\mathbf{x}$ from the centre of the basis function. The most common RBF is a multi-dimensional Gaussian.

As discussed in (Nelles, 2001), if the basis functions are fixed in advance the Md parameters in the $\boldsymbol{\theta}_i$ can be learnt by either global or local linear optimisation. Global methods minimise the sum of the squared errors over all N training examples. If implemented online using the well known recursive least squares (RLS) algorithm (Ljung, 1987) the complexity is $O(M^2d^2)$ per training example.

Local optimisation avoids the expensive global optimisation by locally optimising a loss function in each local model. The local loss functions are the weighted sum of squared errors

$$J_i = \sum_{k=1}^{N} \boldsymbol{\Phi}_i(\mathbf{x}_k) e_{i,k}^2 = \sum_{k=1}^{N} \boldsymbol{\Phi}_i(\mathbf{x}_k)(y_k - \mathbf{x}_k^T \boldsymbol{\theta}_i)^2 \quad (3)$$

The weighted RLS (wRLS) algorithm can update all M local models in time $O(Md^2)$ per training example.

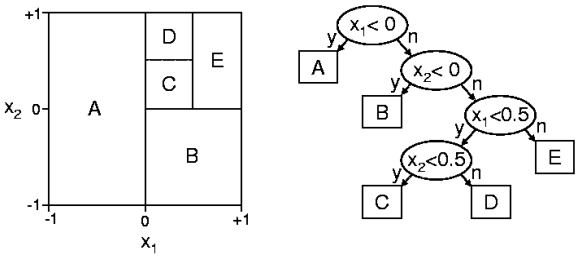

Fig. 1. Partitioning of an input space by a decision tree.

Unfortunately the number of local models M increases exponentially with the number of dimensions if the basis functions are fixed on a grid. In addition the spacing and sizes of the basis functions must be defined in advance, or found using an even more expensive nonlinear optimisation process. Clearly alternative techniques must be found for higher dimensional problems.

2.2 Receptive Field Weighted Regression

The natural answer to this "curse of dimensionality" (Bellman, 1961) is to turn to non-parametric learning algorithms where the complexity of the representation can be varied dynamically. More parameters can be allocated to modelling more intricate parts of the function, and less parameters to simpler regions.

Receptive field weighted regression (RFWR) (Schaal and Atkeson, 1998) dynamically constructs the same nRBF representation as (2). New basis functions are added when the system reaches areas of the input space without sufficient coverage. In addition the size and shape of the basis functions are optimised by online second-order gradient descent. The result is a set of basis functions that are longer in directions of less curvature and shorter in directions of high curvature, therefore improving the prediction accuracy. Each linear model attempts to optimise the local cost function (3) using wRLS with exponential forgetting because of the changing shape of the basis functions.

RFWR has since been adapted to improve its dimensionality reduction capability (Vijayakumar and Schaal, 2000), however the test domains do not contain redundant dimensions and the original algorithm without any ridge regression parameters is more competitive.

Unfortunately the algorithm is sensitive to various parameters, including the initial shape of basis functions and a penalty term γ, and requires a distance metric to be defined over the regressor space (Potts, 2004*a*). Decision trees offer an alternative way to partition the input space that also scale to a high number of dimensions, and require no such metric.

2.3 Online LOLIMOT Algorithm

The locally linear model tree (LOLIMOT) algorithm also uses the nRBF representation (2) (Nelles, 1999). The basis functions are fitted to a rectangular partitioning of the input space performed by a decision tree with axis-orthogonal splits at the internal nodes (see Fig. 1 for an example). For each rectangle, the corresponding basis function has a standard deviation in each dimension that is equal to the width of the rectangle in that dimension multiplied by a constant k_σ. In the experiments $k_\sigma = 0.25$. Each linear model optimises the local cost function (3) using wRLS.

The difficult part is deciding when to grow or prune the tree. For each local model the online LOLIMOT algorithm maintains two background sub-models on either side of each potential axis-orthogonal split bisecting the region. Therefore in each partition in Fig. 1 LOLIMOT maintains four sub-models, one on each side of either a horizontal or vertical split through the centre of the region.

At each time instant k the most active linear model a is considered for splitting, where

$$a = \arg\max_i \left(\mathbf{\Phi}_i(\mathbf{x}_k)\right)$$

Its local loss function J_a is compared with the sum of the local loss functions J_{aj}^- and J_{aj}^+ for the sub-models on each side of the split in dimension j where

$$j = \arg\min_i (J_{ai}^- + J_{ai}^+)$$

is the most promising split. A split is made if $J_a > (J_{aj}^- + J_{aj}^+)k_{\text{improve}}$. An internal node i is considered for pruning if both children l and r are leaves. The leaves are removed if $J_i < (J_l + J_r)$.

Unfortunately it is very difficult to determine the optimal k_{improve} without extensive trial and error, and the value is sensitive to the level of noise in the data (Nelles, 1999). In the experiments it also proved necessary to stop splitting when the input space had fragmented into a predetermined number of regions.

2.4 Incremental Model Tree Induction

In the machine learning community there has also been recent work on learning model trees online (Potts, 2004*a*). The Incremental Model Tree Induction (IMTI) algorithm takes a different approach to the system identification methods described above. The model tree is grown under the assumption that there is no interaction between the local linear models in the leaves. This assumption enables the likelihood that a split is beneficial to be calculated no matter what level of noise is present. Moreover each training example is only passed down the tree to a single leaf, resulting in fast training times.

In each leaf IMTI maintains a linear model, and κ candidate splits along each dimension (therefore generalising the single candidate split in each dimension maintained by the LOLIMOT algorithm). The assumption of no interaction between the models in the leaves means that the loss function is no longer weighted, and is only calculated over the N_i examples observed in each leaf i

$$J_i = \sum_{k=1}^{N_i} e_{i,k}^2 = \sum_{k=1}^{N_i} (y_k - \mathbf{x}_k^T \boldsymbol{\theta}_i)^2 \tag{4}$$

The loss functions, and the linear model parameters in each leaf, are calculated using RLS. The linear sub-models on each side of the $\kappa(d-1)$ candidate splits are also maintained in a similar manner. Denote the loss function on one side of the jth candidate split as J_{ij}^-, and the loss function on the other side as J_{ij}^+. Assuming Gaussian noise with unknown variance the Chow test for homogeneity amongst sub-samples (Chow, 1960) gives the probability that the value of the statistic

$$F_{ij} = \frac{(J_i - J_{ij}^- - J_{ij}^+) \times (N_i - 2d)}{(J_{ij}^- + J_{ij}^+) \times d} \tag{5}$$

occurs under the null hypothesis that the data comes from a single linear model. Under this null hypothesis F_{ij} is distributed according to Fisher's $\mathcal{F}$ distribution with d and $N_i - 2d$ degrees of freedom, and the probability of obtaining F_{ij} is the associated p-value (probability in the tail of the distribution). The best split is found by minimising this probability, and therefore maximising F_{ij}, over every split j. Clearly F_{ij} is maximised when $(J_{ij}^- + J_{ij}^+)$ is minimised, and this strategy corresponds to the split selection in the LOLIMOT algorithm. The advantage of this method is that the minimum probability, α, gives the likelihood that the split is incorrect, and therefore a split is only made when α is less than a predetermined significance level α_{split}. Pruning is performed when the probability α at an internal node rises above the threshold α_{prune}, and a stopping parameter δ_0 controls the asymptotic tree size (Potts, 2004*a*).

Although the tree is grown under the assumption of no interaction between local models, better predictions are obtained by applying a smoothing technique that results in the same representation (2) as nRBF, RFWR and LOLIMOT. Gaussian basis functions are fitted to the rectangular leaf regions in the same manner as for LOLIMOT, with the standard deviation in each dimension equal to the width of the rectangle in that dimension multiplied by the constant k_σ. In the experiments $\alpha_{\text{split}} = 0.01\%$, $\alpha_{\text{prune}} = 0.1\%$, $\kappa = 5$ and $k_\sigma = 0.25$.

3. IMPROVED IMTI ALGORITHM

3.1 Splitting Rule

The derivation of (5) requires that the linear model in each leaf is initially empty, and that the linear models on each side of every candidate split are constructed from the same examples as the leaf model. However in an online setting when a leaf is split and two children are created there is already a fitted linear model for each child (the models that were instrumental in choosing the split). Assume that for a particular node i this initial linear model was created from $N_{i,0}$ examples ($N_{i,0} = 0$ only at the root). As before the jth candidate split partitions additional examples into two sets: N_{ij}^- on one side forming a model with loss function J_{ij}^-, and N_{ij}^+ forming a model with loss function J_{ij}^+ on the other. The leaf model is built from all $N_i = N_{i,0} + N_{ij}^- + N_{ij}^+$ examples and has loss function J_i. Under these more general conditions it can be shown (equation (A-1) in the Appendix with $p = 1$ and $q = 2$) that the statistic

$$S_{ij} = \frac{(J_i - J_{ij}^- - J_{ij}^+) \times (N_{ij}^- + N_{ij}^+ - 2d)}{(J_{ij}^- + J_{ij}^+) \times (N_{i,0} + d)}$$

is distributed according to the $\mathcal{F}$ distribution with $N_{i,0}+d$ and $N_{ij}^- + N_{ij}^+ - 2d$ degrees of freedom. This statistic is used in the online algorithm because it takes into account the $N_{i,0}$ examples already collected before the parent node was split.

3.2 Pruning

Using the same statistic for pruning results in a comparison between the model in each internal node and a combination of the two models in the node's children. A superior pruning statistic can be derived that compares the model in each internal node with the entire piecewise linear approximation constructed by all the leaves in the node's sub-tree. The more general pruning statistic (equation (A-1) in the Appendix)

$$P_i = \frac{(J_i - J_{\mathrm{L}}) \times (N_{\mathrm{L}} - qd)}{J_{\mathrm{L}} \times (N_i - N_{\mathrm{L}} + (q-1)d)}$$

is distributed according to the $\mathcal{F}$ distribution with $N_i - N_{\mathrm{L}} + (q-1)d$ and $N_{\mathrm{L}} - qd$ degrees of freedom, where J_i is calculated from all N_i examples observed at the node and J_{L} is the sum of the loss functions formed from the N_{L} examples observed in the q leaves of the sub-tree rooted at i.

4. EMPIRICAL RESULTS

Model errors are given as the normalised root mean square error (nRMSE) on a set of independent test samples. All results show the mean of 20 trials, with error bars and errors in tables indicating one unbiased estimate of the standard deviation over these trials.

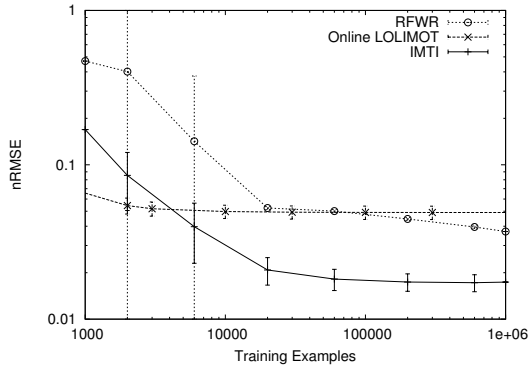

Fig. 2. Model errors on the cart and pendulum.

4.1 Cart and Pendulum

The problem of swinging up a pendulum on a cart is highly nonlinear when the pendulum is allowed to rotate through 360°. The simplified dynamic model (not taking into account frictional effects) is

$$0 = \ddot{x}\cos\theta - l\ddot{\theta} - g\sin\theta$$
$$u = (m+M)\ddot{x} - ml\ddot{\theta}\cos\theta + ml\dot{\theta}^2\sin\theta$$

where x is the position of the cart (limited to ± 2), θ is the angle of the pendulum, $l = 1$ is the length of the pendulum, g is gravity, $M = m = 1$ are the masses of the cart and pendulum respectively, and u is the lateral force applied to the cart (limited to ± 7). The state at time k is $\mathbf{z}_k = [x_k\ \dot{x}_k\ \theta_k\ \dot{\theta}_k]^T$. The next state of the system is calculated using 5 successive Euler integrations with a time step of 0.01 seconds to give an overall sampling rate of 20 times per second. The system is initialised at rest with the pendulum hanging vertically downward. A simple hand-coded control strategy repeatedly swings up the pendulum and balances it for a short period using the observed state vector $\mathbf{y}_k = \mathbf{z}_{k+1} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is a vector of independent zero-mean Gaussian noise with variance σ^2 and $\sigma = 0.1$. The system identification task is to learn the dynamics given examples of $\langle \mathbf{x}_k, \mathbf{y}_k \rangle$. The sequence of states generated is given directly to the learner without changing the order, and is therefore very highly correlated. The algorithms are tested using 10,000 examples randomly drawn from a similar sequence, but without noise.

The RFWR initial distance metric $\mathbf{D}_0 = 10\mathbf{I}$, the penalty $\gamma = 10^{-7}$ and the learning rates are 1000. Online LOLIMOT uses $k_{\mathrm{improve}} = 1.3$ and the number of models is limited to 100. IMTI uses a stopping parameter $\delta_0 = 0.001$. Fig. 2 shows how the approximation error evolves over time for each algorithm, demonstrating the superior model constructed by IMTI. Table 1 shows that IMTI

Table 1. Model sizes and training times per example for the cart and pendulum.

Algorithm	Number of local models	Training time (ms)
RFWR	136 ± 4	37 ± 1
Online LOLIMOT	100 ± 0	25 ± 4
IMTI	98 ± 30	18 ± 0

uses no more local models than RFWR or online LOLIMOT, although the variation in local model numbers over the 20 trials is high.

4.2 Flight Simulator

Learning to fly an aeroplane is a complex high-dimensional task. These experiments use a flight simulator based on a high-fidelity flight model of a Pilatus PC-9 aerobatic aircraft. The model was provided by the Australian Defence Science and Technology Organisation and is based on wind tunnel and in-flight performance data. The same simulator has also been used in previous work (Isaac and Sammut, 2003; Potts, 2004*b*).

The system is sampled 4 times per second, and 9 state variables are recorded (altitude, roll, pitch, yaw rate, roll rate, pitch rate, climb rate, air speed and a Boolean variable indicating whether the plane is on the ground) along with 4 action variables (ailerons, elevator, throttle and the categorical flaps setting which can take the values 'normal', 'take off' and 'landing'). These state variables were selected so that a dynamic model of the plane could be learnt, and therefore the absolute position and heading were disregarded. The combination of states and actions results in a 13 dimensional regressor vector. The learning task is to predict the 8 continuous values of the next state.

The training examples are taken directly from a trace of the aircraft flying so that successive regressors are highly correlated. Simulated turbulence is set to a high level resulting in complex noise characteristics that deviate substantially from the independent Gaussian assumption. Additional noise is also added to the inputs to excite the system and provide a richer source of training examples. The algorithms are tested using 10,000 examples randomly drawn from a similar trace.

The RFWR initial distance metric $\mathbf{D}_0 = 2.5\mathbf{I}$, the penalty $\gamma = 10^{-5}$ and the learning rates are 1000. Online LOLIMOT uses $k_{\text{improve}} = 2.0$ and the number of models is limited to 30. IMTI uses a stopping parameter $\delta_0 = 0.02$. Fig. 3 shows how the approximation error evolves over time for each algorithm, and also for a single linear model updated using recursive least squares (RLS). Clearly RFWR and IMTI form good models of the aircraft

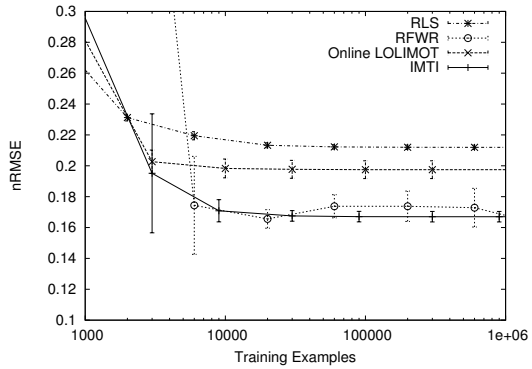

Fig. 3. Model errors on the flight simulator.

Table 2. Model sizes and training times per example for the flight simulator.

Algorithm	Number of local models	Training time (ms)
RFWR	26 ± 5	18 ± 2
Online LOLIMOT	30 ± 0	74 ± 9
IMTI	26 ± 5	108 ± 1

dynamics. Table 2 shows that each algorithm uses approximately the same number of local models, although IMTI and online LOLIMOT are computationally expensive on this higher dimensional problem.

5. CONCLUSIONS

This paper compares three recent online nonlinear function approximation algorithms that can be used for system identification and show potential to scale to a large number of dimensions. All three representations are seen to fit into the basis function paradigm. Receptive field weighted regression dynamically adjusts its basis functions to spread over areas with low curvature. The other two algorithms, incremental model tree induction (IMTI) and the online local linear model tree (LOLIMOT) algorithm use a decision tree to partition the input space.

The tree-based algorithms have fewer parameters and no learning rates to tune, thus avoiding a major cause of instability in many gradient descent systems like RFWR. In addition they require no metric over the input space. IMTI has a single tunable parameter that intuitively controls the trade-off between the number of linear models and the approximation error. Online LOLIMOT has two parameters; the splitting parameter k_{improve} and the maximum number of models. Although the limit on the number of models is easy to set, the algorithm is highly sensitive to k_{improve} and poor learning occurs if it is too large or small.

Improvements to the IMTI algorithm are detailed, allowing it to utilise more data in its splitting and pruning decisions. IMTI is seen to have consistently superior learning characteristics, although

it starts to become computationally expensive in higher dimensions. A better scaling version of the algorithm has been developed although it can learn less effectively (Potts, 2004*b*).

REFERENCES

Bellman, R. (1961). *Adaptive control processes*. Princeton University Press.

Chow, G. (1960). Tests of equality between sets of coefficients in two linear regressions. *Econometrica* **28**(3), 591–605.

Isaac, A. and C. Sammut (2003). Goal-directed learning to fly. In: *Proceedings of the 20th International Conference of Machine Learning*. pp. 258–265.

Kullback, S. and H. Rosenblatt (1957). On the analysis of multiple regression in k categories. *Biometrika* **44**, 67–83.

Ljung, L. (1987). *System identification: Theory for the user*. Prentice-Hall.

Nelles, O. (1999). Nonlinear system identification with local linear neuro-fuzzy models. PhD thesis. TU Darmstadt.

Nelles, O. (2001). *Nonlinear system identification*. Springer.

Potts, D. (2004*b*). Incremental learning of linear model trees. In: *Proceedings of the 21st International Conference on Machine Learning*. pp. 663–670.

Potts, D. (2004*a*). Fast incremental learning of linear model trees. In: *Proceedings of PRICAI-04, Lecture Notes in Artificial Intelligence, 3157*. Springer. pp. 221–230.

Schaal, S. and C. Atkeson (1998). Constructive incremental learning from only local information. *Neural Computation* **10**, 2047–2084.

Slotine, J. and W. Li (1991). *Applied nonlinear control*. Prentice-Hall.

Vijayakumar, S. and S. Schaal (2000). Locally weighted projection regression: Incremental real time learning in high dimensional space. In: *Proceedings of the 17th International Conference on Machine Learning*. pp. 1079–1086.

APPENDIX

This appendix derives the statistical test used in the improved IMTI algorithm. Label the internal nodes of a particular sub-tree rooted at I1 as I1 . . . Ip, and the leaves as L1 . . . Lq (see Fig. 4).

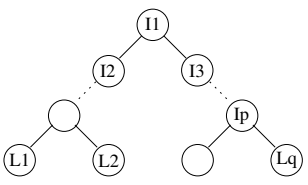

Fig. 4. Labelling of sub-tree nodes.

The sub-tree was grown online, and initially only consisted of the sub-tree root node I1. Therefore there are a number of examples $N_{\mathrm{I}1,0}$ that were observed at I1 but not at any lower node. Similarly at each internal node j there are $N_{\mathrm{I}j,0}$ examples that were observed before the node had any children. A linear model is constructed at each internal node j from the $N_{\mathrm{I}j,0}$ examples observed before the node was split, and the corresponding loss functions (4) $J_{\mathrm{I}j,0}$ are calculated (the residual sums of squares). In each leaf i a linear model is formed from the $N_{\mathrm{L}i}$ examples that reached the leaf, and the loss functions $J_{\mathrm{L}i}$ are also calculated.

If the sum of $N_{\mathrm{L}i}$ over all leaves is denoted as N_{L} and the sum of $N_{\mathrm{I}j,0}$ over all internal nodes is denoted $N_{\mathrm{I},0}$ then the sub-tree has observed a total of $N = N_{\mathrm{L}} + N_{\mathrm{I},0}$ examples. A single linear model is constructed at the root I1 from all N examples and the corresponding loss function J is calculated. Also denote the sum of $J_{\mathrm{L}i}$ over all leaves as J_{L} and the sum of $J_{\mathrm{I}j,0}$ over all internal nodes as $J_{\mathrm{I},0}$.

Making the assumptions that the observation noise is independent, zero-mean and Gaussian with variance σ^2, and that the regressor matrix in each linear regression defined above has full rank d, then we can form the null hypothesis H_0 that all the observed examples were generated from a single linear model. The alternative hypothesis is that the data is better explained by the linear models in the leaves of the tree.

Using standard analysis of covariance techniques generalised to multiple regressions (Kullback and Rosenblatt, 1957) it can be shown that the three expressions on the right-hand side of the identity

$$J \equiv J_{\mathrm{L}} + J_{\mathrm{I},0} + (J - J_{\mathrm{L}} - J_{\mathrm{I},0})$$

are distributed independently as $\chi^2(N_{\mathrm{L}} - qd)\sigma^2$, $\chi^2(N_{\mathrm{I},0} - pd)\sigma^2$ and $\chi^2((p+q-1)d)\sigma^2$. To obtain a better comparison between the null and alternative hypotheses, the effect of the internal nodes is removed. Adding the last two expressions above gives $(J - J_{\mathrm{L}})$ which is distributed as $\chi^2(N_{\mathrm{I},0} + (q-1)d)\sigma^2$ by the summation of two independent χ^2-distributed variables. This distribution is clearly affected if H_0 does not hold, whereas J_{L} has the same distribution regardless. Following Chow (Chow, 1960) H_0 can therefore be tested by the statistic

$$F = \frac{(J - J_{\mathrm{L}}) \times (N_{\mathrm{L}} - qd)}{J_{\mathrm{L}} \times (N_{\mathrm{I},0} + (q-1)d)} \qquad \text{(A-1)}$$

which is distributed according to Fisher's $\mathcal{F}$ distribution with $N_{\mathrm{I},0} + (q-1)d$ and $N_{\mathrm{L}} - qd$ degrees of freedom by the definition of the $\mathcal{F}$ distribution as the ratio of two independent χ^2-distributed variables.